

T.C.
VAN YÜZÜNCÜ YIL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
YAPAY ZEKÂ VE ROBOTİK ANABİLİM DALI

**AŞIRI – AKIM RÖLELERİN KOORDİNASYON PROBLEMİNDE
GELİŞTİRİLMİŞ FİDAN GELİŞİM ALGORİTMASININ UYGULANMASI VE
OPTİMİZASYON ALGORİTMALARI İLE KARŞILAŞTIRILMASI**

YÜKSEK LİSANS TEZİ

Özge ÖZGÜNER ARAS
Danışman: Dr. Öğr. Üyesi Ebubekir SEYYARER

VAN – 2025

T.C.
VAN YÜZÜNCÜ YIL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
YAPAY ZEKÂ VE ROBOTİK ANABİLİM DALI

**AŞIRI – AKIM RÖLELERİN KOORDİNASYON PROBLEMİNDE
GELİŞTİRİLMİŞ FİDAN GELİŞİM ALGORİTMASININ UYGULANMASI VE
OPTİMİZASYON ALGORİTMALARI İLE KARŞILAŞTIRILMASI**

YÜKSEK LİSANS TEZİ

Özge ÖZGÜNER ARAS

Tez Savunma Sınavı Jüri Üyeleri

Dr. Öğr. Üyesi Cengiz HARK (Başkan)

Dr. Öğr. Üyesi Ebubekir SEYYARER (Danışman)

Dr. Öğr. Üyesi Faruk AYATA (Üye)

VAN – 2025

KABUL VE ONAY SAYFASI

Van Yüzüncü Yıl Üniversitesi Fen Bilimleri Enstitüsü, Yapay Zeka ve Robotik Anabilim Dalı'nda Dr. Öğr. Üyesi Ebubekir SEYYARER danışmanlığında, Özge ÖZGÜNER ARAS tarafından sunulan “Aşırı – Akım Rölelerin Koordinasyon Probleminde Geliştirilmiş Fidan Gelişim Algoritmasının Uygulanması ve Optimizasyon Algoritmaları ile Karşılaştırılması” başlıklı bu çalışma Lisansüstü Eğitim ve Öğretim Yönetmeliği'nin ilgili hükümleri gereğince 16/05/2025 tarihinde aşağıdaki jüri tarafından oy birliği / oy çokluğu ile başarılı bulunmuş ve Yüksek Lisans tezi olarak kabul edilmiştir.

Başkan: Dr. Öğr. Üyesi Cengiz HARK

İmza:

Üye: Dr. Öğr. Üyesi Faruk AYATA

İmza:

Üye: Dr. Öğr. Üyesi Ebubekir SEYYARER

İmza:

Üye:

İmza:

Fen Bilimleri Enstitüsü Yönetim Kurulu'nun /.... /..... tarih ve sayılı kararı ile onaylanmıştır.

İmza

.....

Enstitü Müdürü

ETİK BEYAN

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

Özge ÖZGÜNER ARAS



ÖZET

AŞIRI – AKIM RÖLELERİN KOORDİNASYON PROBLEMİNDE GELİŞTİRİLMİŞ FİDAN GELİŞİM ALGORİTMASININ UYGULANMASI VE OPTİMİZASYON ALGORİTMALARI İLE KARŞILAŞTIRILMASI

ÖZGÜNER ARAS, Özge

Yüksek Lisans Tezi, Yapay Zekâ ve Robotik Anabilim Dalı

Danışman: Dr. Öğr. Üyesi Ebubekir SEYYARER

Haziran 2025, 88 sayfa

Optimizasyon algoritmaları, belirli bir amaç fonksiyonunun minimum veya maksimum değerini bulmayı amaçlayan problemleri çözmeye yaygın olarak kullanılmaktadır. Optimizasyon yöntemleri, en iyi alternatifi seçmek için hem matematiksel hem de sezgisel yaklaşımları kullanmaktadır. Bu çalışmada Aşırı Akım Rölelerinin Koordinasyon (Overcurrent Relay Coordination - ORC) problemi ele alınmış ve bu problemin çözümüne yönelik olarak biyolojik büyüme sürecinden esinlenen Fidan Gelişim Algoritması (SGuA) geliştirilmiştir. SGuA, Geliştirilmiş SGuA (ISGuA) versiyonu ile detaylı biçimde analiz edilmiş, özellikle algoritmanın başlangıç popülasyonu üretim yöntemlerinin performansa etkisi araştırılmıştır.

Çalışmada 10 farklı başlatma yöntemi değerlendirilmiştir. Uniform dağılım yöntemi 6.9002 saniyelik en düşük toplam ti süresi ile en başarılı sonuçları vermiştir. Uniform dağılımın parametre uzayını homojen örnekleyerek çözüm alanını dengeli taradığı ve algoritmanın yerel minimumlara takılma riskini azalttığı gözlemlenmiştir. Diğer dağılımlar arasında Beta dağılımları belirli bölgelere odaklanarak erken yakınsama sağlarken, Log-normal, Rayleigh, Weibull ve Exponential gibi asimetric dağılımlar çözüm uzayının bazı bölgelerinde yetersiz örnekleme nedeniyle performansı düşürmüştür. Latin hypercube ve rastgele başlatmalar da Uniform dağılıma kıyasla daha zayıf performans göstermiştir.

Geliştirilen ISGuA algoritması, ayrıca Genetik Algoritma (GA), Balina Optimizasyon Algoritması (WOA), Yapay Arı Kolonisi (ABC), Gri Kurt (GWO), Kelebek (BOA) ve Lig Şampiyonluk Algoritması (LCA) gibi yaygın meta sezgisel yöntemlerle karşılaştırılmıştır. ISGuA, en düşük işletme süresine ulaşarak tüm algoritmalar arasında en başarılı yöntem olmuştur.

Elde edilen sonuçlar, ISGuA'nın hem başlangıç stratejileri hem de genel optimizasyon performansı açısından üstün olduğunu ve ORC gibi çok kısıtlı problemlerde uygulanabilir, kararlı ve verimli bir çözüm sunduğunu göstermektedir.

Anahtar kelimeler: Aşırı – akım röle koordinasyonu, Balina optimizasyon algoritması, Fidan gelişim algoritması, Geliştirilmiş fidan gelişim algoritması, Genetik algoritma, Gri kurt optimizasyon algoritması, Kelebek optimizasyon algoritması, Lig şampiyonluk algoritması, Yapay arı kolonisi algoritması

ABSTRACT

APPLICATION OF THE IMPROVED SAPLING DEVELOPMENT ALGORITHM IN THE COORDINATION PROBLEM OF OVER-CURRENT RELAYS AND COMPARISON WITH OPTIMIZATION ALGORITHMS

ÖZGÜNER ARAS, Özge

M.Sc. Thesis, Department of Artificial Intelligence and Robotics

Supervisor: Asst. Prof. Dr. Ebubekir SEYYARER

June 2025, 88 pages

Optimization algorithms are widely used to solve problems that aim to find the minimum or maximum value of a given objective function. These methods employ both mathematical and heuristic approaches to select the best possible alternative. In this study, the Overcurrent Relay Coordination (ORC) problem is addressed, and a nature-inspired metaheuristic algorithm called the Saplings Growing up Algorithm (SGuA), based on the biological growth process, is proposed. The enhanced version of the algorithm, Improved SGuA (ISGuA), is analyzed in detail, with a particular focus on the impact of initial population generation methods on performance.

Ten different initialization strategies were evaluated. Among them, the uniform distribution method produced the best result with the lowest total operation time of 6.9002 seconds. It was observed that uniform distribution provided balanced exploration of the solution space by homogeneously sampling the parameter domain, thereby reducing the risk of getting trapped in local minima. Among other distributions, beta distributions allowed early convergence by focusing on specific regions, whereas asymmetric distributions such as log-normal, Rayleigh, Weibull, and exponential led to reduced performance due to insufficient sampling in parts of the search space. Latin hypercube and random initialization also showed weaker performance compared to the uniform approach.

The developed ISGuA algorithm was further compared with commonly used metaheuristic methods such as Genetic Algorithm (GA), Whale Optimization Algorithm (WOA), Artificial Bee Colony (ABC), Grey Wolf Optimizer (GWO), Butterfly Optimization Algorithm (BOA), and League Championship Algorithm (LCA). ISGuA outperformed all of these algorithms by achieving the shortest total execution time.

The results demonstrate that ISGuA exhibits superior performance both in terms of initialization strategies and overall optimization capability, providing a stable, efficient, and applicable solution for constrained problems such as ORC.

Keywords: Artificial bee colony algorithm, Butterfly optimization algorithm, Genetic algorithm, Grey wolf optimizer algorithm, Improved saplings growing up algorithm, League championship algorithm, Overcurrent relay coordination, Saplings growing up algorithm, Whale optimization algorithm



TEŞEKKÜR

Bu tez çalışmasının her aşamasında bana destek olan ve katkı sağlayan herkese en içten teşekkürlerimi sunarım. Öncelikle, akademik gelişimime katkılarıyla yolumu aydınlatan ve çalışmam boyunca rehberlik eden değerli danışmanım Dr. Öğr. Üyesi Ebubekir SEYYARER'e derin şükranlarımı sunarım. Onun yönlendirmeleri, sabrı ve bilgi birikimi, bu çalışmanın şekillenmesinde büyük bir rol oynamıştır.

Bu süreçte her zaman yanımda olan, beni motive eden ve sabırla destekleyen canım eşime ve aileme sonsuz teşekkür ederim. Onların sevgisi ve inancı, karşılaştığım her zorluğu aşmamda en büyük güç kaynağım oldu. Ayrıca, bana sevgileriyle huzur ve neşe veren can dostlarım Bide ve Gibson'a da teşekkür etmek istiyorum. Onların varlığı, yoğun geçen akademik süreçte bana eşsiz bir mutluluk ve motivasyon kaynağı oldu. Bu süreçte yanımda olan, fikir alışverişinde bulunduğum, motivasyonumu yüksek tutan tüm dostlarıma teşekkür ederim. Akademik yolculuk, bireysel bir çaba gibi görünse de aslında birçok kişinin desteğiyle anlam kazanmaktadır.

Bu tez, yalnızca akademik bir çalışmanın ürünü değil, aynı zamanda birlikte öğrenmenin, paylaşmanın ve emeğin bir sonucudur.

2025

Özge ÖZGÜNER ARAS



İÇİNDEKİLER

	Sayfa
ÖZET	i
ABSTRACT	iii
TEŞEKKÜR	v
İÇİNDEKİLER	vii
ÇİZELGELER LİSTESİ	ix
ŞEKİLLER LİSTESİ	xi
SİMGELER VE KISALTMALAR	xiii
1. GİRİŞ	1
2. KAYNAK BİLDİRİŞLERİ	3
3. MATERYAL VE YÖNTEM	7
3.1 Meta Sezgisel Algoritmalar	7
3.1.1 Balina Optimizasyon Algoritması	14
3.1.2 Genetik Algoritma	17
3.1.3 Gri Kurt Algoritması	21
3.1.4 Kelebek Algoritması	25
3.1.5 Lig Şampiyonluk Algoritması	29
3.1.6 Yapay Arı Algoritması	33
3.1.7 Fidan Gelişim Algoritması	36
3.1.8 Geliştirilmiş Fidan Gelişim Algoritması	43
3.2 Aşırı Akım Röle Koordinasyonu	46
3.2.1 Aşırı Akım Röle Koordinasyonu Probleminin Formülasyonu	48
3.3 Çok Modlu Kıyaslama Fonksiyonları	49
4. BULGULAR	51
4.1 Fidan Gelişim Algoritması için Farklı Başlatma Yöntemlerinin Performansa Etkisi	51
4.2 Balina Optimizasyon Algoritması Sonuçları	56
4.3 Genetik Algoritma Sonuçları	57
4.4 Gri Kurt Algoritması Sonuçları	57
4.5 Kelebek Optimizasyon Algoritması Sonuçları	58
4.6 Lig Şampiyonluk Algoritması Sonuçları	58
4.7 Yapay Arı Algoritması Sonuçları	59
4.8 Çok Modlu Kıyaslama Fonksiyonlarına Ait Sonuçlar	60
5. TARTIŞMA VE SONUÇ	67
KAYNAKLAR	79

ÖZ GEÇMİŞ.....	87
----------------	----



ÇİZELGELER LİSTESİ

	Sayfa
Çizelge 3.1 Biyoloji tabanlı algoritmalar	9
Çizelge 3.2 Sürü tabanlı algoritmalar	9
Çizelge 3.3 Fizik tabanlı algoritmalar	11
Çizelge 3.4 Kimya tabanlı algoritmalar	11
Çizelge 3.5 Sosyal tabanlı algoritmalar	12
Çizelge 3.6 Spor tabanlı algoritmalar	13
Çizelge 3.7 Müzik tabanlı algoritmalar	13
Çizelge 3.8 Matematik tabanlı algoritmalar	14
Çizelge 3.9 Balina optimizasyon algoritmasının sözde kodu.....	16
Çizelge 3.10 Genetik algoritmanın sözde kodu.....	19
Çizelge 3.11 Gri kurt optimizasyon algoritmasının sözde kodu	22
Çizelge 3.12 Kelebek optimizasyon algoritmasının sözde kodu	26
Çizelge 3.13 Lig şampiyonluk algoritmasının sözde kodu.....	31
Çizelge 3.14 Yapay arı kolonisi optimizasyon algoritmasının sözde kodu.....	34
Çizelge 3.15 Başlangıç popülasyonunu oluşturmak.....	39
Çizelge 3.16 Eşleşme ($S1, S2$)	40
Çizelge 3.17 Dallanma ($S1, S2$)	40
Çizelge 3.18 Aşılama ($S1, S2$)	41
Çizelge 3.19 Sgua-1'in sözde kodu	42
Çizelge 3.20 Sgua-2'nin sözde kodu	42
Çizelge 3.21 Başlatma şeması	44
Çizelge 3.22 Isgua'nın sözde kodu.....	44
Çizelge 3.23 Çok modlu kıyaslama fonksiyonları	50
Çizelge 4.1 Rand	52
Çizelge 4.2 Betarnd (3, 2).....	52
Çizelge 4.3 Betarnd (2.5, 2.5).....	52
Çizelge 4.4 Unifrnd (0, 1).....	53
Çizelge 4.5 Lognrnd (0, 0.5).....	53
Çizelge 4.6 Lognrnd (0, 0.67).....	53

Çizelge 4.7 Exprnd (0.5).....	54
Çizelge 4.8 Raylrnd (0.4)	54
Çizelge 4.9 Wblrnd (1, 1).....	54
Çizelge 4.10 lhsdesign.....	55
Çizelge 4.11 Toplam çalışma süreleri	55
Çizelge 4.12 Balina optimizasyon algoritması	57
Çizelge 4.13 Genetik algoritma	57
Çizelge 4.14 Gri kurt optimizasyon algoritması.....	57
Çizelge 4.15 Kelebek optimizasyon algoritması	58
Çizelge 4.16 Lig şampiyonluk algoritması.....	58
Çizelge 4.17 Yapay arı kolonisi optimizasyon algoritması.....	59
Çizelge 4.18 Diğer algoritmaların sonuçları.....	59
Çizelge 4.19 Algoritmaların çok modlu kıyaslama fonksiyonlarına ait sonuçları	61
Çizelge 4.20 Isgua'nın 2 boyutlu çok modlu kıyaslama fonksiyonlarına ait sonuçları..	62
Çizelge 4.21 Isgua'nın 10 boyutlu çok modlu kıyaslama fonksiyonlarına ait sonuçları.....	63
Çizelge 4.22 Çok modlu kıyaslama fonksiyonları sonucu optimuma ulaşan algoritmalar.....	65
Çizelge 4.23 Çok modlu kıyaslama fonksiyonları sonucu optimuma en yakın sonuca ulaşan algoritmalar.....	65

ŞEKİLLER LİSTESİ

	Sayfa
Şekil 3.1 Meta sezgisel algoritmaların sınıflandırılması (Akyol ve Alatas, 2017).....	8
Şekil 3.2 Kambur balinaların avlanma stratejisi (Mirjalili ve Lewis, 2016)	15
Şekil 3.3 Balina optimizasyon algoritmasının akış diyagramı	17
Şekil 3.4 Genetik algoritmanın genel şeması (Keshav ve Sharma, 2022).....	18
Şekil 3.5 Genetik algoritmanın akış diyagramı	20
Şekil 3.6 Gri kurt optimizasyon algoritması hiyerarşi zinciri ve avlanma mekanizmasının diyagramı (Yuan vd., 2022).....	21
Şekil 3.7 Gri kurt optimizasyon algoritmasının akış diyagramı.....	24
Şekil 3.8 Kelebeklerin hareket davranışı (Makhadmeh vd., 2023)	26
Şekil 3.9 Kelebek optimizasyon algoritmasının akış diyagramı	28
Şekil 3.10 Lig şampiyonluk algoritmasının akış şeması (Kashan, 2014).....	30
Şekil 3.11 Lig şampiyonluk algoritmasının akış diyagramı	32
Şekil 3.12 Arı kolonisinin nektar toplama şeması (Chao ve Li, 2022)	34
Şekil 3.13 Yapay arı kolonisi optimizasyon algoritmasının akış diyagramı	35
Şekil 3.14 Bahçeye fidanların eşit şekilde dağıtılması (Karcı ve Alataş, 2006)	36
Şekil 3.15 SGuA akış diyagramı	43
Şekil 3.16 ISGuA akış diyagramı	46
Şekil 3.17 Örnek trafo merkezi tek hat şeması (Seyyarer vd. 2018)	47
Şekil 4.1 Toplam çalışma süreleri	56
Şekil 4.2 Tüm sonuçlar.....	60



SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış bazı simgeler ve kısaltmalar, açıklamalarıyla aşağıda sunulmuştur.

Simgeler

Açıklama

I_{kd}

Kısa devre akım değeri

I_p

Rölenin başlatma akım değeri

t_d

Çalışma eğrisi değeri

t_i

Rölenin çalışma süresi

Kısaltmalar

Açıklama

ABC

Artificial Bee Colony Algorithm

ACO

Ant Colony Optimization

ACROA

Artificial Chemical Reaction Optimization Algorithm

AHA

Artificial Hummingbird Algorithm

AOA

Arithmetic Optimization Algorithm

ARBO

Athletic Run-Based Optimization

ARO

Artificial Rabbits Optimization

ASO

Atom Search Optimization

BBBC

Big Bang-Big Crunch

BHA

Black Hole Algorithm

BOA

Butterfly Optimization Algorithm

BOA

Base Optimization Algorithm

BSO

Brain Storm Optimization

CapSA

Capuchin Search Algorithm

CBO

Colliding Bodies Optimization

CFA

Chaotic Firefly Algorithm

CFO

Central Force Optimization

CGA	Continuous Genetic Algorithm
COA	Coati Optimization Algorithm
CRO	Chemical Reaction Optimization
CSA	Circle Search Algorithm
CSS	Charged System Search
DE	Differential Evolution
DMO	Dwarf Mongoose Optimization
DOA	Dynastic Optimization Algorithm
DOCR	Directional Overcurrent Relay
EO	Equilibrium Optimizer
ER-WCA	Enhanced Water Cycle Algorithm
EVO	Energy Valley Optimizer
FA	Firefly Algorithm
FRA	Flow Regime Algorithm
GA	Genetic Algorithm
GOA	Gazelle Optimization Algorithm
GSA	Gravitational Search Algorithm
GSA	Golden Sine Algorithm
GWO	Grey Wolf Optimizer Algorithm
HBA	Honey Badger Algorithm
HBO	Heap-Based Optimizer
HBBO	Human Behavior-Based Optimization
HHO	Harris Hawks Optimizer
HMS	Human Mental Search
HOA	Horse Herd Optimization Algorithm
HS	Harmony Search
HUA	Human Urbanization Algorithm
HWOA	Hybrid Whale Optimization Algorithm
ICA	Imperialist Competitive Algorithm
ILCA	Improved League Championship Algorithm
ISGuA	Improved Saplings Growing Up Algorithm
KGMO	Kinetic Gas Molecules Optimization

LA	Lichtenberg Algorithm
LCA	League Championship Algorithm
LFD	Lévy Flight Distribution
LHS	Latin Hypercube Sampling
MBO	Monarch Butterfly Optimization
MFO	Moth-Flame Optimization
MMC	Method Of Musical Composition
MRFO	Manta Ray Foraging Optimization
MSA	Melody Search Algorithm
NNA	Neural Network Algorithm
OF	Objective Function
OIO	Optics-Inspired Optimization
ORC	Overcurrent Relay Coordination
PFA	Path-Finder Algorithm
PO	Political Optimizer
POA	Parliamentary Optimization Algorithm
PRO	Poor And Rich Optimization
PSO	Particle Swarm Optimization
SCA	Society And Civilization Algorithm
SCA	Sine–Cosine Algorithm
SGuA	Saplings Growing Up Algorithm
SHO	Sea-Horse Optimizer
SFLA	Shuffled Frog Leaping Algorithm
SLC	Soccer League Competition
SMO	Starling Murmuration Optimizer
SOA	Seeker Optimization Algorithm
SOA	Skill Optimization Algorithm
SPBO	Student Psychology-Based Optimization
SRS	Special Relativity Search
SSA	Salp Swarm Algorithm
STBO	Sewing Training-Based Optimization
TDS	Time Dial Setting

TEİAŞ	Türkiye Elektrik İletim Anonim Şirketi
TEO	Thermal Exchange Optimization
TLBO	Teaching-Learning-Based Optimization
TSA	Tunicate Search Algorithm
TSM	Time Setting Multiplier
TWO	Tug Of War Optimization
VSA	Vortex Search Algorithm
WCA	Water Cycle Algorithm
WCO	World Cup Optimization
WDO	Wind Driven Optimization
WEO	Water Evaporation Optimization
WOA	Whale Optimization Algorithm
WSO	White Shark Optimizer
WSO	War Strategy Optimization
YDSE	Young Double-Slit Experiment Optimizer

1. GİRİŞ

Optimizasyon algoritmaları, birçok problemi çözmek için sıklıkla kullanılmaktadır. Optimizasyon, bir denklemin en iyi (minimum / maksimum) değerini bulmayı amaçlayan matematik disiplini olarak tanımlanmaktadır. Optimizasyon teorisi ve yöntemleri, verilen amaç fonksiyonu anlamında en iyi alternatifin seçilmesidir (Chong, 2013). Optimizasyon sistemlerinin çözümü için matematiksel yöntemlerle birlikte heuristik yöntemler de kullanılmaktadır (Govan, 2006). Literatürde, Meta-Heuristik terimi Meta ve Heuristik terimlerinin birleşmesiyle oluşturulmaktadır. Bazı kurallar ve prosedürler aracılığıyla problemleri çözen yöntemlerin bütünü olarak tanımlanmaktadır (Aydın, 2009). Bu çalışma Aşırı Akım Rölelerinin Koordinasyon (Overcurrent Relay Coordination - ORC) problemini, Fidan Gelişim Algoritması (SGuA) geliştirilerek Geliştirilmiş Fidan Gelişim Algoritması (ISGuA) ile altı optimizasyon algoritması kullanarak minimize etmek ve çıkan sonuçları karşılaştırmak hedeflenmektedir.

Güç sistemlerinin yapısı, bugünkü teknolojik gelişmelerle birlikte oldukça karmaşık hale gelmektedir (Onat, 2018). Güç sistemlerinin çalışması sırasında, koruyucu röleler büyük önem taşımaktadır (Zeineldin vd., 2015). Koruma rölelerinin ana amacı, elektrik sistemlerinde meydana gelebilecek arızaları hızlı bir şekilde tespit etmek ve bu arızaların neden olduğu zararları azaltmak ya da ortadan kaldırmaktır. Bu sayede, güç sistemlerinin çalışması sırasında meydana gelebilecek zararlar önlenabilmektedir ve elektrik sistemlerinin güvenliği sağlanmaktadır. Koruma röleleri, mesafe koruma, aşırı akım, diferansiyel röleleri gibi çeşitleri bulunmaktadır. Güç sistemlerinin enerjinin sürekliliğinin ve korunmasının sağlanması da mühendislik alanında önemli bir konu olmaktadır. Bu nedenle, güç sistemlerinin korunması için koruma rölelerinin doğru bir şekilde seçimi ve ayarlanması önem arz etmektedir (Akdağ ve Yeroğlu, 2019). Güç sistemlerinin yapıları, hem kalıcı hem de geçici arızalarla karşılaşabilmektedir. Bu arızalar, şebeke ekipmanlarının çalışmasını olumsuz etkileyebilmekte ve hatta yüksek genlikli akımların şebeke ekipmanlarına zarar vermesine neden olabilmektedir. Bu nedenle, güç sistemlerinin yapısında güvenlik önlemlerinin alınması önemlidir (Ortmeyer, 1996). En yaygın koruyucu rölelerden biri aşırı akım rölesidir (Zeineldin vd., 2005).

Aşırı akım röleleri, elektrik dağıtım sistemlerinin ana koruma cihazlarıdır ve güç sistemlerinin arıza durumlarını önlemektedir. Bu röleler, arıza durumlarında çalışarak, sisteme zarar vermemesi için önleyici önlemler almaktadır. ORC, yani birbirleriyle iletişim kurması, geçmişte geleneksel yöntemlerle yapılmaktaydı, daha sonra bilgisayarlı yöntemlerle yapılmaya başlanmıştır (So vd., 2000; Meskin vd., 2015). Geleneksel yöntemler, rölelerin çalışma zamanlarının ve çalışma sırasının belirlenmesi gibi manuel işlemleri gerektirirken, bilgisayarlı yöntemler ise bu işlemlerin bilgisayar ortamında yapılmasını sağlamaktadır. Bu yöntemler sayesinde, ORC daha etkin ve hızlı bir şekilde yapılabilmektedir (So vd., 2000; Acharya, 2022).

ORC, bir elektrik dağıtım sisteminde çok sayıda kısıtın dikkate alınmasını gerektirmektedir (Atsever ve Kocaoğlu, 2022). Bu kısıtlar, rölelerin çalışma zamanları, çalışma sırası, çalışma sıklığı gibi faktörleri içerebilmektedir. Bu sebeple, ORC'nin başarılı bir şekilde gerçekleştirilmesi için çok sayıda kısıttan dahil olduğu bir optimizasyon problemi şeklinde tasarlanması gerekmektedir. Bu optimizasyon problemi, rölelerin çalışma zamanlarını ve çalışma sırasını belirlemeyi amaçlar ve en uygun çözümü bulmayı hedeflemektedir. Bu sayede, ORC daha etkin bir şekilde yapılmaktadır ve sistem güvenliği sağlanmaktadır. Optimum koordinasyon için çeşitli optimizasyon teknikleri önerilmektedir (Noghabi vd., 2009). Meta sezgisel algoritmalar, birbirlerinden farklı yöntemleri birleştirerek optimizasyon problemlerini çözen algoritmalarlardır. Bu algoritmalar, genellikle klasik optimizasyon algoritmalarının çözüm süresini ve/veya çözüm kalitesini artırmak amacıyla kullanılmaktadır.

Bu çalışmada ORC'lerin koordinasyon probleminin çözümü için Geliştirilmiş SGuA (ISGuA) tasarlanmıştır. Elde edilen sonuçlar ile SguA, LCA, GA, WOA ABC, GWO ve BOA'nın performansları ile karşılaştırılmıştır. Yapılan literatür taramasında, algoritmaların bu çalışmalarda kullanıldığı, ancak tamamının kullanıldığı bir ORC problemi çalışmasına rastlanmadığı görülmektedir. Ayrıca, SGuA'yı geliştirerek, ISGuA ile ORC problemi ilk kez çözümlenmiştir. İlerleyen bölümlerde kullanılan materyaller, yöntemler, bulgular ve sonuçlar verilmiştir.

2. KAYNAK BİLDİRİŞLERİ

ORC problemlerinde klasik optimizasyon algoritmalarının yanı sıra meta sezgisel algoritmalar da kullanılmıştır ve bu algoritmaların çözüm süresi ve çözüm kalitesi konusunda olumlu sonuçlar elde edilmiştir. Örneğin Sürekli Genetik Algoritma (Continuous Genetic Algorithm -CGA) (Uthitsunthorn ve Kulworawanichpong, 2010) ve Kaotik Ateşböceği Algoritması (Chaotic Firefly Algorithm - CFA) (Gokhale ve Kale, 2016) gibi meta sezgisel algoritmalar, ORC problemlerinde etkili sonuçlar vermektedir (Atsever ve Hocaoglu, 2022).

Razavi vd. (2008) ORC için GA kullanmışlardır. Sunulan amaç fonksiyon (Objective Function - OF), koordinasyon sorunlarının yanı sıra sürekli veya kesikli zaman ayarlı çarpanların (Time Setting Multiplier - TSM) veya zaman kadranı ayarlarının (Time Dial Setting - TDS) problemlerini de çözebilmektedir. Çalışma iki farklı güç sistemi ağ yapılandırması üzerinde test edilmiştir. Elde edilen sonuçlar, bu yeni yaklaşımın esnek ve başarılı bir yöntem olduğunu göstermektedir.

Alam vd. (2015) yönlü aşırı akım rölelerinin koruma koordinasyon probleminin çözümü için GA, Parçacık Sürü Optimizasyonu (Particle Swarm Optimization - PSO), Harmoni Arama Algoritması (Harmony Search – HS), Diferansiyel Evrim Algoritması (Differential Evolution –DE) ve Arayıcı Optimizasyon Algoritması (Seeker Optimization Algorithm – SOA) gibi beş farklı meta sezgisel algoritmanın performansını incelemişlerdir. Yapılan araştırma DE algoritmasının incelenen beş algoritma arasında en iyi performansı göstermektedir.

Gaikwad vd. (2022) bulut mimarisi için iş planlamasını geliştirmek amacıyla CloudSim simülatöründe bir turnuva şampiyonası stratejisi kullanmışlardır. Pazara sunma süresini azaltmak ve makul fiyatlarla bulut hizmetleri sunmak, bu çalışmanın temel hedefleridir. Verilere göre, en etkili strateji ILCA olarak belirlenmiştir.

Langazane vd. (2022) parçacık sürüsü optimizasyonu ve genetik algoritmaların aşırı akım rölesi duyarlılığı ve hızı üzerindeki etkilerini incelemişlerdir. Çalışma, evrimsel algoritma kontrol parametrelerinin röle performansını etkilediğini göstermiştir. Deneysel sonuçlar, PSO'nun GA'ya göre daha hızlı bir şekilde yakınsadığını ortaya koymuştur.

Rizal vd. (2020) genetik algoritmalar kullanarak gerçekleştirdikleri araştırmaya göre, MATHCAD ile aşırı akım rölesi ayarlarının seçilmesi yöntemi hala ideal değildir ve belirli değişiklikler gerektirmektedir. Bu araştırma, geleneksel optimizasyon stratejileriyle karşılaştırıldığında, aşırı akım rölesi koordinasyon optimizasyonunda bir iyileşme sunduğunu göstermektedir.

Wadood vd. (2019) Yönlü Aşırı Akım Rölelerinin (DOCR) optimal koordinasyon problemini WOA algoritmasını kullanarak değerlendirmişlerdir. Önerilen WOA algoritmasının etkinliğini değerlendirmek amacıyla altı farklı sistem üzerinde uygulanmıştır. WOA yönteminin simülasyon sonuçları, altı problemli modelin başarıyla minimize edildiğini göstermektedir. Bu sonuçlar, önerilen WOA algoritmasının yönlü aşırı akım rölesi koordinasyonu için etkili ve güvenilir bir araç olduğunu doğrulamaktadır. Ayrıca, WOA bulgularının literatürde belirtilen birçok tanınmış ve güncel yöntemi geride bıraktığı belirlenmiştir.

Bedekar vd. (2009) yaptıkları çalışmada dağıtım sistemlerinde aşırı akım rölelerinin optimum koordinasyonu için GA yöntemi kullanılmıştır. Ancak, yeni bir amaç fonksiyonu kullanarak (ceza yöntemiyle) kısıtsız bir optimizasyon sorununa dönüştürülmüş ve ikili değişkenlerin ondalık değerlerine dönüştürülmesi sırasında (haritalama yapılırken), röle çalışma süresi sınırları nedeniyle TMS sınırları, değişken sınırları olarak kullanılmıştır. Her iki örnekte de her röle için minimum çalışma süresi 0.2 s olarak kabul edilmiştir. İllüstrasyon I için CTI değeri 0.57 s olarak alınırken, İllüstrasyon II için 0,6 s olarak alınmıştır. Bu, röle ve kesici özellikleri ile sistem gereksinimlerine bağlı olarak kısıtlamaların oluşturulabileceğini ve en iyi koordinasyonun elde edilebileceğini göstermektedir. Algoritma, çeşitli sistemlerde test edilmiş ve tüm durumlarda tatmin edici sonuçlar vermektedir.

Khurshaid vd. (2019) çalışmalarında, WOA'ya dayalı hibrit meta sezgisel algoritmalar önermiştir. Elde edilen sonuçlar, önerilen HWOA'nın yönlü ORC için etkili ve güvenilir bir araç olduğunu doğrulamaktadır. HWOA kullanılarak elde edilen sonuçlar, WOA ve literatürde belirtilen birçok algorithmadan daha iyi olduğu sonucuna varmaktadır.

Seyyarer vd. (2018) aşırı-akım rölelerinin optimum koordinasyonunu yaparken SGUA ve LCA kullanmışlardır. Çalışmalarında TEİAŞ'ın (Türkiye Elektrik İletim Anonim Şirketi) çalışmalarında, genellikle güç sistemlerindeki ORC'de kullanılan bir

yöntem olarak, faz-faz kısa devre arızalarında ters zamanlı ayar esas alınarak ORC sonuçlarına odaklanılmıştır.

Atsever vd. (2021) ORC için farklı geleneksel yöntemlere ek olarak, Path-Finder Algoritması (Path-Finder Algorithm - PFA) ve Sinir Ağı Algoritması (Neural Network Algorithm - NNA) ilk kez uygulamışlardır. Sonuçlar, PFA ve NNA'nın daha optimal çözümler sağladığını, özellikle PFA'nın en düşük toplam işletme süresine ulaştığını göstermektedir. Ayrıca, Geliştirilmiş Su Döngüsü Algoritması (Enhanced Water Cycle Algorithm - ER-WCA) en hızlı yakınsama süresine sahip algoritma olarak tespit edilmiştir. Çalışma, PFA ve NNA gibi yeni meta sezgisel yöntemlerin aşırı akım röle koordinasyonunda etkili bir optimizasyon tekniği olarak kullanılabileceğini ortaya koymuştur ve gelecekte gerçek dağıtım sistemleri üzerinde uygulanabileceğini ortaya koymaktadır.

3. MATERYAL VE YÖNTEM

Tezin bu bölümünde, meta sezgisel algoritmaların sınıflandırılması, çalışmada kullanılan algoritmalar, ORC probleminin tanımı, ORC probleminin formülasyonu, literatüre yeni kazandırılacak Geliştirilmiş SGUA'ya ve çok modlu kıyaslama fonksiyonlarında yer verilmiştir.

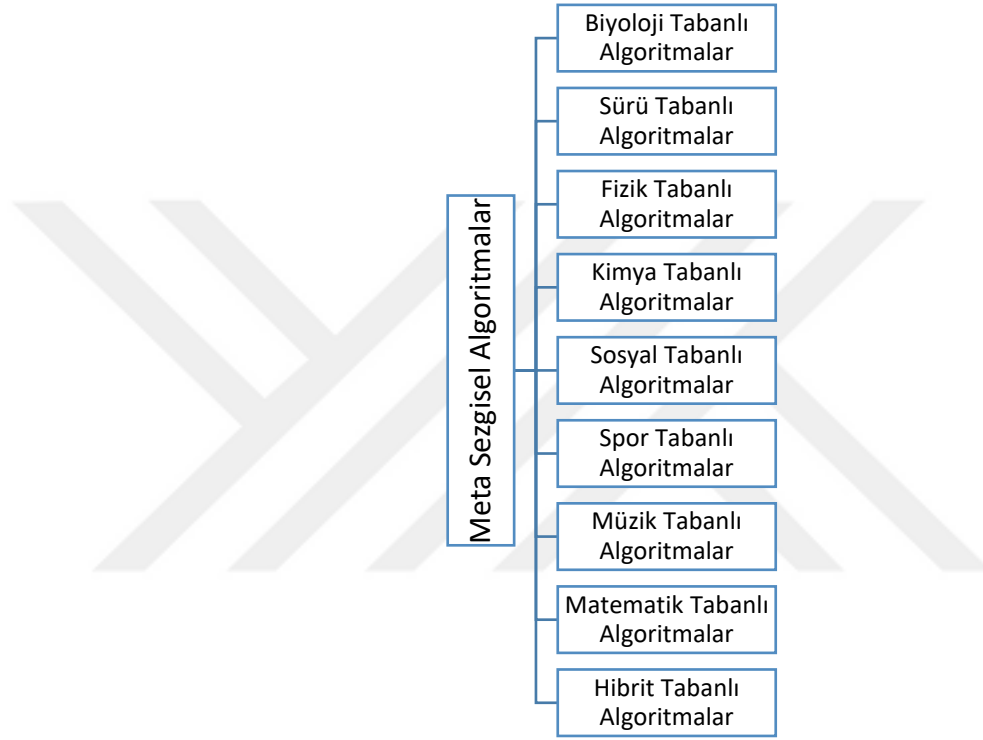
3.1 Meta Sezgisel Algoritmalar

Meta sezgisel algoritmalar, doğal fenomenlerden, biyolojik davranışlardan veya matematikten ilham alan ve problemde bağımsız bir algoritmik çerçeve sağlayan yaklaşımlardır (Abdel-Basset vd., 2018). Kesin yöntemlerden farklı olarak, meta sezgisel algoritmalar, yaklaşık yöntemler kullanarak kabul edilebilir çözümler elde etmeye yönelik tasarlanmıştır. Bu algoritmaların çeşitli avantajları bulunmaktadır. Meta sezgisel algoritmalar büyük ölçekli problemlerin çözümünde etkin olup, hesaplama sürecini hızlandırarak kararlı ve güvenilir sonuçlar üretebilmektedir. Ayrıca basit yapıları, esnek çerçeveleri ve kolay uygulanabilirlikleri sayesinde geniş bir problem yelpazesinde kullanılabilmektedirler. Bu özellikleri nedeniyle, meta sezgisel algoritmalar, geleneksel yöntemler açısından çözümü zorlayıcı olan optimizasyon problemleri için önemli bir alternatif olarak öne çıkmaktadır (Wang vd., 2024). Meta sezgisel algoritmalar birçok probleme uyum sağlayabildiği için geneldir (Blocho, 2020).

Son yirmi yılda, meta sezgisel algoritmalar çeşitli bilimsel alanlardaki karmaşık mühendislik problemlerini çözmek için yaygın olarak kullanılmaktadır. Özellikle, geleneksel matematiksel optimizasyon yaklaşımlarıyla karşılaştırıldığında, meta sezgisel algoritmalar, gradyan gerektirmeyen yapıları ve basit mekanizmaları sayesinde dışbükey olmayan, yüksek derecede doğrusal olmayan, pürüzsüz olmayan ve hatta dinamik gerçek dünya problemlerini ele alırken daha genel bir çözüm sunmaktadır (Li vd., 2020; Su vd., 2023). Bu algoritmalar, çözüm uzayında önceden tanımlanmış kurallara göre arama sürecini başlatmaktadır.

Meta sezgisel algoritmalar, ilham kaynaklarına göre dokuz ana türe ayrılabilir. Bunlar; Biyoloji Tabanlı Algoritmalar, Sürü Tabanlı Algoritmalar, Fizik Tabanlı Algoritmalar, Kimya Tabanlı Algoritmalar, Sosyal Tabanlı Algoritmalar, Spor Tabanlı

Algoritmalar, Müzik Tabanlı Algoritmalar, Matematik Tabanlı Algoritmalar ve Hibrit Tabanlı Algoritmalar'dır (Akyol ve Alataş, 2017). Genel amaçlı meta sezgisel algoritmalar, bu sekiz temel grupta değerlendirilirken, farklı yöntemlerin birleşiminden oluşan hibrit algoritmalar da önemli bir kategori oluşturmaktadır. Şekil 3.1'de meta sezgisel algoritmaların sınıflandırılması verilmiştir.



Şekil 3.1 Meta sezgisel algoritmaların sınıflandırılması (Akyol ve Alataş, 2017)

Biyoloji tabanlı algoritmalar, doğal seçim, kalıtım ve biyolojiyle ilgili diğer evrimsel mekanizmalardan ilham alarak geliştirilen en eski meta sezgisel algoritmalar. Bu kategorideki en popüler algoritmalar, Darwin'in doğal seçim ilkesini simüle eden genetik algoritmadır. GA'da, bir birey grubu rastgele olarak arama uzayında başlatılır ve belirli bir iterasyon sürecinde seçim, mutasyon ve yeniden birleşim gibi çeşitli evrimsel operatörler aracılığıyla evrimleşir. İterasyonların sonunda, en iyi birey optimal çözüm olarak kabul edilir. Diferansiyel evrim (Differential Evolution – DE), GA ile bazı evrimsel operatörleri paylaşan klasik bir biyoloji tabanlı algoritmadır (Zhao vd., 2024). Biyoloji tabanlı bazı algoritmalar Çizelge 3.1'de verilmiştir.

Çizelge 3.1 Biyoloji tabanlı algoritmalar

Algoritmalar	Yıl
Evrin Stratejisi (ES) (Beyer ve Schwefel, 2002)	1973
Genetik Programlama (GP) (Angeline, 1994)	1992
Dilbilimsel Evrim (GE) (Ryan vd., 1998)	1998
Bencil Gen Algoritması (SGA) (Corno vd., 1998)	1998
Yapay Bağışıklık Sistemi (AIS) (Castro ve Timmis, 2003)	2003
Biyocoğrafya Tabanlı Optimizasyon (BBO) (Simon, 2008)	2008

Sürü tabanlı algoritmalar, en hızlı gelişen meta sezgisel algoritmalarından biridir ve doğadaki biyolojik popülasyonların grup davranışlarından, hayvanlardan, bitkilerden ve mikroorganizmalardan ilham alarak geliştirilmiştir (Zhao vd., 2024). Bazı sürü tabanlı algoritmaların optimizasyon üzerinde büyük etkileri bulunmaktadır. Örneğin; Balina optimizasyon algoritması (WOA), kambur balinaların avlarını baloncuk ağlarıyla kuşatma ve avlanma stratejilerini taklit etmektedir (Mirjalili ve Lewis, 2016). Yapay arı kolonisi (ABC) algoritması, arıların koloni içinde besin arama ve paylaşma sürecinde gösterdiği işbirliğini modellemektedir (Karaboga ve Akay, 2009). Gri kurt optimizasyonu (Grey Wolf Optimizer - GWO) algoritması, gri kurtların avlanma ve sosyal hiyerarşi içindeki işbirliği mekanizmalarından ilham alarak geliştirilmiştir (Mirjalili vd., 2014). Kelebek optimizasyon algoritması (BOA), kelebeklerin feromon yayılımı ve koku alma duyularına dayalı işbirlikçi davranışlarını modellemektedir (Arora ve Singh, 2019). Çizelge 3.2’de yaygın olarak kullanılan sürü tabanlı algoritmalar verilmiştir.

Çizelge 3.2 Sürü tabanlı algoritmalar

Algoritmalar	Yıl
Parçacık Sürü Optimizasyonu (PSO) (Kennedy ve Eberhart, 1995)	1995
Karışık Kurbağa Sıçrama Algoritması (SFLA) (Eusuff ve Lansey, 2003)	2003
Ateşböceği Algoritması (FA) (Geem vd., 2001)	2010
Rüzgar Tabanlı Optimizasyon (WDO) (Bayraktar v vd., 2013)	2013
Güve-Alev Optimizasyonu (MFO) (Mirjalili, 2015)	2015
Salp Sürüsü Algoritması (SSA) (Mirjalili vd., 2017)	2017

Çizelge 3.2 Sürü tabanlı algoritmalar (devam)

Algoritmalar	Yıl
Monark Kelebek Optimizasyonu (MBO) (Wang vd., 2019)	2019
Harris Şahin Optimizasyonu (HHO) (Heidari vd., 2019)	2019
Karınca Kolonisi Optimizasyonu (ACO) (Dorigo ve Stützle, 2019)	2019
Tulumba Arama Algoritması (TSA) (Kaur vd., 2020)	2020
Manta Vatozu Beslenme Optimizasyonu (MRFO) (Zhao vd., 2020)	2020
Kapuçin Maymun Arama Algoritması (CapSA) (Braik vd., 2021)	2021
At Sürüsü Optimizasyon Algoritması (HOA) (MiarNaeimi vd., 2021)	2021
Yapay Sinekuşu Algoritması (AHA) (Zhao vd., 2022b)	2022
Cüce Firavun Faresi Optimizasyonu (DMO) (Agushaka vd., 2022a)	2022
Denizati Optimizasyonu (SHO) (Zhao vd., 2022a)	2022
Yapay Tavşan Optimizasyonu (ARO) (Wang vd., 2022c)	2022
Sığırcık Sürüsü Optimizasyonu (SMO) (Zamani vd., 2022)	2022
Beyaz Köpekbaliğı Optimizasyonu (WSO) (Braik vd., 2022)	2022
Ceylan Optimizasyon Algoritması (GOA) (Agushaka vd., 2022b)	2023
Bal Porsuğu Algoritması (HBA) (Hashim vd., 2022)	2022
Koati Optimizasyon Algoritması (COA) (Dehgha vd., 2023)	2023

Fizik tabanlı algoritmalar, meta sezgisel algoritmaların ana dallarından biri olup fiziksel modellerden ilham almaktadır. Bu fiziksel modeller genellikle mekanik, ısı, elektromanyetizma, optik ve atom fiziği gibi çeşitli fizik yasaları, süreçleri, fenomenleri, kavramları ve hareketleri içermektedir (Zhao vd., 2024). Büyük Patlama-Büyük Çöküş (BBBC) algoritması (Erol ve Eksin, 2006), evrenin genişlemesi ve büzülmesi prensiplerine dayanan bir fizik tabanlı meta sezgisel optimizasyon yöntemidir. Young'un Çift Yarık Deneyi (YDSE) optimizasyon algoritması, ışığın dalga doğasından ilham alarak keşif ve sömürü dengesini sağlayan fizik tabanlı bir meta sezgisel yöntemdir (Abdel-Basset vd., 2023). Diğer yaygın fizik tabanlı algoritmalar Çizelge 3.3'te verilmiştir.

Çizelge 3.3 Fizik tabanlı algoritmalar

Algoritmalar	Yıl
Merkezi Kuvvet Optimizasyonu (CFO) (Formato, 2009)	2007
Yerçekimi Arama Algoritması (GSA) (Rashedi vd., 2009)	2009
Ücretli Sistem Araması (CSS) (Kaveh ve Talatahari, 2010)	2010
Su Döngüsü Algoritması (WCA) (Eskandar vd., 2012)	2012
Kara Delik Algoritması (BHA) (Hatamlou, 2013)	2013
Kinetik Gaz Molekülleri Optimizasyonu (KGMO) (Moein ve Logeswaran, 2014)	2014
Çarpışan Cisimlerin Optimizasyonu (CBO) (Kaveh ve Mahdavi, 2014)	2014
Girdap Arama Algoritması (VSA) (Doğan ve Ölmez, 2015)	2015
Optikten İlham Alan Optimizasyon (OIO) (Kashan, 2015)	2015
Su Buharlaştırma Optimizasyonu (WEO) (Kaveh ve Bakhshpoori, 2016)	2016
Termal Değişim Optimizasyonu (TEO) (Kaveh ve Dadras, 2017)	2017
Akış Rejimi Algoritması (FRA) (Tahani ve Babayan, 2019)	2019
Atom Arama Optimizasyonu (ASO) (Zhao vd., 2019)	2019
Denge Optimize Edici (EO) (Faramarzi vd., 2020b)	2020
Lichtenberg Algoritması (LA) (Pereira vd., 2021)	2021
Enerji Vadisi Optimizatörü (EVO) (Azizi vd., 2021)	2021
Özel Görelilik Araması (SRS) (Goodarzimehr vd., 2023)	2023

Kimya tabanlı algoritmalar, genellikle kimyasal kinetik, termodinamik ve kimyasal reaksiyon prensiplerine dayanır. Kimyasal Reaksiyon Optimizasyonu (CRO) (Lam ve Li, 2012), moleküllerin çarpışması sonucu kimyasal reaksiyonların potansiyel enerji yüzeyinde daha düşük ve kararlı bir duruma yönelmesi fenomenine dayanarak geliştirilen temsilci bir kimya tabanlı algoritma yöntemidir. Enerji korunum yasasına göre dört temel çarpışma reaksiyonu modellenmiştir (Zhao vd., 2024). Diğer kimya tabanlı algoritmalar Çizelge 3.4'te listelenmiştir.

Çizelge 3.4 Kimya tabanlı algoritmalar

Algoritmalar	Yıl
Yapay Kimyasal Reaksiyon Optimizasyonu (ACROA) (Alataş, 2011)	2011

Yeni geliştirilen meta sezgisel algoritmalar olarak sosyal tabanlı algoritmalar, literatürde giderek daha fazla ilgi görmektedir. Sosyal tabanlı algoritmalar, iki ana faktörden türetilmiştir: insan davranışları ve fiziksel olmayan sosyal aktiviteler (Zhao vd., 2024). Emperyalist Rekabet Algoritması (ICA), ülkeler arasındaki emperyalist genişleme ve rekabet süreçlerini taklit eden, kolonilerin en güçlü imparatorluklara asimile olması prensibine dayanan bir sosyal tabanlı meta sezgisel optimizasyon yöntemidir (Atashpaz-Gargari ve Lucas, 2007). Savaş Stratejisi Optimizasyonu (WSO), savaş alanındaki askerlerin stratejik hareketlerinden esinlenerek geliştirilen bir meta sezgisel optimizasyon algoritmasıdır. Algoritma, saldırı ve savunma stratejilerini modelleyerek keşif ve sömürü dengesini sağlamayı hedeflemektedir (Ayyarao vd., 2022). Diğer sosyal tabanlı algoritmalar Çizelge 3.5'te listelenmiştir.

Çizelge 3.5 Sosyal tabanlı algoritmalar

Algoritmalar	Yıl
Parlamento Optimizasyon Algoritması (POA) (Borji ve Hamidi, 2009)	2009
Beyin Fırtınası Optimizasyonu (BSO) (Shi, 2011)	2011
Öğretmen-Öğrenci Tabanlı Optimizasyon (TLBO) (Rao vd., 2011)	2011
Toplum ve Medeniyet Algoritması (SCA) (Ray ve Liew, 2003)	2003
İnsan Zihinsel Arama (HMS) (Mousavirad ve Ebrahimpour-Komleh, 2017)	2017
İnsan Davranışına Dayalı Optimizasyon (HBBO) (Ahmadi, 2017)	2017
Fakir ve Zengin Optimizasyonu (PRO) (Moosavi ve Bardsiri, 2019)	2019
Hanedan Optimizasyon Algoritması (DOA) (Wagan ve Shaikh, 2020)	2020
Yığın Tabanlı Optimizasyon (HBO) (Askari vd., 2020a)	2020
İnsan Kentleşme Algoritması (HUA) (Ghasemian vd., 2020)	2020
Öğrenci Psikolojisi Tabanlı Optimizasyon (SPBO) (Das vd., 2020)	2020
Politik Optimizasyon (PO) (Askari vd., 2020b)	2020
Dikiş Eğitimi Tabanlı Optimizasyon (STBO) (Dehghani vd., 2022)	2022
Beceri Optimizasyon Algoritması (SOA) (Givi ve Hubalovska, 2023)	2023

Spor tabanlı algoritmalar, insanlarla ilgili spor etkinlikleri ve fitness programlarından türetilmiştir. Lig Şampiyonası Algoritması (LCA) (Kashan, 2009), spor liglerinde takımlar arasındaki rekabetten ilham alan klasik bir spor tabanlı algoritmadır. Lig Şampiyonluk Algoritması (LCA), spor ligi takımlarının rekabet

mekanizmasından ilham alarak doğrusal olmayan sürekli fonksiyonları optimize etmek için geliştirilmiş bir meta sezgisel algoritmadır. Yapay bir lig içerisinde bireysel spor takımları belirli haftalar boyunca yarışır, her hafta eşleşmeler yaparak karşılaşmalara katılır ve kazanma/kaybetme durumları, belirli bir formasyona dayanan fitness değerleri sonucunda belirlenir. İyileşme döneminde, takımlar bir sonraki hafta için formasyonlarını ve oyun tarzlarını yeni bir çözüm olarak ayarlar; bu süreç, belirlenen durma kriterine ulaşılan kadar sezonlar boyunca devam ederek optimizasyon sürecini sürdürür (Kashan, 2009). Diğer spor tabanlı algoritmalar Çizelge 3.6'da listelenmiştir.

Çizelge 3.6 Spor tabanlı algoritmalar

Algoritmalar	Yıl
Futbol Ligi Yarışması (SLC) (Moosavian ve Roodsari, 2014)	2014
Halat Çekme Optimizasyonu (TWO) (Kaveh ve Zolghadr, 2016)	2016
Dünya Kupası Optimizasyonu (WCO) (Razmjoooy vd., 2016)	2016
Atletizm Koşu Tabanlı Optimizasyon (ARBO) (Barik ve Das, 2021)	2021

Müzik tabanlı algoritmalar, doğadaki müzik ve melodilerden ilham alan yeni bir meta sezgisel algoritma türüdür. Harmoni Arama (Harmony Search - HS) (Geem vd., 2001), müzik tabanlı algoritmalar kategorisinin en bilinen temsilcilerinden biridir ve müzisyenlerin en iyi uyumu bulmak için enstrümanlarının tonlarını doğaçlama yoluyla ayarlamalarına benzer bir süreç izlemektedir. Bazı müzik tabanlı algoritmalar Çizelge 3.7'de gösterilmektedir.

Çizelge 3.7 Müzik tabanlı algoritmalar

Algoritmalar	Yıl
Melodi Arama Algoritması (MSA) (Ashrafi ve Dariane, 2011)	2011
Müzikal Kompozisyon Yöntemi (MMC) (Mora-Gutiérrez vd., 2014)	2014

Matematik tabanlı algoritmalar, meta sezgisel algoritmaların önemli ve hızla gelişen bir alt dalıdır. Bu algoritmalar, belirli matematiksel fonksiyonlardan, kurallardan, formüllerden ve teorilerden türetilmiştir. Bu kategoriye ait iki önemli örnek Sinüs-Kosinüs Algoritması (SCA) (Mirjalili, 2016) ve Aritmetik Optimizasyon

Algoritması (AOA) (Abualigah vd., 2021) olarak öne çıkmaktadır. SCA, matematikteki sinüs ve kosinüs fonksiyonlarının dalgalanma ve periyodikliğini taklit ederken, AOA çarpma, bölme, çıkarma ve toplama gibi dört temel aritmetik işlemin dağılım davranışlarını baz almaktadır. Matematik tabanlı algoritmalar kategorisi henüz diğer meta sezgisel algoritma türleriyle rekabet edecek seviyede olmasa da, gelecek vadeden bir alan olarak değerlendirilmektedir (Zhao vd., 2024). Bu kategoriye ait algoritmalar Çizelge 3.8’de sunulmuştur.

Çizelge 3.8 Matematik tabanlı algoritmalar

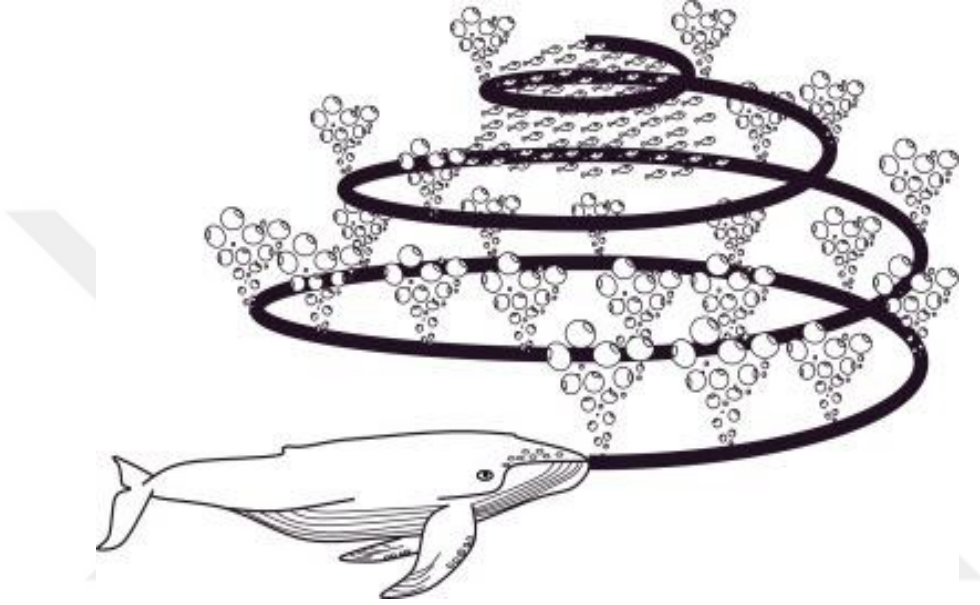
Algoritmalar	Yıl
Taban Optimizasyon Algoritması (BOA) (Salem, 2012)	2012
Altın Sinüs Algoritması (GSA) (Tanyildizi ve Demir, 2017)	2017
Lévy Uçuş Dağılımı (LFD) (Houssein vd., 2020)	2020
Çember Arama Algoritması (CSA) (Qais vd., 2022)	2022

Hibrit tabanlı algoritmalar, iki veya daha fazla farklı optimizasyon yöntemini birleştirerek daha iyi performans ve sonuçlar elde etmeyi amaçlayan optimizasyon algoritmalarıdır. Her yöntemin avantajlarından yararlanırken, sınırlamalarını minimize ederek, tek bir optimizasyon yöntemiyle çözülemeyen karmaşık problemleri etkili bir şekilde çözebilmektedir. WOAGWO (Mohammed ve Rashid, 2020), GWO'nun avlanma stratejisinin Balina Optimizasyon Algoritması (WOA)'nın sömürü sürecine entegre edilmesiyle oluşturulan tipik bir hibrit algoritmadır ve her iki yöntemin eksikliklerini telafi edebilmektedir. Diğer hibrit algoritmalar arasında Hibrit ACO-ABC Algoritması (HACO-ABCA) (Comert ve Yazgan, 2023), Gelişmiş CS ve Uyarlanabilir Gauss Kuantum Davranışlı PSO Tabanlı Hibrit Algoritma (AGQPSO) (Kumar vd., 2021) ve Hibrit Karınca Kolonisi ile ABC Algoritması (AC-ABC) (Shunmugapriya ve Kanmani, 2017) gibi birçok örnek bulunmaktadır (Zhao vd., 2024).

3.1.1 Balina Optimizasyon Algoritması

Mirjalili ve Lewis tarafından 2016 yılında önerilen WOA, kambur balinaların balık avlama stratejilerine dayalı olarak geliştirilen bir algoritmadır. Kambur balinalar

genellikle su yüzeyine yakın küçük balık sürüleriyle beslenmektedirler ve yaklaştıklarında balıkların bir arada kalmasını sağlama amacıyla hava kabarcıkları oluşturmaktadırlar. Bu durum balıklara fark edilmeden yaklaşabilmelerine olanak tanımaktadır (Goldbogen vd., 2013). Kambur balinaların avlanma stratejisi Şekil 3.2'de gösterilmiştir (Mirjalili ve Lewis, 2016).



Şekil 3.2 Kambur balinaların avlanma stratejisi (Mirjalili ve Lewis, 2016)

WOA'da bu avlanma yöntemine göre matematiksel denklemler oluşturulmuştur. t , mevcut iterasyonu, A^+ ve C^+ katsayı vektörlerini ve X^* en iyi çözüm vektörlerini temsil etmektedir. a iterasyon boyunca 2'den 0'a doğrusal olarak düşürülür ve r ise $[0,1]$ arasında rastgele vektör değişkenlerini içermektedir (Mirjalili ve Lewis, 2016).

WOA'da 3 ana kavram vardır. Bunlar;

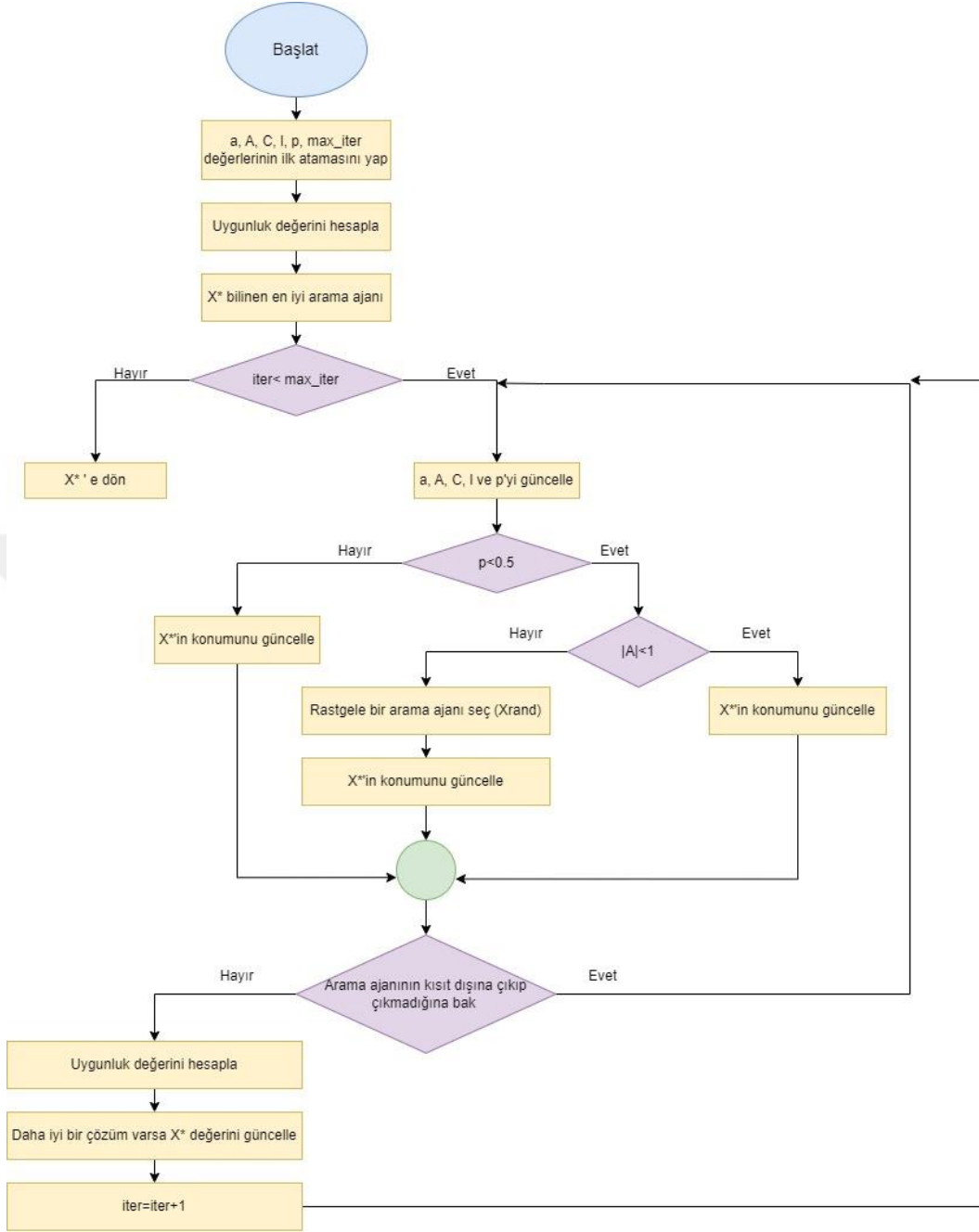
- Avı kuşatma
- Bubble-net saldırı yöntemi (istismar aşaması)
- Av arama (keşif aşaması)

Çizelge 3.9'da WOA yönteminin sözde kodu verilmektedir.

Çizelge 3.9 Balina optimizasyon algoritmasının sözde kodu

1. a, A, C, l, p gibi kontrol parametreleri ile maksimum iterasyon sayısı ve popülasyon büyüklüğü tanımlanır ve başlangıç popülasyonu rastgele şekilde oluşturulur.
2. for: her bir arama ajanı için
3. Uygunluk değerini hesapla
4. end
5. X^* =Bilinen en iyi arama ajanı
6. t=0
7. while (t< maksimum iterasyon sayısı)
8. for: her bir arama ajanı için
9. if (p<0.5)
10. if ($|A| < 1$)
11. else if ($|A| \geq 1$)
12. Rastgele bir arama ajanı seç (Xrand)
13. Arama ajanının konumunu güncelle.
14. end if
15. else if (p $\geq$ 0.5)
16. Arama ajanının konumunu güncelle.
17. end if
18. end for
19. Arama ajanının kısıt dışına çıkıp çıkmadığına bak çıkıyorsa sınır değerini ver
20. Her bir arama ajanının uygunluk değerini hesapla, daha iyi bir çözüm varsa X^* değerini güncelle.
21. t=t+1
22. end while
23. X^* değerini döndür

Şekil 3.3'te WOA'nın akış diyagramı yer almaktadır.



Şekil 3.3 Balina optimizasyon algoritmasının akış diyagramı

3.1.2 Genetik Algoritma

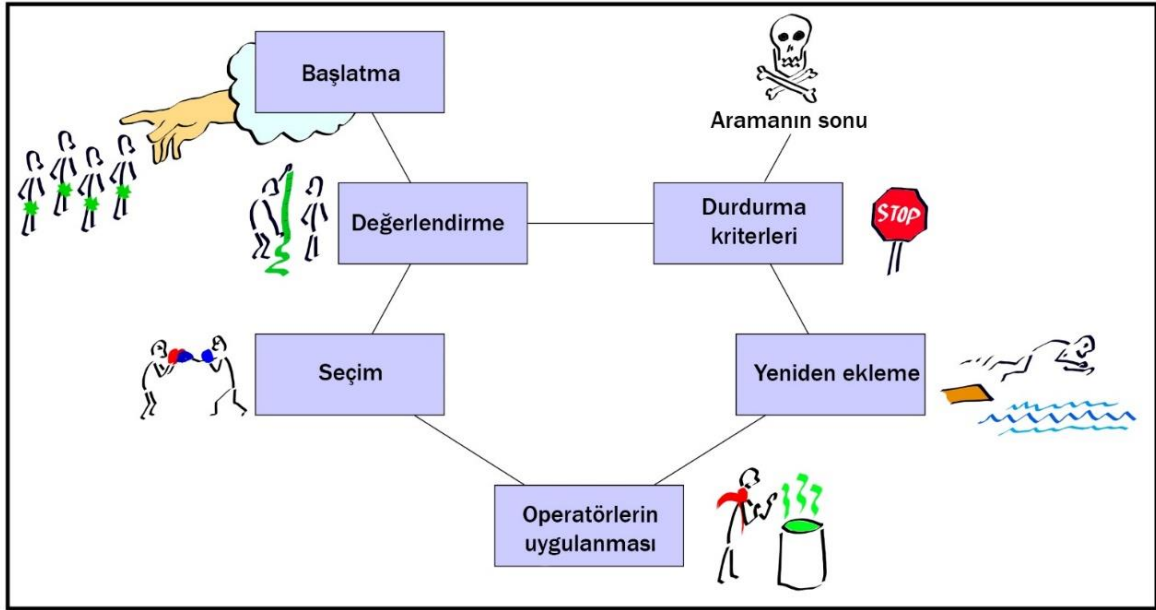
GA'lar, evrimden ilham alan bir bilgisayar modeli ailesi olarak tanımlanabilmektedir. Bu algoritmalar, belirli bir problemin potansiyel çözümünü basit bir kromozom benzeri veri yapısında kodlar ve bu yapılar üzerinde rekombinasyon operatörleri uygularlar ve kritik bilgileri korumak için gerekli adımları atmaktadırlar.

GA'nın uygulaması, genellikle rastgele olan kromozomların bir popülasyonu ile başlamaktadır. Daha sonra, bu yapılar değerlendirilir ve hedef problemin daha iyi bir çözümünü temsil eden kromozomların, daha kötü çözümlerden daha fazla 'üreme' fırsatı verildiği şekilde üreme fırsatları tahsis edilir. Bir çözümün 'iyiliği' genellikle mevcut popülasyon ile ilgili olarak tanımlanmaktadır (Mathew, 2012).

GA'da dört ana kavram vardır. Bunlar;

- İlk Değer Atama
- Seçim
- Çoğaltma
- Sonlandırma

GA'nın genel şeması Şekil 3.4'te gösterilmiştir (Keshav ve Sharma, 2022).



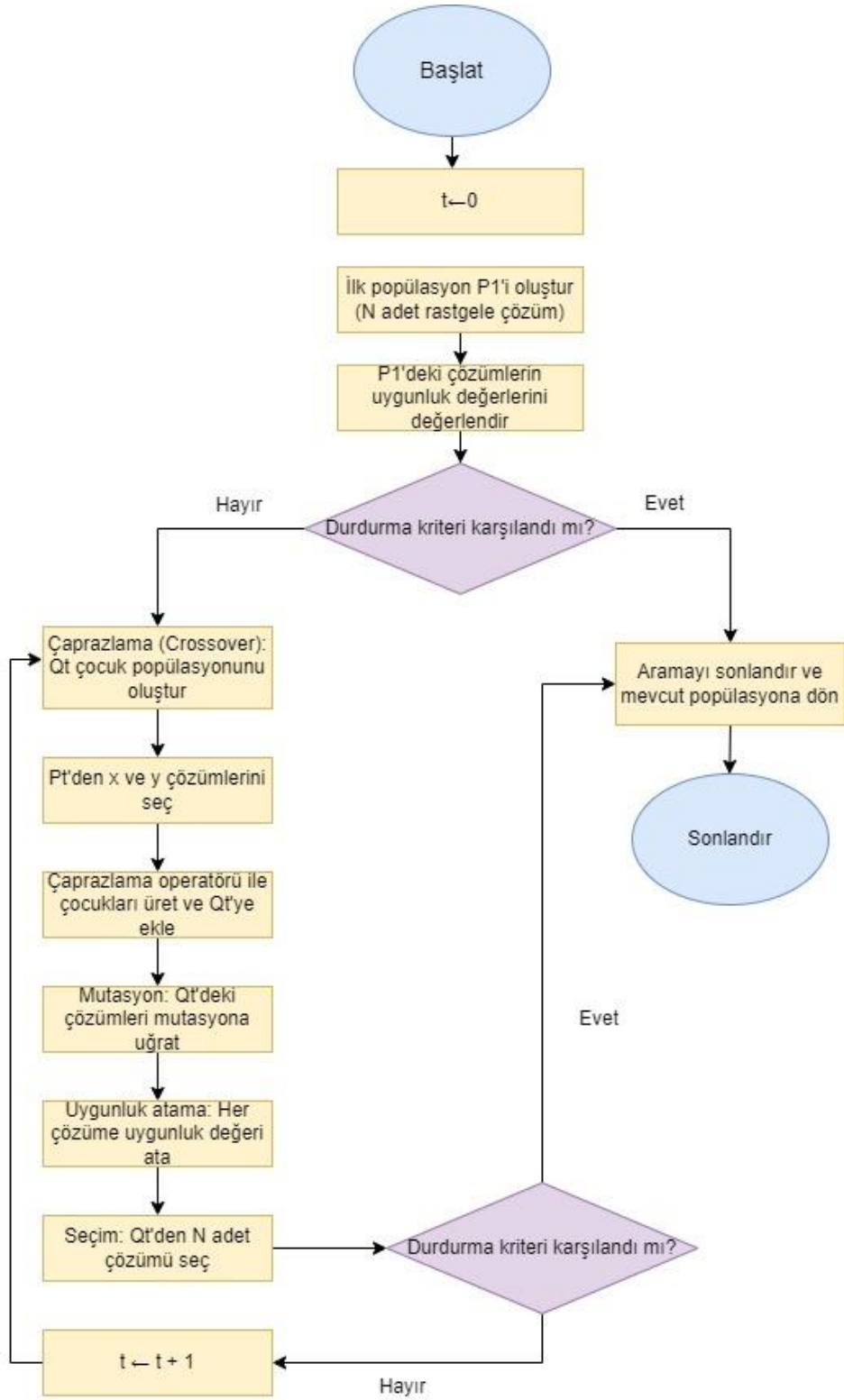
Şekil 3.4 Genetik algoritmanın genel şeması (Keshav ve Sharma, 2022)

Bu kuşaksal süreç, sonlandırma koşulu karşılanana kadar tekrarlanmaktadır (Kumar vd., 2010). Çizelge 3.10'da GA yönteminin sözde kodu verilmektedir.

Çizelge 3.10 Genetik algoritmanın sözde kodu

-
1. $t=1$ olarak başlat. İlk populasyon P_1 'i oluşturmak için N adet çözümü rastgele üret. P_1 'deki çözümlerin uygunluk değerlerini değerlendir.
 2. Çaprazlama (Crossover): Aşağıdaki adımları takip ederek Q_t adında bir çocuk popülasyonu oluştur:
 3. P_t 'den uygunluk değerlerine dayanarak x ve y adlı iki çözümü seç.
 4. Bir çaprazlama operatörü kullanarak çocukları üret ve Q_t 'ye ekle.
 5. Mutasyon: Q_t içindeki her bir çözümü, önceden belirlenmiş bir mutasyon oranıyla mutasyona uğrat.
 6. Uygunluk atama: Her bir çözüme, hedef fonksiyon değeri ve uygunsuzluk durumuna bağlı olarak bir uygunluk değeri ata.
 7. Seçim: Q_t 'deki çözümler arasından uygunluk değerlerine dayanarak N adet çözümü seç ve P_{t+1} 'e kopyala.
 8. Durdurma kriteri karşılanırsa, aramayı sonlandır ve mevcut popülasyona dön, aksi takdirde t 'yi bir artır ve Adım 2'ye geç.
-

Şekil 3.5'te GA'nın akış diyagramı yer almaktadır.

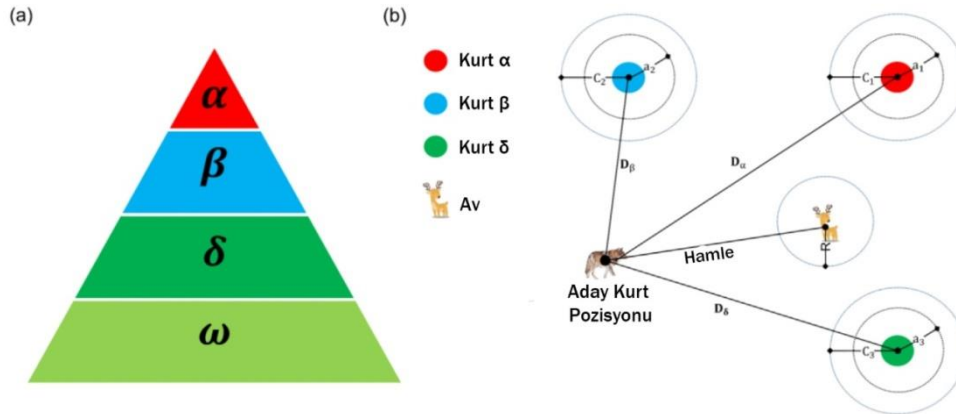


Şekil 3.5 Genetik algoritmanın akış diyagramı

3.1.3 Gri Kurt Algoritması

Gri Kurt Optimizasyon Algoritması (GWO), doğadaki gri kurtların avlanma davranışları ve sosyal hiyerarşi yapısından ilham alınarak tasarlanmış bir meta-sezgisel optimizasyon yöntemidir (Mirjalili vd., 2014; Mirjalili, 2015). Sürü içindeki bireyler, karar verme sürecinde hiyerarşik olarak alfa, beta, delta ve omega rollerine göre konumlanmaktadır (Şekil 3.6) (Yuan vd., 2022).

Gri kurt sürüsünde en yüksek otoriteye sahip bireyler alfa olarak adlandırılmaktadır ve tüm sürü davranışları üzerinde doğrudan belirleyici rol oynamaktadırlar. Beta kurtlar, alfaların en yakın yardımcıları olup, hem diğer bireylerle iletişimi sağlar hem de sürü düzeninin korunmasında aktif rol üstlenmektedir. Hiyerarşinin en alt basamağında yer alan omega kurtlar, genellikle baskıya en fazla maruz kalan bireylerdir ve av sonrasında beslenme sırasının en sonuna bırakılmaktadırlar. Delta kurtlar ise, beta ve alfa tarafından belirlenen, ancak omega seviyesinden daha yüksek konumda olan kurtlardır. Bu sınıf, doğrudan liderlik yapmasa da, sürü içi düzenin sağlanmasında destekleyici bir ara kademeyi temsil etmektedir (Tunç, 2016).



Şekil 3.6 Gri kurt optimizasyon algoritması hiyerarşi zinciri ve avlanma mekanizmasının diyagramı (Yuan vd., 2022)

Avlanma

Gri kurtlar üç aşamada avlanırlar. Bunlar;

- İzleme, takip ve ava yaklaşma

- Avı çevreleme ve av yorulup duruncaya kadar avı hareket ettirme
- Ava saldırma

GWO'da alfa kurtlar mevcut en iyi çözümü, beta ve delta kurtlar ise sırasıyla ikinci ve üçüncü en iyi çözümleri temsil etmektedir. Diğer tüm bireyler, yani omega kurtlar, potansiyel aday çözümler olarak değerlendirilmektedir (Tunç, 2016).

Arama

GWO algoritmasında, arama süreci başlangıçta rastgele oluşturulmuş bir popülasyonla başlamaktadır. Her bireyin uygunluk değeri tanımlanan maliyet fonksiyonuna göre değerlendirilmektedir. Uygunluk değeri en yüksek olan üç birey sırasıyla alfa, beta ve delta rollerine atanmaktadır. Avın yerini bulma ve yönlendirme görevi öncelikle alfa kurdu tarafından üstlenilmektedir. Ancak beta ve delta kurtları da sürece katkıda bulunur ve grup içindeki rehberliği desteklemektedir. Optimizasyon süreci sırasında bu en iyi üç birey her yinelemede güncellenerek algoritmanın çözüme daha hızlı ve daha verimli şekilde ilerlemesi sağlanmaktadır (Tunç, 2016).

Saldırma

GWO'da, avın konumu tespit edildikten sonra saldırı aşamasına geçilmektedir. Gerçek doğadaki davranışa benzer şekilde, bu saldırı, avın direnç kaybederek hareketini durdurduğu noktada gerçekleşmektedir. Algoritma bağlamında bu süreç, hedef çözüme yakınsama evresini temsil etmektedir. Arama ve saldırı (keşif ve sömürü) işlemleri, belirlenen maksimum iterasyon sayısına ulaşıncaya veya önceden tanımlanmış bir durdurma kriteri karşılanıncaya dek sürdürülür. (Tunç, 2016). Çizelge 3.11'de Gri kurt optimizasyon algoritmasının sözde kodu verilmiştir.

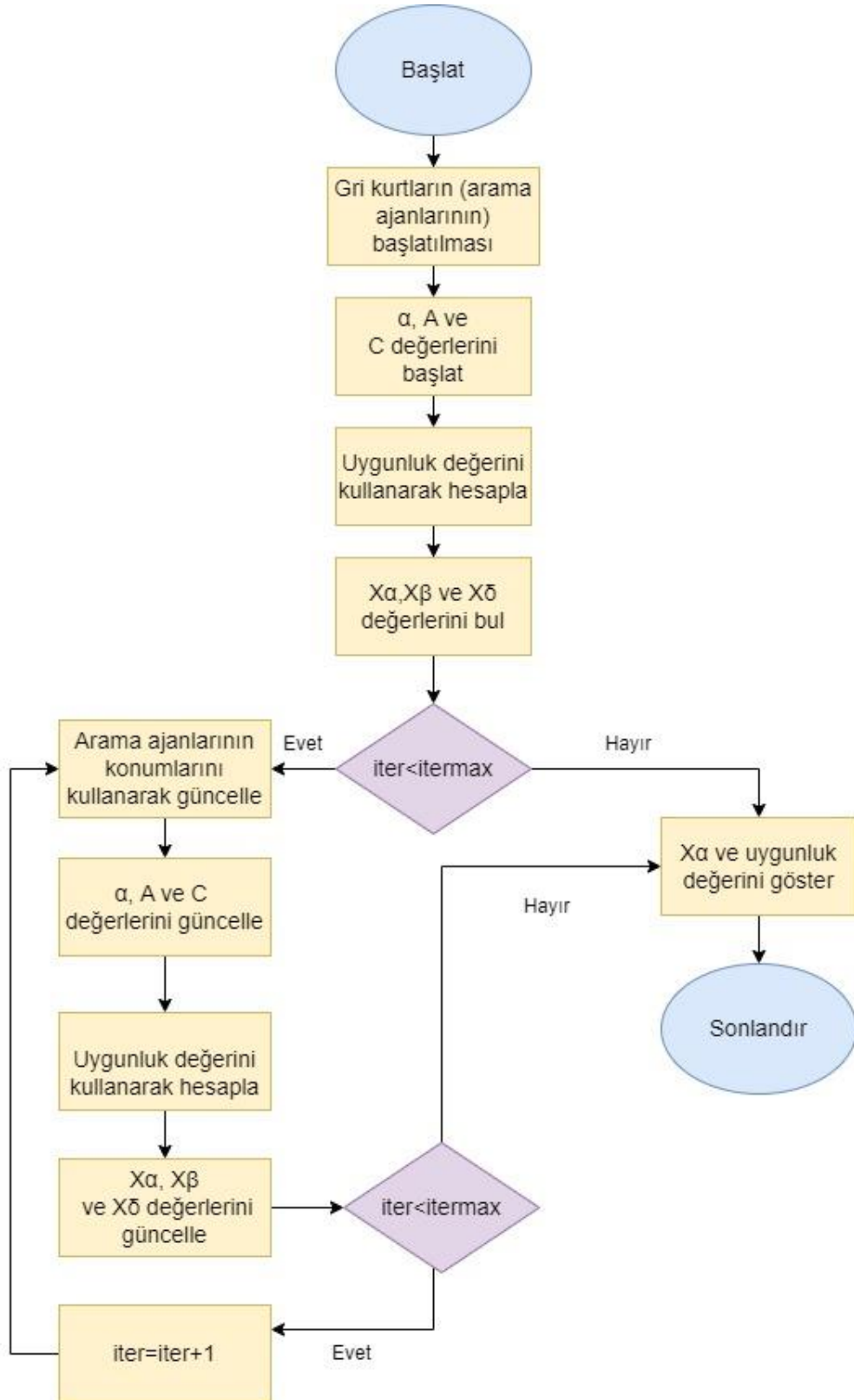
Çizelge 3.11 Gri kurt optimizasyon algoritmasının sözde kodu

-
1. Gri Kurt Populasyonunu Başlat $X_i = (i=1,2,...,n)$
 2. a, A C parametrelerini ata
 3. Her bir ajanın uygunluk değerini hesapla
 4. $X\alpha$, $X\beta$, ve $X\delta$ değerlerini bul
 5. $X\alpha$ = Populasyondaki en iyi konuma sahip ajan
 6. $X\beta$ = Populasyondaki en iyi ikinci konuma sahip ajan
 7. $X\delta$ = Populasyondaki en iyi üçüncü konuma sahip ajan
 8. while ($t < \text{Maksimum İterasyon sayısı}$)
-

Çizelge 3.11 Gri kurt optimizasyon algoritmasının sözde kodu (devam)

9. for Her bir ajan
 10. Mevcut arama ajanların konumlarını güncelle.
 11. end for
 12. a, A ve C parametlerini güncelle
 13. Her bir ajanın uygunluk değerini hesapla
 14. $X\alpha$, $X\beta$ ve $X\delta$ parametlerini hesapla
 15. $t = t+1$
 16. end while
 17. return $X\alpha$
-

Şekil 3.7’de GWO’nun akış diyagramı yer almaktadır.

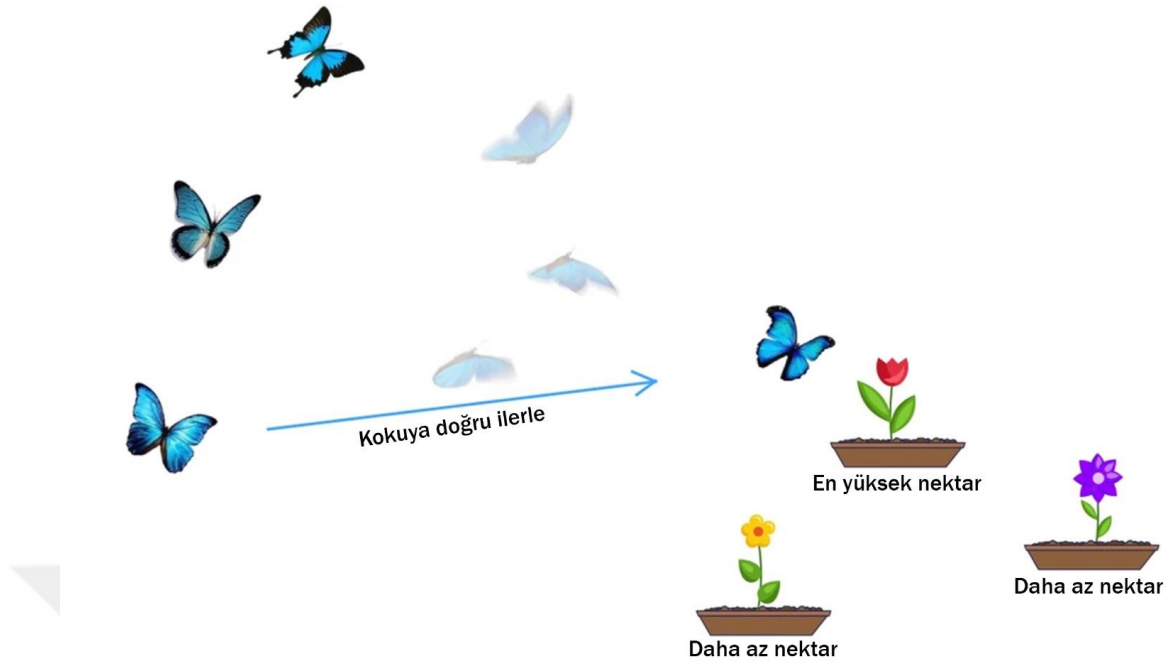


Şekil 3.7 Gri kurt optimizasyon algoritmasının akış diyagramı

3.1.4 Kelebek Algoritması

Kelebek optimizasyon algoritması (Butterfly optimization algorithm (BOA)) kelebeklerin yiyecek arama ve çiftleşme davranışını taklit eden popülasyon tabanlı meta sezgisel algoritmadır. BOA'daki esas motivasyon, kelebeklerin yiyecek arama ve çiftleşme davranışlarını taklit etmektir (Arora ve Singh, 2019). BOA'da bütün kelebekler, arama uzayında optimizasyon işlemini gerçekleştirmek için arama ajanı olarak kabul edilir. Kelebekler, hem besin kaynaklarını tespit etmek hem de eş bulmak amacıyla büyük ölçüde koku alma duyularına dayanırlar. BOA'nın diğer meta sezgisel algoritmalarından temel farklarından birisi her kelebeğin kendine özgü kokuya sahip olmasıdır. Kelebekler, farklı koku yoğunluklarını çok hassas algılayabilen ve sınıflandırabilen canlılardır. Her bir kelebek, gıda kalitesinin uygunluğu veya çiftleşme hakkında bilgi vermek için diğer kelebeklere belirli yoğunlukta koku yaymaktadır. Yiyecekler, çözüme karşılık gelen arama uzayındaki bir konumla da ilgilidir. Her kelebek, koku yayarak gıda kalitesinin uygunluğunu paylaşabilir ve gıdanın bulunduğu konumu etki alanındaki diğer kelebekler tarafından herhangi bir yerden algılanabilir. Böylelikle sosyal bir ağ kurulmuş olmaktadır (Arora ve Singh, 2019).

Kelebekler genel olarak koku kaynağının yerini doğru bir şekilde tespit edebilmektedir ve farklı kokuları ayırt edebilmektedirler. Kelebekler bulundukları yerden daha fazla nektarla başka yerlere hareket etmektedir. Kelebekler bulundukları konumu ve kişisel bilgilerini diğer kelebeklerle paylaşmak için hareket ederken koku üretirler. Daha sonra kelebekler, daha fazla nektar içeren daha iyi bir yer bulmak için üretilen kokuyu takip edeceklerdir. Kelebeklerin genel davranışı Şekil 3.8'de gösterilmektedir (Makhadmeh vd., 2023).



Şekil 3.8 Kelebeklerin hareket davranışı (Makhadmeh vd., 2023)

BOA'da üç parametreye bağlı temel bir algılama kavramı vardır: duyuşal modalite (c), uyarıcı yoğunluğu (I) ve güç üssü (a). Duyuşal modalitesi, uyarıcı tarafından enerji gücünü ölçer. BOA'da modalite kokudur. Uyarıcı yoğunluğu (I), uyarıcının büyüklüğünü gösterir ve arama uzayındaki çözümlerin amaç değeri ile doğrudan bağlantılıdır. Amaç değeri daha yüksek (koku) olan bir kelebek varsa, çevresindeki diğer kelebeklerin bunu hissedebileceği ve ona çekileceği anlamına gelir. Kelebekler yiyecek ve çiftleşme ortağı aranması, hem lokal hem de global aramada gerçekleşebilir. Eş durumu ve yağmur, rüzgar gibi çeşitli koku almayı zorlaştıran diğer durumlar göz önüne alındığında, arama faaliyetlerinde önemli bir p payı olabilir. Bu nedenle, BOA'da global arama ile yerel arama arasında geçiş yapmak için bir geçiş olasılığı p kullanılır (Arora ve Singh, 2019). Çizelge 3.12'de Kelebek optimizasyon algoritmasının sözde kodu verilmiştir.

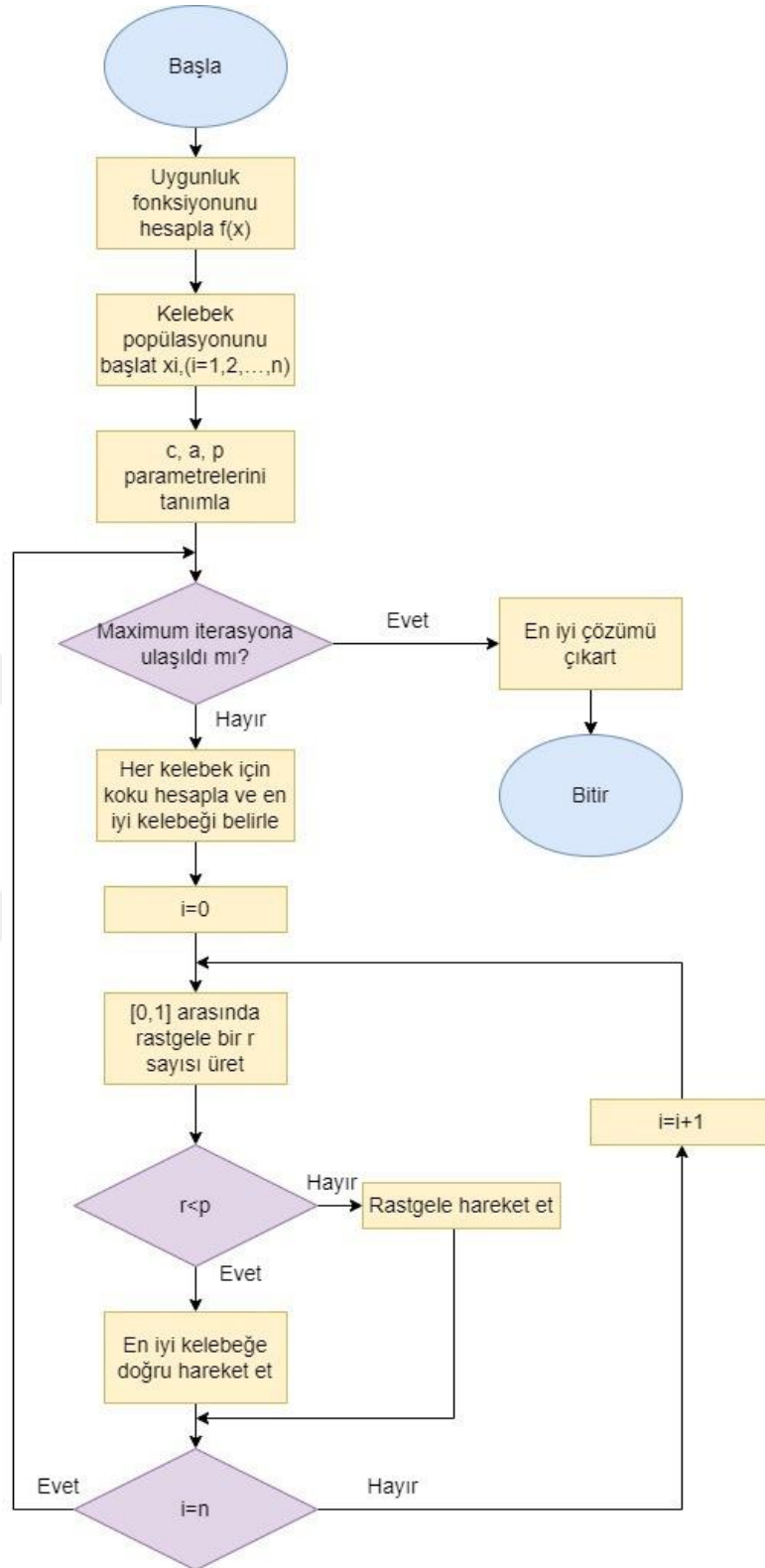
Çizelge 3.12 Kelebek optimizasyon algoritmasının sözde kodu

-
1. Uygunluk fonksiyonu $f(x)$, $x=(x_1, x_2, \dots, x_{dim})$, dim =boyut parametrelerini belirle
 2. n kelebeğin başlangıç popülasyonu oluştur $x_i=(i=1, 2, \dots, n)$
 3. x_i 'deki uyarıcı yoğunluğu I_i , $f(x_i)$ kullanarak belirle
-

Çizelge 3.12 Kelebek optimizasyon algoritmasının sözde kodu (devam)

4. Sensör modalitesi c 'yi, güç katsayısı a 'yı ve değiştirme (anahtarlama) olasılığı p 'yi tanımla
 5. while durdurma kriteri karşılanmadı do
 6. for each popülasyondaki her bf kelebeği için do
 7. Denklem 1 kullanarak bf için koku hesapla
 8. end for
 9. En iyi bf^* 'yi bul
 10. for popülasyondaki her bf kelebeği için do
 11. $[0,1]$ arasında rastgele bir r sayısı üret
 12. if $r < p$ then
 13. Denklem 2'yi kullanarak en iyi kelebeğe/çözüme doğru ilerle
 14. else
 15. Denklem 3'ü kullanarak rastgele hareket et
 16. end if
 17. end for
 18. a 'nın değerini güncelle
 19. end while
-

Şekil 3.9'da BOA'nın akış diyagramı yer almaktadır.



Şekil 3.9 Kelebek optimizasyon algoritmasının akış diyagramı

3.1.5 Lig Şampiyonluk Algoritması

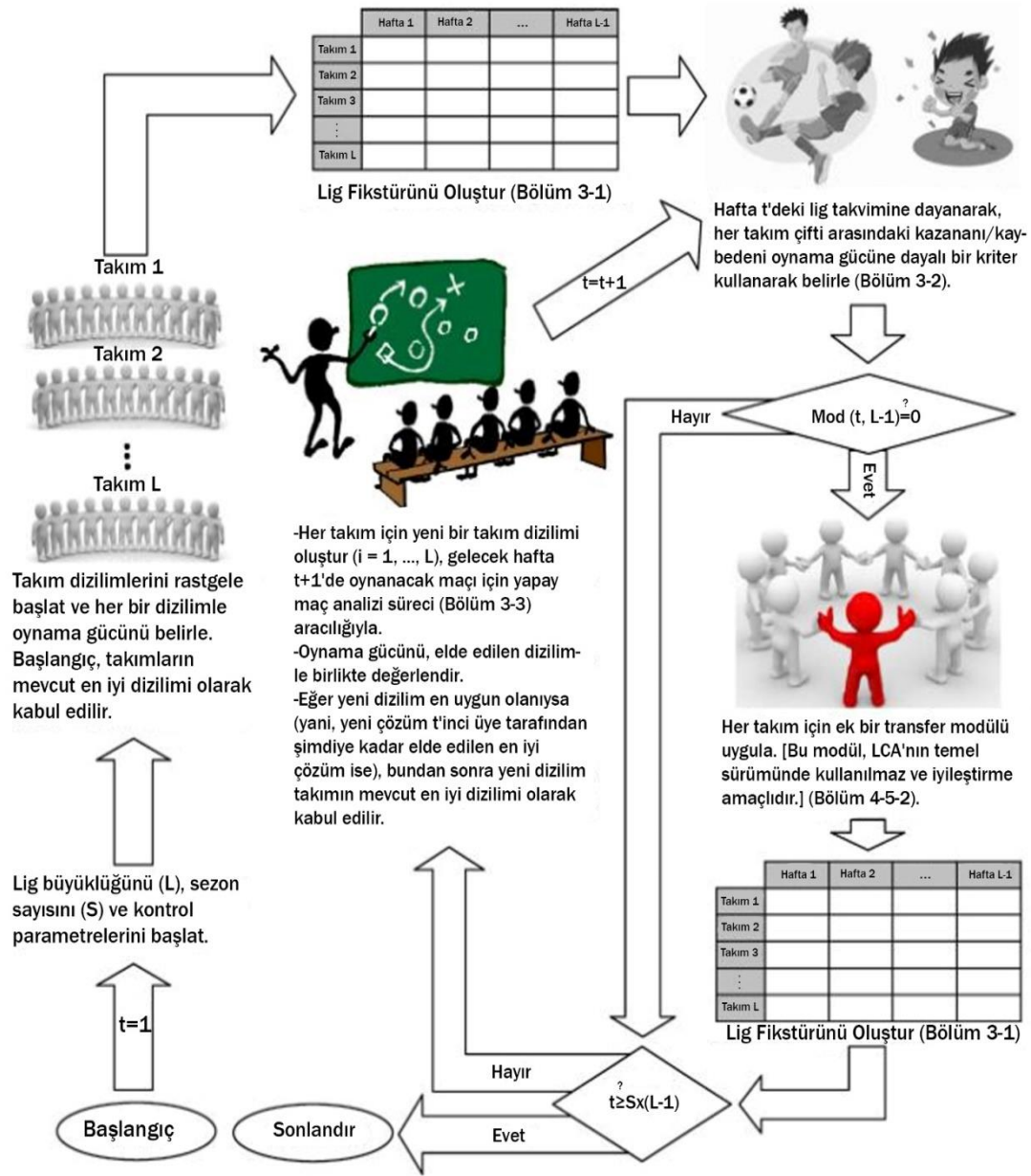
Kashan tarafından önerilen LCA için öncelikle takım sporlarıyla ilgili terimlere göz atmak gerekmektedir. Spor ligleri, belirli sayıda takımın rekabet ettiği organizasyonlardır ve genellikle takım sporlarına yöneliktir. Lig şampiyonluğuna ise farklı kriterler göz önünde bulundurularak itiraz edilebilmektedir. Takımlar, diğer takımlarla belirli sayıda maç yaparak en iyi rekora sahip olmaya çalışmaktadır. Bunun için sıkı bir kazan-kaybet-eşitlik sistemi veya belirli sayıda puan verilen bir puanlama sistemi kullanılmaktadır, böylece kazanma, kayıp veya eşitlik durumları belirlenebilmektedir. Bazı durumlarda ise, belirli kriterleri karşılayan takımlara bonus puanlar da verilebilmektedir (Kashan, 2009; Seyyarer vd., 2017).

LCA diğer doğa esinli algoritmalara benzer şekilde, bir çözümler popülasyonunun optimal çözüme evrimleşmesi şeklinde çalışmaktadır. Ligdeki her takım (birey), çözülecek problem için geçerli bir çözüm olup, n oyuncudan oluşur ve bu, değişken sayısına karşılık gelmektedir. Yapay haftalık lig programına göre takım i , oynama gücü (uygunluğu) ile ilgili olan takım j 'ye karşı oynamaktadır. Lig programına dayanarak, bahsi geçen takımlar, $S \times (L - 1)$ hafta boyunca çiftler halinde rekabet ederler, burada S mevsim sayısıdır ve hafta t olarak belirtilmektedir. Kazanan veya kaybeden takımı belirlemek için kullanılmaktadır. Sonraki maça hazırlanmak için, her takım önceki haftanın maç sonuçlarını kontrol etmektedir ve bu bilgiyi yeni takımını oluşturmak için kullanılmaktadır. Beklenen verimli takım oluşumu, şimdiki en iyi takımdan daha iyi oynama gücüne sahip takım oluşumundan rehberlik olarak seçilmektedir (Kashan, 2009; Wangchamhan vd., 2017).

LCA'da üç ana kavram vardır. Bunlar;

- Lig programının nasıl oluşturulacağı.
- Kazanan veya kaybeden takımın nasıl belirleneceği.
- Yeni bir takımın nasıl oluşturulacağı.

LCA'nın temel aşamalarını göstermek için Şekil 3.10'daki gibi şematik bir akış şeması kullanılabilir (Kashan, 2014).



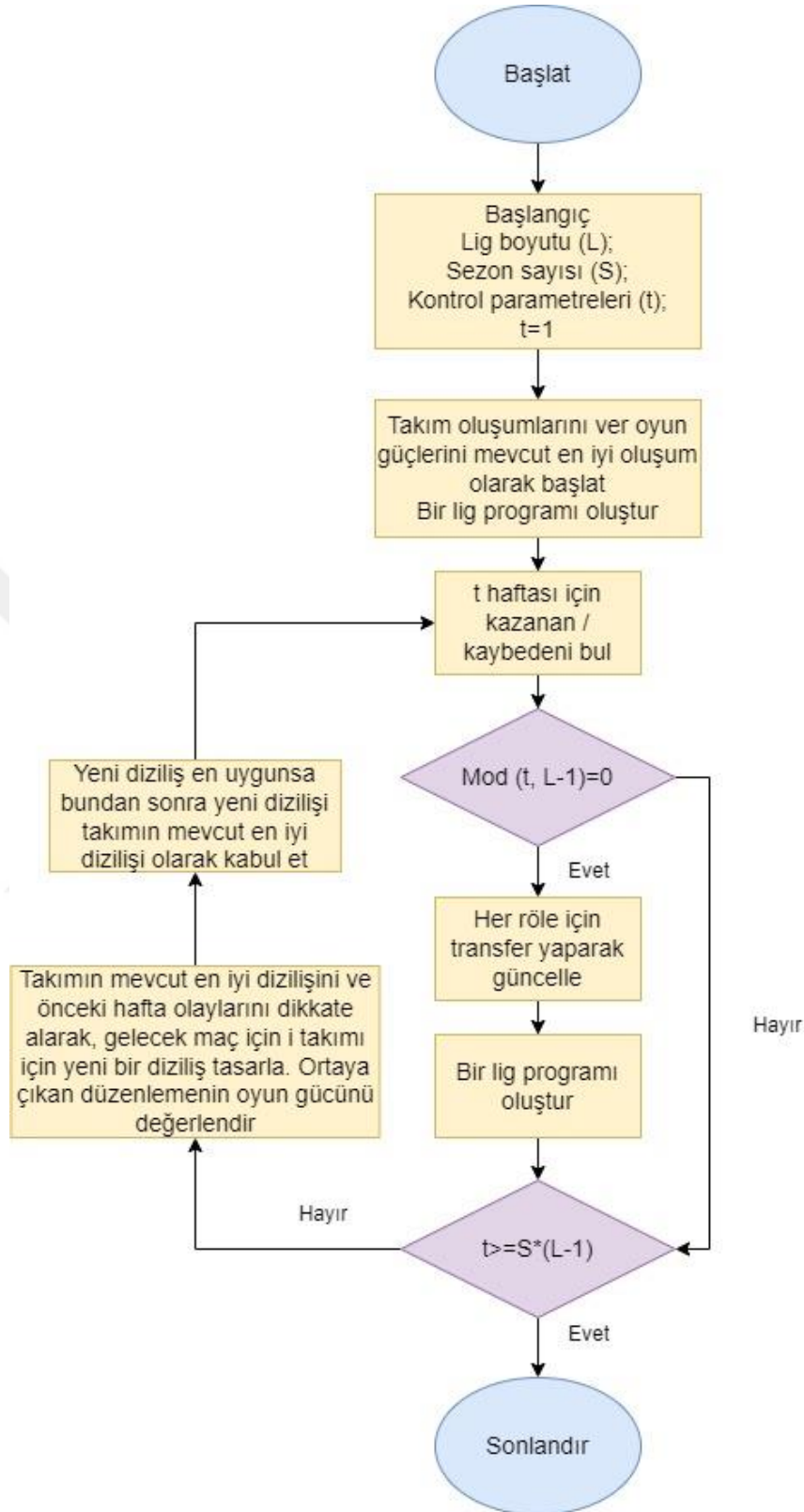
Şekil 3.10 Lig şampiyonluk algoritmasının akış şeması (Kashan, 2014)

Çizelge 3.13'te LCA yönteminin sözde kodu verilmektedir.

Çizelge 3.13 Lig şampiyonluk algoritmasının sözde kodu

-
1. Lig boyutunu (L) ve sezon sayısını (S) başlat; $t=1$;
 2. Bir lig programı oluştur;
 3. Takım oluşumlarını başlat (L çözümlerinden oluşan bir popülasyon oluştur) ve onlarla birlikte oynama güçlerini (işlev veya uygunluk değeri) belirle. Başlatmanın aynı zamanda takımların mevcut en iyi oluşumu olmasına izin ver;
 4. While $t \leq S.(L-1)$
 5. t haftasındaki lig programına göre, oyun gücüne dayalı bir kriter kullanarak her takım çifti arasında kazananı/kaybedeni belirle;
 6. $t=t+1$;
 7. For $i=1$ to L
 8. Takımın mevcut en iyi dizilişini ve önceki hafta olaylarını dikkate alarak, gelecek maç için i takımı için yeni bir diziliş tasarla. Ortaya çıkan düzenlemenin oyun gücünü değerlendir;
 9. Yeni diziliş en uygunsa (yani yeni çözüm, popülasyonun i. üyesi için şimdiye kadar elde edilen en iyi çözümse), bundan sonra yeni dizilişi takımın mevcut en iyi dizilişi olarak kabul et;
 10. End for
 11. If $\text{mod}(t, L-1)=0$
 12. Bir lig programı oluştur;
 13. End if
 14. End while
-

Şekil 3.11’de LCA’nın akış diyagramı yer almaktadır.



Şekil 3.11 Lig şampiyonluk algoritmasının akış diyagramı

3.1.6 Yapay Arı Algoritması

Sezgisel algoritmalar, özellikle çözüm uzayı geniş ve hesaplama açısından karmaşık olan problemlerde, makul sürelerde optimuma yakın çözümler üretebilen yöntemlerdir. Yakınsama eğilimleri bulunsada sezgisel yapıları nedeniyle kesin çözüme ulaşma garantisi vermemektedir (Karaboğa, 2005). Bu algoritmaların bir alt kategorisi olan sürü zekâsı tabanlı yöntemler, karınca, arı, kuş, balık ve termit gibi sosyal organizasyonlara sahip canlıların kolektif davranış modellerinden esinlenerek geliştirilmiştir (Karaboğa ve Akay, 2007).

Sürü zekâsı temelli algoritmalarından biri olan ABC algoritması, son yıllarda çeşitli optimizasyon problemlerinde yaygın olarak kullanılmaktadır. Karaboğa (2005) tarafından geliştirilen ABC algoritması, çok boyutlu ve çok modlu optimizasyon problemlerine arıların doğal ortamda yiyecek arama davranışları taklit edilerek uygulanmaktadır. Bu algoritmada, arıların ziyaret ettiği kaynaklar olası çözümleri, kaynaklardaki nektar miktarı ise çözümün uygunluk (fitness) değerini temsil etmektedir.

ABC algoritmasında üç tür arı yer alır: işçi arılar, gözcü arılar ve kâşif arılar (Karaboğa ve Akay, 2007). Her bir kaynak için bir işçi arı görevlidir. Algoritmanın temel amacı, maksimum nektar içeriğine sahip kaynağı yani problemin en iyi çözümünü bulmaktır (Akay, 2009).

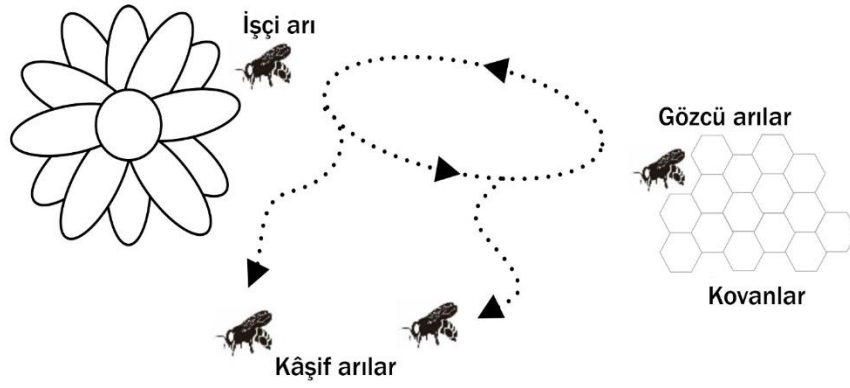
Yapay Arı Kolonisi algoritması, arı kolonisinin yiyecek arama yaparken yapmış olduğu iş bölümünden esinlenilerek geliştirilen sürü zekâsı temelli bir algoritmadır (Karaboğa, 2005). Arı kolonisinde arılar, işçi arı, gözcü arı ve kâşif arı olarak üçe ayrılmaktadır:

İşçi Arılar: İşçi arılar daha fazla nektarın olduğu besin kaynağını araştırırlar. Elde ettikleri bilgileri gözcü arılara aktarırlar.

Gözcü Arılar: Gözcü arılar kovanda bekler ve diğer arıların besin kaynağıyla ilgili dansla paylaştığı bilgileri değerlendirerek besin miktarının fazla olduğu besin kaynağına yönelirler.

Kâşif Arılar: Kâşif arılar yiyecek aramayı rastgele dağılarak yaparlar.

Arı kolonisinin nektar toplama şeması Şekil 3.12'de gösterilmiştir (Chao ve Li, 2022).



Şekil 3.12 Arı kolonisinin nektar toplama şeması (Chao ve Li, 2022)

Diğer popülasyon temelli algoritmelerde olduğu gibi, ABC algoritmasında da iteratif bir süreç bulunmaktadır (Karaboğa, 2005; Karaboğa ve Baştürk, 2007). Çizelge 3.14’te Yapay arı kolonisi optimizasyon algoritmasının sözde kodu verilmiştir.

Çizelge 3.14 Yapay arı kolonisi optimizasyon algoritmasının sözde kodu

-
1. begin:
 2. Parametreleri belirle
 3. yap
 4. döngü kontrolü {
 5. Yeni besin kaynaklarının bulunması için işçi arı safhası
 6. İşçi arılardan almış oldukları bilgilere göre besin arayan gözcü arı safhası
 7. Görevli arılardan etkilenmeyerek yeni besin kaynağı arayan kaşif arı safhası
 8. En iyi besin kaynağını hafızaya al
 9. } son
 10. en iyi besin sonucunu ver ve SONLANDIR
-

Şekil 3.13’te ABC’nin akış diyagramı yer almaktadır.

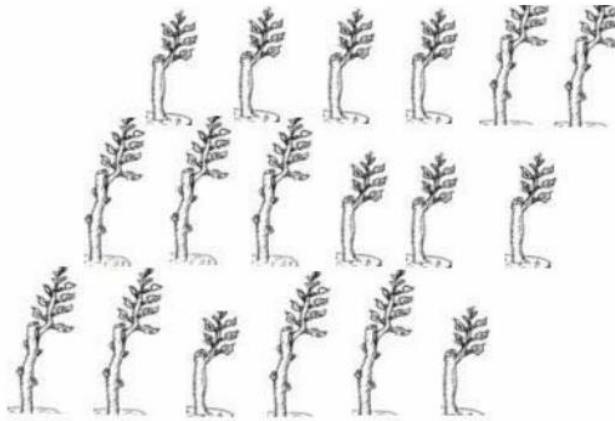


Şekil 3.13 Yapay arı kolonisi optimizasyon algoritmasının akış diyagramı

3.1.7 Fidan Gelişim Algoritması

SGuA, fidanların gelişim süreçlerinden esinlenerek tasarlanmış bir meta sezgisel optimizasyon algoritmasıdır. Bu yöntem, fidanların gelişim süreçlerini taklit ederek, problemin çözümü için gerekli olan parametreleri fidanlara yerleştirmektedir ve işlemleri bu fidanlar üzerinde gerçekleştirmektedir. Algoritmadaki operatörler, fidanların gelişim süreçlerini taklit edebilmek için tasarlanmıştır. Fidan ekimi operatörü, arama uzayında düzenli bir şekilde dağıtılmış fidanların oluşturulmasını sağlamaktadır. Dallanma operatörü, yerel arama yaparken eşleştirme operatörü ise küresel arama yapmayı sağlamaktadır. Aşılama operatörü benzer fidanlar arasında bilgi değişimini sağlamaya yardımcı olmaktadır (Karci, 2007; Seyyarer vd., 2018).

Bu şekilde, SGuA’da aday çözümler başlangıçta düzenli olarak arama uzayına dağıtılmaktadır. Daha sonra hem yerel hem de küresel arama gerçekleştirilmekte ve benzer çözümler arasında bilgi değişimi yapılmaktadır (Karci, 2007). Çözüm alanındaki tüm fidanlar bitkilere karşılaştırılabileceği için hepsinin bahçede eşit şekilde dağılması gerekmektedir (Şekil 3.14). Birden fazla koşul içermeyen bir problemde, her fidan potansiyel bir çözümü temsil etmektedir. Birden fazla koşulun bulunduğu bir durumda, tüm fidanlar bir çözüm olarak kabul edilir. Fidanların daha hızlı gelişebilmesi için, bir çiftçi onları eşit aralıklarla yerleştirmek isteyecektir (Şekil 3.14) (Karcı ve Alataş, 2006).



Şekil 3.14 Bahçeye fidanların eşit şekilde dağıtılması (Karcı ve Alataş, 2006)

Ekim Aşaması

Düzgün bir ekim örneklemesiyle, fidanlar çözüm uzayına homojen bir şekilde dağıtılır.

Büyüme Aşaması

Bu aşama üç temel operatörü kapsamaktadır:

- Eşleşme Operatörü: Var olan fidanlar arasında eşleştirme yaparak yeni bireyler oluşturmayı hedeflemektedir. Bu operatör, iki fidan arasında bilgi alışverişi sağlayan ve küresel düzeyde arama yapan bir yöntemdir.
- Dallanma Operatörü: Fidanların mevcut dal konumlarına dayanarak, olasılıksal bir yaklaşımla yeni dal konumları belirlenir ve yeni fidanlar oluşturulur.
- Aşılama Operatörü: Mevcut fidanlara benzeyen yeni bireyler üretmeyi amaçlar. Bu operatör, benzer özellikteki fidanları temel alarak arama yapan bir tekniktir (Seyyarer vd., 2018).

Ekim Aşaması

İlk olarak, iki fidan $S_0 = \{u_1, u_2, \dots, u_n\}$, $S_1 = \{I_1, I_2, \dots, I_n\}$ olsun, n fidanın boyudur ve bu durumda $k = 1$ olarak kabul edilir, burada u_i , $1 < i < n$, karşılık gelen değişken için üst sınır değeridir ve I_i , $1 < i < n$, karşılık gelen değişken için alt sınır değeridir. Daha sonra, k 'nin bölünme faktörü olduğunu gösteren bir bölen faktör belirlenir. Öncelikle, $k = 2$ ve iki ilave S_3, S_4 fidanı S_0 ve S_1 'den türetilir. Fidan S_0 iki parçaya bölünür (mümkünse eşit uzunlukta), bu durumda 4 fidan ($2^2 = 4$) S_0 'dan türetililebilir. Ancak bunlardan biri S_0 ile aynıdır ve diğeri S_1 ile aynıdır. Daha sonra S_0 ve S_1 'den farklı iki fidan türetililebilir (Seyyarer vd., 2018).

$$S_2 = \left\{ I_1 + (u_1 - I_1) * r, I_2 + (u_2 - I_2) * r, \dots, I_{\frac{n}{2}} + \left(u_{\frac{n}{2}} - I_{\frac{n}{2}} \right) * r, I_{\frac{n}{2}+1} + \left(u_{\frac{n}{2}+1} - I_{\frac{n}{2}+1} \right) * (1 - r), I_{\frac{n}{2}+2} + \left(u_{\frac{n}{2}+2} - I_{\frac{n}{2}+2} \right) * (1 - r) \right\}$$
$$S_3 = \left\{ I_1 + (u_1 - I_1) * (1 - r), I_2 + (u_2 - I_2) * (1 - r), \dots, I_{\frac{n}{2}} + \left(u_{\frac{n}{2}} - I_{\frac{n}{2}} \right) * (1 - r), I_{\frac{n}{2}+1} + \left(u_{\frac{n}{2}+1} - I_{\frac{n}{2}+1} \right) * r, I_{\frac{n}{2}+2} + \left(u_{\frac{n}{2}+2} - I_{\frac{n}{2}+2} \right) * r \right\}$$

Burada r , $0 < r < 1$ gibi rastgele bir sayıdır. Bahçede kalan fidanlar, k 'nin değerinin yükseltilmesiyle aynı yöntemi uygulayarak elde edilecektir. $k = 3$ durumunda,

S_1 'den 6 türemiş fidan olacaktır.

$$\begin{aligned}
S_4 &= \left\{ I_1 + (u_1 - I_1) * r, I_2 + (u_2 - I_2) * r, \dots, I_{\frac{2n}{3}} + \left(\frac{u_{2n}}{3} - I_{\frac{2n}{3}} \right) * r, I_{\frac{2n}{3}+1} + \right. \\
&\quad \left. \left(\frac{u_{2n+1}}{3} - I_{\frac{2n+1}{3}} \right) * r, \dots, I_n + (u_n - I_n) * r \right\} \\
S_5 &= \left\{ I_1 + (u_1 - I_1) * r, I_2 + (u_2 - I_2) * r, \dots, I_{\frac{n}{3}} + \left(\frac{u_n}{3} - I_{\frac{n}{3}} \right) * r, I_{\frac{n}{3}+1} + \right. \\
&\quad \left(\frac{u_{\frac{n}{3}+1}}{3} - I_{\frac{n}{3}+1} \right) * (1 - r), \dots, I_{\frac{2n}{3}} + \left(\frac{u_{2n}}{3} - I_{\frac{2n}{3}} \right) * (1 - r), I_{\frac{2n}{3}+1} + \left(\frac{u_{2n+1}}{3} - I_{\frac{2n+1}{3}} \right) * \\
&\quad \left. r, \dots, I_n + (u_n - I_n) * r \right\} \\
S_6 &= \left\{ I_1 + (u_1 - I_1) * r, I_2 + (u_2 - I_2) * r, \dots, I_{\frac{n}{3}} + \left(\frac{u_n}{3} - I_{\frac{n}{3}} \right) * r, I_{\frac{n}{3}+1} + \right. \\
&\quad \left. \left(\frac{u_{\frac{n}{3}+1}}{3} - I_{\frac{n}{3}+1} \right) * (1 - r), \dots, I_n + (u_n - I_n) * (1 - r) \right\} \\
S_7 &= \left\{ I_1 + (u_1 - I_1) * (1 - r), I_2 + (u_2 - I_2) * (1 - r), \dots, I_{\frac{n}{3}} + \left(\frac{u_n}{3} - I_{\frac{n}{3}} \right) * (1 - r), \right. \\
&\quad \left. I_{\frac{n}{3}+1} + \left(\frac{u_{\frac{n}{3}+1}}{3} - I_{\frac{n}{3}+1} \right) * r, \dots, I_n + (u_n - I_n) * r \right\} \\
S_8 &= \left\{ I_1 + (u_1 - I_1) * (1 - r), I_2 + (u_2 - I_2) * (1 - r), \dots, I_{\frac{n}{3}} + \left(\frac{u_n}{3} - I_{\frac{n}{3}} \right) * (1 - r), \right. \\
&\quad \left. I_{\frac{n}{3}+1} + \left(\frac{u_{\frac{n}{3}+1}}{3} - I_{\frac{n}{3}+1} \right) * r, \dots, I_{\frac{2n}{3}} + \left(\frac{u_{2n}}{3} - I_{\frac{2n}{3}} \right) * r, I_{\frac{2n}{3}+1} + \left(\frac{u_{2n+1}}{3} - I_{\frac{2n+1}{3}} \right) * (1 - \right. \\
&\quad \left. r), \dots, I_n + (u_n - I_n) * (1 - r) \right\} \\
S_9 &= \left\{ I_1 + (u_1 - I_1) * (1 - r), I_2 + (u_2 - I_2) * (1 - r), \dots, I_{\frac{2n}{3}} + \left(\frac{u_{2n}}{3} - I_{\frac{2n}{3}} \right) * (1 - \right. \\
&\quad \left. r), I_{\frac{2n}{3}+1} + \left(\frac{u_{2n+1}}{3} - I_{\frac{2n+1}{3}} \right) * r, \dots, I_n + (u_n - I_n) * r \right\}
\end{aligned}$$

İkili durumda geçerli değer, rasgele bir değerle çarpmak yerine tamamlanır. Fidanların türetilmesi bu şekilde popülasyon tamamlanana kadar devam eder. Örneğin, değişken değerlerinin aralığı [(10, -10), (10, -10), (10, -10), (10, -10)] şeklindedir. O zaman $S_0 = \{10, 10, 10, 10\}$ ve $S_1 = \{-10, -10, -10, -10\}$. $k = 2$ durumunda: $r = 0.7, 0.4, 0.2, 0.8, S_2 = \{4, -2, 6, -6\}$ ve $r = 0.5, 0.0, 0.1, 0.9, S_3 = \{0, -10, -8, 8\}$.

Çizelge 3.15 Başlangıç popülasyonunu oluşturmak

```
//G bir bahçe, I indis kümesi ve Ie genişletilmiş indis kümesi.
1- Dallanma değerlerinin alt sınırlarını içeren G[1] ve üst sınırlarını içeren G[2]
gibi iki tane fidan oluştur.
2- Indis←-3
3- k←-2
4- While P doymamış do
    ie, Ie'nin bir elemanı olsun, her ie bit değeri ile genişletilmiş ve bu bit değeri
    parçaya karşılık gelir.
    i←-1
    While P doymamış ve özel bir k ( $i \leq 2k-2$ ) değeri için tüm fidanlar üretilmemiş
    do
        i bir k-bit sayı ve ie i'nin genişletilmiş değerine karşılık gelir. i'nin her biti
        G[0] ve G[1]'in karşılık gelen kısmının uzunluğuna kadar genişletilir.
        For j←-1 to n do
            if ie'nin j. biti=1 then P[indis]'in j. dallanması=G[1]*r
            else G[indis]'in j. dallanması=G[2]*r
            r=random[0,1] ve reel sayıdır.
        indis←indis+1
        i←i+1
    k←k+1
```

Fidanların Büyüme Süreci:

Bu aşamada büyüme, üç ana işlemle gerçekleştirilir: eşleştirme, dallanma ve aşılama. Eğer bahçede toplamda (G) |G| adet fidan bulunuyorsa, mevcut fidanlar ve bu işlemlerle elde edilen yeni fidanlar (yavru fidanlar) bir araya getirilir. Ardından, bu fidanlar arasından en iyi |G| tanesi seçilerek yeni bahçe oluşturulur. Bu seçim sürecinde herhangi bir rastgelelik yoktur; tamamen en iyi performansı gösteren fidanlar belirlenerek bahçede yer almaktadır (Seyyarer vd., 2018).

Eşleşme:

Eşleşme operatörü, geçici çözümler olan fidanlar arasında bilgi alışverişi yaparak yeni bir fidan üretmeyi amaçlar (Çizelge 3.1). İki fidan $S_1 = \{s_{11}, s_{12}, \dots, s_{1i},$

..., s_{1n} } ve $S_2 = \{s_{21}, s_{22}, ..., s_{2i}, ..., s_{2n}\}$ olarak tanımlansın. Bu iki fidan arasındaki mesafe, eşleştirme sürecini etkileyebilir veya etkilemeyebilir; bu etki mevcut fidanlar arasındaki mesafeye bağlıdır. S_1 ve S_2 fidanlarının eşleşme olasılığı $Pm(S_1, S_2)$ ile gösterilir ve bu olasılık fonksiyonu lineer ya da üstel formda olabilir (Seyyarer vd., 2018).

Çizelge 3.16 Eşleşme (S_1, S_2)

-
1. $j \leftarrow 1, \dots, n$
 2. Hesapla $Pm(S_1, S_2) = 1 - \frac{(\sum_{j=1}^n (s_{1,j} - s_{2,j})^2)^{1/2}}{R}$
 3. $i \leftarrow 1, \dots, n$
 4. if $Pm(S_1, S_2) \text{ random}(0,1)$ then
 $S_1 \leftarrow S_1 - s_{1,j}$ ve $S_2 \leftarrow S_2 - s_{2,i}$, $S_1 \leftarrow S_1 + s_{2,j}$ ve $S_2 \leftarrow S_2 + s_{1,i}$
-

Dallanma:

$S_1 = \{s_{11}, s_{12}, ..., s_{1i}, ..., s_{1n}\}$ biçiminde tanımlanan bir fidan ele alındığında, s_{1i} noktasında bir dal oluştuğunda, bu durumun s_{1j} noktasında da bir dal oluşmasına yol açma olasılığı ($i \neq j$ olması koşuluyla), doğrusal veya doğrusal olmayan yöntemlerle hesaplanabilir. Bu hesaplamalarda, s_{1i} ve s_{1j} noktaları arasındaki mesafe genellikle $|i - j|$ ya da $|j - i|$ olarak ifade edilir (Çizelge 3.17).

Çizelge 3.17 Dallanma (S_1, S_2)

-
1. $i \leftarrow 1, \dots, n$
 2. $j \leftarrow i+1, \dots, n$
 3. if dallanma yoksa $P(s_{1,j}|s_{1,i})=1$ ve dallanma işlemi uygulanır.
 4. else

$$P(s_{1,j}|s_{1,i}) = 1 - \frac{1}{(|j-i|)^2}, i \neq j$$
veya

$$P(s_{1,j}|s_{1,i}) = 1 - \frac{1}{e^{(|j-i|)^2}}$$
 5. if $P(s_{1,j}|s_{1,i}) \geq \text{random}(0,1)$ then $s_{1,j}$ bir dallanma olacak.
-

Aşılama:

Aşılama işlemi, bu algorithma fidanların birbirine benzememesi durumunda iki farklı fidan arasında gerçekleşir. Fidanların farklılığı, aşılama işleminin başarısını etkiler ve ayrıca aşılama başarısı, her iki fidanın farklılığı ile orantılıdır (Çizelge 3.28).

Çizelge 3.18 Aşılama (S_1, S_2)

```
1.  $i \leftarrow 1, \dots, n$ 
2.  $\text{Dis}(S_1, S_2) = \sum_{i=1}^n s_{1,i} \oplus s_{2,i}$ 
3. if  $\text{Dis}(S_1, S_2) \geq \text{random}$  then
4.  $S'_1 \begin{cases} s_{1,i} \text{ if } s_{1,i} = s_{2,i} \\ \text{random if } s_{1,i} \neq s_{2,i} \end{cases}$ 
   ve
    $S'_2 \begin{cases} s_{2,i} \text{ if } s_{2,i} = s_{1,i} \\ \text{random if } s_{2,i} \neq s_{1,i} \end{cases}$ 
```

SGuA, daha kaliteli çözümler (fidanlar) üretmek için bahçedeki bireyler arasındaki benzerlikten yararlanmaktadır. Bu süreç eşleştirme operatörü ile gerçekleştirilmektedir ve bu yönüyle hesaplamalı kültürel etkileşim olarak değerlendirilebilir. Ayrıca, aşılama işlemi eşleştirmenin tersidir; çünkü bu operatör, fidanlar arasındaki farklılıklardan yararlanarak yeni bireyler üretmektedir. Bu bağlamda, aşılama aynı zamanda kültürel etkileşimin de tersidir, zira benzer bireylerin değil, farklı bireylerin etkileşimini temel almaktadır.

Dallanma süreci, tek bir operatörle yönetilir ve amacı yeni çözümleri mevcut çözüm havuzuna dahil ederken, aynı zamanda eski ve düşük kaliteli çözümleri sistemden uzaklaştırmaktır. Bu şekilde çözüm havuzu sürekli yenilenmekte ve gelişmektedir. Fidanların kalitesi, genetik algoritmalarından farklı olarak doğrudan hedef fonksiyon üzerinden değerlendirilmektedir. Her bir fidan, problem için bir çözüm önerisi olarak ele alınır ve bu çözümün başarısı, hedef fonksiyon değeriyle ölçülür (Seyyarer vd., 2018).

SGuA iki farklı yoldan uygulanabilir; Çizelge 3.19'da SGuA-1 ve Çizelge 3.20'de SGuA-2 yöntemlerinin sözde kodu verilmektedir.

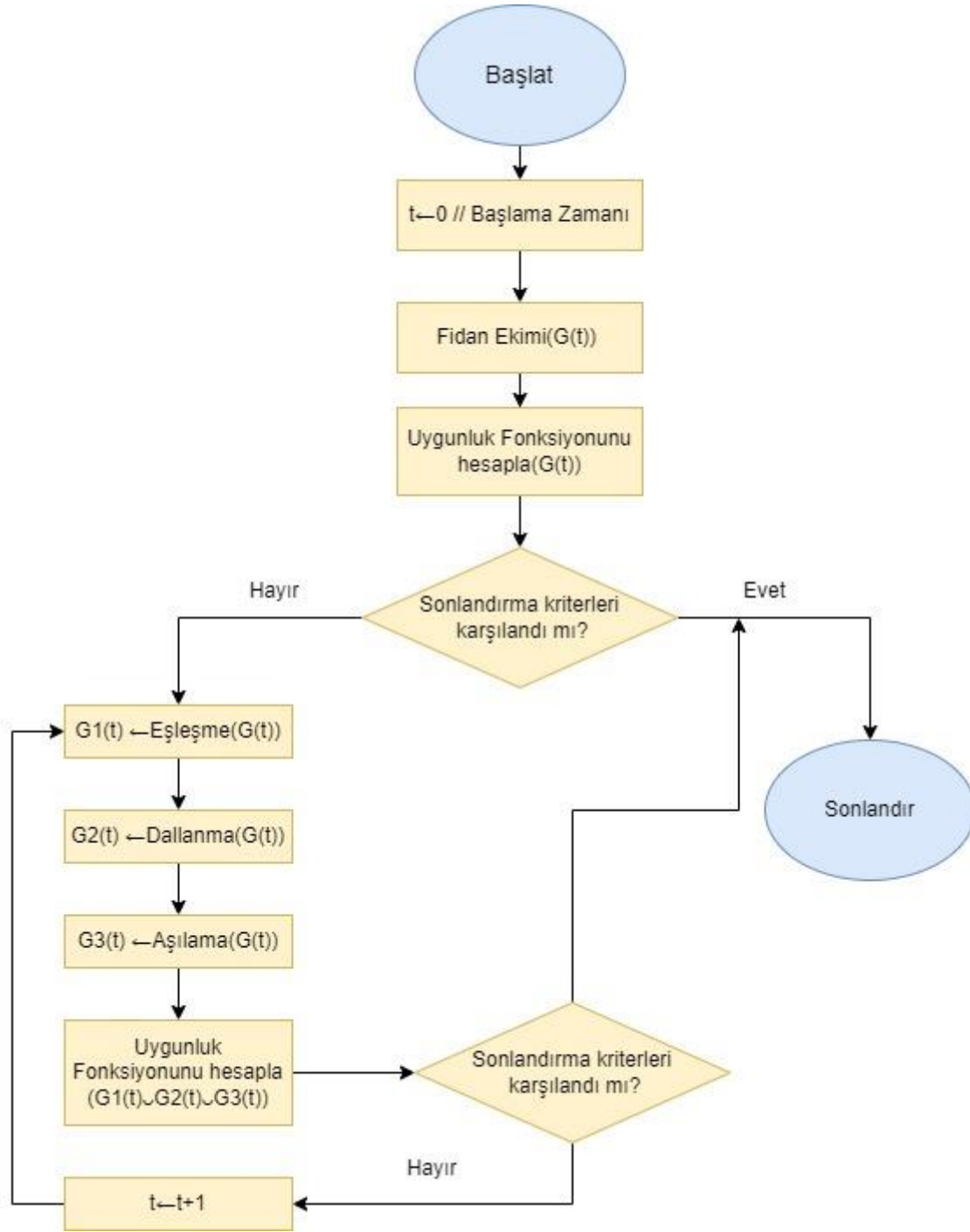
Çizelge 3.19 Sgua-1'in sözde kodu

```
1. t←0 // Başlama Zamanı
2. Fidan Ekimi(G(t)) // Başlangıç Popülasyonu
3. Uygunluk Fonksiyonunu hesapla(G(t))
4. while Sonlandırma kriteri ile karşılanmadı do
5. G1(t) ←Eşleşme(G(t))
6. G2(t) ←Dallanma(G(t))
7. G3(t) ←Aşılama(G(t))
8. Uygunluk Fonksiyonunu hesapla (G1(t)∪G2(t)∪G3(t))
9. G(t+1) ←Seç(G1(t)∪G2(t)∪G3(t))
10. t←t+1
```

Çizelge 3.20 Sgua-2'nin sözde kodu

```
1. t←0 // Başlama Zamanı
2. Fidan Ekimi(G(t)) // Başlangıç Popülasyonu
3. Uygunluk Fonksiyonunu hesapla(G(t))
4. while Sonlandırma kriteri ile karşılanmadı do
5. G1(t) ←Eşleşme(G(t))
6. Uygunluk Fonksiyonunu hesapla (G1(t))
7. Gm(t) ← Seç(G1(t)∪G(t))
8. G2(t) ←Dallanma(Gm(t))
9. Uygunluk Fonksiyonunu hesapla (G2(t))
10. Gb(t) ← Seç(G2(t)∪ Gm(t))
11. G3(t) ←Aşılama(Gb(t))
12. Uygunluk Fonksiyonunu hesapla (G3(t))
13. G(t+1) ←Seç(G3(t)∪Gb(t))
14. t←t+1
```

Şekil 3.15'te SGuA'nın akış diyagramı yer almaktadır.



Şekil 3.15 SGuA akış diyagramı

3.1.8 Geliştirilmiş Fidan Gelişim Algoritması

Geliştirilmiş Fidan Gelişim Algoritması (ISGuA), Fidan Gelişim Algoritması (SGuA) temel alan bir optimizasyon yöntemidir. 3.1.7 başlığında SGuA ile ilgili detaylı bilgiler verilmiştir. Klasik SGuA, belirli optimizasyon problemlerinde yakınsama hızı

ve küresel optimuma ulaşma kapasitesi açısından sınırlamalar içermektedir. Optimizasyon algoritmalarının performansı, başlangıç koşulları, kullanılan test fonksiyonlarının özellikleri ve algoritmanın iterasyon sayısına bağlı olarak önemli ölçüde değişmektedir.

Bu bölümde başlangıç popülasyon oluşturma yöntemleri tanıtılmıştır. Çalışmanın temel amacı, farklı başlatma yöntemlerinin SGuA üzerindeki etkisini test etmektir.

Çizelge 3.21 ve Çizelge 3.22’de çalışmada kullanılan 10 farklı başlangıç popülasyon oluşturma fonksiyonunun özeti ve sözde kodu verilmektedir. Dağılım parametrelerinde yapılan değişikliklerle, iki (2) farklı Beta ve Logaritmik varyantı, birer tane olmak üzere Rayleigh, Uniform, Exponential, Weibull, Random ve Latin hiperküp örnekleme yöntemleri kullanılmıştır (Agushaka vd., 2023).

Çizelge 3.21 Başlatma şeması

S/N	Başlatma Şeması	Fonksiyon
1	Random	Rand
2	Beta	Betarnd (3, 2)
3	Beta	Betarnd (2.5, 2.5)
4	Uniform	Unifrnd (0, 1)
5	Logarithmic normal	Lognrnd (0, 0.5)
6	Logarithmic normal	Lognrnd (0, 0.67)
7	Exponential	Exprnd (0.5)
8	Rayleigh	Raylrnd (0.4)
9	Weibull	Wblrnd (1, 1)
10	Latin hypercube sampling	lhsdesign

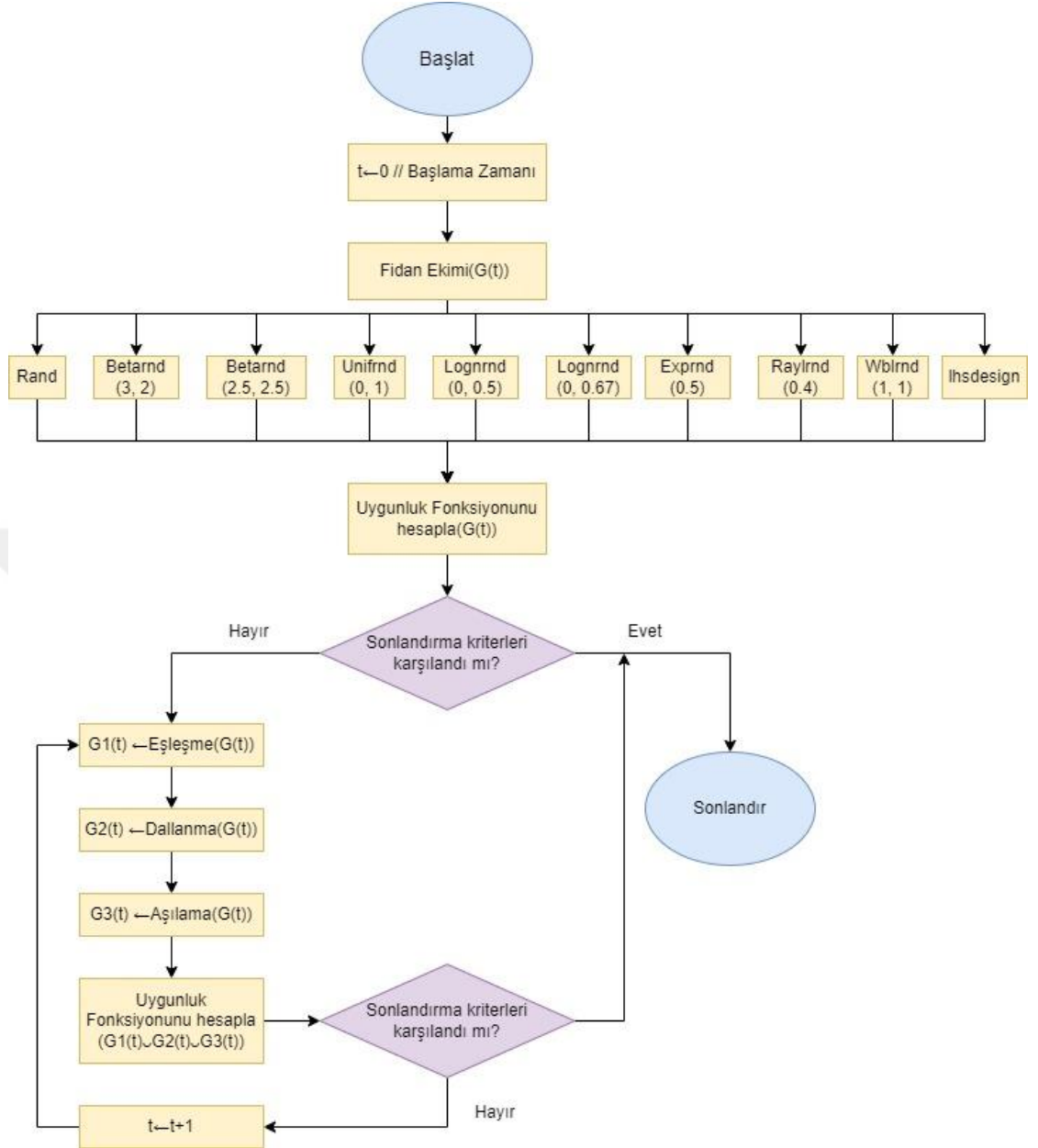
Çizelge 3.22 Isgua’nın sözde kodu

1. $t \leftarrow 0$ // Başlama Zamanı
2. Fidan Ekimi($G(t)$) // Başlangıç Popülasyonu
Rand
Betarnd (3, 2)
Betarnd (2.5, 2.5)

Çizelge 3.22 Isgua'nın sözde kodu (devam)

Unifrnd (0, 1)
Lognrnd (0, 0.5)
Lognrnd (0, 0.67)
Exprnd (0.5)
Raylrnd (0.4)
Wblrnd (1, 1)
lhsdesign
3. Uygunluk Fonksiyonunu hesapla(G(t))
4. while Sonlandırma kriteri ile karşılanmadı do
6. $G1(t) \leftarrow \text{Eşleşme}(G(t))$
7. $G2(t) \leftarrow \text{Dallanma}(G(t))$
8. $G3(t) \leftarrow \text{Aşılama}(G(t))$
9. Uygunluk Fonksiyonunu hesapla ($G1(t) \cup G2(t) \cup G3(t)$)
10. $G(t+1) \leftarrow \text{Seç}(G1(t) \cup G2(t) \cup G3(t))$
11. $t \leftarrow t+1$

Şekil 3.16'da ISGuA'nın akış diyagramı yer almaktadır.



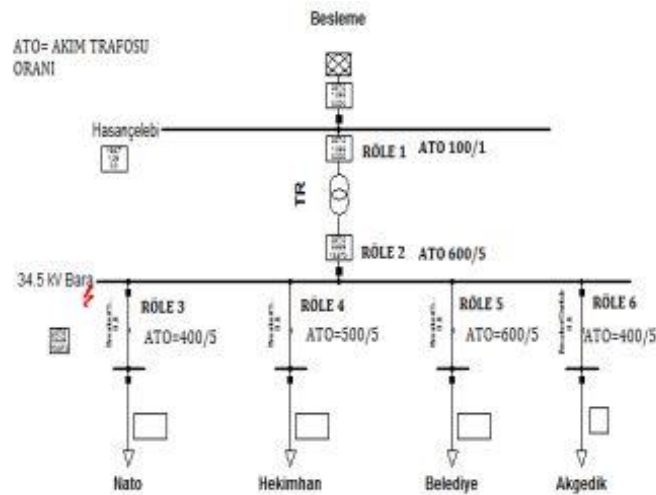
Şekil 3.16 ISGuA akış diyagramı

3.2 Aşırı Akım Rôle Koordinasyonu

Güç sistemlerinde, çeşitli sebeplerden dolayı arızalar meydana gelebilir. Bu arızaların çoğu, nominal seviyeden daha fazla akımın güç sistemine akmasından kaynaklanmaktadır (Seyyarer vd., 2018; Karababa, 2013). Koordinasyon problemi, birçok güç sisteminde karşılaşılan önemli bir sorundur. Bu problem, doğrusal veya doğrusal olmayan bir programlama problemi olarak formüle edilebilmektedir. Doğrusal

koordinasyon modelinde, yalnızca zaman kadranı ayarlanarak optimize edilmektedir ve başlatma akımı ayarları, maksimum yük akımı ile minimum arıza akımı arasındaki değerlerde sabitlenmektedir. Bu model, rölelerin çalışma karakteristiğine ve sistem koşullarına bağlı olarak, güç sisteminde meydana gelebilecek arıza durumlarının önüne geçmeyi amaçlamaktadır. Doğrusal olmayan koordinasyon modelinde ise, rölelerin karakteristiğine bağlı olarak, hem zaman kadranı hem de başlatma akımı ayarları aynı anda optimize edilmektedir. Bu model, daha esnek bir yaklaşım sunar ve güç sisteminde meydana gelebilecek çeşitli arıza durumlarını daha etkin bir şekilde önleyebilmektedir. Ancak, doğrusal olmayan modelin optimizasyonu daha zor ve zaman alıcı olabilmektedir (Amraee, 2012).

Aşırı akım röleleri, ani ve zaman gecikmeli olarak iki çeşit çalışma karakteristiğine dayanmaktadır. Ülkemizde, faz-faz arızaları için ters zamanlı olarak ayarlanırken, faz-toprak arızaları için ise sabit zamanlı olarak ayarlanmaktadır. Bu röleler, gücün akışına göre ileri ya da geri yönde koruma sağlayabilmektedir (Akdağ ve Yeroğlu, 2019). Bu çalışmada, Türkiye Elektrik İletim Anonim Şirketi (TEİAŞ) güç sisteminde ORC için kullanılan aşırı akım rölelerinin ayarları incelenmiştir. Bu sistemde, faz-faz kısa devre (154 kV/OG 3-Faz Aşırı Akım) arızalarına karşı ters zamanlı ayar temel alınarak koordinasyon yapılmıştır (Şekil 3.17) (Seyyarer vd. 2018). Bu koordinasyon sayesinde, sistemde meydana gelebilecek aşırı akım arızalarının önüne geçilebilir ve sistem çalışırken oluşabilecek zararlar azaltılabilmektedir (Seyyarer vd. 2017).



Şekil 3.17 Örnek trafo merkezi tek hat şeması (Seyyarer vd. 2018)

3.2.1 Aşırı Akım Röle Koordinasyonu Probleminin Formülasyonu

Bu çalışmanın ana amacı, aşırı akım rölelerinin çalışma zamanlarını minimize etmektir. Bu amaçla, tüm rölelerin karakteristik özellikleri aynı kabul edilerek bir optimizasyon yapılmıştır. Bu çalışma sonucunda, aşırı akım rölelerinin çalışma zamanlarının en minimum seviyeye indirilebileceği gösterilmiştir. Bu, elektrik sistemlerinin güvenliğini sağlamada önemli bir adımdır (Seyyarer vd., 2017).

ORC’de kullanılacak amaç fonksiyonu Eşitlik 3.1’de gösterildiği gibidir.

$$\text{Amaç Fonksiyonu} = \min \sum_{i=1}^n (t_i) \quad (3.1)$$

Bu çalışmada, n ; güç sisteminde bulunan aşırı akım rölelerinin sayısı ve t_i ; her bir aşırı akım rölesinin çalışma zamanı gösterilmiştir. Bu çalışmada, faz-faz arızaları dikkate alınarak, rölelerin standart ters zamanlı eğrisine göre toplam minimum zaman hesaplanmıştır.

Standart ters zaman eğrili çalışma karakteristiğine sahip, bağıntı Eşitlik 3.2’de görüldüğü gibidir (Seyyarer vd., 2017; Pınar, 1985).

$$t_i = \frac{0,14 \cdot t_d}{\left(\frac{I_{kd}}{I_p}\right)^{0,02} - 1} \quad (3.2)$$

Eşitlik 3.2’deki I_{kd} , kısa devre akım değerini ifade etmektedir ve bu değer sabittir. I_p değeri ise rölenin başlatma akım değeridir. t_i , rölenin çalışma süresini ve t_d ise çalışma eğrisinin değerlerini ifade etmektedir. Optimizasyon için kullanılan kısıtlamalar, TEİAŞ sisteminde kullanılan değerler baz alınarak öngörülen bir aralıkta ifade edilebilir. Bunlar;

I_{kd} değeri sabit değerdir. Bu değer röle 1 için $I_{kd} = 1263A$, röle 2, 3, 4, 5, 6 için $= 5639A$.

$$1.kısıt \quad t_{i_min} \leq t_i \leq t_{i_max} = 1s \leq t_i \leq 2.2s$$

1.kısıtta altı rölenin zaman sıralamasını $t_6, t_5, t_4, t_3 \leq t_2 \leq t_1$ olacak şekilde belirlemektedir. Röle 1’in zamanı olan t_1 , röle 2 zamanı olan t_2 ’den büyük olacağını

belirtmektedir. Ayrıca Röle 2 zamanı olan t_2 , diğer rölelerin zamanlarından büyük olması gerekmektedir. Ancak Röle 3, Röle 4, Röle 5 ve Röle 6 zamanları kendi aralarında sıralanmamaktadır. Sadece bu röle değerleri t_2 ve t_1 'den küçük olmalıdır. t_1 t_2 'den 0.3 sn, t_2 'de diğer röle zamanlarından 0.3 saniye daha büyük olmalıdır.

2.kısıt $td_{min} \leq td \leq td_{max}$ ($0.2 \leq td \leq 1$)

Bu kısıt TEİAŞ'ın kullandığı sistemdeki yaklaşım doğrultusunda belirlenmiştir.

3. kısıt bu kısıt başlama zamanları içindir yani I_p içindir. Bu değer her rölede farklı bir sınıra sahiptir.

Röle 1 $117.152 \leq I_p \leq 128.65$

Röle 2 $523 \leq I_p \leq 575$

Röle 3 $400 \leq I_p \leq 420$

Röle 4 $500 \leq I_p \leq 510$

Röle 5 $600 \leq I_p \leq 610$

Röle 6 $400 \leq I_p \leq 420$

Bu kısıtlar doğrultusunda t_i değerleri hesaplanacaktır (Seyyarer vd., 2017).

3.3 Çok Modlu Kıyaslama Fonksiyonları

Bu tez çalışmasında kullanılan optimizasyon algoritmalarının performansını ölçmek için çok modlu kıyaslama fonksiyonlarından 3 fonksiyon seçilmiştir. Çok modlu kıyaslama fonksiyonlarının birden fazla optimum değeri vardır. Bu durum onları tek modlu fonksiyonlara oranla daha zorlu hale getirmektedir. Bir problemde en iyi çözüm küresel optimum olarak tanımlanırken, diğer çözümler yerel optimum olarak adlandırılmaktadır. Bir algoritmanın asıl amacı, bu yerel optimumları aşarak küresel optimumu bulabilmektir. Bu nedenle, algoritmaların arama yetenekleri ve yerel optimumlardan kaçınma becerileri çok modlu test fonksiyonları aracılığıyla sınanmaktadır. Kullanılan F_8 fonksiyonu hariç F_9 ve F_{10} fonksiyonlarının minimumları 0'a eşittir (Mirjalili, 2015). Çizelge 3.23'te bu çok modlu kıyaslama fonksiyonları verilmiştir.

Çizelge 3.23 Çok modlu kıyaslama fonksiyonları

Fonsiyon Adı	Fonksiyon Denklemi	Değişken boyutu	Aralık	Optimum Değer
Generalized Schwefel	$F_8(x) = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	10	[-500, 500]	-418.98xn
Generalized Rastrigin	$F_9(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	10	[-5.12, 5.12]	0
Ackley	$F_{10}(x) =$ $-20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) -$ $\exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$	10	[-32,32]	0

4. BULGULAR

Bu bölümde, ISGuA algoritmasında kullanılan 10 farklı başlangıç popülasyonu oluşturma yöntemine ait elde edilen sonuçlar detaylı bir şekilde sunulmuştur. Çalışmada, rölenin başlatma akım değeri (I_p), çalışma süresi (t_i) ve çalışma eğrisi değeri (t_d) temel alınarak her yöntemin algoritma performansına olan etkileri kapsamlı olarak incelenmiştir. Farklı başlangıç popülasyonu yöntemlerinin, çözüm kalitesi, hesaplama süresi ve optimizasyon başarısı üzerindeki etkileri analiz edilerek, hangi yöntemin daha verimli olduğu belirlenmeye çalışılmıştır.

Bunun yanı sıra, önerilen ISGuA algoritmasının farklı meta sezgisel algoritmalar ile karşılaştırmalı analizi yapılmıştır. Bu kapsamda, WOA, GA, GWO, BOA, LCA, ABC ve SGuA algoritmaları kullanılarak elde edilen sonuçlar değerlendirilmiştir.

Yapılan karşılaştırmalarda, her algoritma için I_p , t_i ve t_d değerleri analiz edilmiş ve elde edilen sonuçlara göre algoritmaların optimizasyon başarısı belirlenmiştir. Özellikle, t_i değerleri üzerinden yapılan karşılaştırmalar, algoritmaların röle koordinasyonu açısından etkinlik düzeylerini belirlemeye yardımcı olmuştur. ISGuA'nın, diğer meta sezgisel algoritmalarla kıyasla ne ölçüde iyileştirme sağladığı, çözüm kalitesi ve hesaplama süresi bağlamında ayrıntılı olarak incelenmiştir.

Sonuçlar, ISGuA'nın belirli durumlarda daha yüksek optimizasyon başarısı sağladığını ve başlangıç popülasyonu seçiminde kullanılan yöntemlerin algoritmanın performansına doğrudan etki ettiğini göstermektedir. Özellikle, belirli popülasyon oluşturma yöntemleri ile ISGuA'nın daha hızlı yakınsama gösterdiği ve çözüm kalitesinin arttığı tespit edilmiştir. Tüm algoritmaların performans analizleri, elde edilen t_i değerleri üzerinden detaylı çizelgeler ve şekiller ile sunulmuştur.

4.1 Fidan Gelişim Algoritması için Farklı Başlatma Yöntemlerinin Performansa Etkisi

Çizelge 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, 4.8, 4.9 ve 4.10'da farklı başlatma yöntemlerinin SGuA'nın çalışma süresini, başlatma akımını ve tepki eğrisini nasıl etkilediğini göstermektedir. Elde edilen veriler ışığında, hangi başlatma yöntemlerinin

daha verimli olduđu belirlenerek, algoritmanın optimizasyon sürecine katkıları detaylandırılmıştır.

Çizelge 4.1 Rand

Röle	td	Ip	ti
Röle-1	0.5486	122.76	1.6092
Röle-2	0.4471	542.89	1.3062
Röle-3	0.3836	415.66	1.0031
Röle-4	0.3536	507.30	1.003
Röle-5	0.3271	604.95	1.003
Röle-6	0.3890	401.14	1.003

Toplam ti: 6.9275

Çizelge 4.2 Betarnd (3, 2)

Röle	td	Ip	ti
Röle-1	0.5350	128.51	1.6015
Röle-2	0.4517	525.40	1.3010
Röle-3	0.3832	414.02	1.0005
Röle-4	0.43519	509.98	1.0005
Röle-5	0.3273	600.99	1.0005
Röle-6	0.3857	407.17	1.0005

Toplam ti: 6.9045

Çizelge 4.3 Betarnd (2.5, 2.5)

Röle	td	Ip	ti
Röle-1	0.5559	118.08	1.6034
Röle-2	0.4457	543.04	1.3022
Röle-3	0.3837	413.14	1.0011
Röle-4	0.3536	504.96	1.0011
Röle-5	0.3266	604.65	1.0011
Röle-6	0.3839	412.79	1.0011

Toplam ti: 6.9101

Çizelge 4.4 Unifrnd (0, 1)

Röle	td	Ip	ti
Röle-1	0.5383	126.49	1.6001
Röle-2	0.4376	564.02	1.3001
Röle-3	0.3812	419.10	1.0000
Röle-4	0.3536	503.62	1.0000
Röle-5	0.3270	601.59	1.0000
Röle-6	0.3872	402.74	1.0000

Toplam ti: 6.9002

Çizelge 4.5 Lognrnd (0, 0.5)

Röle	td	Ip	ti
Röle-1	0.5437	124.57	1.6053
Röle-2	0.4464	542.32	1.3035
Röle-3	0.3842	412.44	1.0018
Röle-4	0.3550	500.83	1.0018
Röle-5	0.3262	607.10	1.0018
Röle-6	0.3877	403.19	1.0018

Toplam ti: 6.9159

Çizelge 4.6 Lognrnd (0, 0.67)

Röle	td	Ip	ti
Röle-1	0.5594	125.78	1.6586
Röle-2	0.4503	565.97	1.3397
Röle-3	0.3925	409.24	1.0202
Röle-4	0.3609	502.95	1.0202
Röle-5	0.3317	608.84	1.0202
Röle-6	0.3928	408.42	1.0202

Toplam ti: 7.0791

Çizelge 4.7 Exprnd (0.5)

Röle	td	Ip	ti
Röle-1	0.5496	121.45	1.6046
Röle-2	0.4447	546.73	1.3031
Röle-3	0.3815	419.69	1.0015
Röle-4	0.3548	501.18	1.0015
Röle-5	0.3262	606.66	1.0015
Röle-6	0.3887	400.21	1.0015

Toplam ti: 6.9137

Çizelge 4.8 Raylrnd (0.4)

Röle	td	Ip	ti
Röle-1	0.5427	127.92	1.6215
Röle-2	0.4537	532.60	1.3144
Röle-3	0.3891	405.02	1.0072
Röle-4	0.3554	506.11	1.0072
Röle-5	0.3297	600.27	1.0072
Röle-6	0.3852	415.56	1.0072

Toplam ti: 6.9647

Çizelge 4.9 Wblrnd (1, 1)

Röle	td	Ip	ti
Röle-1	0.5424	125.21	1.6050
Röle-2	0.4442	548.48	1.3033
Röle-3	0.3888	400.0	1.0017
Röle-4	0.3541	503.78	1.0017
Röle-5	0.3261	607.19	1.0017
Röle-6	0.3833	415.03	1.0017

Toplam ti: 6.9151

Çizelge 4.10 lhsdesign

Röle	td	Ip	ti
Röle-1	0.5396	127.19	1.6081
Röle-2	0.4414	558.33	1.3054
Röle-3	0.3861	408.43	1.0027
Röle-4	0.3544	503.96	1.0027
Röle-5	0.3271	604.66	1.0027
Röle-6	0.3830	416.76	1.0027

Toplam ti: 6.9243

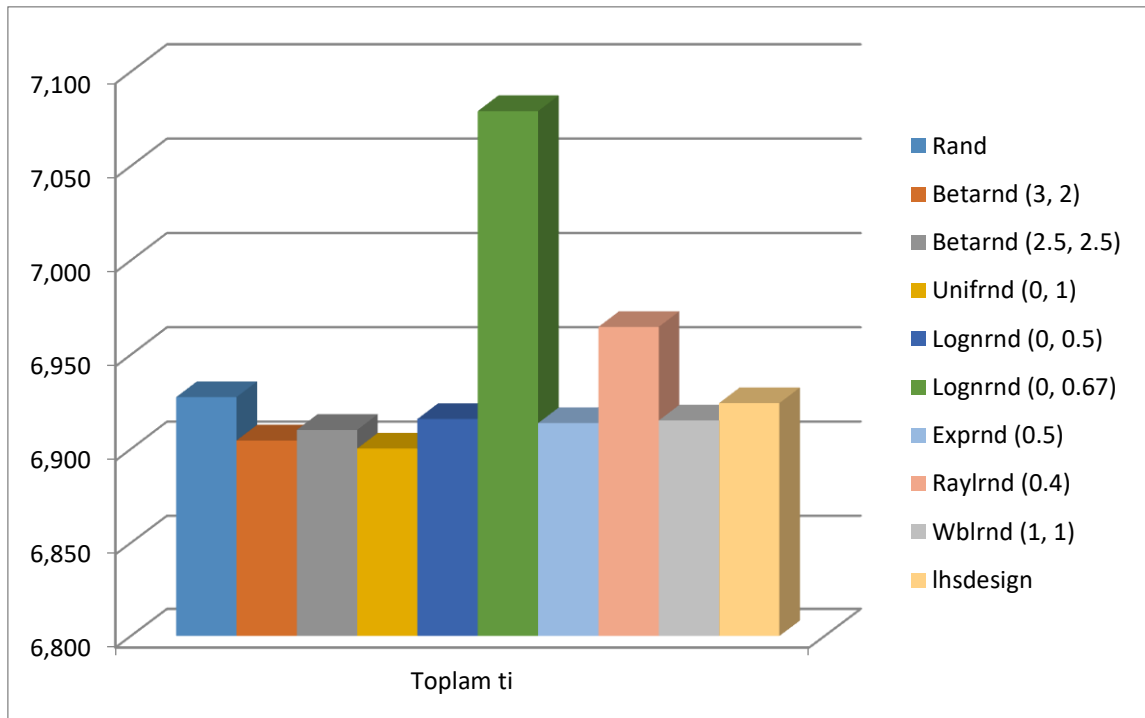
Çizelge 4.11’de tüm başlangıç popülasyon oluşturma yöntemlerinin çalışma süreleri verilmiştir. Şekil 4.1’de ise görsel olarak sunulmuştur.

Çizelge 4.11 Toplam çalışma süreleri

S/N	Fonksiyon	Toplam ti
1	Rand	6.9275
2	Betarnd (3, 2)	6.9045
3	Betarnd (2.5, 2.5)	6.9101
4	Unifrnd (0, 1)	6.9002
5	Lognrnd (0, 0.5)	6.9159
6	Lognrnd (0, 0.67)	7.0791
7	Exprnd (0.5)	6.9137
8	Raylrnd (0.4)	6.9647
9	Wblrnd (1, 1)	6.9151
10	lhsdesign	6.9243

Çizelge 4.11’de, Geliştirilmiş Fidan Gelişim Algoritması’nın farklı başlangıç dağılım yöntemlerine göre toplam çalışma süreleri karşılaştırılmıştır. Elde edilen sonuçlara göre, algoritmanın en kısa sürede çalıştığı dağılım türü 6.9002 saniyelik toplam süreyle Unifrnd (0,1) olmuştur. Bu durum, eşit olasılıkla rastgele dağıtım yapılan çözümlerin, algoritmanın daha hızlı yakınsamasına katkı sağladığını göstermektedir. Özellikle Betarnd (3,2) ve Betarnd (2.5, 2.5) gibi simetrik ve orta yoğunluklu dağılımlar da benzer şekilde düşük sürelerle dikkat çekmiş; bu dağılımların

başlangıç popülasyonunu dengeli oluşturması algoritmanın verimliliğini olumlu yönde etkilemiştir. Ayrıca, 7.0791 saniye ile en uzun süreyi veren Lognrnd (0, 0.67) dağılımı, asimetrik yapısı nedeniyle çözümlerin dengesiz dağılmasına yol açarak algoritmanın performansını düşürmüştür. lhsdesign yöntemi ise daha yapılandırılmış bir örnekleme sunmasına rağmen, rastgele yöntemlerle kıyaslandığında daha yüksek bir çalışma süresi üretmiştir. Genel olarak, elde edilen bulgular başlangıç çözüm dağılımının algoritmanın çalışma süresi üzerinde doğrudan etkili olduğunu ve uygun bir dağılım seçiminin algoritma verimliliğini artırabileceğini göstermektedir.



Şekil 4.1 Toplam çalışma süreleri

Hasançelebi trafo merkezinin röle koordinasyon problemine uygulanan WOA, GA, GWO, BOA, LCA ve ABC sonuçları çizelgeler halinde gösterilmektedir.

4.2 Balina Optimizasyon Algoritması Sonuçları

Çizelge 4.12’de Balina optimizasyon algoritmasına ait sonuçlar yer almaktadır.

Çizelge 4.12 Balina optimizasyon algoritması

Röle	td	Ip	ti
Röle-1	0.5617	121.533	1.640
Röle-2	0.4535	535.434	1.317
Röle-3	0.2632	419.136	1.031
Röle-4	0.2945	509.018	1.016
Röle-5	0.2933	608.715	1.010
Röle-6	0.3726	414.884	1.003

Toplam ti: 7.0170

4.3 Genetik Algoritma Sonuçları

Çizelge 4.13'te Genetik algoritmaya ait sonuçlar yer almaktadır.

Çizelge 4.13 Genetik algoritma

Röle	td	Ip	ti
Röle-1	0.5636	126.319	1.674
Röle-2	0.4680	531.242	1.354
Röle-3	0.2943	412.982	1.023
Röle-4	0.3313	509.032	1.006
Röle-5	0.2320	604.166	1.029
Röle-6	0.3791	405.641	1.002

Toplam ti: 7.0884

4.4 Gri Kurt Algoritması Sonuçları

Çizelge 4.14'te Gri kurt algoritmasına ait sonuçlar yer almaktadır.

Çizelge 4.14 Gri kurt optimizasyon algoritması

Röle	td	Ip	ti
Röle-1	0.5479	122.08	1.603
Röle-2	0.4456	541.46	1.3
Röle-3	0.3575	418.28	1.006
Röle-4	0.3535	503.24	1.0

Çizelge 4.14 Gri kurt optimizasyon algoritması (devam)

Röle	td	Ip	ti
Röle-5	0.3234	604.89	1.001
Röle-6	0.3741	405.46	1.003

Toplam ti: 6.9137

4.5 Kelebek Optimizasyon Algoritması Sonuçları

Çizelge 4.15'te Kelebek optimizasyon algoritmasına ait sonuçlar yer almaktadır.

Çizelge 4.15 Kelebek optimizasyon algoritması

Röle	td	Ip	ti
Röle-1	0.5389	128.2	1.611
Röle-2	0.4408	556.84	1.302
Röle-3	0.3602	418.62	1.006
Röle-4	0.3333	502.56	1.006
Röle-5	0.2984	608.86	1.008
Röle-6	0.3826	407.13	1.001

Toplam ti: 6.9339

4.6 Lig Şampiyonluk Algoritması Sonuçları

Çizelge 4.16'da Lig şampiyonluk algoritmasına ait sonuçlar yer almaktadır.

Çizelge 4.16 Lig şampiyonluk algoritması

Röle	td	Ip	ti
Röle-1	0.5693	117.741	1.605
Röle-2	0.4569	524.616	1.305
Röle-3	0.4925	406.211	1.001
Röle-4	0.4552	501.01	1.001
Röle-5	0.4195	600.432	1
Röle-6	0.3902	400.003	1.005

Toplam ti: 6.9175

4.7 Yapay Arı Kolonisi Algoritması Sonuçları

Çizelge 4.17’de Yapay arı kolonisi algoritmasına ait sonuçlar yer almaktadır.

Çizelge 4.17 Yapay arı kolonisi optimizasyon algoritması

Röle	td	Ip	ti
Röle-1	0.5919	122.942	1.737
Röle-2	0.4653	558.895	1.377
Röle-3	0.2677	419.453	1.030
Röle-4	0.3043	508.993	1.014
Röle-5	0.2298	608.489	1.029
Röle-6	0.3962	406.520	1.027
Toplam ti: 7.2139			

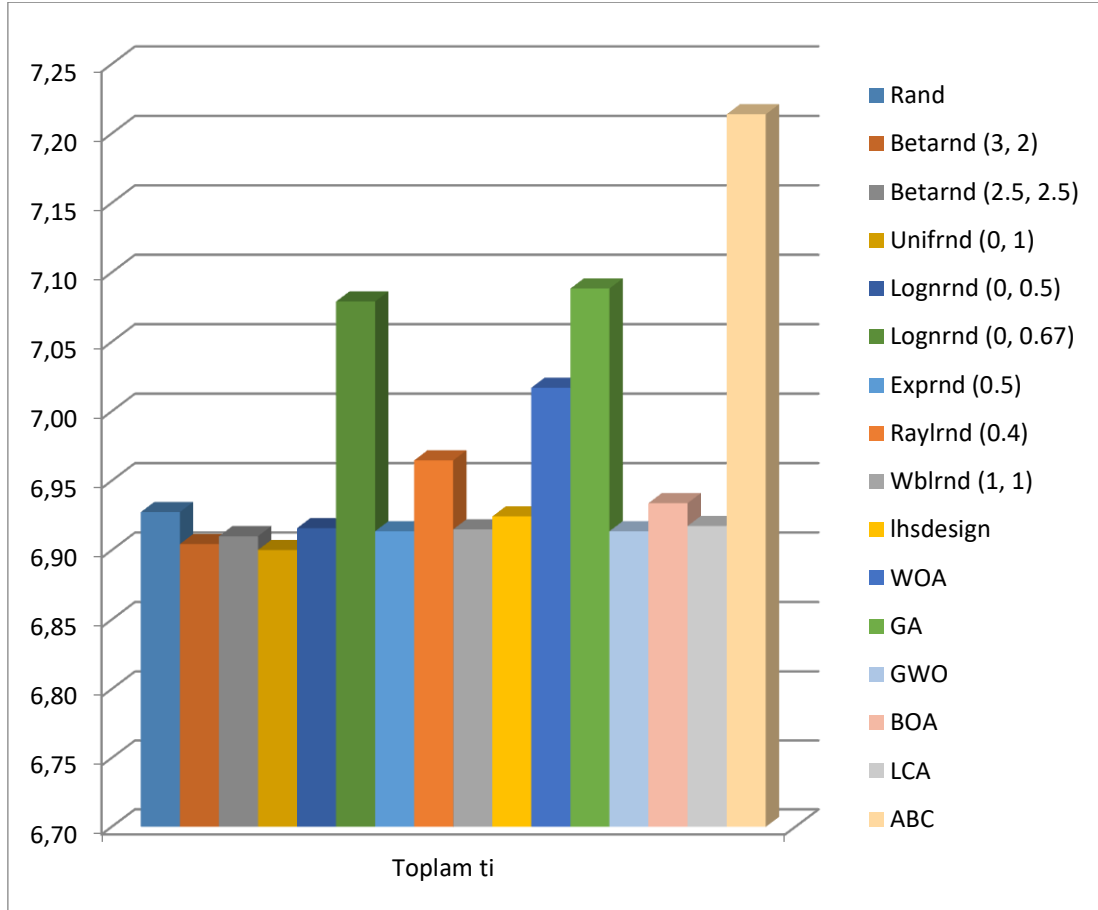
Çizelge 4.18’de diğer algoritmaların çalışma süreleri verilmiştir. Şekil 4.2’de ise tüm başlangıç popülasyon oluşturma yöntemlerinin ve algoritmaların sonuçları görsel olarak sunulmuştur.

Çizelge 4.18 Diğer algoritmaların sonuçları

Algoritma	Toplam ti
WOA	7.0170
GA	7.0884
GWO	6.9137
BOA	6.9339
LCA	6.9175
ABC	7.2139

Çizelge 4.18’de sunulan sonuçlara göre, ORC probleminde kullanılan altı farklı algoritmanın performansları karşılaştırılmıştır. Elde edilen "toplam ti" değerleri göz önünde bulundurulduğunda, GWO 6.9137 ile en düşük değeri elde etmiş ve bu yönüyle en başarılı algoritma olarak öne çıkmıştır. GWO’yu sırasıyla LCA ve BOA algoritmaları 6.9175 ve 6.9339 değerleriyle takip etmektedir. Bu üç algoritmanın oldukça yakın sonuçlar vermesi, söz konusu problem üzerinde benzer düzeyde etkin olduklarını göstermektedir. WOA ve GA ise sırasıyla 7.0170 ve 7.0884 değerleriyle orta

düzeyde performans sergilemiştir. Ayrıca ABC algoritması 7.2139 ile en yüksek toplam ti değerine sahip olmuş ve bu nedenle en düşük performansı gösteren algoritma olarak değerlendirilmektedir. Genel olarak, GWO algoritması bu problem için en uygun çözümleme yeteneğine sahipken, ABC algoritmasının etkinliğinin sınırlı olduğu söylenebilmektedir (Şekil 4.2).



Şekil 4.2 Tüm sonuçlar

4.8 Çok Modlu Kıyaslama Fonksiyonlarına Ait Sonuçlar

Çizelge 4.19’da algoritmaların çok modlu kıyaslama fonksiyonları olan F_8 , F_9 ve F_{10} üzerinde, 10 boyutlu olarak gerçekleştirilen testler sonucunda elde ettikleri performans sonuçları yer almaktadır.

Çizelge 4.19 Algoritmaların çok modlu kıyaslama fonksiyonlarına ait sonuçları

Algoritmalar	İstatistik	F8	F9	F10
WOA	En iyi	-3064.653817	0.000000	0.000000
	En Kötü	-2392.765984	13.929417	0.000000
	Ortalama	-2680.572323	2.785883	0.000000
	Standart Sapma	233.846962	5.571767	0.000000
GA	En iyi	-4189.339208	0.054395	0.219421
	En Kötü	-4188.721351	0.319178	0.316685
	Ortalama	-4189.088042	0.131037	0.271984
	Standart Sapma	0.241227	0.095505	0.032982
GWO	En iyi	-3714.558475	4.781283	0.000000
	En Kötü	-2128.247426	26.239287	0.000001
	Ortalama	-2820.993177	11.286047	0.000001
	Standart Sapma	521.651252	8.386970	0.000000
BOA	En iyi	-2003.863159	30.061203	17.804093
	En Kötü	-1521.276072	40.074684	18.661325
	Ortalama	-1770.096992	35.242423	18.245496
	Standart Sapma	165.671390	3.591669	0.274883
LCA	En iyi	-2954.657230	3.981315	0.004510
	En Kötü	-2307.509854	5.016456	3.026608
	Ortalama	-2531.448996	4.590515	1.389215
	Standart Sapma	221.050869	0.497014	1.139129
ABC	En iyi	-3657.449222	19.609910	7.926481
	En Kötü	-3317.728858	37.747538	12.585693
	Ortalama	-3500.908055	27.796526	9.646055
	Standart Sapma	125.429690	6.537001	1.749673

Çizelge 4.20’de, ISGuA’nın çok modlu kıyaslama fonksiyonları olan F_8 , F_9 ve F_{10} üzerinde, 2 boyutlu olarak gerçekleştirilen testler sonucunda elde ettiği performans sonuçları yer almaktadır.

Çizelge 4.20 Isgua'nın 2 boyutlu çok modlu kıyaslama fonksiyonlarına ait sonuçları

Algoritmalar	İstatistik	F8	F9	F10
Rand	En iyi	-837.965775	0.000000	0.000000
	En Kötü	-837.965775	0.000000	0.000000
	Ortalama	-837.965775	0.000000	0.000000
	Standart Sapma	0.000000	0.000000	0.000000
Betarnd (3, 2)	En iyi	-837.965775	0.000000	0.000000
	En Kötü	-837.965775	0.000000	0.000000
	Ortalama	-837.965775	0.000000	0.000000
	Standart Sapma	0.000000	0.000000	0.000000
Betarnd (2.5, 2.5)	En iyi	-601.089105	0.000000	0.000000
	En Kötü	-601.089105	0.000000	0.000000
	Ortalama	-601.089105	0.000000	0.000000
	Standart Sapma	0.000000	0.000000	0.000000
Unifrnd (0, 1)	En iyi	-837.965775	0.000000	0.000000
	En Kötü	-837.965775	0.000000	0.000000
	Ortalama	-837.965775	0.000000	0.000000
	Standart Sapma	0.000000	0.000000	0.000000
Lognrnd (0, 0.5)	En iyi	-837.965775	0.000000	0.000000
	En Kötü	-837.965775	0.000000	0.000000
	Ortalama	-837.965775	0.000000	0.000000
	Standart Sapma	0.000000	0.000000	0.000000
Lognrnd (0, 0.67)	En iyi	-719.527440	0.000000	0.000000
	En Kötü	-719.527440	0.000000	0.000000
	Ortalama	-719.527440	0.000000	0.000000
	Standart Sapma	0.000000	0.000000	0.000000
Exprnd (0.5)	En iyi	-719.527440	0.000000	0.000000
	En Kötü	-719.527440	0.000000	0.000000
	Ortalama	-719.527440	0.000000	0.000000
	Standart Sapma	0.000000	0.000000	0.000000
Raylrnd (0.4)	En iyi	-601.089105	0.000000	0.000000
	En Kötü	-601.089105	0.000000	0.000000
	Ortalama	-601.089105	0.000000	0.000000
	Standart Sapma	0.000000	0.000000	0.000000

Çizelge 4.20 Isgua'nın 2 boyutlu çok modlu kıyaslama fonksiyonlarına ait sonuçları (devam)

Algoritmalar	İstatistik	F8	F9	F10
Wblrnd (1, 1)	En iyi	-601.089105	0.000000	0.000000
	En Kötü	-601.089105	0.000000	0.000000
	Ortalama	-601.089105	0.000000	0.000000
	Standart Sapma	0.000000	0.000000	0.000000
lhsdesign	En iyi	-719.527440	0.000000	0.000000
	En Kötü	-719.527440	0.000000	0.000000
	Ortalama	-719.527440	0.000000	0.000000
	Standart Sapma	0.000000	0.000000	0.000000

Çizelge 4.21'de, ISGuA'nın çok modlu kıyaslama fonksiyonları olan F_8 , F_9 ve F_{10} üzerinde, 10 boyutlu olarak gerçekleştirilen testler sonucunda elde ettiği performans sonuçları yer almaktadır.

Çizelge 4.21 Isgua'nın 10 boyutlu çok modlu kıyaslama fonksiyonlarına ait sonuçları

Algoritmalar	İstatistik	F8	F9	F10
Rand	En iyi	-2077.050606	0.000000	0.000013
	En Kötü	-2077.050606	0.000000	0.000013
	Ortalama	-2077.050606	0.000000	0.000013
	Standart Sapma	0.000000	0.000000	0.000000
Betarnd (3, 2)	En iyi	-2018.805528	0.000000	0.000017
	En Kötü	-2018.805528	0.000000	0.000017
	Ortalama	-2018.805528	0.000000	0.000017
	Standart Sapma	0.000000	0.000000	0.000000
Betarnd (2.5, 2.5)	En iyi	-1917.381369	0.000000	0.000022
	En Kötü	-1917.381369	0.000000	0.000022
	Ortalama	-1917.381369	0.000000	0.000022
	Standart Sapma	0.000000	0.000000	0.000000
Unifrnd (0, 1)	En iyi	-2448.806987	0.000000	0.000034
	En Kötü	-2448.806987	0.000000	0.000034
	Ortalama	-2448.806987	0.000000	0.000034
	Standart Sapma	0.000000	0.000000	0.000000

Çizelge 4.21 Isgua'nın 10 boyutlu çok modlu kıyaslama fonksiyonlarına ait sonuçları (devam)

Algoritmalar	İstatistik	F8	F9	F10
Lognrnd(0, 0.5)	En iyi	-1654.887168	0.000000	0.000019
	En Kötü	-1654.887168	0.000000	0.000019
	Ortalama	-1654.887168	0.000000	0.000019
	Standart Sapma	0.000000	0.000000	0.000000
Lognrnd (0, 0.67)	En iyi	-2376.949472	0.000000	0.000038
	En Kötü	-2376.949472	0.000000	0.000038
	Ortalama	-2376.949472	0.000000	0.000038
	Standart Sapma	0.000000	0.000000	0.000000
Exprnd (0.5)	En iyi	-2421.309549	0.000000	0.000051
	En Kötü	-2421.309549	0.000000	0.000051
	Ortalama	-2421.309549	0.000000	0.000051
	Standart Sapma	0.000000	0.000000	0.000000
Raylrnd (0.4)	En iyi	-2827.777147	0.000000	0.000020
	En Kötü	-2827.777147	0.000000	0.000020
	Ortalama	-2827.777147	0.000000	0.000020
	Standart Sapma	0.000000	0.000000	0.000000
Wblrnd (1, 1)	En iyi	-2133.688168	0.000000	0.000068
	En Kötü	-2133.688168	0.000000	0.000068
	Ortalama	-2133.688168	0.000000	0.000068
	Standart Sapma	0.000000	0.000000	0.000000
lhsdesign	En iyi	-2405.592744	0.000000	0.000021
	En Kötü	-2405.592744	0.000000	0.000021
	Ortalama	-2405.592744	0.000000	0.000021
	Standart Sapma	0.000000	0.000000	0.000000

Yapılan çalışmada algoritmaların ve ISGuA'nın farklı başlatma yöntemlerinin başarımını değerlendirmek için 3 adet çok modlu kıyaslama fonksiyonu kullanılarak sonuçlar analiz edilmiştir. Analiz sonuçlarına göre 3 adet test fonksiyonundan 2'sinde Çizelge 4.22'de belirtildiği gibi optimum sonuca ulaşılmıştır.

Çizelge 4.22 Çok modlu kıyaslama fonksiyonları sonucu optimuma ulaşan algoritmalar

Optimuma Ulaşan Fonksiyon	Optimuma Ulaşan Algoritmalar ve Başlatma Yöntemleri
F_8	-
F_9	WOA, Rand, Betarnd (3, 2), Betarnd (2.5, 2.5), Unifrnd, Lognrnd (0, 0.5), Lognrnd (0, 0.67), Exprnd, Raylrnd, Wblrnd, lhsdesign
F_{10}	WOA, GWO

Çizelge 4.22 incelendiğinde optimum yakınlığı göz önüne alınarak WOA, 3 fonksiyonun 2'sinde en iyi sonucu elde etmiştir.

Çizelge 4.23 Çok modlu kıyaslama fonksiyonları sonucu optimuma en yakın sonuca ulaşan algoritmalar

Optimuma Yaklaşan Fonksiyon	Optimuma Yaklaşan Algoritmalar ve Başlatma Yöntemleri
F_8	GA, Raylrnd
F_9	-
F_{10}	Rand, Betarnd (3, 2), Betarnd (2.5, 2.5), Unifrnd, Lognrnd (0, 0.5), Lognrnd (0, 0.67), Exprnd, Raylrnd, Wblrnd, lhsdesign

Genel anlamda Raylrnd en iyi sonucu elde etmiştir. F_8 ve F_{10} 'da optimuma yakın sonuca ulaşmıştır. F_9 'da ise en iyi sonuca ulaşmıştır.



5. TARTIŞMA VE SONUÇ

Yapılan çalışma kapsamında aşırı akım rölle koordinasyon problemini çözmede SGuA geliştirilerek ISGuA ile detaylı analizi yapılmıştır. SGuA, biyolojik büyüme sürecinden esinlenerek geliştirilmiş ve çözüm uzayını dengeli bir şekilde tarayabilme potansiyeline sahip meta sezgisel optimizasyon algoritmalar arasında yer almaktadır. Bu potansiyelin etkin bir şekilde değerlendirilebilmesi, algoritmanın başlangıç popülasyonunun nasıl üretildiği ile ilgilidir. Bu bağlamda SGuA’da kullanılan 10 farklı başlatma yöntemi, algoritmanın performansı üzerinde etkili olmuştur.

En düşük t_i değeri 6.9002 ile uniform (eş olasılıklı) dağılım yöntemi bulunmuştur. Uniform dağılım yönteminin, algoritmanın parametre uzayını homojen ve dengeli bir şekilde örneklemesine olanak sağladığını göstermektedir. Uniform dağılım, her bir olası değerin eşit bir olasılıkla seçilmesini olanak tanıdığından, çözüm uzayında yoğun veya seyrek alanların oluşumunu engellemektedir. Bu durum, hem keşif aşamasında hem de çözüm aralığında etkin bir kapsama oranı sağlamaktadır. SGuA’nın yapısı, bu örnekleme yönteminin üzerinden doğrudan beslenmesi arama sürecinde yerel minimumlara takılma riskini azaltmakta ve daha kararlı bir yakınsama süreci sağlamaktadır. Bu bağlamda, uniform dağılım, algoritmanın performansını en üst düzeye çıkarmada önemli bir rol oynamıştır.

Beta dağılımlı başlatmalar, Betarnd (3,2) ve Betarnd (2.5, 2.5), SGuA için uygun sonuçlar verdiği görülmüştür. Beta dağılımları, parametre ayarlarına bağlı olarak örnekleme yoğunluğunu belirli bölgelere kaydırma yeteneğine sahiptir. Betarnd (3,2) dağılımı, arama uzayının alt bölgesine daha fazla odaklanırken, Betarnd (2.5, 2.5) dağılımı simetrik bir yapı sunmaktadır. Bu dağılımlar, parametre uzayının verimli bölgelerinde yoğunlaşmayı artırarak erken yakınsama sağlarken, uniform dağılımlar kadar geniş bir kapsayıcılık sunmadıkları için t_i değerleri sırasıyla 6.9045 ve 6.9101 olarak biraz daha yüksek kalmaktadır.

Log-normal dağılımlar, özellikle Lognrnd (0, 0.67) kullanıldığında, toplam t_i süresinin belirgin bir şekilde arttığını (7.0791) göstermektedir. Bu dağılımlar, verileri pozitif yönde asimetric olarak dağıtarak, çözüm uzayının bazı kısımlarında aşırı örnekleme yol açarken diğer alanların göz ardı edilmesine neden olmaktadır. Bu durum, arama süreçlerinin belirli bölgelerde sıkışmasına ve dolayısıyla yerel

minimumlara saplanma olasılığının artmasına neden olmaktadır. Ayrıca, Lognrnd (0, 0.5) ile yapılan analizlerde elde edilen sonuçlar daha olumlu bir performans sergilemektedir (6.9159), bu da daha düşük varyans durumunun sağladığı avantajı ortaya koymaktadır. Varyansın artması (0.67) performans kaybına sebep olmakta ve bu durum, teorik çerçevenin geçerliliğini desteklemektedir.

Rayleigh, Weibull ve Exponential dağılımlar ile yapılan başlatmalarda toplam ti, sırasıyla Rayleigh dağılımına göre 6.9647, Weibull dağılımına göre 6.9151 ve Exponential dağılımına göre 6.9137 olarak belirlenmiştir. Bu dağılımlar, genellikle doğada görülen enerji yayılımı ve ömür süreleri gibi tek yönlü süreçlerin modellenmesinde uygundur. Ancak, arama uzayında bu tek yönlülük ve asimetrik yapı, çeşitli bölgelerin yetersiz bir şekilde örneklenmesine sebep olarak, algoritmanın dengeli tarama yeteneğini olumsuz etkilemiştir ve dolayısıyla performansın düşmesine yol açmıştır.

Latin hypercube sampling (LHS), teorik olarak çözüm uzayını daha dengeli bir şekilde bölmeyi amaçlamakta; ancak uygulamada her hücreye yalnızca bir örnek düştüğü için rastgeleliğin etkisi altında kalabilmektedir. Bu durum, özellikle çok boyutlu problemler için yeterli derecede tutarlı bir örneklem sağlamada yetersiz kalabilmektedir. Sonuç olarak, LHS 6.9243 ve rastgele başlatma (Rand) 6.9275 değerleri ile uniform dağılımın gerisinde kalmıştır. Bu bulgu, daha sistematik ve dengeli bir örnekleme yapısının (uniform gibi) SGuA performansını önemli ölçüde artırabileceğini göstermektedir.

Başlatma yöntemlerinden elde edilen veriler doğrultusunda ISGuA, güncel ve yaygın olarak kullanılan diğer optimizasyon yöntemleri ile karşılaştırılmıştır. ABC algoritması, en yüksek ti değerine sahip algoritma olmuştur ve bu algoritmanın işlem süresi 7.2139 saniye olarak ölçülmüştür. ABC'nin doğası gereği, çözüm uzayında sık sık yön değiştirmesi ve bilgi paylaşım mekanizmasındaki zayıflıklar, fazla sayıda iterasyon ve tekrar örnekleme gerektirmesine yol açarak işlem süresini uzatmıştır. Ayrıca, arama sürecinde bireylerin birbirinden oldukça bağımsız hareket etmesi, algoritmanın küresel minimuma yakınsama hızını da yavaşlattığı görülmektedir.

GWO algoritması ise lider yönlendirmeli arama stratejisi sayesinde dengeli bir performans sergilemekte olup, 6.9137 saniye gibi bir süre ile önemli bir başarı

göstermiştir. Bu değer, ISGuA sonuçlarına en yakın sonuçlardan biri olarak değerlendirilmektedir.

BOA'nın ti değeri 6.9339 saniye olarak hesaplanmış ve bu sonuç, SGuA, GWO ve LCA algoritmalarının ardından orta düzeyde bir başarı göstermiştir. BOA, kelebeklerin feromon sinyallerini izleyerek optimum noktalara yönelmelerini simüle eden doğadan ilham alan bir algoritmadır. Keşif ve sömürü fazlarını sinyal yoğunluklarına göre dengeleyerek çeşitli optimizasyon problemlerinde rekabetçi sonuçlar elde edebilmekte, ancak aşırı keşif nedeniyle yerel minimumdan çıkışta kararsızlık gösterebilmektedir. Elde edilen süre, teorik olarak hızlı yakınsama kabiliyeti sunan algoritmaların beklenen performansının altında kalmıştır. Bu durumun başlıca nedeni, karmaşık çözüm uzaylarında BOA'nın tekil hareket stratejisinin yetersiz agresifliği nedeniyle sömürü yapamamasıdır.

LCA, WOA ve GA, çözüm kaliteleri ve toplam ti değerleri açısından değerlendirilmiştir. Her bir algoritmanın performansı, hem bu tez kapsamında elde edilen bulgulara hem de literatürdeki verilerle karşılaştırılmaktadır. Tez kapsamında LCA'nın elde ettiği toplam işletme süresi 6.9175 saniye olup, bu sonuç GWO ve ISGuA ile kıyaslandığında en iyi üçüncü sonuç olarak ortaya çıkmaktadır. LCA, rekabetçi yapısı sayesinde çözüm uzayında geniş bir arama gerçekleştirebilmekte ve bu sayede kısa sürede tatmin edici sonuçlara ulaşabilmektedir. Elde edilen bu sonuç, Özgüner ve Seyyarer (2023) tarafından gerçekleştirilen benzer bir çalışmada elde edilen 6.917 saniyelik LCA sonucu ile birebir örtüşmektedir. Her iki çalışmada elde edilen bu benzerlik, LCA'nın farklı başlangıç koşulları altında bile yüksek bir kararlılığa sahip olduğunu ve ORC gibi kısıtlı problemler için güvenilir bir alternatif sunduğunu göstermektedir.

WOA, bu tezde 7.0170 saniyelik ti değeri ile LCA ve SGuA'nın gerisinde kalmıştır. Bu durum, WOA'nın dairesel tarama mantığına dayanan keşif stratejisinin erken aşamalarda yeterince etkili çalışmamasından kaynaklanabileceği düşünülmektedir. Ayrıca, literatürde Özgüner ve Seyyarer (2023) tarafından yürütülen çalışmada WOA'nın toplam ti değeri 6.903 saniye olarak belirlenmiş ve dört algoritma arasında en hızlı çözüm süresi olarak raporlanmıştır. Bu farklılık; başlangıç popülasyonu dağılımı, sistem parametrelerinin sınırları, modelleme yaklaşımı ve kod yapısındaki optimizasyon gibi etkenlerden kaynaklanmaktadır.

GA ise bu tezde 7.0884 saniye ile dört algoritma içinde en yüksek işletme süresine sahip yöntem olmuştur. Bu sonuç, GA'nın geniş çözüm uzayını tararken özellikle mutasyon ve çaprazlama işlemleri sonucunda daha uzun sürede yakınsadığını göstermektedir. Ancak Özgüner ve Seyyarer'in (2023), çalışmasında GA'nın toplam işletme süresi 6.914 saniye olarak raporlanmıştır. Bu fark, algoritmanın doğasından ziyade başlangıç parametrelerine, problem uzayının kısıtlarına ve algoritmanın uygulandığı yazılımsal ortamın farklılıklarından kaynaklanmaktadır. GA gibi popülasyon temelli algoritmalarda rastgelelik faktörü (random seed), gen havuzunun çeşitliliği ve seçim stratejileri toplam süreyi ciddi biçimde etkileyebilmektedir. Bu nedenle, iki çalışmada performans değerleri tam olarak örtüşmemektedir.

Bu çalışmada, aşırı akım rölelerinin koordinasyon sorununun çözümüne yönelik geliştirilen ISGuA farklı başlangıç stratejileri ile değerlendirilmiş ve elde edilen performans değerleri üzerinden kapsamlı analizler gerçekleştirilmiştir. ISGuA'nın uniform dağılım (Unifrnd) ile başlatılması, 6.9002 saniyelik en düşük toplam işletme süresini sağlamıştır. Bu sonuç, ISGuA'nın parametre uzayını etkin bir biçimde tarayabildiğini ve çözüm alanında yüksek verimlilikle işlev gösterdiğini ortaya koymaktadır.

Ayrıca, ISGuA'nın performansı, güncel literatürde sıkça kullanılan Genetik Algoritma, Balina Optimizasyon Algoritması, Gri Kurt Algoritması, Lig Şampiyonluk Algoritması, Kelebek Optimizasyon Algoritması ve Yapay Arı Kolonisi Algoritması gibi diğer meta sezgisel yöntemler ile karşılaştırılmıştır. Yapılan karşılaştırmalı analizler neticesinde, ISGuA tüm algoritmalar arasında en düşük toplam ti süresine ulaşarak, en etkili optimizasyon yöntemi olarak öne çıkmıştır.

ORC problemi üzerindeki değerlendirmelerin ardından, ISGuA ve karşılaştırmaya dahil edilen tüm algoritmaların genel optimizasyon yeteneklerini daha nesnel ve standart bir zeminde analiz edebilmek amacıyla, literatürde sıkça kullanılan test fonksiyonları da çalışmaya dahil edilmiştir. Bu kapsamda seçilen çok modlu benchmark fonksiyonlar, algoritmaların yalnızca belirli bir probleme değil, geniş kapsamlı ve yapısal olarak farklı özellikler taşıyan senaryolara karşı da performanslarını ortaya koyma imkânı sunmuştur.

Çalışmada kullanılan test fonksiyonları olan Generalized Schwefel (F_8), Generalized Rastrigin (F_9) ve Ackley (F_{10}), çok modlu ve zorlu optimizasyon

problemleri olarak literatürde sıkça tercih edilen fonksiyonlardandır. Bu fonksiyonlar, algoritmaların arama-çıkarma dengesini ve küresel minimuma ulaşabilme yeterliliğini değerlendirmek açısından önemli kriterler sunmaktadır. Elde edilen bulgulara göre, F_9 ve F_{10} fonksiyonlarında birden fazla algoritmanın ve dağılım türünün küresel optimum değere ulaşabildiği; ancak F_8 fonksiyonu üzerinde hiçbir yöntemin bu başarıyı sağlayamadığı gözlemlenmiştir. Bu farklılıkların nedenleri, fonksiyonların yapısal özellikleri ve dağılımların/algoritmaların örnekleme stratejileriyle doğrudan ilişkili olduğu söylenebilmektedir.

F_9 ve F_{10} fonksiyonları, çok sayıda yerel minimum içermektedir ve küresel minimum genellikle arama uzayının merkezine yakın konumlanmaktadır. Bu özellik, örnekleme yoğunluğu merkez bölgelerde fazla olan dağılımların; örneğin Beta dağılımı, Log-normal dağılım ve meta sezgisel algoritmaların; örneğin WOA ve GWO'nun başarı oranını artırmaktadır. Özellikle WOA, rastgele örnekleme (Rand), Unifrnd, Beta dağılımları [Betarnd (3,2), Betarnd (2.5, 2.5)], Log-normal dağılımlar [Lognrnd (0,0.5) ve Lognrnd (0, 0.67)], Exponential (Exprnd), Rayleigh (Raylrnd), Weibull (Wblrnd) dağılımları ve Latin Hypercube Sampling (lhsdesign) yöntemi F_9 fonksiyonunda küresel optimuma ulaşmıştır. Bu dağılımların ortak yönü, arama uzayının önemli bölgelerini yeterli çeşitlilikle örnekleyerek algoritmaların yerel minimumlarda sıkışmadan optimum değere yönelmelerini kolaylaştırmasından kaynaklanmaktadır. Ayrıca LHS yöntemi, örneklerin uzaya homojen biçimde dağılmasını sağlayarak keşif yeteneğini artırmış ve bu sayede küresel minimuma erişimi kolaylaştırmıştır.

F_{10} fonksiyonu, üzerinde yürütülen optimizasyon süreçlerinde global optimum değere tam olarak ulaşamamakla birlikte, oldukça yakın sonuçlar üretmiştir. Ackley fonksiyonu; geniş çözüm uzayında çok sayıda sık yerel minimum içeren ve yalnızca merkezde dar bir bölgede konumlanan keskin global minimuma sahip olması nedeniyle, özellikle başlangıç örnekleme duyarlılığı yüksek olan bir fonksiyon türüdür.

Uniform, Latin Hypercube Sampling (LHS), rastgele (Rand) ve Beta, Log-normal gibi dağılımlara dayalı başlatmalar; çözüm uzayının genelinde dengeli veya varyasyonlu dağılım sağlamış olsalar da, çözüm popülasyonunun yeterli yoğunlukta global optimum çevresine düşmesini garanti edememiştir. Bu durum, algoritmanın başlangıç popülasyonuna bağlı olarak global minimumun bulunduğu bölgeye erken

erişimini zorlaştırmış ve sömürü mekanizmasının lokal bölgelerde yoğunlaşmasına neden olmuştur.

Bununla birlikte, kullanılan örnekleme stratejileri ile elde edilen sonuçlar, global optimum değere oldukça yakınsamakta olup; bu durum, algoritmaların genel arama ve yakınsama kapasitesinin yeterli düzeyde olduğunu göstermektedir. Ancak Ackley gibi dar bölgede optimum içeren çok modlu fonksiyonlarda, başlangıç popülasyonunun yalnızca dengeli değil, aynı zamanda problemin karakteristik yapısına özgü şekilde uyarlanması gerektiği bir kez daha ortaya konmuştur.

Ancak F_8 fonksiyonu üzerinde yapılan çalışmalarda, hiçbir yöntemin optimum değere ulaşamadığı görülmüştür. Bu durumun temel nedeni, bu fonksiyonunun oldukça aldatıcı bir yapıya sahip olmasıdır. Fonksiyonun küresel minimumu, arama uzayının sınırlarına yakın bir konumda yer almaktadır. Bu yapı, hem yoğun yerel minimumlara sahip olması hem de küresel minimumun merkezden uzak bir bölgede bulunması nedeniyle, çoğu algoritmanın veya dağılımın bu noktaya ulaşmasını güçleştirmektedir. Özellikle merkezi yoğunlaşmaya sahip dağılımlar; örneğin Gaussian, Log-normal veya küresel yapıdan çok yerel yapıların etkisinde çalışan meta sezgisel algoritmalar bu tip fonksiyonlarda yetersiz kalmaktadır. F_8 fonksiyonu gibi uç noktalarda optimum içeren fonksiyonlar, merkezi yoğunluklu örnekleme stratejilerine sahip yöntemler için oldukça zorlayıcıdır. Bu nedenle, ne WOA ne de GWO gibi başarılı meta sezgisel algoritmalar, bu fonksiyon üzerinde başarılı olamamıştır.

F_8 fonksiyonu için gözlenen bir diğer önemli bulgu ise, GA ve Raylrnd yöntemlerinin optimum değere ulaşamasa da en yakın sonuçları elde etmişlerdir. Bu başarıda, GA'nın sahip olduğu mutasyon ve çaprazlama mekanizmalarının geniş arama uzayını tarayabilme potansiyeli etkili olmaktadır. Ayrıca, Raylrnd yöntemi ile elde edilen sonuçların global optimuma oldukça yakınsaması, dağılımın yüksek keşif kapasitesi sunduğunu ve algoritmanın çözüm uzayının geniş bölgelerini tarayabildiğini göstermektedir. Bu bulgu, uygun örnekleme ile desteklenen meta sezgisel yapıların, uç noktalarda yer alan zorlu global optimumlara dahi yakın çözümler üretebildiğini ortaya koymaktadır. Ancak optimuma tam erişim için, dağılımın çözüm uzayının tamamına yönelik kapsayıcılığının artırılması veya yönlü çeşitlendirilmesi önerilmektedir. Bu iki yöntemin diğerlerinden farklı olarak uç değerlere ulaşma olasılıklarının daha fazla olması, Schwefel fonksiyonunda görece daha iyi performans göstermelerini sağlamıştır.

GA, doğadaki evrimsel süreçleri modelleyerek çözüm arayan popülasyon tabanlı bir optimizasyon yöntemidir. GA, çözüm kümesindeki bireyler üzerinde gerçekleştirdiği çaprazlama, mutasyon ve seçim operatörleriyle arama alanında ilerlemektedir. Bu algoritmanın temel avantajı, geniş çözüm uzaylarında çeşitliliği koruyarak yerel minimumlara sıkışmadan küresel optimuma ulaşabilme potansiyelidir. Ancak yapılan deneysel çalışmalarda, GA'nın her fonksiyon türünde bu potansiyelini eşit düzeyde sergileyemediği gözlemlenmiştir.

Ancak F_8 fonksiyonu gibi karmaşık yapıya sahip, çok sayıda yerel minimum barındıran ve optimum değeri uzayın kenarlarında bulunan fonksiyonlar için GA'nın kısmi başarısı dikkat çekmektedir. F_8 fonksiyonu, fonksiyon yüzeyinin yanıltıcı yapısı ve optimum noktanın merkezden uzak olması nedeniyle klasik optimizasyon yöntemleri açısından oldukça zorlayıcıdır. Ancak GA, sahip olduğu mutasyon operatörü sayesinde çözüm uzayının ekstrem noktalarını keşfetme eğilimindedir. Bu mekanizma, optimumun uzayın sınır bölgelerinde yer aldığı F_8 gibi fonksiyonlarda GA'nın diğer algoritmalarla kıyasla daha yakın sonuçlar elde etmesini sağlamıştır. Ayrıca çaprazlama operatörü ile bireyler arasında yeni kombinasyonların oluşturulması, çeşitli çözümlerin denenmesini mümkün kılmış ve fonksiyonun farklı bölgelerinin keşfini desteklemiştir. GA, F_8 fonksiyonunda tam olarak optimum değere ulaşamasa da, en yakın değerlere ulaşabilen yöntemlerden biri olması, algoritmanın geniş kapsamlı arama yeteneğinin bir göstergesi olmaktadır.

Bununla birlikte, F_9 ve F_{10} fonksiyonlarında GA'nın performansı başarısız olmuştur. Bu fonksiyonların ortak özelliklerinden biri, küresel minimumun genellikle arama uzayının merkezinde yer alması ve çevresinde yoğun şekilde yerel minimumların bulunmasıdır. GA, çözüm havuzunun evrimsel süreçle ilerlemesine dayandığı için yönlendirilmiş bir merkez arama mekanizmasına sahip değildir. Dolayısıyla, optimum değer merkezde yer aldığı bu tür fonksiyonlarda GA'nın bireyleri merkez etrafında yoğunlaşmak yerine farklı yönler dağılabilmekte ve bu durum, çözümün yerel minimumlarda takılmasına neden olabilmektedir. F_9 fonksiyonu, periyodik ve düzenli yapısıyla birçok algoritmayı yanıltıcı yönler çekebilmektedir. Aynı şekilde, F_{10} fonksiyonu da düz alanlar ile ani değişim bölgeleri arasında geçiş barındırması nedeniyle GA'nın arama sürecinde yetersiz kalmasına yol açmıştır. Ayrıca GA'nın başarısızlığında etkili olan diğer bir unsur, genetik çeşitliliğin zamanla azalmasıyla

ortaya çıkan erken yakınsama problemidir. Özellikle Rastrigin ve Ackley gibi fonksiyonlarda, başlangıçta rastgele dağılan bireyler zamanla benzer hale gelir ve popülasyonun tamamı belirli bir çözüm çevresinde yoğunlaşır. Bu durum, fonksiyon yüzeyindeki keşfedilmemiş alanların göz ardı edilmesine ve algoritmanın küresel optimum yerine yerel çözümlerde sıkışmasına neden olmaktadır. Arama uzayında yönlendirme veya dinamik parametre güncellemesi gibi mekanizmaların bulunmaması, GA'nın özellikle çok modlu fonksiyonlara karşı dayanıklılığını azaltmaktadır.

LCA, BOA ve ABC algoritmalarının F_8 , F_9 ve F_{10} fonksiyonlarında istenilen performansı gösterememelerinin temel nedenleri, bu algoritmaların arama stratejileri ile söz konusu fonksiyonların yapısal özellikleri arasındaki uyumsuzluktan kaynaklanmaktadır.

Literatürde LCA'nın keşif-sömürü dengesini başarıyla koruyamadığı ve yerel minimumlara kolayca takıldığı belirtilmektedir. Bu algoritma, popülasyon içindeki en iyi takımı merkeze alarak diğer çözümleri onun etrafında iyileştirmeye çalışmaktadır. Bu durum aşırı sömürüye yol açarak aramanın erken aşamada dar bir alana yoğunlaşmasına neden olur; dolayısıyla yetersiz keşif sonucu farklı bölgeler yeterince araştırılmaz hale gelmektedir. Sonuç olarak LCA, bir yerel en iyi çözüme ulaştığında bunu küresel optimum sanarak orada sıkışıp kalma eğilimi göstermektedir. Algoritmada parametre ayarları örneğin eşleşme planı, sezon sayısı sabit olup adaptif bir yönlendirme mekanizması bulunmadığından, arama süreci problemin yapısına göre dinamik olarak yönlendirilememektedir. Parametrik esneklik eksikliği, LCA'nın farklı problem yapılarına uyum sağlamasını zorlaştırmaktadır. Bu nedenle LCA, karmaşık ve çok modlu fonksiyonlarda küresel optimumdan uzakta bir çözümde takılı kalabilmektedir. Belirtilen zayıflıklar, özellikle Generalized Schwefel (F_8), Generalized Rastrigin (F_9) ve Ackley (F_{10}) gibi zorlu test fonksiyonlarında LCA'nın küresel optimuma ulaşamamasının temel nedenleridir. F_8 fonksiyonu, arama uzayının uzak bir köşesinde yer alan küresel en iyi çözüme ve bu çözümden oldukça uzakta bulunan yanıltıcı ikinci en iyi çözüm değerine sahiptir; birçok arama algoritması bu ikinci en iyi noktada tuzağa düşmektedir. LCA'da güçlü bir keşif mekanizması olmadığı için bu yanıltıcı durumdan çıkıp uzak bölgedeki gerçek optimuma yönelememektedir.

F_9 fonksiyonu ise her yönde periyodik dalgalanmalardan ötürü sayısız yerel minimum içerir. LCA'nın erken yoğunlaşma eğilimi, bu fonksiyonda bir yerel

minimum çevresinde popülasyonun toplanmasına yol açarak gerçek optimum (tüm değişkenlerin 0 olduğu nokta) etrafındaki engelleri aşmasını engellemektedir. Benzer şekilde F_{10} fonksiyonu da çok modlu bir yapıya sahip olup dış bölgelerinde oldukça düz bir plato ve merkezinde dar bir küresel çukur barındırmaktadır. LCA, keşif yeteneği kısıtlı olduğundan Ackley'nin düz dış bölgesinde ilerleme kaydedemeyip küçük salınımlı yerel minimumlarda takılabilir; küresel optimum bölgesine ulaşması güçleşmektedir. Sonuç olarak, LCA'nın keşif-sömürü dengesindeki bozukluk ve uyarlanabilirlik eksikliği, bu fonksiyonların çok modlu ve hileli yapıları karşısında algoritmanın küresel en iyi çözümü bulamamasına neden olmaktadır.

BOA, kelebeklerin koku alma ve eş bulma davranışlarını taklit eden, popülasyona dayalı nispeten yeni bir meta sezgisel algoritmadır. BOA'nın temelinde, bireylerin küresel en iyi çözüme doğru çekilmesi veya rastgele diğer bireylere doğru yönelmesi gibi iki aşamalı bir arama yaklaşımı yatmaktadır. Ancak BOA algoritmasının nüfus çeşitliliğini hızla yitirme ve erken dönemde yerel minimum tuzağına düşme eğilimi bulunmaktadır. Bunun nedeni, BOA'nın keşif-keşif oranının dengeli olmamasıdır. Algoritma, yüksek koku yoğunluğuna sahip mevcut en iyi bireylere doğru yoğun şekilde sömürü yapar ve keşif bileşeni giderek zayıflar. Özellikle sabit tutulan veya uygun ayarlanmayan parametreler algoritmanın davranışını esnek olmayan bir biçimde belirlemektedir. Sonuçta BOA popülasyonu, arama uzayının sadece sınırlı bir bölgesine toplanıp diğer bölgeleri yeterince araştırmadan hızlı bir yakınsamaya uğrayabilmektedir. Bu yönlendirme esnekliği eksikliği, algoritmanın gerektiğinde arama stratejisini değiştirememesi anlamına gelmektedir ve BOA, kompleks fonksiyonlarda bir kez yerel bir çözümde kilitlenirse bundan kurtulmakta zorlanmaktadır. Algoritmada, yerel optimumdan çıkışı sağlamak üzere tasarlanmış güçlü bir mekanizma bulunmadığı için, yerel minimumlara karşı dayanıklılığı sınırlıdır. Bu yapısal zafiyetler, BOA'nın Schwefel, Rastrigin ve Ackley gibi zorlu fonksiyonlarda küresel en iyi değeri bulamamasına yol açmaktadır. Sonuç olarak BOA, gerekli adaptif arama derinliğini ve genişliğini sağlayamadığında bu test fonksiyonlarının yapısal zorlukları karşısında küresel optimuma ulaşamamaktadır.

ABC algoritması ise, literatürde keşif konusunda başarılı fakat sömürü konusunda zayıf bir yöntem olarak tanımlanmıştır; bu da optimum çevresinde ince ayar yapıp kesin küresel değere ulaşmakta yetersiz kalabileceğini göstermektedir. ABC'de

işçi ve gözcü arıların kullandığı çözüm güncelleme denklemi aynıdır ve mevcut çözüm etrafında rastgele bir yönelim içermektedir. Bu yaklaşım, algoritmanın yönlendirme esnekliğini kısıtlamaktadır; çünkü farklı evrelerde keşif veya sömürü ağırlığını dinamik olarak değiştirememektedir. Parametre seçimlerine duyarlılık yüksektir. Uygun olmayan parametre ayarlarında, ya arılar bir bölgeyi gereğinden fazla sömürüp çeşitliliği kaybeder ya da çok sık rastgele arama yapıp kararsız bir arama süreci sergilemektedirler. Ayrıca ABC algoritması, bulduğu daha iyi çözümün fitness değerini her adımda iyileşme koşuluyla kabul eder; bu nedenle eğer popülasyon bir alt optimumda kitlenmişse, mevcut çözümleri geçici olarak kötüleştirmeyi göze alarak o tuzaktan çıkarmak zordur. Keşifçi arı mekanizması bir miktar rastgele sıçrama sağlasa da, bu genellikle popülasyonun küçük bir kısmını etkilemektedir. Sonuç olarak ABC'nin yerel minimumlara dayanıklılığı sınırlı kalabilir ve hızlı küresel arama gerektiren senaryolarda temel haliyle yetersiz performans gösterebilmektedir. Bu nedenlerle ABC algoritması, Schwefel, Rastrigin ve Ackley gibi zorlu test fonksiyonlarında küresel optimumu bulmakta güçlük çekmektedir.

Sonuç olarak tez kapsamında gerçekleştirilen deneysel çalışmalar, meta sezgisel algoritmaların ve çeşitli örnekleme tabanlı başlangıç stratejilerinin, çok modlu yapıdaki optimizasyon problemlerinde gösterdiği başarıyı karşılaştırmalı olarak ortaya koymuştur. Generalized Schwefel (F_8), Generalized Rastrigin (F_9) ve Ackley (F_{10}) fonksiyonları üzerinde yapılan değerlendirmelerde, algoritmaların fonksiyonların yapısal özellikleriyle olan uyumlarının çözüm başarımını doğrudan etkilediği gözlemlenmiştir.

Başlangıç yöntemleri incelendiğinde, özellikle Rayleigh dağılımı (Raylnd), Schwefel fonksiyonunda optimuma oldukça yakın sonuçlar üretmiş ve bu fonksiyon üzerinde en iyi başarıyı gösteren yöntemlerden biri olmuştur. GA algoritması, aynı fonksiyon üzerinde yine optimuma yakın çözümler üretmiş ve örnekleme ile algoritma etkileşiminin önemini vurgulamıştır. Unifrnd, Betarnd (3,2), Betarnd (2.5, 2.5), Lognrnd (0, 0.5), Lognrnd (0, 0.67), Exprnd, Wblrnd ve lhsdesign gibi diğer örnekleme yöntemleri ise özellikle merkezde optimum barındıran Rastrigin ve Ackley gibi fonksiyonlarda optimuma yaklaşan sonuçlar üretmiştir. Bununla birlikte, aldatıcı yapıya sahip ve optimumu uç bölgelerde yer alan Schwefel fonksiyonu karşısında bu yöntemlerin başarı düzeyinin düştüğü gözlemlenmiştir.

Genel deęerlendirme sonucunda, başarılı bir optimizasyon performansının yalnızca algoritmanın yapısına deęil, aynı zamanda başlangıç koşullarına ve örnekleme stratejisine de baęlı olduęu görölmüştür. Bu çalışma, algoritmaların seçiminin problem türüne özgü olarak yapılması gerektiğini ve örnekleme yöntemlerinin stratejik kullanımının, küresel optimuma ulaşmada belirleyici bir rol oynayabileceğini ortaya koymuştur. Özellikle Rayleigh gibi uç noktalara erişimi yüksek daęılımlar, geniş çözüm uzaylarında yer alan optimumlara ulaşmada daha etkili olmuş, LHS ve beta temelli daęılımlar ise merkezi yoğunluklu fonksiyonlarda anlamlı katkı sağlamıştır.





KAYNAKLAR

- Abdel-Basset, M., Abdel-Fatah, L., & Sangaiah, A. K. (2018). Metaheuristic algorithms: A comprehensive review. *Computational intelligence for multimedia big data on the cloud with engineering applications*, 185-231.
- Abdel-Basset, M., El-Shahat, D., Jameel, M., & Abouhawwash, M. (2023). Young's double-slit experiment optimizer: A novel metaheuristic optimization algorithm for global and constraint optimization problems. *Computer Methods in Applied Mechanics and Engineering*, 403, 115652.
- Abualigah, L., Diabat, A., Mirjalili, S., Abd Elaziz, M., & Gandomi, A. H. (2021). The arithmetic optimization algorithm. *Computer methods in applied mechanics and engineering*, 376, 113609.
- Agushaka, J. O., Ezugwu, A. E., & Abualigah, L. (2022). Dwarf mongoose optimization algorithm. *Computer methods in applied mechanics and engineering*, 391, 114570.
- Agushaka, J. O., Ezugwu, A. E., & Abualigah, L. (2023). Gazelle optimization algorithm: a novel nature-inspired metaheuristic optimizer. *Neural Computing and Applications*, 35(5), 4099-4131.
- Agushaka, J. O., Ezugwu, A. E., Abualigah, L., Alharbi, S. K., & Khalifa, H. A. E. W. (2023). Efficient initialization methods for population-based metaheuristic algorithms: A comparative study. *Archives of Computational Methods in Engineering*, 30(3), 1727-1787.
- Akay, B. (2009). *Nümerik optimizasyon problemlerinde yapay arı kolonisi (artificial bee colony) algoritmasının performans analizi (Performance analysis of artificial bee colony algorithm on numerical optimization problems)* (Doctoral dissertation, Erciyes University, Turkey).
- Akdağ, O., & Yeroğlu, C. (2019). Güç sistemlerinde aşırı akım koruma koordinasyon modelinin oluşturulması, benzetimi ve optimizasyonu. *Erzincan University Journal of Science and Technology*, 12(1), 202-214.
- Akyol, S., & Alatas, B. (2017). Plant intelligence based metaheuristic optimization algorithms. *Artificial Intelligence Review*, 47, 417-462.
- Alam, M. N., Das, B., & Pant, V. (2015). A comparative study of metaheuristic optimization approaches for directional overcurrent relays coordination. *Electric Power Systems Research*, 128, 39-52.
- Amraee, T. (2012). Coordination of directional overcurrent relays using seeker algorithm. *IEEE Transactions on Power Delivery*, 27(3), 1415-1422.
- Angeline, P. J. (1994). Book Review: Genetic programming: On the programming of computers by means of natural selection,. John R. Koza, A Bradford Book, MIT Press, Cambridge MA, 1992, ISBN 0-262-11170-5, xiv+ 819pp., US\$55.00. *Biosystems*, 33(1), 69-73.
- Arora, S., & Singh, S. (2019). Butterfly optimization algorithm: a novel approach for global optimization. *Soft computing*, 23, 715-734.
- Atashpaz-Gargari, E., & Lucas, C. (2007, September). Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. In **2007 IEEE congress on evolutionary computation** (pp. 4661-4667). Ieee.
- Atsever, M. B., Karacasu, O., & Hocaoglu, M. H. (2021, December). Performance Analysis of Different Optimization Algorithms on Overcurrent Relay

- Coordination Problem in Distribution Systems. In **2021 11th Smart Grid Conference (SGC)** (pp. 1-6). IEEE.
- Atsever, M. B., & Hocaoglu, M. H. (2022). Aşırı Akım Röle Koordinasyonu için Bozkurt Algoritmasının Kapsamlı Performans Analizi. *Türk Mühendislik Araştırma ve Eğitimi Dergisi*, 1(2), 52-61.
- Aydın, A. (2009). *Metasezgisel yöntemlerle uçak çizelgeleme problemi optimizasyonu* (Doctoral dissertation, Marmara Üniversitesi (Turkey)).
- Ayyarao, T. S., Ramakrishna, N. S. S., Elavarasan, R. M., Polumahanthi, N., Rambabu, M., Saini, G., ... & Alatas, B. (2022). War strategy optimization algorithm: a new effective metaheuristic algorithm for global optimization. *Ieee Access*, 10, 25073-25105.
- Bayraktar, Z., Komurcu, M., Bossard, J. A., & Werner, D. H. (2013). The wind driven optimization technique and its application in electromagnetics. *IEEE transactions on antennas and propagation*, 61(5), 2745-2757.
- Bedekar, P. P., Bhide, S. R., & Kale, V. S. (2009, December). Optimum coordination of overcurrent relays in distribution system using genetic algorithm. In **2009 International Conference on Power Systems** (pp. 1-6). IEEE.
- Beyer, H. G., & Schwefel, H. P. (2002). Evolution strategies—a comprehensive introduction. *Natural computing*, 1, 3-52.
- Blocho, M. (2020). Heuristics, metaheuristics, and hyperheuristics for rich vehicle routing problems. In **Smart delivery systems** (pp. 101-156). Elsevier.
- Braik, M., Hammouri, A., Atwan, J., Al-Betar, M. A., & Awadallah, M. A. (2022). White Shark Optimizer: A novel bio-inspired meta-heuristic algorithm for global optimization problems. *Knowledge-Based Systems*, 243, 108457.
- Braik, M., Sheta, A., & Al-Hiary, H. (2021). A novel meta-heuristic search algorithm for solving optimization problems: capuchin search algorithm. *Neural computing and applications*, 33(7), 2515-2547.
- Castro, L. D., & Timmis, J. I. (2003). Artificial immune systems as a novel soft computing paradigm. *Soft computing*, 7, 526-544.
- Chao, K. H., & Li, J. Y. (2022). Global maximum power point tracking of photovoltaic module arrays based on improved artificial bee colony algorithm. *Electronics*, 11(10), 1572.
- Chong, E. (2013). *An Introduction to Optimization*. Wiley.
- Comert, S. E., & Yazgan, H. R. (2023). A new approach based on hybrid ant colony optimization-artificial bee colony algorithm for multi-objective electric vehicle routing problems. *Engineering Applications of Artificial Intelligence*, 123, 106375.
- Corno, F., Reorda, M. S., & Squillero, G. (1998, May). A new evolutionary algorithm inspired by the selfish gene theory. In **1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No. 98TH8360)** (pp. 575-580). IEEE.
- Erol, O. K., & Eksin, I. (2006). A new optimization method: big bang–big crunch. *Advances in engineering software*, 37(2), 106-111.
- Eusuff, M. M., & Lansey, K. E. (2003). Optimization of water distribution network design using the shuffled frog leaping algorithm. *Journal of Water Resources planning and management*, 129(3), 210-225.

- Dehghani, M., Montazeri, Z., Trojovská, E., & Trojovský, P. (2023). Coati Optimization Algorithm: A new bio-inspired metaheuristic algorithm for solving optimization problems. *Knowledge-Based Systems*, 259, 110011.
- Dorigo, M., & Stützle, T. (2019). *Ant colony optimization: overview and recent advances* (pp. 311-351). Springer International Publishing.
- Gaikwad, A. D., & Singh, K. (2022). Improved League Championship Algorithm (ILCA) for Load Balancing in Cloud Computing. *International Journal of Next- Generation Computing*, 13(5).
- Geem, Z. W., Kim, J. H., & Loganathan, G. V. (2001). A new heuristic optimization algorithm: harmony search. *simulation*, 76(2), 60-68.
- Gokhale, S. S., & Kale, V. S. (2016). An application of a tent map initiated Chaotic Firefly algorithm for optimal overcurrent relay coordination. *International Journal of Electrical Power & Energy Systems*, 78, 336-342.
- Goldbogen, J. A., Friedlaender, A. S., Calambokidis, J., Mckenna, M. F., Simon, M., & Nowacek, D. P. (2013). Integrative approaches to the study of baleen whale diving behavior, feeding performance, and foraging ecology. *BioScience*, 63(2), 90-100.
- Govan, A. (2006). Introduction to optimization. In **North Carolina State University, SAMSI NDHS, Undergraduate workshop** (Vol. 3, p. 6).
- Hashim, F. A., Houssein, E. H., Hussain, K., Mabrouk, M. S., & Al-Atabany, W. (2022). Honey Badger Algorithm: New metaheuristic algorithm for solving optimization problems. *Mathematics and Computers in Simulation*, 192, 84-110.
- Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., & Chen, H. (2019). Harris hawks optimization: Algorithm and applications. *Future generation computer systems*, 97, 849-872.
- Karababa, S., (2013). "Harmoniklerin dijital aşırı akım röleleri üzerine etkisinin incelenmesi", Yayınlanmış Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, 10-35.
- Karaboga, D. (2005). *An idea based on honey bee swarm for numerical optimization* (Vol. 200, pp. 1-10). Technical report-tr06, Erciyes university, engineering faculty, computer engineering department.
- Karaboga, D., & Akay, B. (2007, June). Artificial bee colony (ABC) algorithm on training artificial neural networks. In **2007 IEEE 15th Signal Processing and Communications Applications** (pp. 1-4). IEEE.
- Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of global optimization*, 39, 459-471.
- Karaboga, D., & Akay, B. (2009). A comparative study of artificial bee colony algorithm. *Applied mathematics and computation*, 214(1), 108-132.
- Karci, A. (2007, October). Natural inspired computational intelligence method: saplings growing up algorithm. In **2007 IEEE International Conference on Computational Cybernetics** (pp. 221-226). IEEE.
- Karci, A., & Alatas, B. (2006, September). Thinking capability of saplings growing up algorithm. In **International Conference on Intelligent Data Engineering and Automated Learning** (pp. 386-393). Berlin, Heidelberg: Springer Berlin Heidelberg.

- Kashan, A. H. (2009, December). League championship algorithm: a new algorithm for numerical function optimization. In **2009 International Conference of Soft Computing and Pattern Recognition** (pp. 43-48). IEEE.
- Kashan, A. H. (2014). League Championship Algorithm (LCA): An algorithm for global optimization inspired by sport championships. *Applied Soft Computing*, 16, 171-200.
- Kaur, S., Awasthi, L. K., Sangal, A. L., & Dhiman, G. (2020). Tunicate Swarm Algorithm: A new bio-inspired based metaheuristic paradigm for global optimization. *Engineering Applications of Artificial Intelligence*, 90, 103541.
- Kennedy, J., & Eberhart, R. (1995, November). Particle swarm optimization. In **Proceedings of ICNN'95-International Conference on Neural Networks** (Vol. 4, pp. 1942-1948). IEEE.
- Keshav R. Sharma [Internet]. 2022 [Cited 2023 Sep 21]. Available from: <https://vidyasheela.com/post/gentle-introduction-to-genetic-algorithm>
- Khurshaid, T., Wadood, A., Farkoush, S. G., Yu, J., Kim, C. H., & Rhee, S. B. (2019). An improved optimal solution for the directional overcurrent relays coordination using hybridized whale optimization algorithm in complex power systems. *IEEE Access*, 7, 90418-90435.
- Kumar, M., Husain, D. M., Upreti, N., & Gupta, D. (2010). Genetic algorithm: Review and application. Available at SSRN 3529843.
- Kumar, N., Shaikh, A. A., Mahato, S. K., & Bhunia, A. K. (2021). Applications of new hybrid algorithm based on advanced cuckoo search and adaptive Gaussian quantum behaved particle swarm optimization in solving ordinary differential equations. *Expert Systems with Applications*, 172, 114646.
- Lam, A. Y., & Li, V. O. (2012). Chemical reaction optimization: a tutorial. *Memetic Computing*, 4, 3-17.
- Langazane, S. N., & Saha, A. K. (2022). Effects of particle swarm optimization and genetic algorithm control parameters on overcurrent relay selectivity and speed. *IEEE Access*, 10, 4550-4567.
- Li, Q., Liu, S. Y., & Yang, X. S. (2020). Influence of initialization on the performance of metaheuristic optimizers. *Applied Soft Computing*, 91, 106193.
- Makhadmeh, S. N., Al-Betar, M. A., Abasi, A. K., Awadallah, M. A., Doush, I. A., Alyasseri, Z. A. A., & Alomari, O. A. (2023). Recent advances in butterfly optimization algorithm, its versions and applications. *Archives of Computational Methods in Engineering*, 30(2), 1399-1420.
- Mathew, T. V. (2012). Genetic algorithm. *Report submitted at IIT Bombay*, 53.
- Meskin, M., Domijan, A., & Grinberg, I. (2015). Optimal co-ordination of overcurrent relays in the interconnected power systems using break points. *Electric Power Systems Research*, 127, 53-63.
- MiarNaeimi, F., Azizyan, G., & Rashki, M. (2021). Horse herd optimization algorithm: A nature-inspired algorithm for high-dimensional optimization problems. *Knowledge-Based Systems*, 213, 106711.
- Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in engineering software*, 69, 46-61.
- Mirjalili, S. (2015). The ant lion optimizer. *Advances in engineering software*, 83, 80-98.
- Mirjalili, S. (2015). How effective is the Grey Wolf optimizer in training multi-layer perceptrons. *Applied intelligence*, 43, 150-161.

- Mirjalili, S. (2015). Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowledge-based systems*, 89, 228-249.
- Mirjalili, S., Gandomi, A. H., Mirjalili, S. Z., Saremi, S., Faris, H., & Mirjalili, S. M. (2017). Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Advances in engineering software*, 114, 163-191.
- Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in engineering software*, 95, 51-67.
- Mohammed, H., & Rashid, T. (2020). A novel hybrid GWO with WOA for global numerical optimization and solving pressure vessel design. *Neural Computing and Applications*, 32(18), 14701-14718.
- Noghabi, A. S., Sadeh, J., & Mashhadi, H. R. (2009). Considering different network topologies in optimal overcurrent relay coordination using a hybrid GA. *IEEE Transactions on Power Delivery*, 24(4), 1857-1863.
- Onat, N. (2018). Trends in Power System Protection Researches: A Review of Fundamental Relays. *Balkan Journal of Electrical and Computer Engineering*, 6(4), 247-256.
- Ortmeyer, T. H., Hiyama, T., & Salehfar, H. (1996). Power quality effects of distribution system faults. *International Journal of Electrical Power & Energy Systems*, 18(5), 323-329.
- Özgüner, Ö., & Seyyarer, E. (2023). Aşırı Akım Röle Optimizasyonunda Optimizasyon Algoritmalarının Karşılaştırmalı Analizi. *Türk Doğa ve Fen Dergisi*, 12(4), 25-32.
- Pınar, M., (1985). “Elektrik sistemlerinde koruma, röleler ve röle koordinasyonu adlı”, Yayınlanmış Yüksek Lisans Tezi, Marmara Üniversitesi, İstanbul, 20-30.
- Razavi, F., Abyaneh, H. A., Al-Dabbagh, M., Mohammadi, R., & Torkaman, H. (2008). A new comprehensive genetic algorithm method for optimal overcurrent relays coordination. *Electric Power Systems Research*, 78(4), 713-720.
- Rizal, A. S., Umam, M. N., Syaputra, A. M., & Gemayel, A. (2020, September). Study of overcurrent relays coordination optimization based on genetic algorithms. In **2020 International Conference On Technology And Policy in Energy and Electric Power (ICT-PEP)** (pp. 100-103). IEEE.
- Ryan, C., Collins, J. J., & Neill, M. O. (1998). Grammatical evolution: Evolving programs for an arbitrary language. In **Genetic Programming: First European Workshop, EuroGP'98 Paris, France, April 14–15, 1998 Proceedings 1** (pp. 83-96). Springer Berlin Heidelberg.
- Salgotra, R., Singh, U., Saha, S., & Gandomi, A. H. (2020, July). Improving cuckoo search: incorporating changes for CEC 2017 and CEC 2020 benchmark problems. In **2020 IEEE Congress on Evolutionary Computation (CEC)** (pp. 1-7). IEEE.
- Seyyarer, E., Hark, C., Ayata, F., Uçkan, T., & Karcı, A. (2018). Fidan Gelişim ve Lig Şampiyonluk Algoritmaları Kullanarak Aşırı-Akım Rölelerin Optimum Koordinasyonu. *Computer Science*, 3(2), 1-14.
- Shunmugapriya, P., & Kanmani, S. (2017). A hybrid algorithm using ant and bee colony optimization for feature selection and classification (AC-ABC Hybrid). *Swarm and evolutionary computation*, 36, 27-36.
- Simon, D. (2008). Biogeography-based optimization. *IEEE transactions on evolutionary computation*, 12(6), 702-713.

- So, C. W., & Li, K. K. (2000). Overcurrent relay coordination by evolutionary programming. *Electric power systems research*, 53(2), 83-90.
- Su, H., Zhao, D., Heidari, A. A., Liu, L., Zhang, X., Mafarja, M., & Chen, H. (2023). RIME: A physics-based optimization. *Neurocomputing*, 532, 183-214.
- Tunç, A. (2016). Finans Sektörü için Yapay Öğrenme Teknikleri Kullanarak Kredi Kullanabilirliğin Tespiti, Selçuk Üniversitesi. Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi.
- Uthitsunthorn, D., & Kulworawanichpong, T. (2010, June). Optimal overcurrent relay coordination using genetic algorithms. In **2010 International Conference on Advances in Energy Engineering** (pp. 162-165). IEEE.
- Wadood, A., Khurshaid, T., Farkoush, S. G., Yu, J., Kim, C. H., & Rhee, S. B. (2019). Nature-inspired whale optimization algorithm for optimal coordination of directional overcurrent relays in power systems. *Energies*, 12(12), 2297.
- Wang, G. G., Deb, S., & Cui, Z. (2019). Monarch butterfly optimization. *Neural computing and applications*, 31, 1995-2014.
- Wang, L., Cao, Q., Zhang, Z., Mirjalili, S., & Zhao, W. (2022). Artificial rabbits optimization: A new bio-inspired meta-heuristic algorithm for solving engineering optimization problems. *Engineering Applications of Artificial Intelligence*, 114, 105082.
- Wangchamhan, T., Chiewchanwattana, S., & Sunat, K. (2017). Efficient algorithms based on the k-means and Chaotic League Championship Algorithm for numeric, categorical, and mixed-type data clustering. *Expert Systems with Applications*, 90, 146-167.
- Wang, X., Snášel, V., Mirjalili, S., Pan, J. S., Kong, L., & Shehadeh, H. A. (2024). Artificial Protozoa Optimizer (APO): A novel bio-inspired metaheuristic algorithm for engineering optimization. *Knowledge-based systems*, 295, 111737.
- Yuan, J., Li, C., Wang, Q., Han, Y., Wang, J., Zeren, Z., ... & Wang, Y. (2022). Lightning Whistler Wave Speech Recognition Based on Grey Wolf Optimization Algorithm. *Atmosphere*, 13(11), 1828.
- Zeineldin, H., El-Saadany, E. F., & Salama, M. A. (2005, June). Optimal coordination of directional overcurrent relay coordination. In **IEEE Power Engineering Society General Meeting, 2005** (pp. 1101-1106). IEEE.
- Zeineldin, H. H., Sharaf, H. M., Ibrahim, D. K., & Abou El-Zahab, E. E. D. (2014). Optimal protection coordination for meshed distribution systems with DG using dual setting directional over-current relays. *IEEE transactions on smart grid*, 6(1), 115-123.
- Zhao, W., Wang, L., & Mirjalili, S. (2022). Artificial hummingbird algorithm: A new bio-inspired optimizer with its engineering applications. *Computer Methods in Applied Mechanics and Engineering*, 388, 114194.
- Zhao, W., Wang, L., Zhang, Z., Fan, H., Zhang, J., Mirjalili, S., ... & Cao, Q. (2024). Electric eel foraging optimization: A new bio-inspired optimizer for engineering applications. *Expert systems with applications*, 238, 122200.
- Zhao, W., Zhang, Z., & Wang, L. (2020). Manta ray foraging optimization: An effective bio-inspired optimizer for engineering applications. *Engineering Applications of Artificial Intelligence*, 87, 103300.
- Zhao, S., Zhang, T., Ma, S., & Wang, M. (2023). Sea-horse optimizer: a novel nature-inspired meta-heuristic for global optimization problems. *Applied Intelligence*, 53(10), 11833-11860.

Zamani, H., Nadimi-Shahraki, M. H., & Gandomi, A. H. (2022). Starling murmuration optimizer: A novel bio-inspired algorithm for global and engineering optimization. *Computer Methods in Applied Mechanics and Engineering*, 392, 114616.





ÖZ GEÇMİŞ

Kişisel Bilgiler

Adı Soyadı : Özge ÖZGÜNER ARAS

Eğitim Bilgileri

Lisans

Üniversite : Van Yüzüncü Yıl Üniversitesi

Fakülte : Eğitim Fakültesi

Bölüm : Bilgisayar ve Öğretim Teknolojileri

Eğitimi Bölümü

Mezuniyet Yılı : 2015

Akademik Yayınlar

Makaleler

Özgüner, Ö., & Seyyarer, E., (2023). Aşırı Akım Rôle Optimizasyonunda Optimizasyon Algoritmalarının Karşılaştırmalı Analizi. Türk Doğa ve Fen Dergisi, vol.12, no.4, 25-32.

Özgüner, Ö., Güler, Ç., & Çavuş, H., (2022). Computer Education and Instructional Technology Teacher Candidates' Ideas and Attitudes Towards Stem Education. Elektronik Eğitim Bilimleri Dergisi, vol.10, no.20, 237-256.

Ataman, F., & Özgüner, Ö., (2021). Determination of Social Media Users' Perceptions About "Black Friday" Activity via Sentiment Analysis. International Journal of Multidisciplinary and Current Educational Research (IJMCER), vol.3, no.3, 361-371.

Hakemli Kongre / Sempozyum Bildiri Kitaplarında Yer Alan Yayınlar

Özgüner, Ö., & Uçkan, T., (2023). Text Summarization Using Tf-Idf and Textrank Algorithm. 4. International Mediterranean Congress (pp.392-397). Lefkoşa, Cyprus (Kktc)

Akay, V., Özgüner, Ö., & Ayata, F., (2023). Big Data Security and Technologies. BILTEK VII-International Symposium on Current Developments in Science, Technology and Social Sciences (pp.494-506). Ankara, Turkey

Özgüner, Ö., Uçkan, T., & Seyyarer, E., (2023). Text Classification Using Machine Learning Algorithms and Deep Learning Models. 15th International Conference On Engineering & Natural Sciences (pp.82-89). Muş, Turkey

Özgüner, Ö., Güler, Ç., & Çavuş, H., (2019). Prospective Teachers' Opinions of Computer and Instructional Technologies for STEM Education. DMS-2019 (Data Science, Machine Learning and Statistics 2019) (pp.262-264). Van, Turkey

Özgüner, Ö., & Güler, Ç., (2019). Technology Integration Process of a Math Teacher. DMS-2019 (Data Science, Machine Learning and Statistics 2019) (pp.259-261). Van, Turkey



VAN YÜZÜNCÜ YIL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
LİSANSÜSTÜ TEZ ORJİNALLİK RAPORU

Tarih 19/06/2025

Tez Başlığı: Aşırı – Akım Rölelerin Koordinasyon Probleminde Geliştirilmiş Fidan Gelişim Algoritmasının Uygulanması ve Optimizasyon Algoritmaları ile Karşılaştırılması

Yukarıda başlığı belirtilen tez çalışmamın, Kabul ve onay sayfası, Teşekkür, İçindekiler, Simge ve kısaltmalar, Gereç ve yöntemler, kapak sayfası, giriş, ana bölümler ve sonuç bölümlerinden oluşan toplam 46(kırk altı) sayfalık kısmına ilişkin, 19/06/2025 tarihinde şahsım/tez danışmanım tarafından ithenticate adlı intihal tespit programından aşağıda belirtilen filtrelemeler uygulanarak alınmış olan orijinallik raporuna göre tezimin benzerlik oranı % 5 (Beş) dir.

Uygulanan filtreler aşağıda verilmiştir:

- Kabul ve onay sayfası hariç,
- Teşekkür hariç,
- İçindekiler hariç,
- Simge ve kısaltmalar hariç,
- Gereç ve yöntemler hariç,
- Kaynakça hariç,
- Alıntılar hariç,
- Tezden çıkan yayınlar hariç,
- 7 kelimedenden daha az örtüşme içeren metin kısımları hariç (Limit match size to 7 words)

Van Yüzüncü Yıl Üniversitesi Lisansüstü Tez Orijinallik Raporu Alınması ve Kullanılmasına İlişkin Yönergeyi inceledim ve bu yönergede belirtilen azami benzerlik oranlarına göre tez çalışmamın herhangi bir intihal içermediğini; aksinin tespit edileceği muhtemel durumda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve yukarıda vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

Gereğini bilgilerinize arz ederim.

19/06/2025

Adı Soyadı: Özge ÖZGÜNER ARAS

Öğrenci No: 22910001097

Anabilim Dalı: Yapay Zekâ ve Robotik Anabilim Dalı

Programı: Yapay Zekâ ve Robotik Anabilim

Statüsü: (X) Yüksek lisans () Doktora

DANIŞMAN
UYGUNDUR

ENSTİTÜ ONAYI
UYGUNDUR