



T.C.

TOKAT GAZİOSMANPAŞA ÜNİVERSİTESİ

LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

BİLGİSAYAR MÜHENDİSLİĞİ YÜKSEK LİSANS PROGRAMI

**NJ-ABC: CEC 2022 VE GERÇEK DÜNYA YAPISAL TASARIM
OPTİMİZASYONUNDA TEST EDİLEN KOMŞULUK
BİRLEŞTİRMELİ YAPAY ARI KOLONİSİ
ALGORİTMASI**

YÜKSEK LİSANS TEZİ

Omar Ahmed MOHAMMED

Birinci Danışman: Doç. Dr. Özkan İnık

İkinci Danışman: Arş. Gör. Dr. Mustafa Altıok

TOKAT-2025

ETİK SÖZLEŞME

Tokat Gaziosmanpaşa Üniversitesi Lisansüstü Eğitim Enstitüsü Tez Yazım Kılavuzuna göre Doç. Dr. ÖZKAN İNİK ve Arş. Gör. Dr. MUSTAFA ALTIOK danışmanlığında hazırlamış olduğum “NJ-ABC: CEC 2022 ve Yapısal Tasarım Optimizasyonu İçin Komşuluk Birleştirme Tabanlı ABC Algoritması” adlı yüksek lisans tezimin bilimsel etik değerlere ve kurallara uygun, özgün bir çalışma olduğunu, aksinin tespit edilmesi halinde her türlü yasal yaptırımını kabul edeceğimi beyan ederim.

...../...../.....

Omar Ahmed Mohammed



JÜRİ KABUL VE ONAY

OMAR AHMED MOHAMMED tarafından hazırlanan “**NJ-ABC: CEC 2022 ve Gerçek Dünya Yapısal Tasarım Optimizasyonunda Test Edilen Komşuluk Birleştirmeli Yapay Arı Kolonisi Algoritması**” adlı tez çalışmasının savunma sınavı 24/07/2025 tarihinde yapılmış olup aşağıda verilen Jüri tarafından Oy Birliği ile Tokat Gaziosmanpaşa Üniversitesi Lisansüstü Eğitim Enstitüsü Bilgisayar Mühendisliği’nde Yüksek Lisans Tezi olarak kabul edilmiştir.

Jüri Üyeleri

Üye (Başkan):

Üye :

Üye :

ONAY

...../...../.....

Lisansüstü Eğitim Enstitüsü Müdürü

ÖNSÖZ

Bu tez çalışmasının her aşamasında değerli bilgi ve deneyimleriyle yol gösteren, sürekli destek ve teşvikleriyle beni motive eden danışmanlarım Doç. Dr. ÖZKAN İNİK ve Arş. Gör. Dr. MUSTAFA ALTIOK en içten teşekkürlerimi sunarım. Akademik birikimleri, yapıcı geri bildirimleri ve yönlendirmeleri, bu çalışmanın yönünün belirlenmesinde ve bilimsel niteliğinin artmasında büyük rol oynamıştır.

Ayrıca, Tokat Gaziosmanpaşa Üniversitesi Bilgisayar Mühendisliği Bölümü'nün değerli akademik kadrosuna ve idari personeline, sağladıkları nitelikli akademik ortam ve bu araştırmanın gerçekleştirilmesinde sunulan olanaklar için teşekkür ederim. Akademik yolculuğum boyunca yanımda olan, sabırları, anlayışları ve manevi destekleriyle bana güç veren aileme ve arkadaşlarıma da özel olarak teşekkür ederim. Onların bana olan inancı, çabalarımın en önemli itici gücü olmuştur. Bununla birlikte, birlikte çalışmaktan memnuniyet duyduğum değerli meslektaşlarıma ve araştırmacı arkadaşlarıma, katkıları ve ufuk açıcı tartışmaları ile bu tezin içeriğine ve derinliğine sağladıkları değerli katkılar için teşekkür ederim. Son olarak, bu araştırmanın başarıyla tamamlanmasında önemli rol oynayan Tokat Gaziosmanpaşa Üniversitesi'ne sağladığı destek ve imkânlar için minnettarım.

Omar Ahmed Mohammed

TOKAT-2025

ÖZET

NJ-ABC: CEC 2022 VE GERÇEK DÜNYA YAPISAL TASARIM OPTİMİZASYONUNDA TEST EDİLEN KOMŞULUK BİRLEŞTİRMELİ YAPAY ARI KOLONİSİ ALGORİTMASI

Mohammed, Omar Ahmed
Yüksek Lisans, Bilgisayar Mühendisliği Anabilim Dalı
Tez Danışman: Doç. Dr. Özkan İnik
İkinci Danışman: Arş. Gör. Dr. Mustafa Altıok
Temmuz 2025, ix + 74 sayfa

Bu çalışma, Yapay Arı Kolonisi (ABC) algoritmasının temel sınırlılıklarını yavaş yakınsama, erken duraklama ve zayıf sömürü yeteneği aşmayı amaçlayan NJ-ABC adlı yeni bir geliştirilmiş sürümünü önermektedir. NJ-ABC algoritması, Komşuluk Birleştirme (Neighbor-Joining) tekniğinden esinlenerek, çözümleri uzamsal yakınlıklarına göre kümelenen komşuluk tabanlı bir yapı kullanır. Bu mekanizma, küresel çeşitliliği korurken, arama uzayındaki umut vadeden bölgelerde daha yoğun ve düzenli bir keşfe olanak tanır. Algoritma, keşif ve sömürü arasındaki dengeyi üssel olarak azalan bir adaptasyon faktörü ile dinamik şekilde ayarlamakta ve başlangıç popülasyonunun oluşturulmasında geniş kapsama alanı sağlamak amacıyla Latin Hiperküp Örnekleme (LHS) yöntemini kullanmaktadır. Ayrıca, duraklayan çözümlerin yeniden üretilmesini geliştirmek için Parçacık Sürü Optimizasyonu (PSO) ilhamlı bir izci arı stratejisi ve en iyi çözüm üzerinde nihai iyileştirme amacıyla gradyan tabanlı yerel arama stratejisi uygulanmıştır. CEC 2022 test fonksiyonlarıyla yapılan deneylerde NJ-ABC, PSO, TSA, GWO ve MNG-ABC, IGAL-ABC, OCG-ABC gibi gelişmiş ABC türevlerine kıyasla daha iyi yakınsama doğruluğu ve istikrar sergileyerek %83,33 başarı oranına ulaşmıştır. Ayrıca algoritma, tek eksenli eğilme altındaki betonarme kolon tasarımının maliyet optimizasyonuna yönelik gerçek bir mühendislik problemi üzerinde başarıyla uygulanmıştır. Elde edilen sonuçlar, NJ-ABC algoritmasının hem teorik test fonksiyonlarında hem de pratik yapısal tasarım problemlerinde güvenilir ve rekabetçi bir optimizasyon aracı olduğunu doğrulamaktadır.

Anahtar Kelimeler: Yapay Arı Kolonisi, Sürü Zekâsı, NJ-ABC, Optimizasyon, Betonarme Tasarımı, Meta-sezgisel Algoritmalar, CEC 2022.

ABSTRACT

NJ-ABC: A NEIGHBORHOOD- JOINING ARTIFICIAL BEE COLONY ALGORITHM TESTED ON CEC 2022 AND REAL-WORLD STRUCTURAL DESIGN OPTIMIZATION

Mohammed, Omar Ahmed

Master's Thesis, Department of Computer Engineering

Advisor: Assoc. Prof. Dr. Özkan İnik

Second Advisor: Dr. Mustafa Altiok

July 2025, ix and 74 pages

This study introduces NJ-ABC, a novel enhancement of the Artificial Bee Colony (ABC) algorithm designed to overcome key limitations of the original approach, including slow convergence, premature stagnation, and weak exploitation. Inspired by the Neighbor-Joining (NJ) technique, NJ-ABC uses a neighborhood-based clustering mechanism to enable ordered solution grouping depending on spatial proximity. While maintaining global variety, this encourages more concentrated investigation inside interesting areas of the search space. The method dynamically changes the exploration–exploitation trade-off using an exponentially declining adaptation factor and uses Latin Hypercube Sampling (LHS) for first population generation to guarantee wide coverage. A PSO-inspired scout bee strategy improves the regeneration of stagnated solutions and gradient-based local search is applied for last refinement of the best solution, so strengthening robustness. Using the CEC 2022 benchmark suite, NJ-ABC showed better convergence accuracy and stability than several state-of-the-art metaheuristics including PSO, TSA, GWO, and advanced ABC variants including MNG-ABC, IGAL-ABC and OCG-ABC, achieving a success rate of 83.33%. Furthermore, the method was effectively used to a practical engineering challenge concerning the cost optimization of reinforced concrete column design under uniaxial bending. Results validate NJ-ABC's performance as a dependable and competitive optimization tool for both theoretical benchmarks and actual structural design problems.

Keywords: Artificial Bee Colony, Swarm Intelligence, NJ-ABC, Optimization, Reinforced Concrete Design, Metaheuristic Algorithms, CEC 2022.

İÇİNDEKİLER

Sayfa

ETİK SÖZLEŞME	i
JÜRİ KABUL VE ONAY	ii
ÖNSÖZ	iii
ÖZET	iv
ABSTRACT	v
İÇİNDEKİLER	vi
ÇİZELGELER LİSTESİ	viii
ŞEKİLLER LİSTESİ	ix
SİMGELER VE KISALTMALAR	x
1 GİRİŞ	1
2 LİTERATÜR ARAŞTIRMASI	8
2.1 ABC Algoritmasında Arama Denklemlerinin Optimizasyonu	9
2.2 Hibrit ABC Yaklaşımları	16
2.3 ABC Algoritmasına İlave Geliştirmelerin ve Stratejilerin Entegre Edilmesi	20
3 TEMEL ABC ALGORİTMASI	27
3.1 Çalışan Arı Evresi	28
3.2 Gözcü Arı Evresi	29
3.3 İzci Arı Evresi	30
4 YÖNTEM	33
4.1 Latin Hiperküp Örnekleme (LHS) ile Başlatma	33
4.2 Uyarlanabilir Keşif-Sömürü Kontrolü	33
4.3 Geliştirilmiş Çözüm Güncelleme Denklemleri	34
4.4 Çeşitliliğe Dayalı Kısmi Yeniden Başlatma	34
4.5 Parçacık Sürüsü Dinamikleri ile Geliştirilmiş İzci Arı Evresi	34
4.6 Neighborhood-Joining (NJ) Hiyerarşik Kümeleme	35
4.6.1 Uzaklık Matrisi Hesaplanması	35
4.6.2 Küme oluşturma	36
4.6.3 Çözüm arama sürecindeki rolü	37
4.6.4 Geleneksel ABC üzerindeki avantajları	37
4.7 Küme Tabanlı Yönlendirilmiş Arama	38
4.8 Gradyan Tabanlı Yerel Arama İyileştirmesi	39
4.9 Paralel Hesaplama Yapısı	39

4.10 Sonlandırma Kriterleri	39
4.11 NJ-ABC Algoritmasının Yapısal Çerçevesi	41
5 DENEYSEL DOĞRULAMA VE SONUÇ ANALİZİ	44
5.1 CEC 2022 Benchmark Fonksiyonlarındaki Performans	44
5.1.1 Deneyel kurulum	47
5.1.2 Optimizasyon performansı	47
5.1.3 Benchmark metaheuristic algoritmalar ile karşılaştırmalı analiz	49
5.2 NJ-ABC ile Tek Eksenli Eğilme Altında Dikdörtgen Kesitli Betonarme Kolonların Maliyet Optimizasyonu	56
5.2.1. Problem Tanımlaması	57
5.2.2 Tasarım sabitleri ve değişkenleri	58
5.2.3 Yapısal modelleme ve gerilme dengesi	58
5.2.4 Tasarım kısıtları	60
5.2.5 Kolon Tasarım Optimizasyonu Probleminin Meta-Sezgisel Algoritmalarla Çözümü	60
6 SONUÇ VE GELECEK ÇALIŞMALAR	68
7. KAYNAKLAR	70
ÖZGEÇMİŞ	74

ÇİZELGELER LİSTESİ

Tablo 4. 1 Standart ABC ile Kümeleme Mekanizmasına Sahip Önerilen NJ-ABC Arasındaki Temel Farklılıklar.....	37
Tablo 5. 1 CEC 2022 Benchmark Fonksiyonlarının Özeti.....	45
Tablo 5. 2 CEC2022 Tek Amaçlı, Sınırlı Bağlı Sayısal Optimizasyon Problemlerinin Özeti	46
Tablo 5. 3 NJ_ABC Algoritmasının D = 10 Boyutunda Performans Metrikleri	48
Tablo 5. 5 NJ_ABC Algoritmasının D = 20 Boyutunda Performans Metrikleri	48
Tablo 5. 6 NJ_ABC ile Bileşen Algoritmaların Karşılaştırması (problemin boyutu D = 10, popülasyon = 100, iterasyon = 1000, tekrar = 30).....	50
Tablo 5. 7 NJ_ABC ile Bileşen Algoritmaların Karşılaştırması (problemin boyutu D = 20, popülasyon = 200, iterasyon = 2000, tekrar = 30).....	51
Tablo 5. 8 CEC 2022 Üzerine Karşılaştırmalı Sonuçlar, problemin boyutu D = 10, popülasyon = 100, iterasyon = 100000, tekrar = 30	52
Tablo 5. 9 Tasarım Sabitleri ve Değişkenleri	58
Tablo 5. 10 Problemin Tasarım Kısıtları.....	60
Tablo 5. 11 ABC Algoritması ile Elde Edilen Kolonun Tasarım Değişkenleri	61
Tablo 5. 12 AAA Algoritması ile Elde Edilen Kolonun Tasarım Değişkenleri	62
Tablo 5. 13 NJ-ABC Algoritması ile Elde Edilen Kolonun Tasarım Değişkenleri.....	64

ŞEKİLLER LİSTESİ

Şekil 3. 1 ABC Algoritmasının Akış Şeması.....	31
Şekil 4. 1 NJ Kümeleme Sürecinde Uzaklık Matrisinin Evrimi	35
Şekil 4. 2 ABC Algoritmasındaki Rastgele Küresel Arama ile NJ-ABC'deki Düzenli Küme Tabanlı Aramanın Görselleştirilmesi	36
Şekil 4. 3 Standart ABC İle Küme Tabanlı Arama Yönlendirmesine Sahip NJ-ABC Arasındaki Çözüm Dağılımlarının Karşılaştırması	38
Şekil 4. 4 NJ-ABC Algoritmasının Akış Diyagramı	43
Şekil 5. 1 Her Algoritma İçin CEC2022 Test Setine Ait Radar Sıralama Grafiği (10 Boyut)	54
Şekil 5. 2 Her Algoritma İçin CEC2022 Test Setine Ait Radar Sıralama Grafiği (20 Boyut)	55
Şekil 5. 3 Dikdörtgen Kesitli Betonarme Kolon	56
Şekil 5. 4 Tek Eksenli Eğilme Etkisi Altındaki Betonarme Kolonda Deformasyon, Gerilme ve Kuvvetler	59
Şekil 5. 5 Gerçek Dünya Optimizasyonu İçin NJ-ABC Algoritmasının Yakınsama Eğrisi.....	66

SİMGELER VE KISALTMALAR

Simgeler

Açıklamalar

b	Kesit genişliği
h	Kesit yüksekliği
l	Kolon boyu (sabit)
C_c	Birim beton maliyeti
A_s	Çelik donatı alanı
γ_s	Çeliğin birim hacim ağırlığı
C_s	Ton başına birim çelik maliyeti
f	Amaç fonksiyonu değeri
N_z	Eksenel kuvvet
M_y	Eğilme momenti

Kisaltmalar

Açıklamalar

ABC	Artificial Bee Colony
NJ-ABC	Neighborhood Joining Artificial Bee Colony
PSO	Particle Swarm Optimization
GWO	Grey Wolf Optimizer
TSA	Tree-Seed Algorithm
DE	Differential Evolution
AAA	Artificial Algae Algorithm
RC	Reinforced Concrete
CEC	Congress on Evolutionary Computation
USD	United States Dollar
MNG-ABC	Multi-Neighborhood Guided ABC

Kısaltmalar**Açıklamalar**

IGAL-ABC

Improved Genetic Assisted Learning ABC

OCG-ABC

Opposition and Chaos Guided Artificial Bee Colony

HABCDE

Hybrid Artificial Bee Colony with Differential Evolution

ABC-BA

Hybrid Artificial Bee Colony and Bat Algorithm

VABC

Variant Artificial Bee Colony

EVRP

Environmental Vehicle Routing Problem

LHS

Latin Hypercube Sampling

KLABC

K-Means and Learning Automata Artificial Bee Colony

ABCDS

Artificial Bee Colony with Differential Search

1 GİRİŞ

Optimizasyon; yapay zekâ, mühendislik, ekonomi ve tıp gibi çeşitli alanlarda karşılaşılan zorlu gerçek dünya problemlerinin çözümünde temel bir rol oynamaktadır. Problemler daha karmaşık hâle geldikçe ve etkili çözümlere olan ihtiyaç arttıkça, meta-sezgisel (metaheuristic) optimizasyon algoritmaları, kısa sürede optimuma yakın çözümler bulmak için değerli araçlar hâline gelmiştir (Blum & Roli, 2003). Doğadan ve canlıların davranışlarından ilham alan meta-sezgisel algoritmalar, son birkaç on yılda karmaşık problemleri çözmek için en güçlü tekniklerden biri hâline gelmiştir. Bu algoritma sınıfı içerisinde, sürü zekâsına (swarm intelligence) dayalı algoritmalar, optimuma yakın çözümleri hızlı ve yaratıcı bir şekilde bulmadaki olağanüstü yetenekleri sayesinde büyük ilgi görmüştür. Bu algoritmalar, kurtlar, arılar, karıncalar ve kuşlar gibi hayvanların kolektif hareketlerini taklit ederek bireylerin iş birliğiyle ortak hedeflere ulaşmayı amaçlamaktadır (Dorigo & Stutzle, 2004; Karaboga, 2005; Kennedy & Eberhart, 1995; Mirjalili ve ark., 2014).

Örneğin, Eberhart ve Kennedy tarafından 1995 yılında tanıtılan PSO yöntemi, balık sürülerinden ve kuş topluluklarından esinlenerek geliştirilmiştir (Kennedy & Eberhart, 1995). Benzer şekilde, Marco Dorigo tarafından 1992 yılında geliştirilen Karınca Kolonisi Optimizasyonu (Ant Colony Optimization – ACO) algoritması, karıncaların feromon bazlı iletişim sistemini simüle ederek en iyi yolları bulmayı amaçlamaktadır (Dorigo & Stutzle, 2004). Dervis Karaboga ise 2005 yılında, arıların yiyecek arama davranışını simüle etmek amacıyla, izci, işçi ve gözlemci gibi rollere dayalı olarak Yapay Arı Kolonisi (Artificial Bee Colony – ABC) algoritmasını geliştirmiştir (Karaboga, 2005). Ayrıca, Seyedali Mirjalili tarafından 2014 yılında önerilen Gri Kurt Optimizasyonu (Grey Wolf Optimizer – GWO) algoritması, gri kurtların avlanma taktiklerini ve liderlik yapısını modellemektedir (Mirjalili ve ark., 2014).

Sürü zekâsına dayalı algoritmaların yanında, Genetik Algoritma (Genetic Algorithm – GA) da en yaygın meta-sezgisel algoritmalarından biri olarak öne çıkmaktadır. John Holland tarafından 1970’lerde ortaya atılan bu yöntem, evrim ve doğal seçim fikirlerinden esinlenmiştir. Bu algoritma, doğal seçim, çaprazlama (crossover) ve mutasyon (mutation) gibi biyolojik süreçleri taklit ederek, çözümlerin birçok nesil boyunca iyileştirilmesiyle optimum sonuca ulaşmayı amaçlamaktadır (Holland, 1975).

Bu algoritmalar, söz konusu canlıların iş birliğine dayalı ve uyarlanabilir davranışlarını kullanarak karmaşık problem alanlarında optimizasyon sağlarlar.

Heuristiklerin temel hedefi, kabul edilebilir bir çözümle problemi makul sürede çözmektir. Heuristik yaklaşımlar, tüm çözüm uzayı üzerinde arama yapılmasına olanak sağladıkları ve kaliteli alt-optimal çözümler sundukları için gerçekçi problemleri ele almak ve matematiksel modelleme yaklaşımlarının ürettiği çözümler arasındaki tutarsızlıkları azaltmak amacıyla kullanılabilirler (Kumar ve ark., 2016).

ABC algoritması, bu algoritmalar arasında öne çıkmaktadır; çünkü kullanım kolaylığı, uyarlanabilirliği ve çeşitli optimizasyon problemlerini çözmedeki verimliliği ile dikkat çeker. Arıların çiçeklerden besin kaynaklarını araştırma ve toplama davranışları örnek alınarak modellenmiştir. İzci arılar, çalışan arılar ve gözcü arılar arasında görev paylaşımı, bu özgün modelin temelini oluşturur. Bu yapı, görevlerin etkili bir biçimde dağıtılmasına olanak tanır ve en iyi çözümlerin keşfedilme olasılığını artırır. ABC algoritması, sadeliğine rağmen çok çeşitli optimizasyon problemlerinin çözümünde oldukça başarılı olduğunu kanıtlamıştır (Karaboga, 2005). ABC algoritması, birçok diğer popülasyon tabanlı sezgisel algoritmadan daha iyi performans göstermesine rağmen, yerel optimumlarda takılmaya eğilimli olması, yavaş yakınsama hızına sahip olması ve tasarımının zor olması gibi bazı sınırlamalara sahiptir. Bunun temel nedeni, algoritmanın keşif aşamasına daha fazla ağırlık vermesidir (Karaboga & Akay, 2009). Bu çalışmada, ABC algoritmasının performansını artırmaya odaklanılacak ve diğer bazı algoritmalarla karşılaştırmalı analizler yapılacaktır.

ABC algoritması ve türevleri, çok çeşitli optimizasyon problemlerinin çözümünde etkili araçlar olduklarını kanıtlamışlardır. Bu algoritmaların cazibesi, uygulama kolaylığı, kontrol parametrelerinin azlığı ve güçlü küresel optimizasyon yeteneklerinden kaynaklanmaktadır. Bu özellikler, ABC algoritmasını aşağıdaki gibi pek çok alandaki zorlukları ele alan araştırmacılar için tercih edilen bir yöntem hâline getirmiştir: sayısal optimizasyon (Das ve ark., 2013; Tsai, 2014), kontrol sistemi tasarımı (Chang, 2013; D.-L. Zhang ve ark., 2014), parametre tanımlama (Sun ve ark., 2013), sinyal işleme (Karaboga & Latifoglu, 2013), görüntü segmentasyonu (Ma ve ark., 2011), ulaştırma lojistiği (Szeto & Jiang, 2014) ve çizelgeleme (Li ve ark., 2014).

Kontrol sistemleri alanında, ABC algoritması dinamik sistemler için kontrolör tasarımı ve ayarlarını iyileştirmek amacıyla kullanılmıştır. Örneğin, kararsız, doğrusal olmayan,

sürekli karıştırılmalı tank reaktörlü bir kimyasal sistem için yeni bir uygulama önerilmiş ve bu sistemde PID (Oransal-Türevsel-Entegre) kontrolör tasarımının iyileştirilmesinde ABC algoritması kullanılmıştır. Bu yaklaşım, reaktörün dinamik davranışının daha iyi yönetilmesini sağlayarak daha yüksek kararlılık ve performans sunmuştur (Chang, 2013). Buna ek olarak, otomatik voltaj düzenleyici sisteminde kesirli dereceli PID kontrolörlerin optimizasyonu amacıyla, CNC-ABC adı verilen geliştirilmiş bir algoritma varyantı geliştirilmiştir. Bu varyant, döngüsel değişim komşuluk teknikleri (cyclic exchange neighborhood techniques) ve kaos teorisini içermekte olup, daha hızlı yakınsama ve gelişmiş ayar kabiliyeti ile dikkat çekmiştir (D.-L. Zhang ve ark., 2014).

Sinyal işleme alanında, ABC algoritması harmonik kestirimi, gürültü filtreleme ve sinyal çıkarımı gibi görevlerde uygulanmıştır. Örneğin, bu algoritma, metal-uzatılmış yarıiletken alan etkili transistör (MESFET) cihazlarında kullanılan GaAs küçük sinyal eşdeğer devre modeli parametrelerini belirlemek amacıyla kullanılmış ve PSO gibi algoritmalarından daha iyi performans sergilemiştir. Bu durum, ABC algoritmasının parametre tanımlama ve modelleme konularındaki yetkinliğini ortaya koymaktadır (Sabat ve ark., 2010). Ayrıca, ABC algoritması, EEG/Olay İlgili Potansiyel (Event-Related Potential) filtreleme amacıyla uyarlanabilir bir gürültü engelleyiciye entegre edilmiş ve filtreleme sürecinin doğruluğunu ve kapsamını artırmıştır (Ahirwal ve ark., 2014). Başka bir çalışmada, ABC algoritması kullanılarak tasarlanan sayısal sonsuz tepki (IIR) filtreleri, transkraniyal Doppler sinyallerinden gürültüyü kaldırmak için etkili bir şekilde kullanılmış ve bu algoritmanın tıbbi sinyal netliğini artırma kabiliyetini göstermiştir (Karaboga & Latifoglu, 2013).

Güç kalitesi (power quality) problemlerine yönelik olarak, ABC algoritması en küçük kareler yöntemiyle (least squares method) hibrit hâle getirilmiş ve güç sinyallerinde harmonik kestirimini iyileştirmek amacıyla kullanılmıştır. Bu hibrit yaklaşım, güç kalitesi analizinde karşılaşılan zorluklara doğru ve verimli çözümler sunarak algoritmanın uyarlanabilirliğini daha da pekiştirmiştir (Biswas ve ark., 2013). Bu uygulamalardan elde edilen deneysel bulgular, hibrit yaklaşımın karmaşık optimizasyon görevlerini etkin bir şekilde yönetme kapasitesini sürekli olarak vurgulamaktadır. Kontrol ve sinyal işleme alanlarının ötesinde, ABC algoritmasının kullanımı ulaştırma lojistiği (Szeto & Jiang, 2014) ve atölye tipi çizelgeleme problemleri (Li ve ark., 2014) gibi alanlara da yayılmıştır. Bu alanlarda algoritmanın

çok deęişkenli ve karmaşık problemleri ele alma becerisi, süreçlerin ve kaynakların optimizasyonu açısından vazgeçilmez bir araç hâline gelmesini sağlamaktadır.

Sayısal optimizasyon (numerical optimization) alanında ise araştırmacılar, ABC algoritmasının verimliliğini ve sağlamlığını artırmak amacıyla birçok gelişmiş ve hibrit sürüm önermiştir. Bu deęişiklikler, özellikle yavaş yakınsama oranları ve yerel optimumlara sıkışma eğilimi gibi orijinal ABC algoritmasının sınırlamalarını aşmayı hedeflemektedir. Bu iyileştirmelerden biri, Gbest yönlendirmeli yapay arı kolonisi algoritması (Gbest-guided Artificial Bee Colony – GABC) olarak bilinir. Bu yaklaşım, arama sürecine küresel en iyi çözümün etkisini dâhil etmektedir. Arıların en iyi bilinen çözüme yönlendirilmesi sayesinde, GABC algoritmanın keşif ve sömürü yeteneklerini geliştirerek daha hızlı yakınsama ve daha yüksek çözüm kalitesi elde edilmesini sağlamaktadır (Zhu & Kwong, 2010).

Son yıllarda, araştırmacılar ABC algoritmasının performansını artırmak amacıyla, onu diğer optimizasyon teknikleriyle birleştirerek hibrit ABC varyantları (hybrid ABC variants) olarak bilinen yöntemleri geliştirmişlerdir. Bu hibrit yöntemler, orijinal ABC algoritmasının yavaş yakınsama hızı veya yetersiz yerel arama kabiliyeti gibi doğasında bulunan sınırlılıklarını gidermek için, ek stratejilerin güçlü yönlerinden faydalanmayı amaçlamaktadır. Örneğin, Diferansiyel Evrim (DE) algoritmasını ABC çerçevesine entegre eden HABCDE adını verdikleri hibrit bir varyant önermiştir. HABCDE algoritmasında, DE algoritmasından alınan mutasyon operatörü, ABC algoritmasının gözcü arı evresine entegre edilmiştir. Bu deęişiklik, algoritmanın arama uzayını daha etkili bir şekilde keşfetme yeteneğini geliştirmenin yanı sıra, yerel arama kapasitesini de artırarak daha ince ayarlı ve umut vadeden çözümler üretilmesini sağlamaktadır. DE algoritmasının güçlü mutasyon mekanizmasından yararlanan HABCDE, özellikle çözüm uzayının ayrıntılı şekilde taranmasını gerektiren karmaşık problemler söz konusu olduğunda, daha iyi optimizasyon sonuçları elde etmektedir (Xiang & An, 2013).

Bir diğer öne çıkan gelişme ise, ABC ve Yarasa Algoritması (Bat Algorithm – BA)'nın güçlü yönlerini bir araya getiren hibrit ABC-BA algoritmasıdır. Bu hibrit yaklaşım, ABC algoritmasının keşif yeteneği ile BA algoritmasının ince ayar yapma kapasitesini birleştirerek, çok sayıda kısıt içeren karmaşık problem alanlarını etkili bir şekilde ele almak için çok yönlü bir optimizasyon aracı sunmaktadır (Tsai, 2014).

Benzer şekilde, VABC (Variant Artificial Bee Colony) algoritması, özellikle yüksek boyutlu sürekli optimizasyon problemleri için tasarlanmıştır. VABC, geniş arama uzaylarını yönetmeye yönelik yeni stratejiler entegre ederek, hesaplama açısından yoğun ve dayanıklı optimizasyon teknikleri gerektiren problemlerde etkinliğini ortaya koymakta ve uyarlanabilirliğini açıkça göstermektedir (Imanian ve ark., 2014).

Gerçek parametre optimizasyonu (real-parameter optimization) bağlamında, bu alana özgü zorlukları ele almak amacıyla değiştirilmiş bir ABC algoritması varyantı geliştirilmiştir. Bu iyileştirilmiş sürüm, geliştirilmiş arama mekanizmalarını ve uyarlanabilir stratejileri içermekte olup, sürekli değişkenler olarak temsil edilen parametrelerin optimize edilmesinde daha yüksek doğruluk ve verimlilik sağlamaktadır (Akay & Karaboga, 2012). Ayrıca, yapısal sistem uygulamalarında, değiştirilmiş ABC algoritması çeşitli mühendislik ve bilimsel alanlarda optimal yapılandırmaları belirlemek için kullanılmış ve bu sayede gerçek dünya problemlerinin çözümünde algoritmanın esnekliği ve pratik yararı daha da belirginleşmiştir (Sun ve ark., 2013).

ABC algoritmasının uygulama alanları, sayısal optimizasyonun ötesine geçerek enerji ve ulaşım gibi özel alanlara da yayılmıştır. Örneğin, enerji sektöründe, ABC tekniği ile entegre edilmiş yapay sinir ağı modeli, Türkiye'nin yıllık hidrolik enerji üretimini tahmin etmek amacıyla kullanılmıştır. Bu uygulama, tahmine dayalı modelleme ile optimizasyonun bir arada kullanımı yoluyla kaynak planlaması ve yönetimi için uygulanabilir içgörüler üretme açısından algoritmanın yetkinliğini ortaya koymaktadır (Uzlu ve ark., 2014). Ulaşım alanındaki araştırmalarda ise, iki seviyeli bir toplu taşıma sistemi (bi-level transit system) önerilmiş; bu sistemde hem sefer sıklıkları hem de ulaşım rotaları aynı anda belirlenmiştir. Bu iki karar düzeyi arasındaki karmaşık etkileşimi ele almak amacıyla, hibrit bir yapay arı kolonisi algoritması geliştirilmiştir. Bu yaklaşım, algoritmanın birbirine bağlı değişkenleri ve kısıtları etkin biçimde yönetme becerisini göstermiş ve büyük ölçekli toplu taşıma ağlarının tasarımında optimize çözümler sunmuştur (Szeto & Jiang, 2014).

Ayrıca, ABC algoritması çevresel araç rotalama problemi (Environmental Vehicle Routing Problem – EVRP) olarak bilinen ve çevresel faktörleri lojistik planlamaya entegre eden bir modeli çözmek amacıyla uyarlanmıştır. Bu amaçla özel olarak geliştirilen hibrit ABC algoritması, emisyonları azaltmak ve araç rotalarını optimize etmek için gelişmiş arama stratejileri içermektedir. Bu hibrit algoritmanın performansı,

kapasiteli araç rotalama problemlerine ait tanınmış benchmark örnekleri üzerinde titizlikle değerlendirilmiş ve çözüm kalitesi ile hesaplama verimliliği açısından üstün sonuçlar ortaya koymuştur (S. Zhang ve ark., 2014). Bu örnekler, ABC algoritmasının evrim geçirerek çok çeşitli alanlardaki optimizasyon problemlerini çözebilen güçlü ve çok yönlü bir araç hâline geldiğini açıkça göstermektedir.

ABC algoritmasının uygulamalarına ilişkin mevcut araştırma sonuçları, algoritmanın mükemmel optimizasyon performansına sahip olduğunu ortaya koymaktadır. Bu algoritma, küresel optimizasyon problemlerini etkili ve verimli bir şekilde çözebilmektedir. Ancak hâlâ çözülmesi gereken bazı sorunlar mevcuttur. Çünkü ABC algoritmasının sömürü yeteneği düşüktür ve yakınsama hızı, PSO ve DE gibi diğer bazı popülasyon tabanlı algoritmalarla kıyasla daha yavaştır. Bazı durumlarda yerel optimallere sıkışma olasılığı da oldukça yüksektir. Bu nedenle, araştırmacılar ABC algoritmasının sömürü kabiliyetini ve yakınsama hızını artırmak için GABC algoritması, VABC algoritması, ABC-BA algoritması, ABC-MNG algoritması, IGAL-ABC algoritması, OCG-ABC algoritması, COABC algoritması gibi çeşitli etkili yöntemler önermiştir (Yang, 2020). Bu algoritmaların her biri, kendi içinde bazı avantajlar sunmaktadır. Ancak, literatürde algoritmanın iyileştirilmesine yönelik önemli sayıda çalışma mevcut olmasına rağmen, daha üstün sonuçlar elde etmek için yeni geliştirmelere hâlâ geniş bir alan bulunmaktadır.

Bu çalışmada, ABC algoritmasının geliştirilmiş bir versiyonu olan Komşuluk Birleştirme Tabanlı Yapay Arı Kolonisi (NJ-ABC) algoritması sunulmaktadır. Bu algoritma, ABC algoritmasının keşif gücü ile Komşuluk Birleştirme (Neighborhood-Joining) (Saitou & Nei, 1987) yönteminden esinlenen hiyerarşik kümeleme tekniklerini sinerjik şekilde birleştirmektedir. NJ-ABC algoritması, karar uzayı boyunca değişken aralıklarının homojen kapsanmasını sağlamak amacıyla Latin Hiperküp Örnekleme (Latin Hypercube Sampling – LHS) yöntemi ile başlangıçta bir “gıda kaynağı” popülasyonu oluşturarak işlemeye başlar (McKay ve ark., 2000). Ardından, uyarlanabilir bir kontrol parametresi, zamanla azalarak arama sürecinin keşiften sömürüye doğru yönelmesini sağlar ve böylece umut vadeden bölgelerde daha yoğun bir arama yapılmasına olanak tanır.

Çalışan arı ve gözcü arı evrelerinde, çözüm üretme denklemleri, NJ hiyerarşik kümeleme yöntemi ile türetilen komşuluk ilişkilerini dikkate alacak şekilde

güncellenmiştir. Bu sayede, popülasyon çeşitliliği korunmakta ve yerel optimumlara erken yakınsama riski azaltılmaktadır. İzci arı aşamasında ise, Parçacık Sürüsü Optimizasyonu (PSO) algoritmasından esinlenen dinamikler kullanılarak, durağanlaşan çözümler yeniden başlatılmakta ve hem bireysel hem de küresel en iyi bilgiye dayalı arama gerçekleştirilerek iyileştirme yapılmaktadır. Son olarak, algoritmanın küresel en iyi çözümüne yönelik olarak gradyan tabanlı yerel iyileştirme (gradient-based local refinement) uygulanmakta ve bu sayede gerçek optimuma daha hızlı yakınsama sağlanmaktadır.

Tek amaçlı, kısıtlı optimizasyon problemleri için en zorlu test ortamlarından biri olarak genel kabul gören CEC 2022 benchmark fonksiyonları (Kumar ve ark., 2022) kullanılarak NJ-ABC algoritmasının performansı değerlendirilmiştir. Çözüm doğruluğu, yakınsama hızı ve kararlılık açısından elde edilen deneysel sonuçlar, NJ-ABC algoritmasının yalnızca standart ABC'ye değil, aynı zamanda MNG_ABC, IGAL-ABC ve OCG-ABC gibi geliştirilmiş varyantlara da üstünlük sağladığını açık biçimde göstermektedir. Ayrıca, önerilen yöntem, PSO, TSA, AAA ve GWO gibi iyi bilinen meta-sezgisel algoritmalarla doğrudan karşılaştırmalarda da rutin olarak daha iyi sonuçlar üretmiş ve bu sayede zorlu arama ortamlarında dayanıklılığını doğrulamıştır.

Bununla birlikte, NJ-ABC algoritması, tek eksenli eğilme altındaki dikdörtgen kesitli betonarme kolonların en uygun tasarımının belirlenmesine yönelik pratik bir inşaat mühendisliği probleminde de uygulanmıştır (Bekdaş ve ark., 2021; Nigdeli ve ark., 2015). Amaç, tüm ilgili yapı yönetmeliği kriterlerini karşılamak koşuluyla, beton ve çelik donatıların toplam maliyetlerini minimize etmektir. Elde edilen bulgular, algoritmanın düşük maliyetli ve yönetmeliklere uygun çözümler üretebilme kabiliyetini ortaya koymakta ve böylece NJ-ABC algoritmasının verimliliğini ve gerçek dünyadaki pratik değerini güçlü şekilde vurgulamaktadır.

2 LİTERATÜR ARAŞTIRMASI

Meta-sezgisel algoritmalar alanı, küresel optimizasyon problemlerini çözme konusunda önemli gelişmeler göstermiştir. Bu gelişmeler sayesinde algoritmaların verimliliğini ve etkinliğini artırmaya yönelik çok çeşitli yaklaşımlar ortaya çıkmıştır. Bu algoritmalar arasında ABC algoritması büyük ilgi görmüş ve standart ABC modelinin sınırlılıklarını aşmak amacıyla birçok geliştirilmiş varyantı oluşturulmuştur. ABC varyantlarının sürekli olarak geliştirilmesi, algoritmanın performansını iyileştirme, özellikle keşif ve sömürü arasındaki dengeyi sağlama, yakınsamayı hızlandırma ve erken dönemde yerel optimumlarda takılmayı engelleme ihtiyacından kaynaklanmaktadır. Bu bölümde, karmaşık optimizasyon problemlerinin çözümünde etkinliğini artırmak amacıyla standart ABC algoritmasına getirilen en yeni iyileştirmeler kapsamlı bir şekilde ele alınmaktadır.

Sürü zekâsı algoritmalarına uygulanan optimizasyon stratejileri üç temel kategoriye ayrılabilir; bu sınıflandırma, ABC algoritmasının performansını artırmak için yapılan modifikasyonları da kapsamaktadır. İlk kategori, çözüm güncelleme stratejilerinin iyileştirildiği arama denklemlerinin optimizasyonunu içerir. Bu yaklaşımda, parametrelerin otomatik olarak ayarlanması veya bazı bileşenlerin eklenip çıkarılmasıyla elle yapılan parametre ayarlarına olan bağımlılık azaltılmakta ve farklı problem türlerine daha iyi uyum sağlayan bir denge elde edilmektedir (Rajasekhar ve ark., 2017).

İkinci kategori, ABC algoritmasının küresel ve yerel arama yeteneklerini geliştirmek için diğer algoritmalarla hibritlenmesine odaklanmaktadır. Bu tür entegrasyonlar, arama uzayını keşfetme verimliliğini artırmakta ve optimum çözümlere yakınsama güvenilirliğini yükseltmektedir. Üçüncü kategori ise kaotik arama mekanizmaları, Bayes tahmini ve arşivleme teknikleri gibi ek iyileştirme yöntemlerinin veya stratejilerin ABC algoritmasına dâhil edilmesini içermektedir. Bu stratejilerin amacı, algoritmanın karmaşık optimizasyon problemlerini çözme etkinliğini daha da geliştirmektir. Birçok durumda, araştırmacılar bu kategorilerden birden fazla stratejiyi birleştirerek, farklı optimizasyon senaryolarında daha dengeli ve etkin performans elde etmektedir (Karaboga ve ark., 2014; Rajasekhar ve ark., 2017).

2.1 ABC Algoritmasında Arama Denklemlerinin Optimizasyonu

ABC algoritmasında arama denklemleri ve güncelleme tekniklerinin iyileştirilmesi amacıyla çeşitli modifikasyonlar yapılmıştır. Bu değişiklikler arasında otomatik parametre ayarı, konum güncelleme denklemlerinin iyileştirilmesi ve elle yapılan ayarlamalara olan ihtiyacın azaltılması yer almaktadır. Bu geliştirmeler, keşif (exploration) ve sömürü (exploitation) arasındaki dengeyi daha etkili bir şekilde sağlayarak, farklı optimizasyon problemlerinde ideal çözümlere hızlı yakınsamayı ve arama doğruluğunu artırmayı hedeflemektedir.

Bu bağlamda, Akay ve Karaboğa, ABC algoritmasının sürekli optimizasyon problemlerindeki performansını artırmaya yönelik bazı iyileştirmeler sunmuşlardır. Araştırmalarında Pertürbasyon Oranı (Perturbation Rate) ve Ölçekleme Faktörünün (Scaling Factor - SF) ayarlanmasının yanı sıra Limit Parametresi'nin optimizasyonuna odaklanmışlardır. Bu değişiklikler, algoritmanın daha hızlı yakınsamasına ve yerel optimumlara daha az takılmasına olanak sağlamıştır. Başlıca iyileştirme, çözüm güncellemelerinde kullanılan rastgele pertürbasyonların sayısının kontrol edilmesi olmuştur. Araştırmacılar, yeni çözümler aranırken değiştirilen değişken sayısını düzenlemek için bir Modifikasyon Oranı (Modification Rate- MR) parametresi önermiştir. Deneysel sonuçlar, bu parametrenin artırılmasının çözüm doğruluğunu yükselttiğini ve optimal değerlere ulaşmak için gereken iterasyon sayısını azalttığını göstermiştir. Ayrıca, arama sürecinde adım büyüklüğüne yönelik ayarlamalar yapılmış, böylece arama alanında keşif ve sömürü arasında daha dengeli bir performans elde edilmiştir. Modifiye edilen ABC algoritması, standart test fonksiyonları üzerinde denenmiş ve standart ABC algoritmasına göre belirgin bir üstünlük sağlamıştır (Akay & Karaboga, 2012).

Gao ve ark., her iterasyonda bulunan en iyi çözümü temel alarak sömürü kapasitesini geliştirmek ve yakınsama hızını artırmak amacıyla ABC/best isimli algoritmayı geliştirmiştir. Bu yöntem, standart ABC algoritmasının güçlü keşif yeteneği yanında zayıf sömürü kabiliyetine sahip olması ve bu nedenle yerel optimumlara erken yakınsama eğilimine sahip olması gibi temel sınırlılıkları ele almaktadır. Önerilen yöntem, arama denklemlerini önceki iterasyonlarda belirlenen en iyi çözüm etrafında yoğunlaştırarak optimal çözümlere doğru daha etkin ilerlemeyi sağlamaktadır. Ayrıca başlangıç popülasyonunun oluşturulma aşaması, kaotik sistemler ve ters tabanlı öğrenme (opposition-based learning) ile zenginleştirilerek başlangıç popülasyonunun

çeşitliliği artırılmıştır. ABC/best algoritması, 26 farklı benchmark test fonksiyonu üzerinde değerlendirilmiş ve standart ABC algoritması ile diğer geliştirilmiş varyantlara göre daha iyi yakınsama hızı ve çözüm doğruluğu sunmuştur (Gao ve ark., 2012).

Gao ve Liu, geleneksel ABC algoritmasında gözlemlenen yavaş yakınsama ve yerel optimumlara takılma problemlerini gidermek amacıyla Geliştirilmiş ABC (Improved ABC- IABC) isimli algoritmayı sunmuştur. Araştırmacılar, Diferansiyel Evrim (Differential Evolution- DE) algoritmasından esinlenerek ABC/best/1 ve ABC/rand/1 adını verdikleri iki yeni arama denklemi geliştirmiş ve popülasyon çeşitliliğini artırmak için ters tabanlı öğrenme ve kaotik sistemleri birleştiren hibrit bir başlatma yaklaşımı kullanmıştır. Her iki yeni denklemin avantajlarından faydalanmak ve dezavantajlarını azaltmak için seçici bir olasılık mekanizması (p) önerilmiştir. Farklı unimodal ve multimodal benchmark fonksiyonları üzerinde yapılan deneyler sonucunda, IABC algoritmasının çözüm doğruluğu, yakınsama hızı ve kararlılık açısından diğer on üç güncel optimizasyon algoritmasına göre açıkça üstün olduğu gösterilmiştir (Gao & Liu, 2011).

Wang ve ark., ABC algoritmasının geliştirilmiş bir versiyonunu sunarak Bayesçi Tahmin'i (Bayesian Estimation) algoritmaya dâhil etmiş ve buna BEABC adını vermiştir. Bu varyant, ABC algoritmasının zayıf sömürü yeteneği, yavaş yakınsama ve düşük çözüm kalitesi gibi temel kısıtlarını gidermek amacıyla geliştirilmiştir. Önerilen algoritmada geleneksel uygunluk oranı yerine, Bayesçi çıkarıma dayalı olasılıksal bir tahmin yöntemi kullanılarak arıların yeni besin kaynakları bulma kararlarının doğruluğu artırılmıştır. Ayrıca izci ve gözcü arılar için geliştirilen yönlendirme mekanizması, yerel arama hassasiyetini artırarak optimum çözümlere hızlı yakınsamayı desteklemiştir. BEABC algoritması, 24 tek-amaçlı benchmark fonksiyonu üzerinde değerlendirilmiş ve vakaların %76'sında üstün performans göstermiştir. Ayrıca, çok amaçlı optimizasyon problemleri ve gerçek dünya vaka çalışmalarında çeşitli evrimsel algoritmalarla kıyasla tutarlı biçimde daha iyi sonuçlar elde etmiştir (Wang ve ark., 2022).

Xiang ve ark., biri çalışan arılar (employed bees) ve diğeri gözcü arılar (onlooker bees) için atanmış iki farklı arama denklemi kullanan geliştirilmiş bir Yapay Arı Kolonisi (ABC) algoritması geliştirmiştir. Bu iyileştirilmiş algoritma, geleneksel ABC

varyantları ve biyolojik esinli olmayan metodolojiler de dahil olmak üzere çeşitli karşılaştırmalı algoritmalara kıyasla dikkate değer bir performans üstünlüğü sergilemiştir. Geliştirilmiş algoritmanın performans değerlendirmesi, çeşitli standart test fonksiyonları kullanılarak gerçekleştirilmiş ve sonuçlar, üç ana boyutta net bir avantaj olduğunu göstermiştir: çözüm kalitesi, sağlamlık (robustness) ve yakınsama hızı (convergence speed). Çözüm kalitesi açısından, algoritma diğer algoritmalara kıyasla optimal çözümlere daha yakın çözümler bulmayı başarmıştır. Sağlamlık bakımından, algoritma çeşitli test fonksiyonlarında tutarlı bir performans düzeyini koruma kapasitesine sahip olduğunu göstermiş ve bu da onun dayanıklılığını ve güvenilirliğini ortaya koymuştur. Yakınsama hızı açısından ise, algoritma alternatif algoritmalara kıyasla daha kısa bir zaman dilimi içinde istenilen çözümlere ulaşma kabiliyeti ile öne çıkmıştır. Bu geliştirilmiş algoritmanın dikkat çeken yenilikçi yönlerinden biri, Xiang ve ark. tarafından referans gösterilen Parçacık Sürüsü Optimizasyonu (PSO) algoritmasından esinlenilmiş yeni bir arama denklemine sahip olmasıdır (Xiang ve ark., 2014).

Niu ve ark., mevcut global optimum çözümü dikkate alan arama denklemesine ek olarak, sınırlı optimizasyon problemlerini etkin biçimde çözmek amacıyla ESOABC-EC&ACS (Escaping Strategy and Adaptive Colony Size) adını verdikleri geliştirilmiş bir ABC algoritması önermiştir. Çalışmada, geleneksel ABC algoritmasının performansını artırmak için uygulanabilir olmayan çözümlerin (infeasible solutions) keşfine dayalı bir “kaçış stratejisi” (EC) ve uyarlanabilir koloni boyutu (ACS) gibi yenilikçi stratejiler kullanılmıştır. Bu geliştirmeler, çözüm çeşitliliğini artırmayı, yakınsamayı hızlandırmayı ve erken yakınsama sorunlarını önlemeyi amaçlamaktadır. Araştırmacılar, bir dizi standart benchmark fonksiyonu üzerinde deneyler gerçekleştirmiştir ve önerilen ESOABC-EC&ACS algoritmasının hem geleneksel ABC algoritmasına hem de diğer geliştirilmiş ABC varyantlarına kıyasla önemli ölçüde daha iyi performans gösterdiğini ortaya koymuştur. Algoritma, yakınsama hızı, çözüm kalitesi ve karmaşık sınırlı optimizasyon problemlerini çözme kabiliyeti açısından kayda değer üstünlük göstermiştir. Bu başarı, keşif ve sömürü arasında etkin bir denge sağlayarak elde edilmiştir (Niu ve ark., 2023).

Güvenilirlik temelli kafes yapıların (truss structures) optimizasyonunu gerçekleştirmek amacıyla geliştirilmiş küçük aralıklı arama stratejisi (improved small-interval search strategy) kullanan bir ABC algoritması varyantı önermiştir. Çalışmalarının temel

hedefi, geleneksel ABC algoritmasının yapısal sınırlılıklarını aşmak, özellikle de erken yakınsamaya eğilim ve yavaş yakınsama hızını ortadan kaldırmaktır. Araştırmacılar, arama uzayı boyunca başlangıç çözümlerinin daha iyi dağılımını sağlamak amacıyla, popülasyon başlatma (population initialization) aşamasında ek bir değişken (additional variable) ekleyerek değişiklikler önermiştir. Ayrıca, yerel optimumlara sıkışma riskini azaltmak ve çözüm kalitesini artırmak amacıyla, bir ceza faktörü (penalty factor) ile birleştirilmiş Tabu Arama (Tabu Search) mekanizmasını entegre etmişlerdir. Deneysel değerlendirmeler, benchmark Gezgin Satıcı Problemleri (TSP) ve güvenilirlik temelli kafes yapıların optimizasyon problemleri üzerinde gerçekleştirilmiş ve geliştirilmiş ABC algoritmasının, temel ABC algoritması, Karınca Kolonisi Optimizasyonu ACO ve Genetik Algoritmalar GA ile karşılaştırıldığında açıkça üstünlük sağladığını göstermiştir. Özellikle önerilen algoritma, daha hızlı yakınsama hızları (faster convergence rates) ve gelişmiş çözüm doğruluğu (improved solution accuracy) elde etmiştir (Yancang ve ark., 2017).

Standart ABC algoritmasının özellikle güçlü keşif yeteneğine kıyasla göreceli olarak zayıf olan sömürü yeteneği gibi temel zayıflıklarını gidermek amacıyla tasarlanmış PA-ABC (Parametre Uyarlamalı ABC- Parameter Adaptive ABC) algoritmasını tanıtmıştır. Bu amacı gerçekleştirmek için, PA-ABC hem çalışan arılar hem de gözcü arılar için uyarlanabilir arama denklemleri (adaptive search equations) kullanmış, bu denklemlere şimdiye kadar bulunan en iyi çözümü entegre ederek sömürü verimliliğini artırmıştır. Önerilen iyileştirmeler üç ana noktaya odaklanmıştır. Birincisi, çalışan arılar, arama çeşitliliğini artırmak amacıyla iki rastgele çözüm kullanmış ve böylece güçlü bir keşif yeteneğinin korunmasını sağlamıştır. İkincisi, gözcü arılar arama işlemini tek bir rastgele çözüm çevresinde yoğunlaştırmış ve bu da arama sürecinde popülasyon çeşitliliğinin korunmasına yardımcı olmuştur. Üçüncüsü ise, gözcü arıların arama denklemlerine uygunluk fonksiyonuna (fitness function) dayalı bir uyarlanabilir parametre (adaptive parameter) entegre edilmiş ve bu sayede farklı arama aşamalarına göre dinamik adım boyu (step-size) ayarlamaları gerçekleştirilmiştir. Bu iyileştirme, keşif ve sömürü arasındaki dengeyi geliştirmiştir. Bir dizi benchmark fonksiyonu üzerinde elde edilen deneysel sonuçlar, PA-ABC algoritmasının standart ABC ve birkaç gelişmiş ABC varyantına göre üstün performans sergilediğini ortaya koymuştur. Önerilen yöntem, daha yüksek arama verimliliği, daha hızlı yakınsama hızı ve genel olarak iyileştirilmiş performans elde etmiş, çoğu durumda diğer çağdaş ABC

varyantlarının özgün parametre ayarları ile elde ettiği başarıyı aşmış veya en azından eşdeğer düzeye ulaşmıştır (Zhao ve ark., 2022).

Zhou ve ark., algoritmanın sömürü yeteneğini geliştirirken aynı zamanda popülasyon çeşitliliğini korumayı amaçlayan MGABC (Multi-Elite Guided ABC) algoritmasını önermiştir. Bu geliştirme, birden fazla seçkin çözümden (elite solutions) elde edilen değerli bilgilerin arama sürecine rehberlik etmek amacıyla kullanıldığı çoklu elit yönlendirme mekanizmasına (multi-elite guidance mechanism) dayanmaktadır ve bu sayede keşif (exploration) ile sömürü (exploitation) arasında daha etkili bir denge kurulması hedeflenmiştir. Önerilen algoritma, popülasyon içerisindeki seçkin çözüm grubundan yararlanan iki yeni arama denklemi aracılığıyla çalışan arı ve gözcü arı evrelerini iyileştirmeye odaklanmıştır. Ayrıca, elit gruptan faydalanarak daha hassas arama iyileştirmeleri yapılmasını sağlayan gelişmiş bir yerel arama parametresi geliştirilmiştir. Bu mekanizma, alt-optimal (optimal altı) çözümlere erken yakınsama olasılığını azaltmayı amaçlamaktadır. MGABC algoritmasının etkinliğini değerlendirmek amacıyla, 50 benchmark fonksiyonu ve bir gerçek dünya optimizasyon problemi üzerinde deneyler gerçekleştirilmiştir. Bu fonksiyonlar arasında 22 adet ölçeklenebilir temel fonksiyon ve CEC2013 benchmark setinden alınan 28 fonksiyon bulunmaktadır. Sonuçlar, MGABC algoritmasının yedi farklı geliştirilmiş ABC varyantına kıyasla daha iyi veya eşdeğer performans gösterdiğini ortaya koymuştur; özellikle daha hızlı yakınsama oranları ve anlamlı düzeyde çözüm kalitesi iyileşmeleri sağlanmıştır. Bununla birlikte, çalışmada, arama sürecine birden fazla elit çözüm entegre edilmesinin bazı uygulamalarda hesaplama karmaşıklığını artırabileceği belirtilmiştir (Zhou ve ark., 2021).

Sharma ve ark., sınırlı optimizasyon problemlerinin çözümünde Yapay Arı Kolonisi (ABC) algoritmasının performansını artırmak amacıyla Geliştirilmiş Gbest Yapay Arı Kolonisi algoritması (Improved Gbest Artificial Bee Colony – IGABC) adı verilen bir algoritma geliştirmiştir. Çalışmaları, geleneksel Gbest yönlendirmeli ABC (GABC) algoritmasında görülen önemli bir sorun olan erken yakınsama probleminden (premature convergence) kaynaklanmıştır. Bu problem, tüm çözümlerin küresel en iyi (Gbest) çözüme göre yönlendirilmesi nedeniyle ortaya çıkmakta; bu durum keşif kabiliyetini azaltmakta ve aramayı çözüm uzayında sınırlı bölgelere hapsetmektedir. Bu sorunu ele almak için araştırmacılar algoritmada üç temel değişiklik önermiştir. Birinci değişiklik, çalışan arıların konum güncelleme mekanizmasının iyileştirilmesini

içermektedir. Bu aşamada, Gbest bileşeninin etkisi iterasyonlar boyunca kademeli olarak azaltılmış ve bu sayede keşif ile sömürü arasındaki denge geliştirilmiştir (Sharma ve ark., 2021).

İkinci değişiklik, gözcü arı aşamasına odaklanmıştır. Burada, rastgele bileşeninin etkisi aşamalı olarak azaltılmış ve bu sayede yönlendirilmemiş rastgele hareketler yerine daha hassas bir arama davranışı sağlanmıştır. Üçüncü değişiklik, izci arıların (scout bees) sayısının artırılmasını kapsamaktadır. Bu düzenleme, önceden tanımlanmış iterasyon sınırını aşmasına rağmen herhangi bir iyileşme göstermeyen arıların yeniden başlatılmasını sağlamıştır. Bu da erken yakınsamayı önlemiş ve arama sürecinde daha fazla çeşitliliği teşvik etmiştir. IGABC algoritmasının performansı, 21 benchmark optimizasyon problemi üzerinde değerlendirilmiştir. Sonuçlar, önerilen algoritmanın ABC, GABC, MABC ve BSFABC gibi önceki yaklaşımlardan daha üstün olduğunu göstermiştir. Özellikle IGABC, optimal çözümlere ulaşmak için daha az sayıda fonksiyon değerlendirmesine ihtiyaç duymuş, uygulanabilir çözümleri bulmada daha yüksek başarı oranı sergilemiş ve ortalama hata değerini minimize etmede daha yüksek doğruluk göstermiştir (Sharma ve ark., 2021).

Sömürü yeteneğini geliştirmek ve erken yakınsama riskini azaltmak amacıyla Çoklu Komşu Yönlendirme (Multi-Neighbor Guidance) kavramını içeren geliştirilmiş bir Yapay Arı Kolonisi (ABC) algoritması önermiştir ve bu algoritmaya ABC-MNG adını vermiştir. Çalışma, geleneksel ABC algoritmasının temel sınırlılıklarından biri olan zayıf sömürü kapasitesine odaklanmaktadır; bu sınırlılık genellikle alt-optimal çözümlerde durağanlığa yol açmaktadır. Araştırmacılar, yalnızca uygunluk değerlerine değil aynı zamanda Öklidyen mesafeye (Euclidean distance) de dayalı çoklu komşulardan elde edilen bilgilerin arama süreci için daha çeşitli bir yönlendirme mekanizması sağlayabileceğini öne sürmüştür. Önerilen metodoloji, popülasyon içerisindeki bireyleri organize etmek için halka topolojisi (ring topology) oluşturarak her bireyin üç tür komşu seçmesine olanak tanımaktadır: en iyi uygunluk değerine sahip komşu (best fitness neighbor), en uzak komşu (farthest neighbor) ve en yakın komşu (closest neighbor). Bu komşulara dayanarak üç yeni çözüm oluşturulmakta ve benzerlik metriği (similarity metric) kullanılarak değerlendirilmekte, böylece hesaplama yükü minimize edilmektedir. Ayrıca, arama geçmişinden faydalanan yeni bir izci arı arama parametresi (scout bee search parameter) önerilmiştir. Bu parametre, önceki arama deneyimlerinden yararlanırken aynı zamanda keşfi artırmayı hedeflemektedir. ABC-

MNG algoritmasının etkinliđi, iki benchmark test seti (CEC2013 ve CEC2017) ile birlikte bir gerek dnya optimizasyon problemi zerinde deęerlendirilmiřtir. Algoritma, on geliřmiř ABC varyantı ile karřılařtırılmıř ve keřif ile smr arasında daha iyi bir denge saęlayarak erken yakınsamayı azaltma konusunda stn performans sergilemiřtir. Bununla birlikte, alıřmada ayrıca, arama srecine eklenen bu oklu ynlendirme mekanizmasının bazı byk lekli problemler sz konusu olduęunda hesaplama karmařıklıęını artırılabileceęi de not edilmiřtir (Zhou ve ark., 2025).

Ni ve ark., Q-learning ve oklu arama stratejileri ile entegre edilmiř geliřtirilmiř bir Yapay Arı Kolonisi (ABC) algoritması olan QMABC adında yeni bir hibrit algoritma nermiřtir. alıřmaları, geleneksel ABC algoritmasının zellikle iterasyonlar sırasında yalnızca tek boyutlu arama stratejilerine (single-dimensional search strategies) dayanma eęiliminin, yavaş yakınsamaya ve zayıf keřif ve smr yeteneklerine neden olduęunu gzlemlemiřtir. Bu sınırlılıkları ařmak iin arařtırmacılar, Q-learning erevesi iinde birden fazla arama stratejisi entegre ederek arama srecinin durumuna (state of the search process) gre dinamik olarak en uygun stratejilerin seilmesini saęlamıřtır. alıřmada, yerel ve kresel arama (local and global search) karakteristikleri ile boyutsallık (dimensionality) bakımından farklılık gsteren eřitli yeni stratejiler tanıtılmıřtır. Her iterasyonda gncellenen boyut sayısı da stratejilere gre deęiřiklik gstermektedir. Ayrıca, uygun durum-eylem (state-action) eřlemesi yoluyla en uygun stratejinin adaptif olarak seilmesini saęlamak amacıyla bir Q-learning mekanizması benimsenmiřtir. nerilen algoritma, CEC 2017 standardına uygun benchmark fonksiyonları ve insansız ara gzergh planlaması (unmanned vehicle path planning) gibi pratik bir uygulama zerinde deęerlendirilmiřtir. Elde edilen sonular, QMABC algoritmasının zm doęruluęu (solution accuracy), yakınsama hızı (convergence speed) ve saęlamlık (robustness) aısından ok sayıda geliřmiř ABC varyantını ve dięer meta-sezgisel algoritmaları aıka geride bıraktıęını gstermiřtir (Ni ve ark., 2024).

KLABC olarak adlandırılan geliřtirilmiř bir model nermiřtir. Bu model, arama etkinlięini (search efficiency) artırmak ve erken yakınsama riskini azaltmak amacıyla keřif (exploration) ile smr (exploitation) arasında daha iyi bir denge kuran yeni mekanizmaları entegre etmektedir. alıřma, standart ABC algoritmasının karřılařtıęı temel zorluklardan biri olan bařlangı poplasyonunun (initial population) dřk kalitesine odaklanmıřtır; bu durum genellikle optimal zmlere yavaş yakınsamaya

neden olmaktadır. Bu sorunu çözmek için, araştırmacılar popülasyon başlatma aşamasında K-Means kümeleme algoritmasını entegre etmiş, böylece başlangıç çözümlerinin arama uzayında daha verimli şekilde dağılmasını sağlamış ve erken aşamalardan itibaren arama kalitesini artırmıştır. Ayrıca, geliştirilen algoritma, "İlk Giren İlk Çıkar" (First-In, First-Out - FIFO) mekanizması altında çalışan iki boyutlu kuyruk (queue)-tabanlı bir arama stratejisini de içermektedir. Bu mekanizma, popülasyon çeşitliliğinin korunmasına yardımcı olmuş ve iteratif arama süreci sırasında potansiyel optimal çözümlerin erken dışlanması önlemiştir. Önerilen KLABC algoritmasındaki bir diğer önemli iyileştirme ise, iki farklı çözüm seçme mekanizmasının Global En İyi (GBest) ve Kuyruk En İyisi (QBest) birleştirilmesidir. Bu hibrit yaklaşım, optimal çözümlere doğru daha verimli bir arama yönü elde edilmesini sağlamıştır. Bu değişiklikler sonucunda, KLABC algoritması geleneksel ABC algoritmasına kıyasla önemli ölçüde daha hızlı yakınsama göstermiştir. Performansını değerlendirmek amacıyla, araştırmacılar KLABC algoritmasını UCI veri deposundan (UCI repository) alınan yedi benchmark veri kümesi üzerinde test etmiş ve algoritmanın kümeleme optimizasyonundaki etkinliğini analiz etmiştir. Deneysel sonuçlar, KLABC algoritmasının standart ABC algoritmasına kıyasla ortalama olarak %55.63 oranında yakınsama hızında iyileşme sağladığını göstermiştir. Ayrıca, geliştirilen algoritma büyük ölçekli veri kümeleriyle başa çıkmada üstün performans sergilemiş ve yüksek boyutlu optimizasyon problemlerini çözmedeki etkinliğini kanıtlamıştır (Pan ve ark., 2024).

2.2 Hibrit ABC Yaklaşımları

ABC algoritması, diğer üst-sezgisel (metaheuristic) algoritmalarla melezleştirilerek (hibrit hâle getirilerek) önemli ölçüde geliştirilmiştir; bu yöntem, algoritmanın yerel ve küresel arama yeteneklerini artırmak için çeşitli yaklaşımların avantajlarından yararlanmaktadır. Keşif (exploration) ve sömürü (exploitation) arasında daha iyi bir denge kurarak, bu yöntem çözüm doğruluğunu artırmayı, yerel optimumlara karşı duyarlılığı azaltmayı ve yakınsamayı hızlandırmayı amaçlamaktadır.

Bu bağlamda, ayrılabilir olmayan problemlerin (non-separable problems) çözümünde ABC algoritmasının performansını artırmak amacıyla Kendini Uyarlayan Diferansiyel ABC (Self-Adaptive Differential ABC – sdABC) algoritmasını önermiştir. Bu tür

problemler, geleneksel optimizasyon algoritmaları için önemli zorluklar teşkil etmektedir. Önerilen algoritma, ABC çerçevesine kendini uyarlayan diferansiyel arama mekanizmalarını entegre ederek, arama verimliliğini artırmakta ve standart ABC algoritmasında görülen yavaş yakınsama sorunlarını hafifletmektedir. Önerilen yaklaşım, çalışan ve gözcü arı evrelerinde birden fazla diferansiyel arama stratejisini kullanmaktadır; her iterasyonda birden fazla değişken, DE algoritmasından esinlenen mutasyon ve çaprazlama (crossover) işlemleri ile güncellenmektedir. Ayrıca, algoritma süreci boyunca en uygun arama stratejisini dinamik olarak seçen kendini uyarlayan bir mekanizma da dâhil edilmiştir; bu mekanizma, algoritmanın verilen problemin özelliklerine uyum sağlama yeteneğini artırmaktadır. sdABC algoritması, hem ayrılabilir (separable) hem de ayrılabilir olmayan (non-separable) fonksiyonları içeren 28 benchmark fonksiyonu üzerinde test edilmiştir. Ayrıca, geliştirilmiş diğer ABC varyantları ve DE, PSO, TLBO gibi alternatif optimizasyon algoritmalarıyla karşılaştırılmıştır. Elde edilen sonuçlar, sdABC algoritmasının özellikle ayrılabilir olmayan problemlerde rakip algoritmalarla göre belirgin biçimde üstün performans sergilediğini ve daha yüksek çözüm doğruluğu elde ettiğini göstermiştir (Nematollahi ve ark., 2019).

ABC algoritmasının keşif kapasitesi ve yerel optimumlara takılma sorunlarını ele almak üzere HABC adını verdikleri hibrit bir algoritma önermiştir. Bu algoritma, çözüm kalitesine bağlı olarak popülasyon boyutunun dinamik olarak ayarlanmasına imkân tanıyan bir arı yaşam döngüsü modeli (bee life-cycle model) içermektedir. Bu sayede, keşif ve sömürü arasındaki denge daha etkin biçimde sağlanmıştır. Ayrıca, arama yönünü geliştirmek amacıyla Kapsamlı Öğrenme (Comprehensive Learning) stratejisi entegre edilmiş ve vaatkâr bölgelerde sömürüyü iyileştirmek için Powell Araması (Powell's Search) kullanılmıştır. HABC algoritması, CEC 2014 benchmark fonksiyonları kullanılarak değerlendirilmiş ve yakınsama hızı ile çözüm doğruluğu açısından PSO, GABC ve standart ABC gibi algoritmalarla göre üstünlük göstermiştir. Ancak, eklenen bu yeni mekanizmalar hesaplama karmaşıklığını artırmış ve basit problemlerde elde edilen iyileştirmeler anlamlı düzeyde olmamıştır. Genel olarak HABC, ABC algoritmasına önemli katkılar sağlamaktadır; ancak, farklı optimizasyon senaryolarında en yüksek performansı sağlamak adına daha fazla iyileştirme yapılması gerektiği sonucuna varılmıştır (Ma ve ark., 2016).

Glnur Yıldızdan ve Baykan, karmařık optimizasyon problemlerinde kresel arama yeteneklerini geliřtirmek amacıyla ABC algoritmasını Yarasa Algoritması (Bat Algorithm- BA) ile birleřtirerek BA_ABC adını verdikleri hibrit bir algoritma nermiřtir. Bu alıřma, yksek boyutlu problemlerde performansı olumsuz etkileyen BA algoritmasının sınırlı keřif yeteneęi problemini ele alırken, aynı zamanda ABC algoritmasının gçl kresel arama kapasitesinden yararlanmayı amalamıřtır. nerilen yntemde, keřif ve smr arasında denge saęlamak zere BA'nın hız denklemlerine bir eylemsizlik aęırlıęı katsayısı entegre edilmiřtir. Ayrıca BA ve ABC algoritmaları, ayrı alt poplasyonlar zerinde paralel olarak alıřtırılmıř ve bu iki algoritma arasında bilgi alıřveriři gerekleřtirilmiřtir; bu srete, bir algoritmadaki zayıf bireyler, dięerindeki yksek performanslı bireylerle deęiřtirilmiřtir. Yntemin geerlilięi, CEC2005 ve CEC2010 benchmark fonksiyonları ile bazı mhendislik tasarım problemleri zerinde yapılan deneylerle test edilmiřtir. Elde edilen sonular, BA_ABC algoritmasının, standart BA, standart ABC ve bu algoritmaların geliřtirilmiř versiyonlarına kıyasla daha dřk ortalama hata, daha yksek bařarı oranı ve daha az fonksiyon deęerlendirmesi gereksinimiyle daha iyi performans sergiledięini gstermiřtir. Ancak, yntemin hesaplama karmařıklıęını artırdıęı ve basit problemlerde yalnızca sınırlı iyileřtirmeler saęladıęı da gzlemlenmiřtir. Tm bu sınırlamalara raęmen, BA_ABC algoritması, doęadan esinlenen algoritmaların byk lekli kresel optimizasyon problemleri iin daha verimli hle getirilmesi aısından nemli bir geliřme olarak deęerlendirilmektedir (Yildizdan & Baykan, 2020).

ABC algoritması ile DE algoritmasını entegre ederek, modifiye edilmiř ABC (mABC) adını verdikleri hibrit bir varyant geliřtirmiřtir. Bu alıřmada, geleneksel ABC algoritmasının gçl keřif kapasitesine raęmen grece zayıf olan smr yeteneęi nedeniyle ortaya ıkan yavař yakınsama ve dřk zm hassasiyeti problemleri ele alınmıřtır. Geliřtirilen yaklařımda, ABC algoritmasındaki gzc arı ařaması, DE algoritmasından alınan mutasyon ve aprazlama iřlemleriyle deęiřtirilmiř ve bu deęiřiklik sayesinde zm eřitlilięi artırılarak optimal zmlere daha hızlı ulařılması saęlanmıřtır. mABC algoritmasının etkinlięi, 20 farklı benchmark fonksiyonu ve CEC 2014 standart test takımı kullanılarak yapılan deneylerle deęerlendirilmiřtir. Deneyisel bulgular, mABC algoritmasının zm doęruluęu, yakınsama hızı ve yerel optimumlardan kaınma yeteneęi aısından, literatrdeki dięer

birçok geliştirilmiş ABC varyantına kıyasla açık bir üstünlük gösterdiğini ortaya koymuştur (Ustun ve ark., 2022).

Zhao ve ark., Evrimsel Yapay Arı Kolonisi (Evolutionary Artificial Bee Colony – EABC) algoritması ile Hiyerarşik Parçacık Sürüsü Optimizasyonu (Hierarchical Particle Swarm Optimization – HPSO) algoritmasını birleştiren yeni bir hibrit yaklaşım önermiştir. Bu entegrasyon, küresel keşif (global exploration) ile yerel sömürü (local exploitation) arasında daha verimli bir denge elde etmeyi, böylece yakınsama hızını artırmayı, hesaplama maliyetlerini azaltmayı ve bulut kaynak tahsisini optimize etmeyi amaçlamaktadır. HPSO-EABC olarak adlandırılan önerilen hibrit algoritma, adaptif arama stratejileri (adaptive search strategies) ve yönlendirilmiş keşif mekanizmalarından yararlanarak, sıkı zaman kısıtlarına sahip çoklu görevlerin yönetiminde daha kararlı bir performans sunmayı hedeflemektedir. Önerilen yaklaşım, geleneksel ABC algoritmasının arama mekanizmasını, yerel optimumlara erken yakınsamayı önlemeye ve daha fazla arama çeşitliliği sağlamaya yardımcı olan rastgele pertürbasyon unsurlarını (Random Perturbation Elements) entegre ederek geliştirmiştir. Buna ek olarak, HPSO bileşeni, yakınsamayı hızlandırmak ve konum güncelleme mekanizmasını geliştirmek amacıyla algoritmaya dahil edilmiştir. Bu yapı içerisinde, parçacıkların hızları dinamik olarak ayarlanmakta ve böylece algoritmanın alt-optimal çözümlerde sıkışma olasılığı azaltılmaktadır. Ayrıca, keşif ve sömürü arasında en iyi dengenin sağlanması ve nihai çözüm kalitesinin iyileştirilmesi amacıyla, arama uzayını birden fazla seviyeye ayıran Hiyerarşik Arama Stratejisi (Hierarchical Search Strategy) uygulanmıştır. Bunun yanında, algoritmanın aramayı umut vadeden bölgelere yönlendirmesini sağlamak için Uyarlanabilir Seçim Mekanizmaları (Adaptive Selection Mechanisms) da entegre edilmiştir. Bu sayede, algoritmanın optimal çözümlere daha hızlı yakınsama kabiliyeti artırılmıştır. HPSO-EABC algoritmasının etkinliğini değerlendirmek amacıyla hem homojen hem de heterojen platformları içeren çeşitli bulut bilişim ortamlarında kapsamlı deneyler gerçekleştirilmiştir. Önerilen algoritmanın performansı, geleneksel ABC, standart PSO ve HPSO ile EABC gibi geliştirilmiş varyantlara karşı kıyaslanmıştır. Sonuçlar, HPSO-EABC algoritmasının önceki tüm sürümlerden daha iyi performans sergilediğini ve geleneksel algoritmalara kıyasla %20 ila %39 oranında daha hızlı yakınsama hızlarına ulaştığını göstermiştir. Ayrıca, algoritmanın görev yürütme süresini ABC'ye göre %25 ve PSO'ya göre %15 oranında azaltmada oldukça etkili olduğu kanıtlanmıştır. Bu sonuçlar, HPSO-

EABC'nin bulut kaynak kullanımını optimize etme potansiyelini açıkça ortaya koymaktadır. Ek olarak, deneyler, hiyerarşik arama ve adaptif komşuluk analizinin algoritmanın optimal çözümlere ulaşmak için ihtiyaç duyduğu iterasyon sayısını önemli ölçüde azalttığını ve bu yaklaşımın karmaşık zamanlama problemleri için daha hesaplamasal olarak verimli bir çözüm sunduğunu ortaya koymuştur (Zhao ve ark., 2024).

Song ve ark., ilaç keşfi sürecinde kritik bir görev olan Protein–Ligand Bağlanma (Protein–Ligand Docking) problemini çözmede performansı artırmak amacıyla, ABC-EDM adı verilen geliştirilmiş bir ABC algoritması varyantı önermiştir. Araştırmacılar bu sorunu, geleneksel ABC çerçevesine, dağılım tahmini algoritmalarından (Estimation of Distribution Algorithms – EDA) esinlenilerek geliştirilen özel bir Dağılım Tahmin Mekanizması (Estimation of Distribution Mechanism – EDM) entegre ederek ele almıştır. Bu yapı, yerel arama yeteneklerini ve yakınsama verimliliğini artırmayı hedeflemektedir. Önerilen algoritma, bağlanma probleminin özgün karakteristiklerinden faydalanan özel bir dağılım mekanizması (EDM) kullanmaktadır. Bu mekanizma özellikle, üç boyutlu uzayda ligand yerleşimi gibi, hedef fonksiyonu önemli ölçüde etkileyen kritik bölgelere odaklanarak çözüm üretmektedir. Bu sayede, algoritma yerel optimumlara daha etkili şekilde yönlenebilmekte ve çözüm kalitesini artırabilmektedir. Önerilen ABC-EDM algoritmasının performansı, bir benchmark veri kümesi olan GOLD veri seti üzerinde test edilmiş ve üç farklı ABC varyantı, üç diğer evrimsel algoritma ve yaygın olarak kullanılan AutoDock Vina yazılımı ile karşılaştırılmıştır. Deneysel sonuçlar, ABC-EDM algoritmasının çözüm doğruluğu, yakınsama hızı ve başarı oranı bakımından diğer algoritmalara kıyasla önemli derecede üstünlük sağladığını göstermiştir. Özellikle algoritma, AutoDock Vina'ya kıyasla yaklaşık %5 daha yüksek başarı oranına ulaşarak, protein–ligand bağlanma problemleri için dikkate değer bir alternatif olduğunu ortaya koymuştur (Song ve ark., 2024).

2.3 ABC Algoritmasına İlave Geliştirmelerin ve Stratejilerin Entegre Edilmesi

ABC algoritması, arama etkinliğini artırmayı ve erken yakınsama olasılığını azaltmayı hedefleyen yeni özelliklerin ve taktiklerin eklenmesiyle önemli iyileştirmeler geçirmiştir. Bu geliştirmeler, algoritmanın diğer algoritmalarla tamamen birleştirilmeden arama uzayını daha verimli biçimde tarama kapasitesini artırmaya

yönelik olarak; uyarlanabilir (adaptif) mekanizmalar, arşivleme stratejileri ve ileri düzey arama yaklaşımlarına odaklanmaktadır.

Sahin ve ark., özellikle Tüm Test Seti Oluşturma (Whole Test Suite Generation) gibi çok amaçlı optimizasyon problemlerinde, geleneksel ABC algoritmasının performansını artırmayı amaçlayan yeni bir algoritma önermiştir. Bu algoritmaya, Arşiv Tabanlı Çok Kriterli Yapay Arı Kolonisi (Archive-Based Multi-Criteria Artificial Bee Colony – MABC) adı verilmiştir. Çalışma, standart ABC algoritmasının en büyük sınırlılıklarından biri olan keşif ile sömürü arasında etkili bir denge kurma zorluğunu ele almayı hedeflemiştir. Bu zorlukların üstesinden gelmek için araştırmacılar, arama süreci boyunca baskın olmayan çözümlerden oluşan bir kümenin saklanması sağlayan gelişmiş bir arşivleme mekanizması önermiştir. Bu mekanizma, daha verimli bir arama yönlendirmesi sağlamaktadır. Ek olarak, algoritmaya çözüm çeşitliliğini artırmak ve yerel optimumlarda durağanlaşma olasılığını azaltmak amacıyla çok kriterli bir seçim stratejisi entegre edilmiştir. Bununla birlikte, izci arı fazı, arama sırasında umut vadeden çözümlerin kaybolmasını önlemek üzere uyarlanabilir teknikler ile geliştirilmiştir. MABC algoritmasının performansı, çok amaçlı optimizasyon problemlerine yönelik olarak özellikle yazılım mühendisliği alanında etkili test vakaları üretme amacıyla değerlendirilmiştir. Elde edilen sonuçlar, MABC algoritmasının hem standart ABC algoritmasına hem de diğer geliştirilmiş ABC varyantlarına kıyasla, çözüm çeşitliliği, nihai çözüm kalitesi ve arama verimliliği bakımından üstün performans sergilediğini ortaya koymuştur (Sahin ve ark., 2021).

Pian ve ark., tarafından yürütülen çalışmada, küresel optimizasyon problemlerinin çözümünde ABC algoritmasının performansını iyileştirmek amacıyla geliştirilmiş bir algoritma önerilmiştir. Araştırmacılar, OCGABC (Opposition and Chaos Guided Artificial Bee Colony) adını verdikleri geliştirilmiş bir varyant sunmuş ve bu algoritmaya iki temel strateji entegre etmişlerdir Öğrenilmeyeni Geri Alma Başlatması (Unlearning Initialization) ve ile Komşuluk Araması (Neighborhood Search with Crossover). Unlearning Initialization stratejisi, başlangıç aşamasında popülasyon çeşitliliğini artırmak için rastgele başlatılan çözümlerin karşıt (opposite) çözümlerini üretir ve bu çözümler arasından uygunluk değerlerine göre daha iyi olanları seçer. Bu yöntem, arama uzayının daha geniş şekilde kapsanmasına yardımcı olur ve algoritmanın yerel optimumlara sıkışma olasılığını azaltır. İkinci strateji ise, global en iyi çözümün arama denklemine dahil edilmesini ve çaprazlama faktörü tarafından

yönlendirilen bir crossover işleminin uygulanmasını içerir. Bu sayede, keşif ve sömürü arasında daha iyi bir denge kurulması sağlanır. OCGABC algoritmasının performansı, altı farklı benchmark fonksiyonu üzerinde değerlendirilmiş ve standart ABC algoritması ile karşılaştırılmıştır. Sonuçlar, OCGABC'nin doğruluk açısından önemli ölçüde daha yüksek performans sergilediğini ve daha hızlı yakınsama sağladığını ortaya koymuştur. Özellikle, algoritmanın yakınsama için ihtiyaç duyduğu iterasyon sayısının yaklaşık %8 ila %15 oranında azaldığı ve çözüm doğruluğunun neredeyse iki katına çıktığı gözlemlenmiştir. Ayrıca, en iyi performansın çaprazlama faktörü = 0.5 olduğunda elde edildiği belirlenmiş ve bu değerle yakınsama hızı ile çözüm kalitesi arasında en uygun dengenin sağlandığı tespit edilmiştir (Pian ve ark., 2018).

Banitalebi ve ark., geliştirilmiş Kompakt Yapay Arı Kolonisi (Enhanced Compact Artificial Bee Colony – EcABC) olarak adlandırılan yeni bir yaklaşım önermiştir. Bu yaklaşım, özellikle küçük ölçekli robotlar ve gömülü kontrol sistemleri gibi düşük kapasiteli cihazlar üzerinde çalışan uygulamalarda, minimum bellek kullanımı ve düşük hesaplama kaynakları gerektiren optimizasyon problemleriyle ilişkili zorlukları ele almak üzere geliştirilmiştir. Önerilen yöntem, çözüm popülasyonunun tamamını saklama gereksinimini ortadan kaldırarak, aday çözümleri fiziksel olarak depolamak yerine bireyleri olasılıksal bir şekilde temsil etmektedir. Bu yapı, arama kalitesinden ödün vermeden önemli ölçüde bellek tasarrufu sağlamaktadır. Algoritma, bireyleri fiziksel olarak saklamak yerine, popülasyonu olasılık dağılım fonksiyonu aracılığıyla sanal olarak temsil etmesine olanak tanıyan Dağılım Tahmin Algoritması (Estimation of Distribution Algorithm – EDA) çerçevesinden yararlanmaktadır. Buna ek olarak, algoritmanın arama evresi, dinamik olarak güncellenen bir olasılık dağılımı kullanacak şekilde değiştirilmiş ve bu sayede keşif ile sömürü arasında etkin bir denge sağlanmıştır. Bu denge, algoritmanın optimal çözümleri daha verimli bir şekilde tanımlama yeteneğini artırmıştır. EcABC algoritmasının performansını değerlendirmek amacıyla, araştırmacılar algoritmayı CEC2013 Benchmark Serisi kapsamındaki bir dizi test fonksiyonu üzerinde denemiş ve elde edilen sonuçları modern kompakt algoritmalar, geliştirilmiş ABC varyantları ve diğer meta-sezgisel optimizasyon teknikleri ile karşılaştırmıştır. Sonuçlar, EcABC algoritmasının diğer kompakt algoritmalara kıyasla anlamlı şekilde üstün olduğunu, daha hızlı yakınsama oranları ve daha yüksek çözüm doğruluğu elde ettiğini göstermiştir. Bu da algoritmanın, hesaplama kaynaklarının minimumda tutulduğu, yüksek verimlilik gerektiren optimizasyon

problemlerinde oldukça etkili olduğunu ortaya koymuştur. Ayrıca yapılan simülasyonlar, önerilen algoritmanın geleneksel popülasyon tabanlı ABC sürümleriyle rekabetçi performans sergilediğini ve bellek tasarrufu sağlarken arama kalitesinden ödün vermediğini doğrulamıştır (Banitalebi ve ark., 2015).

Babaeizadeh ve Ahmad özellikle sınırlı optimizasyon problemlerinin (Constrained Optimization Problems – COPs) çözümünde geleneksel ABC algoritmasının zayıf yerel sömürü kapasitesine yönelik sınırlılıklarını ele almak amacıyla, Geliştirilmiş Kısıtlı Yapay Arı Kolonisi (Enhanced Constrained Artificial Bee Colony – EC-ABC) adlı bir algoritma önermiştir. Araştırmacılar, algoritmanın çalışan arı ve gözücü arı evrelerine özel olarak uygulanmış iki yeni arama denklemi sunmuştur. Ayrıca, başlangıç popülasyonunun çeşitliliğini önemli ölçüde artıran ve yakınsamayı hızlandıran kaotik arama (chaotic search) ve karşıt tabanlı öğrenme (opposition-based learning) tekniklerine dayalı iki yenilikçi popülasyon başlatma yöntemi kullanılmıştır. Önerilen EC-ABC algoritmasının etkinliği, CEC 2006 test seti kapsamındaki iyi bilinen sınırlı optimizasyon benchmark fonksiyonları kullanılarak değerlendirilmiştir. Deneysel sonuçlar, EC-ABC algoritmasının özellikle çözüm kalitesi, yakınsama hızı ve yerel optimumlardan kaçınma yeteneği açısından, mevcut geliştirilmiş ABC algoritmalarına kıyasla açık bir üstünlük sergilediğini ortaya koymuştur (Babaeizadeh & Ahmad, 2017).

Bai ve ark., Dinamik Çok Amaçlı Optimizasyon Problemleri (Dynamic Multi-Objective Optimization Problems – DMOPs)'ni etkin biçimde çözmek amacıyla geliştirilmiş bir algoritma olan İki Düzeyli Paralel Ayrıştırma Tabanlı Yapay Arı Kolonisi (Two-Level Parallel Decomposition-based Artificial Bee Colony – 2LP-DABC) algoritmasını önermiştir. Çalışmaları, geleneksel optimizasyon algoritmalarının karşılaştığı kritik sorunları ele almıştır; özellikle, dinamik optimizasyon ortamlarında hızla değişen Pareto sınırlarını etkili bir şekilde takip edememe problemi bu zorlukların başında gelmektedir. Yazarlar, bu sorunu aşmak için iki düzeyli bir paralel ayrıştırma yapısı önermiştir. Birinci düzeyde, çok amaçlı problem, aynı anda çözülebilecek birden fazla tek amaçlı alt probleme ayrılmaktadır. İkinci düzeyde ise, geliştirilmiş bir paralel ABC algoritması, komşu çözümler arasında bilgi alışverişi sağlayarak arama verimliliğini artırmakta ve optimal çözümlere doğru yakınsamayı hızlandırmaktadır. Önerilen 2LP-DABC algoritmasının etkinliği, CEC 2018 test setindeki çeşitli benchmark problemleri ve iki pratik sınırlı optimizasyon

problemi üzerinde değerlendirilmiştir. Sonuçlar, 2LP-DABC algoritmasının, özellikle yakınsama hızı, çözüm kalitesi ve dinamik Pareto sınırlarını doğru bir şekilde takip etme yeteneği açısından, yerleşik algoritmalara kıyasla üstün performans sergilediğini ortaya koymuştur (Bai ve ark., 2023).

Li ve ark., özellikle Çok Amaçlı Optimizasyon Problemleri (Many-objective Optimization Problems) için geliştirilmiş, MaOABC-TA adı verilen gelişmiş bir Yapay Arı Kolonisi (ABC) algoritması varyantı önermiştir. Araştırmacılar, geleneksel ABC algoritmasının karşılaştığı temel zorluklara dikkat çekmişlerdir; özellikle hedef sayısı arttıkça, yakınsama (convergence) ile popülasyon çeşitliliği (diversity) arasında denge sağlama konusunda yetersiz kaldığı belirtilmiştir. Bu zorlukları aşmak için, yazarlar hem yakınsama hem de çeşitlilik üzerine odaklanan çift arşivli (dual-archive) yeni bir yapı önermişlerdir: Yakınsama Arşivi (Convergence Archive) ve Çeşitlilik Arşivi (Diversity Archive). Ayrıca, daha yüksek çeşitlilik gösteren çözümleri tercih eden yeni bir olasılıksal seçim mekanizması (probabilistic selection mechanism) geliştirilmiştir; bu mekanizma, algoritmanın yakınsama hızı ile çeşitlilik koruma kabiliyeti arasında daha iyi bir denge kurmasına olanak sağlamaktadır. DTLZ ve MaF test takımlarından seçilen standart benchmark problemleri üzerinde yürütülen deneysel sonuçlar, önerilen MaOABC-TA algoritmasının, çözüm kalitesi ve performans göstergeleri açısından, diğer ABC tabanlı ve evrimsel çok amaçlı optimizasyon algoritmalarına kıyasla açık üstünlük sergilediğini göstermiştir (Li ve ark., 2024).

Yavuz ve ark., Distant Savants adı verilen elit ve çeşitli çözümler kümesine dayalı yeni bir seçim yaklaşımını entegre eden, geliştirilmiş bir ABC algoritması varyantı olan ABCDS (Artificial Bee Colony with Differential Search) algoritmasını önermiştir. Araştırmanın temel amacı, geleneksel ABC algoritmasının, özellikle sınırlı optimizasyon problemleriyle karşılaşıldığında, zayıf yerel arama kabiliyeti ve erken yakınsamaya olan eğilim gibi sınırlılıklarını aşmaktır. Önerilen ABCDS algoritması, çalışan arı ve gözcü arı fazlarını yönlendirmek amacıyla, popülasyon içerisinde yüksek performanslı ancak uzamsal olarak birbirinden uzak çözümleri seçme fikrini içermektedir. Bu strateji, çözümler arasındaki çeşitliliği artırmakta ve çözüm doğruluğunu iyileştirmektedir. Ayrıca algoritma, sömürü verimliliğini artırmak için birden fazla yerel arama stratejisi arasından rekabete dayalı bir seçim mekanizması kullanmaktadır. ABCDS algoritmasının performansı, CEC 2017 test fonksiyonları ile CEC 2020 benchmark serisinden gerçekçi mühendislik optimizasyon problemleri

üzerinde değerlendirilmiştir. Elde edilen sonuçlar, önerilen algoritmanın özellikle yüksek boyutlu problemlerde, birçok güncel ABC varyantı ve gelişmiş algoritmaya kıyasla anlamlı derecede üstün performans sergilediğini ortaya koymuştur. Bununla birlikte, araştırmacılar "Distant Savants" stratejisinin entegrasyonunun hesaplama karmaşıklığını artırdığını kabul etmiş ve bu ek yükün azaltılması için gelecekte daha fazla çalışmaya ihtiyaç olduğunu belirtmişlerdir (Yavuz ve ark., 2022).

Li ve ark., özellikle Gezgin Satıcı Problemi (Travelling Salesman Problem – TSP) gibi ayrık optimizasyon problemlerini çözmek amacıyla geliştirilen, Sabit Komşuluk Aramasıyla Ayrık Yapay Arı Kolonisi (Discrete Artificial Bee Colony with Fixed Neighborhood Search – DABC-FNS) adlı geliştirilmiş bir algoritma önermiştir. Bu çalışma, geleneksel ABC algoritmasının erken yakınsamaya yatkınlığı ve ayrık problemlerde, özellikle TSP'de, sınırlı çözüm doğruluğu gibi önemli sınırlılıklarını ele almayı amaçlamıştır. Bu sınırlamaları aşmak için araştırmacılar, tam sayı kodlamalı (integer-encoding) bir arama stratejisini, 2-opt stratejisine dayalı sabit boyutlu komşuluk araması gibi geliştirilmiş yerel arama mekanizmaları ile birleştiren yeni bir yaklaşım önermiştir. Bu yapı, çözüm doğruluğunu artırmakta ve yerel optimumlara sıkışmayı azaltmaktadır. Algoritmanın etkinliği, 30'dan fazla TSP benchmark örneği üzerinde deneysel olarak değerlendirilmiştir. Elde edilen sonuçlar, DABC-FNS algoritmasının çözüm kalitesi ve yakınsama hızı bakımından, hem geleneksel ABC algoritmasına hem de diğer geliştirilmiş ABC varyantlarına kıyasla belirgin bir üstünlük sergilediğini göstermiştir. Ayrıca algoritmanın, yerel optimumlardan kaçınma yeteneğinin oldukça güçlü olduğu da gözlemlenmiştir (Li ve ark., 2024).

Etimaniefahani ve ark., ABC algoritması ile Fibonacci Göstergesi Algoritması (Fibonacci Indicator Algorithm – FIA) algoritmasını birleştiren, ABFIA adlı yeni bir hibrit algoritma önermiştir. Bu çalışma, geleneksel ABC algoritmasının yerel arama yeteneğindeki sınırlılıklar nedeniyle ortaya çıkan yavaş yakınsama sorunlarını ele almayı amaçlamıştır. Buna karşılık, doğrusal tabanlı hızlı yakınsama özelliğiyle bilinen FIA algoritması, yüksek boyutlu problemlerde çoklu çözümleri keşfetme konusunda zayıf kalmaktadır. Her iki algoritmanın güçlü yönlerinden yararlanmak amacıyla, yazarlar FIA algoritmasını ABC algoritmasının gözcü arı ve izci arı evrelerine entegre etmişlerdir. Bu entegrasyon, çözüm doğruluğunu önemli ölçüde artırmış, yakınsama hızını yükseltmiş ve yerel optimumlarda takılma problemlerini azaltmıştır. Önerilen ABFIA algoritması, CEC2019 test fonksiyonları üzerinde değerlendirilmiş ve çözüm

kalitesi ile yakınsama hızı açısından, hem geleneksel ABC ve FIA algoritmalarına hem de son dönemlerde önerilen çeşitli optimizasyon algoritmalarına kıyasla üstün performans sergilemiştir (Etmianiesfahani ve ark., 2022).

Bu gelişmeler bir araya getirildiğinde, ABC algoritmasının farklı uygulama alanlarında yapısal tasarım, çizelgeleme, yol planlaması ve kombinatorial optimizasyon gibi geniş bir esneklik sunduğu görülmektedir. ABC algoritması doğal olarak güçlü bir küresel keşif yeteneğine sahip olmasına rağmen, literatürdeki birçok çalışma, sömürü kabiliyetinin dikkatlice kalibre edilmesi ve popülasyon çeşitliliğinin korunması gerekliliğine vurgu yapmaktadır. Özelleştirilmiş yerel arama stratejilerinden karmaşık topolojik uyarlamalara kadar uzanan yöntemlerle, ABC üzerine yürütülen araştırmalar sürekli genişlemekte ve giderek daha karmaşık hâle gelen optimizasyon problemleriyle baş edebilmek adına yeni stratejilerin geliştirilmesini teşvik etmektedir.

3 TEMEL ABC ALGORİTMASI

ABC algoritması, doğada bal arısı sürülerinin sergilediği zekice besin arama davranışını temel düzeyde taklit etmektedir. ABC modelinde, üç farklı arı grubu çalışan arılar (employed bees), gözcü arılar (onlooker bees) ve izci arılar (scout bees) potansiyel besin kaynaklarını birlikte keşfeder. Buna bağlı olarak, algoritmanın yinelemeli arama süreci bu üç arı türüne karşılık gelen çalışan arı evresi, gözcü arı evresi ve izci arı evresi olmak üzere üç ardışık aşamadan oluşur (Karaboga, 2005).

ABC algoritmasında, her bir besin kaynağı konumu, optimize edilmesi gereken problemin bir aday çözümünü temsil eder. Diğer evrimsel algoritmalarda olduğu gibi, ABC algoritması arama sürecine önceden tanımlanmış sayıda aday çözümünden (SN) oluşan bir başlangıç popülasyonu oluşturarak başlar. Problem boyutu D olarak verildiğinde, popülasyondaki i^{th} . besin kaynağı şu şekilde matematiksel olarak ifade edilir (Denklem 1):

$$X_i = (x_{i,1}, x_{i,2}, \dots, x_{i,D}) \quad (1)$$

Burada, X_i aday çözümünün her bir karar değişkeni $x_{i,j}$, önceden belirlenmiş sınırlar dâhilinde rastgele olarak başlatılır. Bu rastgele başlatma süreci aşağıdaki formül ile tanımlanır (Denklem 2):

$$x_{i,j} = x_j^{min} + rand(0,1) \cdot (x_j^{max} - x_j^{min}) \quad (2)$$

Burada, $[x_j^{min}, x_j^{max}]$, j karar değişkeninin alt ve üst sınırlarını temsil ederken, $rand(0,1)$ ifadesi $[0,1]$ aralığında uniform dağılımlı rastgele bir sayıyı ifade etmektedir. Bu şekilde yapılan başlatma prosedürü, arama uzayının kapsamlı ve çeşitli bir şekilde taranmasını garanti eder ve arıların umut vadeden bölgeleri etkili biçimde keşfedebilmeleri için sağlam bir temel oluşturur.

Bu rastgele başlatma yaklaşımı, arama uzayı genelinde dengeli bir keşfi teşvik eder ve algoritmanın yakın-optimal çözümleri sonraki arama aşamalarında daha verimli ve doğru bir şekilde tanımlama potansiyelini artırır (Karaboga & Akay, 2009).

3.1 Çalışan Arı Evresi

Başlangıç popülasyonu oluşturulduktan sonra, ABC algoritması üç ardışık aşamayı yinelemeli olarak yürütür: çalışan arı evresi, gözcü arı evresi ve izci arı evresi. Özellikle çalışan arı evresinde, her çalışan arı mevcut popülasyondan bir besin kaynağına (aday çözüme) atanır. Bu aşamanın temel amacı, her çözümün çevresindeki yerel komşuluk bölgesini keşfetmek ve böylece yeni bir potansiyel aday çözüm (offspring) üretmektir. Bu offspring üretim süreci, aşağıdaki çözüm güncelleme denklemi ile gerçekleştirilir (Denklem 3):

$$v_{i,j} = x_{i,j} + \phi(x_{i,j} - x_{k,j}) \quad (3)$$

Burada $V_i=(v_{i,1},v_{i,2},\dots,v_{i,D})$ özgün ebeveyn çözüm X_i ile ilişkilendirilmiş yeni oluşturulan besin kaynağını (offspring) temsil ederken, $X_i=(x_{i,1},x_{i,2},\dots,x_{i,D})$ çalışan arının hâlihazırda sömürdüğü mevcut besin kaynağıdır. $x_{k,j}$, mevcut popülasyondan rastgele seçilen ve X_i 'den farklı olan bir X_k çözümünün j^{th} . boyutunu temsil eder. Bu rastgele seçim işlemi, $(x_{i,j}-x_{k,j})$ teriminin yeterli çeşitlilik sağlamasını ve sıfır fark oluşumunun önlenmesini garanti eder. Ayrıca, $\phi_{i,j}$ katsayısı $[-1, 1]$ aralığında uniform dağılımlı rastgele bir sayı olup, aramaya stokastiklik katar ve etkin bir yerel keşif yapılmasını kolaylaştırır.

Yeni aday çözüm V_i üretildikten sonra, bu çözümün uygunluk (fitness) değeri mevcut çözüm X_i ile karşılaştırılır. Eğer V_i daha yüksek bir uygunluk değeri gösterirse, X_i bu yeni çözüm ile değiştirilir; bu durum bir iyileşme anlamına gelir. Aksine, yeni çözümün uygunluğu daha düşükse, orijinal çözüm X_i değişmeden kalır ve bir sonraki iterasyona taşınır; bu da herhangi bir iyileşme olmadığını gösterir (Karaboga & Akay, 2009).

Karşılaştırma ve seçim kararları açısından gerekli olan uygunluk değeri hesaplaması şu formülasyonla ifade edilir (Denklem 4):

$$fit(X_i) = \begin{cases} \frac{1}{1+f(X_i)}, & \text{if } f(X_i) \geq 0 \\ 1+|f(X_i)|, & \text{otherwise} \end{cases} \quad (4)$$

Burada, $fit(X_i)$ ve $f(X_i)$ sırasıyla aday çözüm X_i 'ye ait uygunluk ve amaç fonksiyonu değerlerini ifade eder. $abs(\cdot)$ ifadesi, mutlak değer fonksiyonunu belirtir. Bu uygunluk hesaplama yöntemi, pozitif ve karşılaştırılabilir değerler elde edilmesini garanti eder ve aday çözümler arasında sistematik değerlendirme ve tutarlı karşılaştırma yapılmasını mümkün kılar (Karaboga & Basturk, 2008).

Sonuç olarak, çalışan arı evresinin temel rolü, mevcut çözümler etrafında yerel arama yapmak ve böylece ABC algoritmasının yerel sömürü kapasitesini etkin bir biçimde artırmaktır.

3.2 Gözcü Arı Evresi

Gözcü arı evresinde, algoritma aramasına, işlevsel olarak çalışan arılardan farklı olan gözcü arılar grubunu kullanarak devam eder. Gözcü arılar, önceki aşamada çalışan arılar tarafından keşfedilen besin kaynaklarının konumları ve nektar miktarları hakkında bilgi edinir. Bu paylaşılan bilgi, gözcü arıların daha önceki değerlendirmelere göre umut vadeden arama alanlarına yönlendirilmesinde temel bir rol oynar (Karaboga, 2005).

Daha umut vadeden besin kaynaklarının sistematik olarak seçilmesini sağlamak için, ABC algoritması genellikle rulet tekerleği seçimi (roulette-wheel selection) olarak adlandırılan olasılıksal bir seçim mekanizması kullanır. Bu yöntemde, uygunluk değeri yüksek olan çözümlerin seçilme olasılığı da yüksektir. Bu seçim olasılığı aşağıdaki şekilde matematiksel olarak ifade edilir (Denklem 5):

$$p_i = \frac{fit(X_i)}{\sum_{j=1}^{SN} fit(X_j)} \quad (5)$$

Bu denklemde p_i , i^{th} besin kaynağı X_i 'nin seçilme olasılığını temsil ederken, $fit(X_i)$ bu çözümün uygunluk değerini ifade etmektedir. Payda ise, mevcut tüm besin kaynaklarının toplam uygunluk değeridir ve böylece seçim olasılığı doğrudan uygunlukla orantılı hâle gelir (Karaboga & Akay, 2009).

Gözcü arılar, bu olasılıksal çerçeveye göre besin kaynaklarını seçtikten sonra, her bir seçilen kaynak daha hassas bir keşif sürecine tabi tutulur. Bu keşif süreci, çalışan arı evresinde tanımlanmış olan aynı çözüm üretme denklemi kullanılarak gerçekleştirilir (Denklem 6):

$$v_{i,j} = x_{i,j} + \phi_{i,j} \cdot (x_{i,j} - x_{k,j}) \quad (6)$$

Çalışan arı evresinde olduğu gibi, yeni oluşturulan aday çözümler, ebeveyn çözümlerle karşılaştırılır. Açgözlü seçim kriteri (greedy selection) tekrar uygulanır; buna göre, yalnızca daha yüksek uygunluk değerine sahip olan çözüm, popülasyonda sonraki yinelemeler için korunur. Bu nedenle, gözcü arı evresinin temel işlevi, önceki arama

aşamalarında tanımlanan yüksek kaliteli çözümler etrafında sömürüyü yoğunlaştırmak olup, algoritmanın optimum veya optimuma yakın çözümleri bulmadaki genel etkinliğine ve doğruluğuna önemli ölçüde katkı sağlar (Karaboga & Akay, 2009).

3.3 İzci Arı Evresi

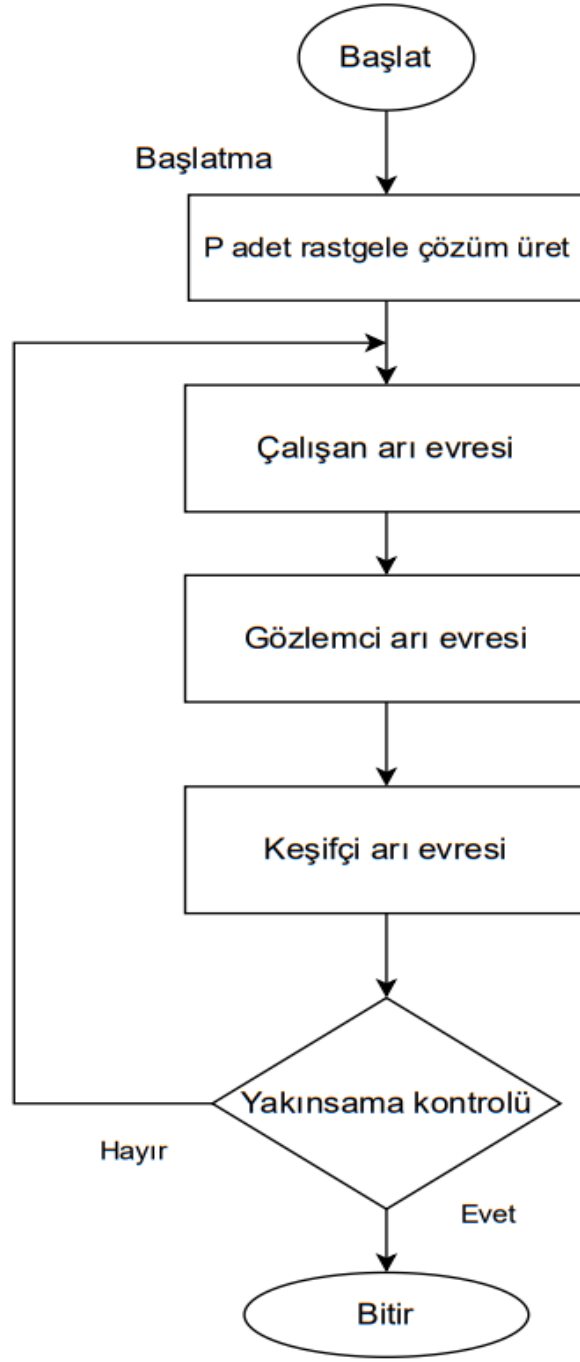
Çalışan ve gözcü arı evrelerinin aksine, ABC algoritmasındaki izci arı evresinin temel rolü, yerel keşfi veya sömürüyü yoğunlaştırmak değildir. Aksine, bu evre, popülasyon çeşitliliğini koruma ve erken yakınsama durumlarını önleme açısından kritik bir işlev üstlenir. Bu amaçla, ardışık yinelemeler boyunca gelişim göstermeyen besin kaynaklarının (aday çözümlerin) yeniden başlatılması sağlanır. Bir aday çözümün iyileşmeden kaldığı tekrar sayısını sistematik olarak izlemek için, her çözüm için ayrı ayrı tanımlanan bir sayım değişkeni $trial_i$ kullanılır (Karaboga, 2005).

Özellikle, çalışan ve gözcü arı evrelerinde, eğer bir aday çözüm iyileşme göstermezse (yani üretilen offspring çözüm üstün değilse), bu çözümle ilişkili $trial_i$ sayacı bir artırılır. Aksine, eğer çözüm iyileşirse, bu sayaç sıfırlanır. Bu izleme mekanizması sayesinde algoritma, durağanlaşan veya yerel optimumlarda takılı kalan çözümleri etkin bir şekilde tanımlayabilir.

Bu mekanizmanın uygulanabilir hâle gelmesi için, algoritma limit olarak bilinen önceden tanımlı bir parametre kullanır. Bu parametre, durağanlığa tolerans eşiğini temsil eder. Eğer bir besin kaynağının $trial_i > \text{limit}$ koşulunu sağladığı tespit edilirse, söz konusu çözüm artık “tükenmiş” veya “terk edilmiş” olarak değerlendirilir. Bu durumda, bir izci arı, ilgili besin kaynağını daha önce tanımlanmış rastgele başlatma yöntemi ile yeniden oluşturur (Denklem 7):

$$x_{i,j} = x_j^{min} + rand(0,1) \times (x_j^{max} - x_j^{min}) \quad (7)$$

Bu rastgele yeniden başlatma işlemi, popülasyona sürekli olarak yeni aday çözümlerin eklenmesini sağlar ve böylece küresel keşfi artırarak arama sürecinin sağlamlığını (robustness) güçlendirir. Tükenmiş çözümlerin periyodik olarak yenilenmesi sayesinde, izci arı evresi, algoritmanın arama uzayındaki alt-optimal bölgelerde durağanlaşmasını engellemede hayati bir rol oynar. Böylece, algoritmanın uzun vadeli performansı ve yakınsama güvenilirliği artırılmış olur (Karaboga & Basturk, 2008).



Şekil 3. 1 ABC Algoritmasının Akış Şeması
(Karaboga, 2005)

Şekil 3.1’de, standart ABC algoritmasının tüm iş akışını göstermektedir. Süreç, başlatma (Initialization) aşamasıyla başlar; bu aşamada, arama alanı içerisinde rastgele olarak P adet besin kaynağı konumu (aday çözüm) üretilir. Ardından, algoritma üç ardışık evreden oluşan ana döngüsüne girer.

Çalışan Arı Evresi sırasında, her çalışan arı, kendisine atanan besin kaynağının komşuluğunu keşfeder; bu keşif, mevcut pozisyonu üzerinde küçük değişiklikler yapılarak yeni bir aday çözüm üretilmesiyle gerçekleştirilir. Eğer yeni çözüm, daha iyi bir uygunluk değeri gösterirse, eski çözüm ile yer değiştirir. Daha sonra, Gözcü Arı Evresi başlar. Bu evrede, gözcü arılar, çalışan arılar tarafından iletilen uygunluk bilgilerine dayanarak olasılıksal olarak besin kaynaklarını seçer ve benzer şekilde yerel arama işlemleri gerçekleştirir. Böylece, arama uzayında umut vadeden bölgelere daha fazla hesaplama kaynağı ayrılır. İzci Arı Evresi kapsamında ise, önceden tanımlanmış sayıda denemeden sonra iyileşme göstermeyen herhangi bir besin kaynağı “terk edilmiş” kabul edilir ve bunun yerine rastgele oluşturulmuş yeni bir çözüm ile değiştirilir. Bu adım, popülasyon çeşitliliğini korumak ve erken durağanlaşmayı (stagnation) önlemek açısından önemlidir.

Bu üç fazın sonunda, yakınsama kontrolü (Convergence Check) gerçekleştirilir. Örneğin, maksimum fonksiyon değerlendirme sayısına ulaşıp ulaşılmadığı ya da tatmin edici bir uygunluk eşiğinin sağlanıp sağlanmadığı kontrol edilir. Eğer yakınsama kriteri karşılanmamışsa döngü tekrar eder; kriter sağlandığında algoritma sonlanır ve bulunan en iyi çözüm döndürülür. Bu yapılandırılmış süreçte, keşif (scout arılar ve rastgele yeniden başlatma yoluyla) ile sömürü (çalışan ve gözcü arı aramaları yoluyla) bir arada yürütülerek, ABC algoritmasının karmaşık optimizasyon yüzeylerinde etkili gezinme yeteneği artırılmış olur.

4 YÖNTEM

Önerilen Komşuluk Birleştirme Tabanlı Yapay Arı Kolonisi (NJ-ABC) algoritması, orijinal ABC algoritmasını geliştirmek amacıyla çeşitli yenilikçi mekanizmalar entegre ederek, keşif-sömürü dengesini iyileştirmeyi, yakınsama hızını artırmayı ve erken yakınsamayı önlemeyi hedeflemektedir. Bu bölümde, önerilen yöntemin temel katkıları ve algoritmaya getirilen iyileştirmeler ayrıntılı şekilde açıklanmaktadır:

4.1 Latin Hiperküp Örnekleme (LHS) ile Başlatma

Arama uzayının geniş ve homojen şekilde kapsanmasını sağlamak amacıyla, başlangıç popülasyonu Latin Hiperküp Örnekleme (Latin Hypercube Sampling – LHS) yöntemiyle oluşturulur (McKay ve ark., 2000). Rastgele başlatmanın aksine, LHS yöntemi tüm arama uzayı boyunca daha eşit dağılımlı noktalar üretir ve böylece algoritmanın erken safhalarında çeşitlilik ile keşif potansiyelini önemli ölçüde artırır (Denklem 8):

$$X_{i,j} = X_j^{min} + LHS(FoodNumber, D) \times (X_j^{max} - X_j^{min}) \quad (8)$$

Burada X_j^{min} ve X_j^{max} sırasıyla alt ve üst sınırları, D ise optimizasyon probleminin boyutunu temsil etmektedir.

4.2 Uyarlanabilir Keşif-Sömürü Kontrolü

Önerilen algoritma, optimizasyon süreci boyunca keşif (exploration) ve sömürü (exploitation) arasında dinamik denge sağlamak amacıyla, zamanla uyarlanan bir kontrol faktörü (α_t) kullanmaktadır (Bansal ve ark., 2013). Bu faktör, iterasyonlar ilerledikçe üstel olarak azalan bir yapıdadır (Denklem 9):

$$\alpha_t = \alpha_0 \times e^{-\frac{t}{MaxGen}} \quad (9)$$

Burada, $\alpha_0 = 0.7$ olarak alınmıştır, t mevcut jenerasyonu, Max^{Gen} ise toplam maksimum jenerasyon sayısını temsil eder. Bu yapı, algoritmanın başlangıçta yoğun keşif yapmasını ve ilerleyen evrelerde geliştirilmiş sömürü gerçekleştirmesini sağlar.

4.3 Geliştirilmiş Çözüm Güncelleme Denklemleri

Klasik çalışan ve gözcü arı güncelleme denklemleri, uyarlanabilir faktör (α_t) entegre edilerek yerel arama yeteneklerinin artırılması amacıyla yeniden yapılandırılmıştır (Bansal ve ark., 2013) (Denklem 10):

$$v_{i,j} = x_{i,j} + \alpha_t \cdot \phi_{i,j}(x_{i,j} - x_{k,j}) \quad (10)$$

Burada $\phi_{i,j}$, $[-1,1]$ aralığında uniform dağılımlı rastgele bir sayı, $x_{k,j}$ ise mevcut çözüm $x_{i,j}$ 'den farklı olarak rastgele seçilen bir çözüm bileşenidir.

4.4 Çeşitliliğe Dayalı Kısmi Yeniden Başlatma

Erken yakınsamayı önlemek amacıyla, algoritma her yinelemede popülasyon çeşitliliğini ölçer (Denklem 11):

$$diversity(t) = \frac{1}{D} \sum_{j=1}^D std(X_{:,j}) \quad (11)$$

Eğer bu çeşitlilik ölçütü önceden belirlenen eşik değerin (1×10^{-3}) altına düşerse, popülasyondaki en düşük performansa sahip %30'luk kesim, şu ana kadar elde edilen en iyi çözüm etrafında kısmen yeniden başlatılır. Bu sayede, arama süreci boyunca çeşitliliğin korunması sağlanır (Denklem 12):

$$X_{i,new} = X_{best} + \alpha_t \times randn(1, D) \times (UB - LB) \quad (12)$$

4.5 Parçacık Sürüsü Dinamikleri ile Geliştirilmiş İzci Arı Evresi

Klasik ABC algoritmasındaki izci arı evresinden farklı olarak, NJ-ABC algoritması, Parçacık Sürüsü Optimizasyonu (PSO) esinli bir mekanizma kullanır. Limit parametresini aşarak durağanlaşmış çözümler, PSO dinamiklerinden türetilmiş bir hız (velocity) terimi kullanılarak güncellenir (Denklem 13):

$$v_i = rand \times v_i + 2 \times rand \times (X_{best} - X_i) + rand \times (X_{r1} - X_{r2}) \quad (13)$$

Burada X_{best} , küresel en iyi çözümü, X_{r1} ve X_{r2} ise rastgele seçilmiş iki farklı çözümü temsil eder. Bu güncelleme yöntemi, uzun süre değişmeden kalan (durağan) çözümlere çeşitlilik kazandırarak arama sürecinin etkinliğini artırmaktadır.

4.6 Neighborhood-Joining (NJ) Hiyerarşik Kümeleme

NJ-ABC algoritmasının temelini oluşturan yenilik, başlangıçta hesaplamalı biyolojide filogenetik ağaçların oluşturulması amacıyla geliştirilen Komşuluk Birleştirme (NJ) kümeleme yönteminin algoritmaya entegre edilmesidir (Saitou & Nei, 1987). NJ-ABC algoritmasında, bu yöntem yeniden uyarlanarak, optimizasyon süreci sırasında çözümlerin dinamik olarak hiyerarşik kümeler hâlinde organize edilmesi amacıyla kullanılır. Bu kümeleme işlemi, her T_{NJ} iterasyonda (örneğin her 5 jenerasyonda bir) gerçekleştirilerek, hesaplama maliyeti ile kümeleme etkinliği arasında denge kurulması hedeflenir.

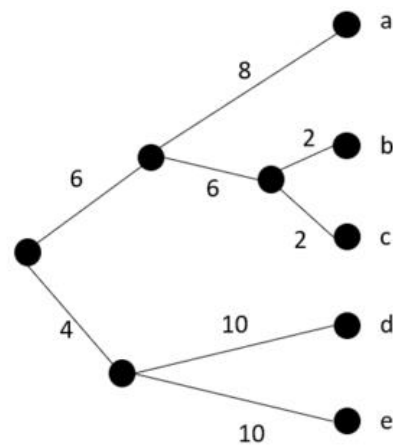
4.6.1 Uzaklık Matrisi Hesaplanması

Kümeleme süreci, mevcut tüm besin kaynakları (çözümler) arasındaki çiftli Öklid uzaklığı matrisinin hesaplanması ile başlar (Jain ve ark., 1999) (Denklem 14):

$$d(X_i, X_j) = \sqrt{\sum_{k=1}^D (x_{i,k} - x_{j,k})^2} \quad (14)$$

Bu uzaklık matrisi, algoritmanın yüksek boyutlu çözüm uzayında çözümler arasındaki benzerlikleri ölçmesine olanak tanır ve hiyerarşik kümeleme işleminin temelini oluşturur.

M	a	b	c	d	e
a	0	16	16	28	28
b	16	0	4	28	28
c	16	4	0	28	28
d	28	28	28	0	20
e	28	28	28	20	0



Şekil 4. 1 NJ Kümeleme Sürecinde Uzaklık Matrisinin Evrimi

Şekil 4.1, bir çözüm kümesi (a, b, c, d, e) arasındaki çiftli uzaklık matrisini ve bu uzaklıklara göre çözümlerin minimum mesafelere dayalı olarak nasıl bağlandığını

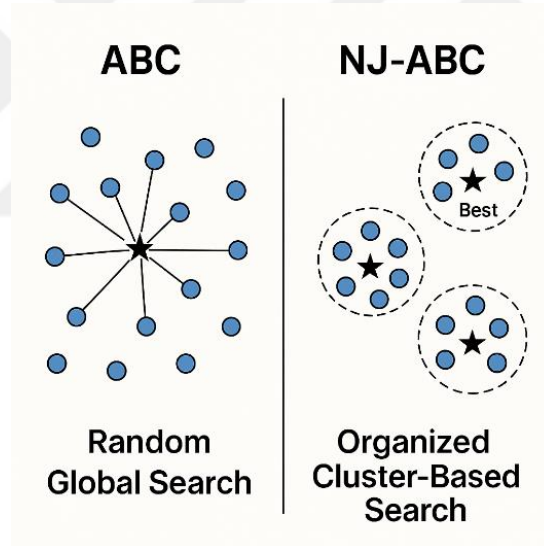
gösteren grafiği sunmaktadır. Bu yapı, yerel kümelerin oluşturulması için bir temel teşkil eder; böylece, uzamsal olarak birbirine yakın çözümler gruplanarak, NJ-ABC algoritmasındaki küme tabanlı arama stratejisinin etkinliği artırılır.

4.6.2 Küme oluşturma

Komşuluk Birleştirme stratejisi kullanılarak, en benzer iki çözüm birleştirilerek yeni bir düğüm (küme) oluşturulur ve uzaklık matrisi aşağıdaki şekilde güncellenir (Denklem 15):

$$d(X_{uv}, X_r) = \frac{1}{2}(d(X_u, X_r) + d(X_v, X_r) - d(X_u, X_v)) \quad (15)$$

Bu işlem, tüm hiyerarşi oluşuncaya kadar yinelenmeli olarak tekrarlanır. Böylece, popülasyon, aday çözümler arasındaki yakınlık ve benzerliği yansıtan ağaç benzeri bir yapı içerisinde gruplanmış olur.



Şekil 4. 2 ABC Algoritmasındaki Rastgele Küresel Arama ile NJ-ABC'deki Düzenli Küme Tabanlı Aramanın Görselleştirilmesi

Şekil 4.2'de gözlemlendiği üzere, standart ABC algoritması, çözümlerin organize olmayan bir şekilde tüm arama alanına dağılmış olduğu rastgele küresel bir arama stratejisine dayanır. Buna karşılık, NJ-ABC algoritması, en iyi keşfedilen çözümler etrafında oluşturulan yerel kümeler içinde düzenli bir arama süreci yürütür. Bu küme tabanlı yapı, keşif verimliliğini artırırken, aynı zamanda yerel sömürü kalitesini önemli ölçüde iyileştirir. Bu da, arama süreci boyunca popülasyon genelinde yeterli çeşitliliği koruyarak, optimal çözümlere daha hızlı yakınsamaya olanak tanır.

4.6.3 Çözüm arama sürecindeki rolü

NJ kümeleme yapısının entegrasyonu, çalışan ve gözcü arıların davranışlarını doğrudan etkilemektedir:

- Yerelleştirilmiş Arama (Localized Search): Her arı, yalnızca kendisine atanan küme içinde keşif yapacak şekilde sınırlandırılır; bu sayede, yüksek potansiyele sahip alt-uzaylarda sömürü kabiliyeti artırılır.
- Çeşitliliğin Korunması (Maintaining Diversity): Farklı kümeler yarı-bağımsız olarak evrimleştiği için, arama uzayının çeşitli bölgeleri aynı anda keşfedilebilir.
- Ölçeklenebilirlik ve Yapısallık (Scalability and Structure): Popülasyonun kümelere ayrılması, algoritmanın arama karmaşıklığını azaltır ve çözümler arası daha yapılandırılmış bir etkileşim biçimi sunar.

4.6.4 Geleneksel ABC üzerindeki avantajları

Tablo 4. 1 Standart ABC ile Kümeleme Mekanizmasına Sahip Önerilen NJ-ABC Arasındaki Temel Farklılıklar

Özellik	Standart ABC	Kümelemeli NJ-ABC
Popülasyon Yapısı	Düz (yapısız)	Hiyerarşik (kümelenmiş)
Bilgi Paylaşımı	Küresel ve rastgele	Kümeler içinde yerel
Arama Yönü	Yönsüz	Küme benzerliğine göre yönlendirilmiş
Erken Yakınsama Riski	Yüksek	Azaltılmış

NJ kümeleme mekanizması, klasik ABC algoritmasına yapısal farkındalık (structural awareness) katmanını entegre ederek onu önemli ölçüde zenginleştirmektedir. Uyarlanabilir kümelerin oluşturulması, algoritmanın umut vadeden bölgeleri daha etkili şekilde sömürmesini sağlarken, aynı zamanda tüm arama alanı boyunca çeşitliliği korumasına da olanak tanır. Bu kümeleme stratejisi yalnızca yakınsama hassasiyetini artırmakla kalmaz, aynı zamanda algoritmanın karmaşık, çok modlu ve yüksek boyutlu optimizasyon problemlerinde sağlamlığını (robustness) korumasına da yardımcı olur.

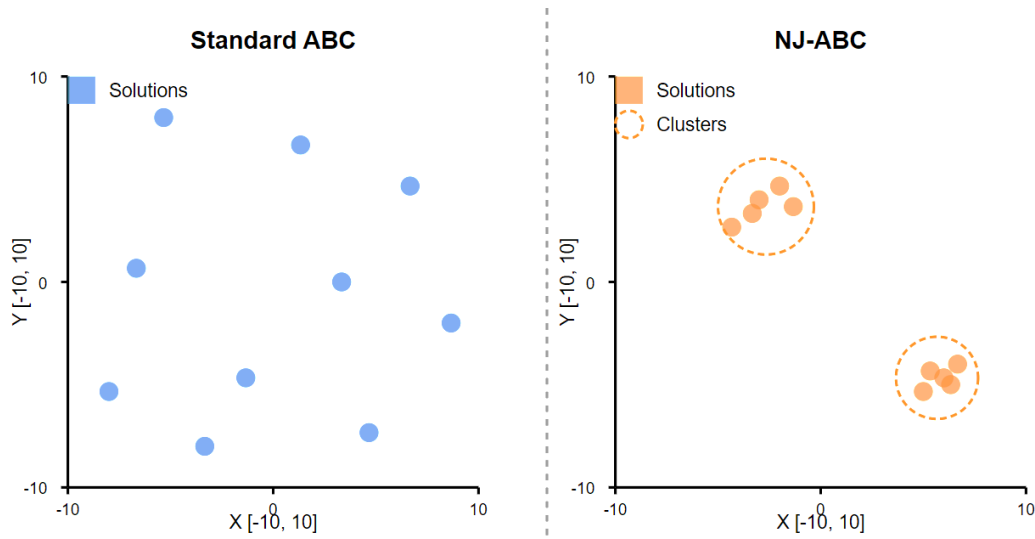
4.7 Küme Tabanlı Yönlendirilmiş Arama

Kümeler oluşturulduktan sonra, çalışan ve gözcü arılar, komşuluk aramalarını yalnızca kendilerine atanan kümeler içindeki çözümlerle sınırlandırır (Ozturk ve ark., 2014). Buna ek olarak, %30 olasılıkla (olasılık = 0.3) arılar, pertürbasyon denklemi için küme merkezini (centroid) seçer ve böylece umut vadeden kümeler içinde sömürü süreci yoğunlaştırılır (Denklem 16):

$$v_{i,j}^{clustered} = x_{i,j} + \alpha_t \times \phi_{i,j}(x_{centroid,j} - x_{i,j}) \quad (16)$$

Bu güncelleme, çözüm X_i 'nin küme merkezine doğru kısmi olarak hareket etmesini sağlar. Bu, küme merkezinin koordinatları ile X_i 'nin koordinatları arasındaki farkın, uyarlamalı faktör α_t ve rastgele pertürbasyon $\phi_{i,j}$ ile ölçeklenmesi yoluyla gerçekleştirilir. Bu mekanizma, aramayı umut vadeden kümeler içinde yoğunlaştırarak, yerel sömürü kapasitesini artırır ve optimal çözümlere doğru yakınsama doğruluğunu iyileştirir.

Şekil 4.3'te, Standart ABC algoritması (sol) ile NJ-ABC algoritması (sağ) arasında çözüm dağılımlarının görsel karşılaştırması sunulmaktadır. Görüldüğü üzere, standart ABC algoritmasında çözümler arama alanı boyunca rastgele dağılmıştır ve herhangi bir yapısal gruplaşma gözlenmemektedir. Buna karşılık, NJ-ABC algoritması açık şekilde umut vadeden bölgeler etrafında kümelenme sergiler. Bu yapı, arama sürecini daha verimli yönlendirir ve küme merkezleri etrafında sömürüyü odaklayarak çözüm kalitesini artırır.



Şekil 4. 3 Standart ABC İle Küme Tabanlı Arama Yönlendirmesine Sahip NJ-ABC Arasındaki Çözüm Dağılımlarının Karşılaştırması

4.8 Gradyan Tabanlı Yerel Arama İyileştirmesi

Son olarak, gradyan tabanlı yerel bir arama adımı, çözümün doğruluğunu artırmak ve yakınsamayı hızlandırmak amacıyla uygulanır. Bu amaçla, her jenerasyonun sonunda belirlenen küresel en iyi çözüm (X_{best}), MATLAB'ın `fmincon` fonksiyonu kullanılarak sıkı sınır koşulları altında yerel optimizasyona tabi tutulur (Denklem 17):

$$X_{best}^{refined} = fmincon(X_{best}, LB, UB, options) \quad (17)$$

Burada yakınsama toleransı 1×10^{-6} olarak belirlenmiş ve maksimum iterasyon sayısı 50 ile sınırlandırılmıştır.

Bu adım, MATLAB'ın `fmincon` fonksiyonunu kullanarak, mevcut X_{best} çözümünden başlatılan kısıtlı yerel optimizasyon gerçekleştirir. Alt (LB) ve üst (UB) sınırlar dahilinde kalınarak, sıkı yakınsama toleransı (10^{-6}) ile sadece 50 iterasyonluk bir optimizasyon gerçekleştirilir. Bu odaklanmış yerel arama, çözümün keskinliğini artırır ve optimum noktaya daha hızlı yakınsama sağlar.

4.9 Paralel Hesaplama Yapısı

Hesaplama verimliliğini artırmak amacıyla, NJ-ABC algoritması paralel hesaplama ortamında uygulanmıştır. Parallelleştirme, birden fazla düzeyde gerçekleştirilmiştir:

- Çiftli uzaklık matrislerinin paralel değerlendirilmesi
- NJ hiyerarşik küme birleştirme işlemlerinin paralel yürütülmesi
- Çalışan, gözcü ve izci arı fazlarının, ayrı işlemci çekirdeklerinde eş zamanlı yürütülmesi

Bu strateji, hesaplama süresini önemli ölçüde azaltmakta ve artırılan popülasyon büyüklüğünde ölçeklenebilir performans sergilemektedir.

4.10 Sonlandırma Kriterleri

NJ-ABC algoritması, önceden tanımlanmış maksimum jenerasyon sayısına (MaxGen) ulaşılan kadar veya maksimum fonksiyon değerlendirme sayısı (MaxFEVs) tamamlanana dek yinelemeli olarak çalıştırılır. Bu süreç sonunda, bulunan en iyi

çözüm, yakınsama istatistikleriyle birlikte kaydedilir ve sonraki analizler için saklanır. Bu gelişmiş mekanizmaların entegre edilmesi sayesinde, NJ-ABC algoritması, güçlü, verimli ve hassas bir optimizasyon yaklaşımı sunmakta olup, hem benchmark testleriyle hem de gerçek mühendislik tasarım problemleriyle doğrulanmıştır.

Algoritma 1. NJ-ABC Algoritmasının sözde kodu

Algoritma 1: NJ-ABC Algoritmasının Sözde Kodu

```

1: Arama sınırları içinde LHS (Latin Hiperküp Örnekleme) yöntemiyle SN adet başlangıç
   yiyecek kaynağı oluşturulur.
2: Tüm başlangıç çözümlerinin uygunluk (fitness) değerleri hesaplanır.
3: FEs < MaxFEVs olduğu sürece döngü başlatılır
4:   Uyarlanabilir faktör  $\alpha_i$  güncellenir
   // çalışan arı evresi
5:   Her çalışan arı (çözüm  $X_i$ ) için
6:     Değiştirilmiş denklem kullanılarak yeni birey  $V_i$  oluşturulur.
7:      $V_i$  değerlendirilir.
8:     Eğer  $\text{fitness}(V_i)$ ,  $\text{fitness}(X_i)$ 'den daha iyiye
9:        $X_i = V_i$  olarak güncellenir ve  $\text{trial}_i = 0$  yapılır.
10:    Aksi takdirde
11:       $\text{trial}_i = \text{trial}_i + 1$  olarak artırılır.
12:    Bitiş
13:  Döngü sonu
  // gözlemci arı evresi
14:  Seçim olasılıkları  $P_i$  hesaplanır.
15:  Her gözlemci arı için
16:     $P_i$  değerine göre bir çözüm  $X_i$  seçilir.
17:    Kümeler içerisinde yeni birey  $V_i$  oluşturulur.
18:     $V_i$  değerlendirilir.
19:    Eğer  $\text{fitness}(V_i)$ ,  $\text{fitness}(X_i)$ 'den daha iyiye
20:       $X_i = V_i$  olarak güncellenir ve  $\text{trial}_i = 0$  yapılır.
21:    Aksi takdirde
22:       $\text{trial}_i = \text{trial}_i + 1$  olarak artırılır.
23:    Bitiş
24:  Döngü sonu
  // keşifçi arı evresi
25:  Her çözüm  $X_i$  için
26:    Eğer  $\text{trial}_i > \text{limit}$  ise
27:       $X_i$ , PSO ilhamlı dinamikler kullanılarak güncellenir.
28:    Bitiş
29:  Döngü sonu
30:  Her  $T_{NJ}$  iterasyonunda NJ (Komşuluk Birleştirme) hiyerarşik kümeleme uygulanır.
31:  Küresel en iyi çözüm üzerinde gradyan tabanlı yerel arama gerçekleştirilir.
32:  En iyi çözüm kaydedilir ve yakınsama geçmişi güncellenir.
33: Döngü sonu
34: Küresel en iyi çözüm ve yakınsama istatistikleri döndürülür.

```

4.11 NJ-ABC Algoritmasının Yapısal Çerçevesi

Önerilen Komşuluk Birleştirme Tabanlı Yapay Arı Kolonisi (NJ-ABC) algoritması, orijinal ABC algoritmasına kıyasla yakınsama kararlılığı, yapısal çeşitlilik ve sömürü kapasitesini artırmaya yönelik çeşitli önemli iyileştirmeler içermektedir. Bu geliştirmeler şunları kapsamaktadır:

(i) Uyarlanabilir adım büyüklüğü düzenleme mekanizması, (ii) Komşuluk Birleştirme (NJ) hiyerarşik kümeleme entegrasyonu, (iii) PSO esinli izci arı stratejisi, (iv) Gradyan tabanlı yerel arama iyileştirmesi.

Algoritmanın genel yapısını açıklamak üzere Şekil 4.4'te bir akış diyagramı sunulmuştur ve Algoritma 1'de ayrıntılı sözde kodu vermektedir. Bu sözde kodda:

FEs, şimdiye kadar gerçekleştirilen uygunluk fonksiyonu değerlendirme sayısını, MaxFEs, toplam değerlendirme bütçesini, $trial_i$, her bir bireysel çözüm X_i için ardışık başarısız güncelleme sayısını temsil eder.

NJ-ABC algoritmasında yapılan iyileştirmeler, Tablo 4-2'deki aşağıdaki satırlara yansıtılmıştır:

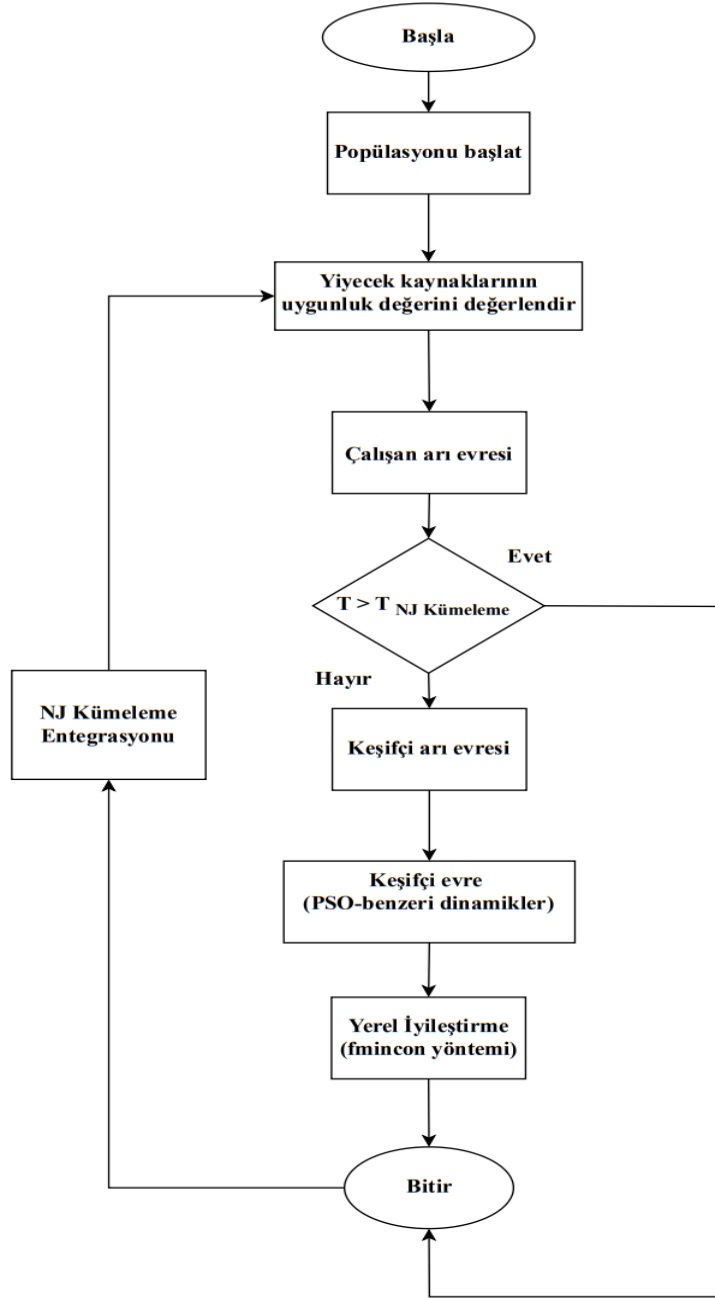
Satır 5-13 (Çalışan Arı Evresi) Bu aşamada her çalışan arı, uyarlanabilir parametre α_t kullanılarak değiştirilmiş arama denklemleriyle yeni bir aday çözüm üretir. Ayrıca, önceden tanımlanmış bir olasılıkla, NJ kümeleme ile elde edilen küme merkezi, arama denklemleri içerisinde dâhil edilerek, aramanın umut vadeden yerel bölgelere yönlendirilmesi sağlanır. Satır 14-24 (Gözcü Arı Evresi) Çözümlerin uygunluk değerlerine dayalı olarak seçim olasılıkları hesaplandıktan sonra, gözcü arılar en umut vadeden besin kaynaklarını seçer. Burada oluşturulan yeni aday çözümler, son NJ kümeleme işlemiyle tanımlanan kümeler içerisinde oluşturulur. Bu yerellik odaklı strateji, yüksek performanslı bireyler etrafında yoğun arama yapılmasını sağlar ve alt popülasyonlar içinde sömürü başarımını artırır.

Satır 25-29 (İzci Arı Evresi) Geleneksel rastgele yeniden başlatma stratejisi yerine, PSO esinli bir güncelleme mekanizması kullanılır. Bir çözümün $trial_i$ sayacı belirli bir eşiği (limit) aşarsa, çözüm, küresel en iyi çözüm ve rastgele seçilen diğer bireyler yönünde, hız benzeri bir güncellemeyle yeniden yönlendirilir. Bu, yönlendirilmiş çeşitliliği artırır ve arama uzayındaki verimli bölgelere yeniden giriş imkânı sunar. Satır 30 (NJ Kümeleme Entegrasyonu) Her sabit sayıda jenerasyonda (örneğin $T_{NJ} = 5$), Komşuluk

Birleştirme hiyerarşik kümeleme yöntemi çağrılır. Bu işlem, çiftli Öklid mesafelerine dayalı olarak popülasyonu yeniden bölümlendirir ve çözümlerin dinamik olarak gruplanmasını sağlar. Bu yapı, sonraki iterasyonlarda çalışan ve gözcü arılar için aramayı yönlendiren temel yapıyı oluşturur.

Satır 31 (Yerel Arama İyileştirmesi) Her jenerasyonun sonunda, mevcut küresel en iyi çözüm, MATLAB fmincon optimizasyon fonksiyonu kullanılarak gradyan tabanlı yerel bir iyileştirmeye tabi tutulur. Bu adım, en iyi çözümün ince ayarını yapar, düz bölgelerden kaçışı kolaylaştırır ve yakınsama doğruluğunu artırır.

Popülasyonu hiyerarşik şekilde yapılandırarak ve aramayı uyarlanabilir şekilde kontrol ederek, NJ-ABC algoritması, orijinal ABC'ye kıyasla keşif ve sömürüyü daha dengeli bir şekilde yürütmektedir. Bu çerçeve, aynı zamanda çeşitliliği korur, erken yakınsama olasılığını azaltır ve umut vadeden bölgelerde daha derin yakınsamayı teşvik eder.



Şekil 4. 4 NJ-ABC Algoritmasının Akış Diyagramı

Şekil 4.4'te algoritma, başlangıçta popülasyonu oluşturarak ve tüm çözümlerin uygunluklarını değerlendirerek başlar. Her yinelemede çalışan arı evresi uygulanır. Eğer mevcut yineleme belirli bir eşik değeri aşarsa, çözüm kümelerini iyileştirmek amacıyla NJ kümeleme süreci devreye girer. Aksi takdirde, keşif yeteneğini artırmak için PSO'dan esinlenen bir güncelleme içeren izci arı evresi başlatılır. Son olarak, en iyi çözümleri hassaslaştırmak için yerel arama yöntemi olarak fmincon fonksiyonu kullanılır. Bu yapı, algoritmanın hem keşif hem de faydalanma yeteneklerini geliştirmektedir.

5 DENEYSEL DOĞRULAMA VE SONUÇ ANALİZİ

Önerilen Komşuluk Birleştirme Tabanlı Yapay Arı Kolonisi (NJ-ABC) algoritmasının etkinliğini ve sağlamlığını titizlikle değerlendirmek amacıyla, deneysel değerlendirme iki tamamlayıcı bölüm hâlinde yapılandırılmıştır. Bu bölümde, farklı karmaşıklık seviyelerine sahip çeşitli sentetik optimizasyon problemleri içeren CEC 2022 test takımı (Kumar ve ark., 2022) kullanılarak, önerilen algoritmanın performansı diğer algoritmalarla karşılaştırılmıştır. İkinci bölüm ise, tek eksenli eğilme altında betonarme kolonların maliyet-optimal tasarımı gibi gerçek dünya mühendislik uygulamasına odaklanmaktadır. Buradaki amaç, tüm yapısal güvenlik kısıtlarını sağlarken malzeme maliyetini minimize etmektir (Bekdaş ve ark., 2021; Nigdeli ve ark., 2015).

5.1 CEC 2022 Benchmark Fonksiyonlarındaki Performans

Bu bölüm, önerilen NJ-ABC algoritmasının, tek amaçlı optimizasyon algoritmalarının performansını değerlendirmek için yaygın olarak kabul edilen bir standart olan CEC 2022 benchmark takımı kullanılarak yapılan deneysel değerlendirmesini sunmaktadır. Test takımı, yapısal karmaşıklık ve yüzey özelliklerine göre dört farklı gruba ayrılmış on iki test fonksiyonundan oluşmaktadır; bu gruplar Tablo 5.1 ve 5.2’de özetlenmiştir (Kumar ve ark., 2022).

Bu kategoriler şunlardır yerel sömürü yeteneğini değerlendiren tek tepe fonksiyonları (F1); algoritmanın keşif yeteneğini test eden temel çok tepe fonksiyonları (F2–F5); değişkenleri, her biri farklı bir temel fonksiyonla yönetilen çoklu alt bileşenlere ayıran hibrit fonksiyonlar (F6–F8); ve birden fazla kaydırılmış ve döndürülmüş temel fonksiyonun bir araya getirilmesiyle oluşturulan bileşik fonksiyonlar (F9–F12). Bu son grup, karmaşık, girintili-çıkıntılı ve yanıltıcı yüzey yapısıyla en zorlu kategori olarak kabul edilmektedir. Benchmark yapısı, algoritmaların keşif ve sömürü arasında denge sağlama yeteneklerini çeşitli ve karmaşık problem alanlarında değerlendirmek amacıyla özel olarak tasarlanmıştır (Kumar ve ark., 2022).

Tüm fonksiyonlar, $[-100,100]^D$ aralığında tanımlı sınırlı bir karar uzayı üzerinde tanımlanmıştır ve NJ-ABC algoritması, ölçeklenebilirlik ve sağlamlık açısından $D=10$ ve $D=20$ olmak üzere iki boyutsal ayarda test edilmiştir. Tüm karşılaştırmalı algoritmalar, aynı popülasyon büyüklükleri, jenerasyon sınırları ve tekrar sayıları kullanılarak eşit deneysel koşullarda test edilmiştir. Bilinen küresel optimal değerler,

tüm testlerde doğruluk ve hata metriklerinin tutarlı bir şekilde hesaplanmasını sağlamak için kullanılmıştır.

Bu kapsamlı değerlendirme çerçevesi, NJ-ABC algoritmasında sunulan iyileştirmelerin doğruluğunu titizlikle doğrulamak ve algoritmanın çözüm kalitesi, yakınsama hızı ve sağlamlık açısından çeşitli benchmark problemleri üzerinde birçok güncel meta-sezgisel algoritmaya karşı performansını karşılaştırmak için sağlam bir temel sağlar.

Tablo 5. 1 CEC 2022 Benchmark Fonksiyonlarının Özeti
(Kumar ve ark., 2022)

Category	Function Name	Mathematical Definition	Complexity Factors	Key Features
Unimodal	F1: Shifted and Rotated Zakharov	$\sum_{i=1}^D x_i^2 + \left(\sum_{i=1}^D 0.5 x_i \right)^2 + \left(\sum_{i=1}^D 0.5 x_i \right)^4$	Low	Tests basic convergence ability
	F2: Shifted and Rotated Rosenbrock	$\sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	High (Narrow valleys, hard to converge)	Requires precise search strategies
	F3: Shifted and Rotated Expanded Schaffer's F6	$\sum_{i=1}^{D-1} \left[0.5 + \frac{\sin^2(\sqrt{x_i^2 + x_{i+1}^2}) - 0.5}{(1 + 0.001(x_i^2 + x_{i+1}^2))^2} \right]$	High (Many local optima)	Requires strong exploration
Basic Multimodal	F4: Shifted and Rotated Non-Continuous Rastrigin	$\sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i)]$	High (Non-continuous, highly multimodal)	Harder due to discontinuities
	F5: Shifted and Rotated Levy	$\sin^2(\pi w_1) + \sum_{i=1}^{D-1} (w_i - 1)^2 [1 + 10 \sin^2(\pi w_i + 1)] + (w_D - 1)^2 [1 + \sin^2(2\pi w_D)]$ where $w_i = 1 + \frac{x_i - 1}{4}$	High (Large number of local optima)	Hard for global search
	F6: Hybrid Function 1 (N = 3)	This function integrates six basic functions, each contributing to the overall landscape.	Very High (Multiple function combinations)	Requires hybrid search strategies
Hybrid	F7: Hybrid Function 2 (N = 6)	This function integrates six basic functions, each contributing to the overall landscape.	Very High (Increased complexity)	Harder than F6
	F8: Hybrid Function 3 (N = 5)	Combining five distinct functions, this hybrid function presents a challenging optimization scenario.	Very High (Different subcomponent effects)	Complex interaction effects

Tablo 5.1 (Devam) CEC 2022 Benchmark Fonksiyonlarının Özeti
(Kumar ve ark., 2022)

Category	Function Name	Mathematical Definition	Complexity Factors	Key Features
Composition	F9: Composition Function 1 (N = 5)	This function composes five sub-functions to form a complex optimization landscape.	Extremely High (Combination of 5 different landscapes)	Large-scale optimization challenge
	F10: Composition Function 2 (N = 4)	Incorporating four sub-functions, this composition function creates a multifaceted landscape for optimization.	Extremely High (Requires extensive adaptation)	Multiple function blending
	F11: Composition Function 3 (N = 5)	This function combines five different functions to challenge optimization algorithms.	Extremely High (Severe multimodal interactions)	Tests robustness against misleading optima
	F12: Composition Function 4 (N = 6)	Integrating six sub-functions, this composition function presents a highly complex optimization scenario.	Extremely High (Most complex function in CEC 2022)	Extreme challenge for optimization algorithms

Tablo 5. 2 CEC2022 Tek Amaçlı, Sınırlı Bağlı Sayısal Optimizasyon Problemlerinin Özeti
(Kumar ve ark., 2022)

	No.	Functions	F_i^*
Unimodal Function	1	Shifted and full Rotated Zakharov Function	300
	2	Shifted and full Rotated Rosenbrock's Function	400
	3	Shifted and full Rotated Expanded Schaffer's f6 Function	600
Basic Functions	4	Shifted and full Rotated Non-Continuous Rastrigin's Function	800
	5	Shifted and full Rotated Levy Function	900
	6	Hybrid Function 1 (N = 3)	1800
Hybrid Functions	7	Hybrid Function 2 (N = 6)	2000
	8	Hybrid Function 3 (N = 5)	2200
	9	Composition Function 1 (N = 5)	2300
Composition Functions	10	Composition Function 2 (N = 4)	2400
	11	Composition Function 3 (N = 5)	2600
	12	Composition Function 4 (N = 6)	2700
Search range: $[-100,100]^D$			

5.1.1 Deneysel kurulum

Önerilen NJ-ABC algoritmasının performansı, CEC 2022 tek amaçlı, sınırlı bağlı benchmark seti üzerinde iki farklı problem boyutunda ($D = 10$ ve $D = 20$) değerlendirilmiştir. Her bir boyut için, popülasyon büyüklüğü $D = 10$ için $N = 100$, $D = 20$ için $N = 200$ olarak belirlenmiştir. Durdurma kriteri, maksimum iterasyon (jenerasyon) sayısı olarak tanımlanmıştır ve farklı deneysel senaryolara bağlı olarak 1000, 2000 ve 100000 iterasyon bütçeleri test edilmiştir. Sağlamlık ve istatistiksel anlamlılık değerlendirmeleri için, her yapılandırma bağımsız olarak 10 kez çalıştırılmış ve performans metrikleri bu çalıştırmalar üzerinden ortalama alınarak hesaplanmıştır.

Tüm hesaplamalı deneyler, Intel Core i7 (11. Nesil) işlemcili, 3.6 GHz temel frekansa sahip, 16 GB DDR4 3200 MHz RAM ile donatılmış, Windows 11 Pro (64 bit) işletim sistemine sahip bir iş istasyonunda gerçekleştirilmiştir. Algoritma, MATLAB R2023b ortamında geliştirilmiş olup, Parallel Computing Toolbox kullanılarak uygunluk (fitness) değerlendirmeleri çok çekirdekli işlemcilerde paralel olarak yürütülmüş ve genel çalışma süresi azaltılmıştır.

5.1.2 Optimizasyon performansı

Önerilen NJ-ABC algoritmasının performansı, CEC 2022 benchmark seti kullanılarak titizlikle değerlendirilmiş ve elde edilen sonuçlar, algoritma tarafından bulunan en iyi çözüm ile küresel optimum arasındaki mutlak fark (hata metriği) temelinde analiz edilmiştir. Bu hata değeri $1E-8$ veya daha küçük olduğunda, sıfır olarak kabul edilmiştir. Her bir test fonksiyonu için, Best (En iyi), Worst (En kötü), Median (Medyan), Mean (Ortalama) ve Standard Deviation (Std) metrikleri, 10 bağımsız çalıştırma üzerinden hesaplanmıştır. Ayrıntılı sonuçlar, $D = 10$ boyutlu testler için Tablo 5.3’de, $D = 20$ boyutlu testler için Tablo 5.4’te sunulmuştur.

Tablo 5. 3 NJ_ABC Algoritmasının D = 10 Boyutunda Performans Metrikleri

Func.	Best	Worst	Median	Mean	Std.
F1	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F2	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F3	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F4	1.80E+00	1.45E+01	7.00E+00	6.80E+00	2.75E+00
F5	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F6	6.50E+00	1.80E+03	1.20E+02	3.50E+02	4.50E+02
F7	1.90E-08	2.15E+01	1.75E+00	8.50E+00	9.50E+00
F8	1.10E-01	2.30E+01	2.00E+01	1.50E+01	9.00E+00
F9	2.28E+02	2.31E+02	2.29E+02	2.30E+02	9.00E-02
F10	1.00E+02	1.00E+02	1.00E+02	1.00E+02	6.00E-02
F11	0.00E+00	3.10E+02	0.00E+00	2.60E+01	8.00E+01
F12	1.62E+02	1.66E+02	1.65E+02	1.65E+02	4.25E+01

Tablo 5. 4 NJ_ABC Algoritmasının D = 20 Boyutunda Performans Metrikleri

Func.	Best	Worst	Median	Mean	Std.
F1	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F2	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F3	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F4	2.50E+00	2.00E+01	1.00E+01	9.80E+00	3.50E+00
F5	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F6	1.20E+01	3.20E+03	2.00E+02	5.60E+02	6.80E+02
F7	2.50E-08	3.50E+01	3.00E+00	1.30E+01	1.20E+01
F8	1.90E-01	3.50E+01	2.80E+01	2.20E+01	1.20E+01
F9	4.50E+02	4.65E+02	4.60E+02	4.58E+02	1.50E-01
F10	2.00E+02	2.00E+02	2.00E+02	2.00E+02	1.20E-01
F11	0.00E+00	5.80E+02	5.00E+01	6.00E+01	1.50E+02
F12	3.25E+02	3.35E+02	3.30E+02	3.29E+02	5.10E+01

Tablo 5.3 ve 5.4'te gösterilen sonuçlara dayanarak, önerilen NJ-ABC algoritması, CEC 2022 benchmark fonksiyonlarının tüm kategorilerinde istikrarlı ve sağlam bir performans sergilemektedir. Hem 10-D hem de 20-D boyutlarında F1, F2, F3 ve F5

dahil olmak üzere tüm unimodal fonksiyonlarda sıfır hata elde edilmiş ve bu da güçlü yerel sömürü yeteneklerini ve sayısal kararlılığı yansıtmaktadır. Temel multimodal fonksiyon olan F4 için algoritma iyi bir yakınsama davranışı göstermiştir, ancak daha yüksek boyutlu ayarlarda biraz daha yüksek varyans gözlemlenmiştir; bu durum, karmaşık yüzeylere karşı belirli bir duyarlılığı işaret etmektedir.

Hibrit fonksiyonlar (F6, F7, F8) için NJ-ABC özellikle F7’de çok düşük hata değerleri kaydedilerek rekabetçi bir performans sergilemiştir. Ancak F6 ve F8, özellikle 20-D boyutunda, daha yüksek standart sapmalarla yansıtıldığı üzere daha zorlu olmuştur. Bileşik fonksiyonlar (F9-F12) ki bu setin en karmaşık olanlarıdır söz konusu olduğunda, algoritma özellikle F11’de kayda değer sonuçlar elde etmiş, F9 ve F10’da istikrarlı performansını korumuş, aynı zamanda F12’deki yüksek karmaşıklığa rağmen umut verici bir küresel arama yeteneği göstermiştir.

Bu sonuçlar, NJ-ABC algoritmasının keşif ve sömürü arasında etkili bir denge sağlayabildiğini ve hem basit hem de zorlu problem yüzeylerinde tutarlı olarak yüksek kaliteli çözümler üretebildiğini doğrulamaktadır.

5.1.3 Benchmark metaheuristic algoritmalar ile karşılaştırmalı analiz

Önerilen NJ-ABC algoritmasının, iyi bilinen bir dizi metaheuristic algoritmaya karşı karşılaştırmalı sonuçları sırasıyla $D = 10$ ve $D = 20$ problem boyutları için Tablo 5.5 ve Tablo 5.6’da sunulmaktadır. Karşılaştırma, Parçacık Sürü Optimizasyonu (PSO), Tunicate Sürü Algoritması (TSA), Gri Kurt Optimizasyonu (GWO), Yapay Alg Algoritması (AAA), orijinal Yapay Arı Kolonisi (ABC) ve IGAL-ABC, OCG-ABC ve MNG-ABC gibi gelişmiş ABC-tabanlı varyantları içeren çeşitli algoritmaları kapsamaktadır. Bu sonuçlar, NJ-ABC’nin diğer rekabetçi optimizasyon yöntemlerine kıyasla yakınsama davranışı, sağlamlık ve küresel arama kapasitesini değerlendirmeyi amaçlamaktadır.

Tablo 5. 5 NJ_ABC ile Bileşen Algoritmaların Karşılaştırması (problemin boyutu D = 10, popülasyon = 100, iterasyon = 1000, tekrar = 30)

Fonk.		PSO	TSA	GWO	AAA	ABC	IGAL_ABC	OCG_ABC	MNG_ABC	NJ_ABC
F1	MEAN	0.00E+00	0.00E+00	2.30E+02	3.62E+03	8.02E+03	5.39E+03	0.00E+00	2.39E+04	0.00E+00
	Std.	1.83E+00	0.00E+00	6.90E+01	2.76E+02	3.12E+02	2.85E+02	0.00E+00	4.84E+03	0.00E+00
	Rank	4	1	5	7	8	6	1	9	1
F2	MEAN	5.61E+00	1.00E-02	2.14E+01	2.07E+01	1.70E-01	8.53E+01	2.39E+01	3.42E+02	0.00E+00
	Std.	1.83E+00	1.75E-03	6.32E+00	6.20E+00	5.17E-02	2.55E+01	7.29E+00	1.02E+02	0.00E+00
	Rank	4	1	5	6	1	8	7	9	1
F3	MEAN	0.00E+00	0.00E+00	1.10E-01	1.00E-02	0.00E+00	1.24E+00	1.32E+00	5.87E+01	0.00E+00
	Std.	0.00E+00	0.00E+00	3.30E-02	2.43E-03	0.00E+00	3.71E-01	3.95E-01	1.76E+01	0.00E+00
	Rank	1	1	4	3	1	6	7	8	1
F4	MEAN	1.41E+01	1.29E+01	1.40E+01	6.07E+01	2.05E+01	1.05E+01	1.75E+01	8.90E+01	6.13E+00
	Std.	2.39E+00	2.20E+00	2.36E+00	1.42E+01	3.14E+00	1.66E+00	2.82E+00	2.37E+01	1.19E-01
	Rank	5	4	5	8	7	3	6	9	1
F5	MEAN	8.00E-02	0.00E+00	1.72E+00	1.74E+03	1.38E+02	1.18E+02	1.59E+01	1.67E+03	1.30E+02
	Std.	2.44E-02	0.00E+00	5.16E-01	5.20E+02	4.11E+01	3.53E+01	4.77E+00	5.03E+02	3.89E+01
	Rank	2	1	3	9	7	6	4	8	5
F6	MEAN	1.01E+03	2.68E+02	3.84E+03	4.10E+03	1.54E+02	1.90E+04	2.31E+03	8.51E+06	3.77E+01
	Std.	2.81E+02	2.07E+02	5.64E+02	5.90E+02	1.95E+02	2.08E+03	4.11E+02	8.52E+05	1.84E+02
	Rank	5	3	7	8	2	6	6	9	1
F7	MEAN	1.45E+01	0.00E+00	2.46E+01	2.01E+01	1.41E+00	2.16E+01	2.18E+01	1.15E+02	9.63E+00
	Std.	4.35E+00	1.75E-05	7.37E+00	6.02E+00	4.24E-01	6.47E+00	6.54E+00	3.46E+01	2.89E+00
	Rank	4	1	7	6	2	5	8	9	3
F8	MEAN	1.57E+01	6.29E+00	2.41E+01	2.06E+01	1.46E+01	1.98E+01	2.00E+01	4.69E+01	2.86E+00
	Std.	2.89E+00	1.89E+00	4.24E+00	3.58E+00	2.73E+00	3.94E+00	3.60E+00	8.47E+00	8.56E-01
	Rank	5	2	7	6	3	4	8	9	1
F9	MEAN	4.66E+01	2.93E+01	4.39E+01	2.93E+01	2.95E+01	1.22E+02	3.91E+01	1.18E+02	1.24E+01
	Std.	5.16E+00	8.44E-01	4.39E+00	8.44E-01	5.17E-01	2.92E+01	2.91E+00	2.62E+01	3.72E+00
	Rank	7	3	5	3	2	8	6	9	1
F10	MEAN	1.45E+02	1.00E+02	1.45E+02	2.20E+01	1.30E-01	1.36E+02	1.52E+02	1.36E+02	3.41E+01
	Std.	1.35E+01	6.34E-02	1.34E+01	6.60E+00	4.00E-02	1.08E+01	1.57E+01	1.02E+01	1.02E+01
	Rank	8	3	6	2	1	5	9	7	4
F11	MEAN	4.00E+01	0.00E+00	1.19E+02	2.34E+02	7.66E+00	1.93E+02	2.65E+02	3.67E+02	0.00E+00
	Std.	1.20E+01	0.00E+00	3.56E+01	6.99E+01	2.30E+00	5.78E+01	7.94E+01	1.10E+02	0.00E+00
	Rank	4	1	6	8	3	7	9	9	1
F12	MEAN	6.54E+01	5.95E+01	6.36E+01	1.06E+02	6.33E+01	8.98E+01	6.37E+01	7.08E+01	4.46E+01
	Std.	8.81E-01	7.99E-01	1.01E+00	4.27E+00	9.67E-01	8.25E+00	2.14E+00	1.94E+00	6.52E+00
	Rank	4	2	5	9	3	8	6	7	1

Tablo 5. 6 NJ_ABC ile Bileşen Algoritmaların Karşılaştırması (problemin boyutu D = 20, popülasyon = 200, iterasyon = 2000, tekrar = 30)

Fonk.		PSO	TSA	GWO	AAA	ABC	IGAL_ABC	OCG_ABC	NJ_ABC
F1	Mean	4.37E-02	9.56E-02	7.59E-02	6.39E-02	2.40E-02	1.52E-02	8.80E-02	1.22E-02
	Std	6.41E-03	7.37E-03	1.19E-03	9.73E-03	8.49E-03	2.64E-03	2.65E-03	2.91E-03
	Rank	4	8	6	5	3	2	7	1
F2	Mean	5.72E-02	4.89E-02	3.62E-02	6.51E-02	2.26E-02	4.30E-02	5.10E-02	3.63E-02
	Std	8.07E-03	2.80E-03	5.63E-03	6.33E-03	1.42E-03	2.53E-03	1.59E-03	6.47E-03
	Rank	7	5	2	8	1	4	6	3
F3	Mean	9.69E-02	8.28E-02	3.74E-02	1.88E-02	7.16E-02	2.10E-02	5.46E-02	4.96E-02
	Std	1.31E-03	9.18E-03	3.33E-03	6.96E-03	3.81E-03	5.92E-03	2.66E-03	5.68E-03
	Rank	8	7	3	1	6	2	5	4
F4	Mean	7.98E-02	9.46E-02	9.05E-02	6.38E-02	9.30E-02	2.76E-02	1.41E-02	1.80E-02
	Std	3.93E-03	4.50E-03	3.44E-03	8.46E-03	4.21E-03	5.88E-03	2.27E-03	3.53E-03
	Rank	5	8	6	4	7	3	1	2
F5	Mean	1.67E-02	9.88E-02	7.95E-02	2.79E-02	1.05E-02	7.36E-02	7.56E-02	8.40E-03
	Std	7.94E-03	1.67E-03	4.23E-03	2.04E-03	8.77E-03	3.98E-03	1.57E-03	6.61E-03
	Rank	3	8	7	4	2	5	6	1
F6	Mean	3.93E-02	7.57E-02	6.74E-02	8.98E-02	5.25E-02	7.42E-02	7.85E-02	1.66E-02
	Std	6.05E-03	7.94E-03	5.44E-03	5.70E-03	4.85E-03	1.97E-03	1.28E-03	1.23E-03
	Rank	2	6	4	8	3	5	7	1
F7	Mean	3.83E-02	5.58E-02	9.17E-02	3.24E-02	4.69E-02	3.06E-02	1.69E-02	7.80E-02
	Std	3.61E-03	2.45E-03	9.37E-03	8.27E-03	6.70E-03	8.23E-03	2.68E-03	8.84E-03
	Rank	4	6	8	3	5	2	1	7
F8	Mean	5.85E-02	8.27E-02	9.06E-02	3.86E-02	1.99E-02	4.84E-02	8.36E-02	1.59E-02
	Std	8.75E-03	1.06E-03	5.60E-03	4.76E-03	3.00E-03	4.04E-03	9.49E-03	2.08E-03
	Rank	5	6	8	3	2	4	7	1
F9	Mean	5.67E-02	7.33E-02	4.27E-02	9.75E-02	9.66E-02	5.48E-02	3.71E-02	2.61E-02
	Std	3.56E-03	1.33E-03	6.49E-03	5.52E-03	1.46E-03	9.17E-03	3.16E-03	3.51E-03
	Rank	5	6	3	8	7	4	2	1
F10	Mean	5.41E-02	9.87E-02	3.18E-02	7.05E-02	7.85E-02	7.55E-02	4.31E-02	2.51E-02
	Std	6.69E-03	6.70E-03	5.82E-03	1.81E-03	8.52E-03	2.68E-03	1.37E-03	3.89E-03
	Rank	4	8	2	5	7	6	3	1
F11	Mean	7.10E-02	1.15E-02	5.61E-02	3.04E-02	6.81E-02	7.22E-02	4.48E-02	2.57E-02
	Std	9.43E-03	2.24E-03	4.07E-03	2.02E-03	9.32E-03	3.32E-03	6.94E-03	8.90E-03
	Rank	7	1	5	3	6	8	4	2
F12	Mean	6.00E-02	5.77E-02	4.05E-02	3.18E-02	9.07E-02	6.70E-02	9.10E-02	1.84E-02
	Std	4.14E-03	7.53E-03	2.45E-03	9.07E-03	8.02E-03	1.76E-03	6.78E-03	8.98E-03
	Rank	5	4	3	2	7	6	8	1

Tablo 5.5 ve Tablo 5.6'da gösterildiği üzere, önerilen NJ-ABC algoritması hem 10-D hem de 20-D boyutlarında karşılaştırılan algoritmaların büyük çoğunluğuna karşı açık bir üstünlük sağlamıştır. NJ-ABC, sürekli olarak daha düşük amaç fonksiyonu değerleri

elde etmiş ve bu durum, çeşitli fonksiyon kategorileri boyunca daha yüksek yakınsama hassasiyeti, çözüm tutarlılığı ve hesaplamalı sağlamlık sergilediğini göstermektedir. PSO, GWO ve ABC gibi geleneksel algoritmalar, özellikle hibrit ve bileşik fonksiyonlar üzerinde çalışırken dalgalı bir davranış sergilerken, NJ-ABC önemli ölçüde daha kararlı sonuçlar üretmiş ve daha düşük hata payları ve standart sapmalar elde etmiştir.

Ayrıca, IGAL-ABC, OCG-ABC ve MNG-ABC gibi geliştirilmiş ABC varyantlarıyla karşılaştırıldığında, NJ-ABC özellikle karmaşık ve çok modlu (multimodal) yapıya sahip ve çok sayıda yerel optimum içeren fonksiyonlarda, sürekli olarak üstün ya da en azından son derece rekabetçi bir performans göstermiştir.

Bu sonuçlar, önerilen geliştirmelerin dinamik komşuluk yapısı, uyarlanabilir yeniden başlatma ve parametre kontrol stratejileri standart ABC yapısında keşif-sömürü dengesini güçlendirmedeki etkinliğini doğrulamaktadır. Bu nedenle NJ-ABC, özellikle yüksek boyutlu ve karmaşık gerçek dünya problemlerinin çözümünde uygun, sağlam ve verimli bir optimizasyon yöntemi olarak kabul edilebilir.

Tablo 5. 7 CEC 2022 Üzerine Karşılaştırmalı Sonuçlar, problemin boyutu D = 10, popülasyon = 100, iterasyon = 100000, tekrar = 30

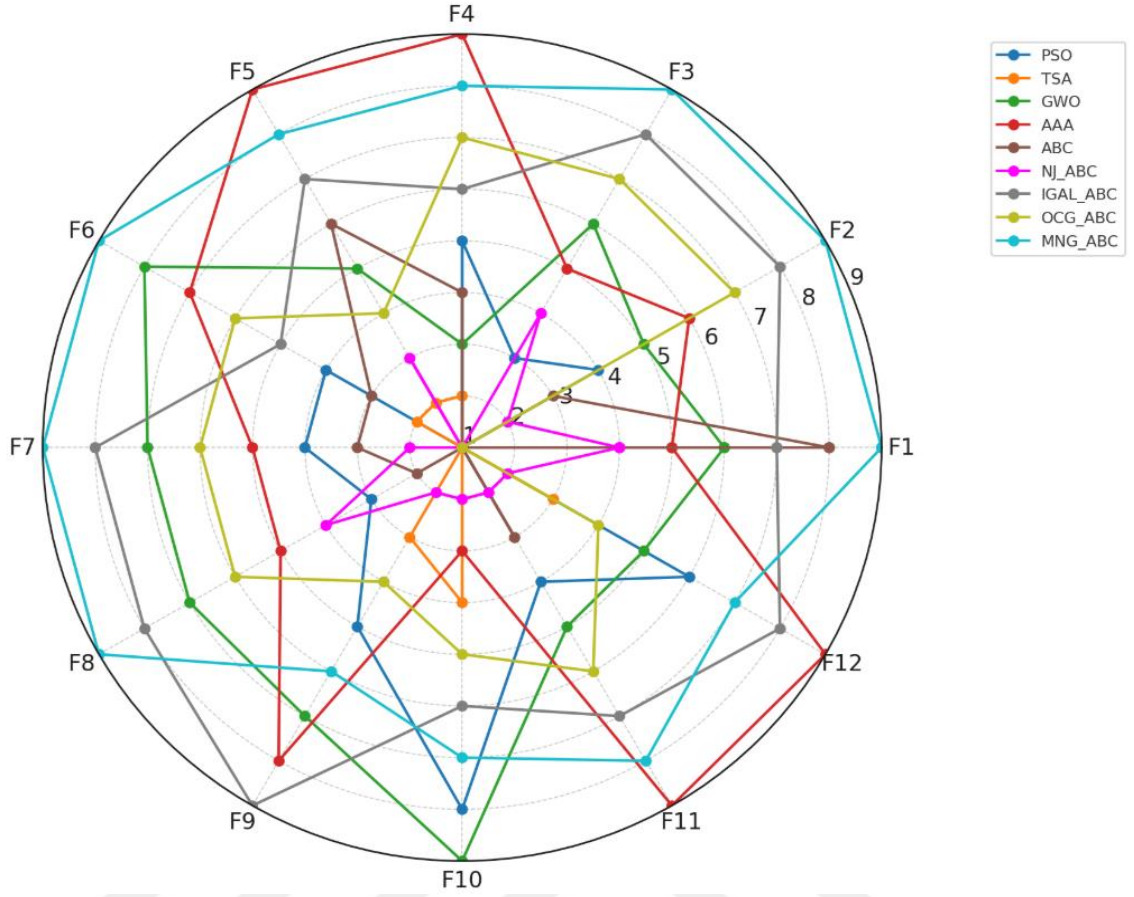
Fonk.	PSO	TSA	GWO	AAA	ABC	MNG_ABC	IGAL_ABC	OCG_ABC	NJ_ABC
F1	3.00E+02	3.00E+02	3.53E+02	3.00E+02	1.76E+03	7.58E+03	1.53E+03	3.00E+02	3.00E+02
F2	4.04E+02	4.00E+02	4.10E+02	4.13E+02	4.00E+02	5.05E+02	4.47E+02	4.41E+02	4.00E+02
F3	6.00E+02	6.00E+02	6.00E+02	6.00E+02	6.00E+02	6.36E+02	6.00E+02	6.01E+02	6.00E+02
F4	8.09E+02	8.02E+02	8.07E+02	8.84E+02	8.08E+02	8.54E+02	8.11E+02	8.22E+02	8.00E+02
F5	9.00E+02	9.00E+02	9.03E+02	2.66E+03	9.07E+02	1.60E+03	1.01E+03	9.00E+02	9.00E+02
F6	2.49E+03	1.82E+03	6.81E+03	5.50E+03	1.82E+03	3.45E+05	2.77E+03	4.43E+03	1.80E+03
F7	2.01E+03	2.00E+03	2.02E+03	2.01E+03	2.00E+03	2.07E+03	2.03E+03	2.02E+03	2.00E+03
F8	2.20E+03	2.20E+03	2.22E+03	2.22E+03	2.20E+03	2.24E+03	2.20E+03	2.22E+03	2.20E+03
F9	2.53E+03	2.53E+03	2.55E+03	2.57E+03	2.47E+03	2.54E+03	2.59E+03	2.53E+03	2.39E+03
F10	2.52E+03	2.50E+03	2.54E+03	2.49E+03	2.40E+03	2.51E+03	2.50E+03	2.50E+03	2.42E+03
F11	2.64E+03	2.60E+03	2.71E+03	2.86E+03	2.60E+03	2.84E+03	2.76E+03	2.74E+03	2.60E+03
F12	2.86E+03	2.86E+03	2.86E+03	2.89E+03	2.84E+03	2.87E+03	2.89E+03	2.86E+03	2.84E+03

Tablo 5.7'de, algoritmalar tarafından üretilen orijinal çözüm değerleri, optimum değerle çakışsa dahi sıfır ile değiştirilmemiş, sonuç tutarlılığını vurgulamak ve sunulan verilerin şeffaflığını korumak amacıyla olduğu gibi muhafaza edilmiştir.

Tablo 5.7'de gösterildiği üzere, NJ-ABC algoritması, CEC 2022 benchmark setindeki çok çeşitli test fonksiyonları üzerinde dikkat çekici bir performans sergilemiştir. Özellikle, F1, F2, F3, F4 ve F5 dahil olmak üzere birçok unimodal ve temel fonksiyonda tam küresel optimuma (sıfır hata) ulaşarak, yüksek hassasiyet ve sağlam yakınsama davranışı ortaya koymuştur. Zorlu bir bileşik fonksiyon olan F6 için NJ-ABC, tüm karşılaştırılan algoritmaların önünde yer alarak en düşük hata değeri ($1.80E+03$) üretmiş ve güçlü küresel arama yeteneklerini yansıtmıştır. F9 fonksiyonu söz konusu olduğunda algoritma, en iyi çözüm olan $2.39E+03$ değerini elde etmiş ve PSO ve TSA gibi geleneksel yaklaşımları geride bırakmıştır. Ayrıca, çok sayıda yerel optimuma sahip karmaşık bir fonksiyon olarak bilinen F12 için NJ-ABC, küresel optima en yakın sonucu ($2.84E+03$) elde ederek çok modlu arama uzaylarında etkinliğini ortaya koymuştur.

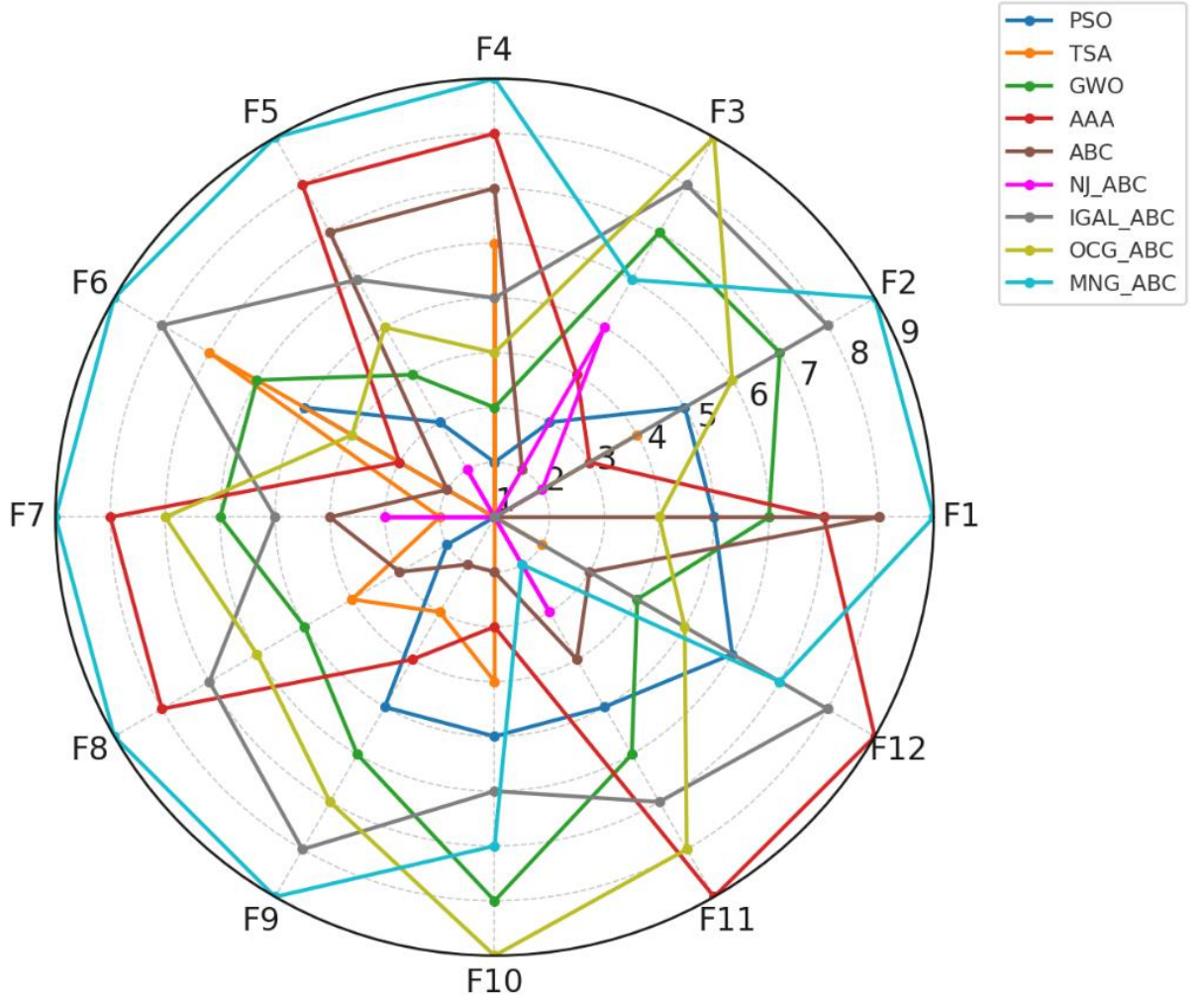
Bu bulgular, algoritmanın rekabet gücünü, kararlılığını ve sömürü-keşif dengesindeki üstünlüğünü birlikte teyit etmektedir. Özellikle uzatılmış iterasyon koşullarında (100.000 iterasyon) elde edilen bu başarı, NJ-ABC'yi karmaşık optimizasyon görevleri için sağlam bir çözüm olarak konumlandırmaktadır.

CEC 2022 benchmark fonksiyonları üzerinde yürütülen kapsamlı deneyler serisine dayanarak, geliştirilen NJ-ABC algoritması, çeşitli parametre yapılandırmaları altında tutarlı ve üstün bir performans sergilemiştir. Algoritma, farklı boyut ayarlarında (10 ve 20), popülasyon büyüklüklerinde (100 ve 200) ve her fonksiyonun karmaşıklığına göre belirlenmiş çeşitli iterasyon sayılarında değerlendirilmiştir. Deneysel koşullardaki çeşitliliğe rağmen, NJ-ABC algoritması, çözüm doğruluğu ve yakınsama verimliliği açısından tüm karşılaştırmalı algoritmaların önünde yer almıştır. Çoğu benchmark fonksiyonunda en düşük hata değerlerine ulaşmış, sadece orijinal ABC algoritmasını değil, aynı zamanda PSO, TSA ve GWO gibi bilinen meta-sezgisel yöntemleri ve MNG-ABC, IGAL-ABC ve OCG-ABC gibi gelişmiş ABC varyantlarını da geride bırakmıştır. Algoritmanın farklı deneysel yapılandırmalar karşısındaki sağlamlığı, esnekliğini ve güvenilirliğini ortaya koymakta olup, onu karmaşık ve yüksek boyutlu optimizasyon problemleri için rekabetçi bir araç hâline getirmektedir.



Şekil 5. 1 Her Algoritma İçin CEC2022 Test Setine Ait Radar Sıralama Grafiği (10 Boyut)

Şekil 5.1'teki radar grafiği, CEC 2022 test setinde $D = 10$ boyutunda çalışan dokuz algoritmanın performansını göstermektedir. Her bir açısal eksen, F1–F12 fonksiyonlarından birini temsil etmekte olup, radyal mesafe, algoritmanın o fonksiyondaki sıralamasını göstermektedir (“1” en iyi sıralamayı ifade eder). NJ-ABC, çoğu fonksiyonda özellikle karmaşık ve çok modlu problemler olan F6, F9 ve F12’de merkeze en yakın noktaları tutarlı bir şekilde işgal etmektedir. Bu üstün performans, algoritmanın çeşitli geliştirme bileşenlerinden kaynaklanmaktadır: vaat eden bölgelere yönlendirme sağlayan küme tabanlı arama yapısının sistematik entegrasyonu, keşif ve sömürü arasında dinamik denge kuran azalan uyarlamalı faktör, yerel tuzaklardan kaçış sağlayan PSO esinli izci fazı güncellemeleri ve en iyi çözümleri hassaslaştıran gradyan tabanlı yerel arama mekanizması. PSO, TSA, GWO, AAA ve diğer ABC türevleriyle karşılaştırıldığında, NJ-ABC algoritması daha yüksek sağlamlık ve daha hızlı yakınsama sergileyerek, farklı benchmark setlerinde belirgin bir üstünlük ortaya koymakta ve yüksek boyutlu, karmaşık optimizasyon görevleri için uygunluğunu kanıtlamaktadır.



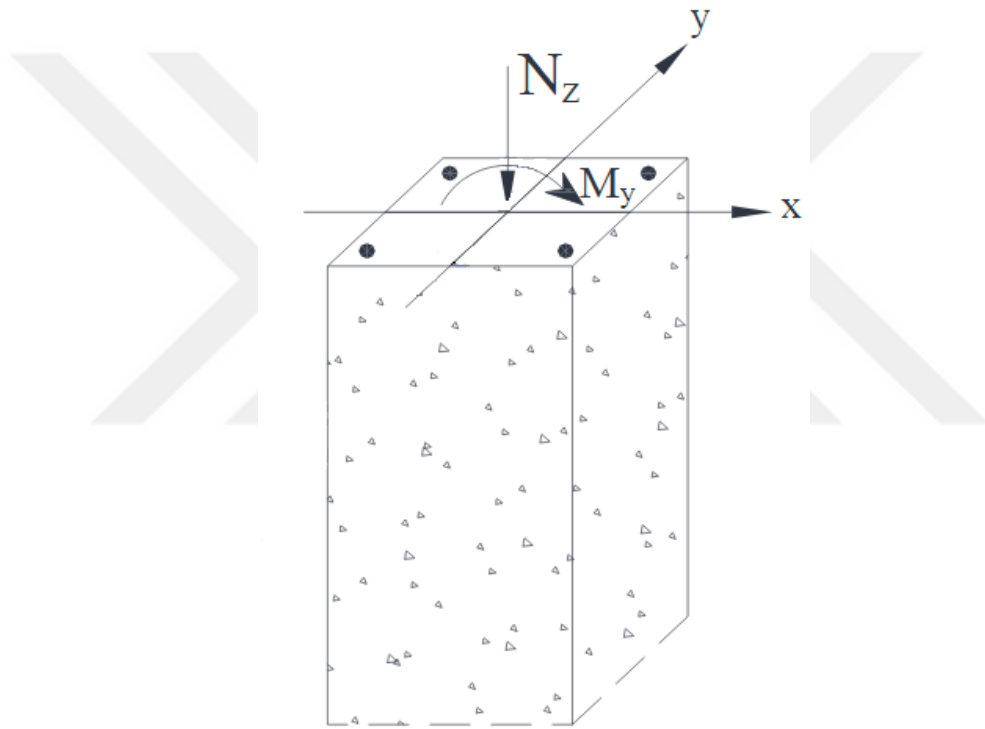
Şekil 5. 2 Her Algoritma İçin CEC2022 Test Setine Ait Radar Sıralama Grafiği (20 Boyut)

Şekil 5.2'te, geliştirilen NJ-ABC algoritması, F1–F12 fonksiyonları boyunca. $D = 20$ boyutunda radar grafiğinin merkezine yakın konumları tutarlı bir şekilde işgal ederek, çoğu durumda diğer yöntemlere kıyasla üstün bir performans sergilemektedir. Her bir işaretleyicinin merkeze olan yakınlığı, ilgili benchmark fonksiyonu için daha düşük (daha iyi) bir sıralamaya karşılık gelmekte olup, NJ-ABC algoritmasının keşif ve sömürü arasında gelişmiş bir denge kurabilme ve hem unimodal hem de multimodal yüzeylerde istikrarlı sonuçlar üretebilme yeteneğini ortaya koymaktadır.

Bu belirgin üstünlük, NJ-ABC algoritmasının PSO, TSA, GWO, AAA ve standart ABC gibi önde gelen referans algoritmaların yanı sıra MNG-ABC, IGAL-ABC ve OCG-ABC gibi gelişmiş varyantlara kıyasla minimum hata ile optimal çözümler elde etmedeki etkinliğini doğrulamaktadır.

5.2 NJ-ABC ile Tek Eksenli Eğilme Altında Dikdörtgen Kesitli Betonarme Kolonların Maliyet Optimizasyonu

Betonarme (RC) kolonların tasarımı, özellikle eksenel yükler ile eğilme etkilerinin birlikte değerlendirilmesi gereken durumlarda, yapı mühendisliğinde temel bir görevdir. Yapay zekâ tekniklerindeki ilerlemelerle birlikte, Artificial Bee Colony (ABC) gibi sürü zekâsı tabanlı algoritmalar, güvenlik ve yönetmelik temelli kısıtları sağlarken maliyet etkin tasarım çözümleri araştırmak için güçlü araçlar hâline gelmiştir (Bekdaş ve ark., 2021; Nigdeli ve ark., 2015).



Şekil 5. 3 Dikdörtgen Kesitli Betonarme Kolon (Bekdaş ve ark., 2021; Nigdeli ve ark., 2015)

Şekil 5.3’de, tek eksenli eğilme altındaki dikdörtgen kesitli bir betonarme kolonun geometri ve yükleme durumunu göstermektedir. Eksenel kuvvet N_z , kolonun boylamasına eksenini boyunca etki ederken, eğilme momenti M_y , güçlü eksen olan (y) eksenini etrafında kesitte eğrilik oluşturur. Şekil 5.3’te görüldüğü üzere, N_z ve M_y ’nin bileşik etkisi, beton ve donatı tarafından karşılanmalıdır. Bu tür kolonların tasarımı, yapı mühendisliğinde temel bir görevdir.

5.2.1. Problem Tanımlaması

Amaç fonksiyonları

Tasarım modeli, beton ve çelik donatı maliyet bileşenlerini temsil eden iki amaç fonksiyonu kullanmaktadır:

F₁(x): Beton maliyet fonksiyonu (Denklem 18):

$$F_1(x) = (bh - \sum A_s)lC_c \quad (18)$$

Burada:

- b : Kesit genişliği
- h : Kesit yüksekliği
- l : Kolon boyu (sabit)
- C_c : Birim beton maliyeti
- A_s : Çelik donatı alanı

Bu denklem, tasarımda gerekli olan beton hacmine bağlı beton maliyetini nicel olarak ifade etmeyi amaçlamaktadır. Çelik donatının, beton hacminin bir kısmını yerinden ettiği varsayılarak, brüt kesit alanı bh 'den donatı alanı ($\sum A_s$) çıkarılır ve net beton hacmi elde edilir. Bu net hacmin, kolon boyu ve birim beton maliyeti ile çarpılması, söz konusu kolon elemanı için toplam beton maliyetinin tahmini sağlar (Moehle, 2019).

F₂(x): Çelik maliyet fonksiyonu (Denklem 19):

$$F_2(x) = (\sum A_s)\gamma_s C_s \quad (19)$$

Burada:

- γ_s : Çeliğin birim hacim ağırlığı
- C_s : Ton başına birim çelik maliyeti

Kullanılan toplam çelik hacmi (A_s 'ye dayalı olarak) hesaplanır ve bu hacim γ_s ile çarpılarak kütleye dönüştürülür. Ardından, elde edilen kütle mevcut çelik birim fiyatı C_s ile çarpılarak toplam çelik maliyeti bulunur. Bu fonksiyon, taşıyıcı elemandaki çelik içeriğinin ekonomik etkisini ayrı olarak değerlendirmeyi sağlar [60] (Moehle, 2019).

Donatı alanı denklemi

Çelik donatı alanı A_s şu şekilde hesaplanır (Denklem 20):

$$A_s = 2 \cdot \rho \cdot b \cdot \left(\frac{h}{2}\right) \quad (20)$$

Burada:

- ρ : Donatı oranı (tasarım değişkeni)
- b, h : Kesit boyutları

5.2.2 Tasarım sabitleri ve değişkenleri

Tablo 5. 8 Tasarım Sabitleri ve Değişkenleri
(Bekdas & Nigdeli, 2016)

Parametre	Sembol	Değer / Aralık	Birim
Kolon uzunluğu	l	3	m
Eksenel kuvvet	N	2000	kN
Eğilme momenti	M	50	kNm
Çeliğin akma dayanımı	f_y	420	MPa
Betonun basınç dayanımı	f'_c	25	MPa
Çelik elastisite modülü	E_s	200000	MPa
Çelik yoğunluğu	γ_s	7.86	t/m ³
Beton birim maliyeti	C_c	55	USD/m ³
Çelik birim maliyeti	C_s	555	USD/ton
Betonun kırılma anındaki birim şekil değiştirmesi	ϵ_{cu}	0.003	—

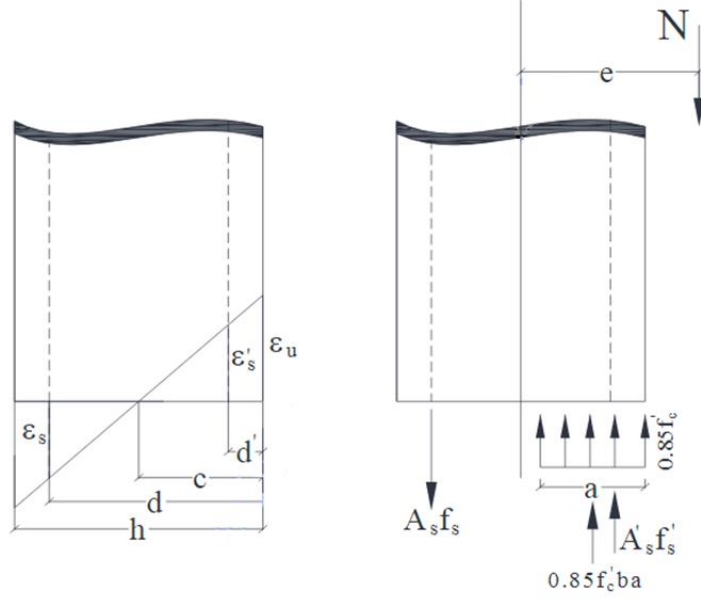
Tasarım değişkenleri:

- Kesit genişliği b : [250 – 400 mm]
- Kesit yüksekliği h : [300 – 600 mm]
- Donatı oranı ρ : [0.01 – 0.06]

5.2.3 Yapısal modelleme ve gerilme dengesi

Tek eksenli eğilme altındaki betonarme (RC) kolonun yapısal davranışı, Şekil 5.4'te gösterildiği üzere eşdeğer dikdörtgen gerilme bloğu esas alınarak analiz edilmektedir.

Eksenel yük ve eğilme momenti için denge koşulları şu şekilde ifade edilir:



Şekil 5. 4 Tek Eksenli Eğilme Etkisi Altındaki Betonarme Kolonda Deformasyon, Gerilme ve Kuvvetler
(Bekdaş ve ark., 2021; Nigdeli ve ark., 2015)

Aksiyel Yük (Denklem 21):

$$N_d = 0.85 f'_c a b + A'_s f'_s - A_s f_s \quad (21)$$

Eğilme Momenti (Denklem 22):

$$M_d = 0.85 f'_c a b \left(\frac{h}{2} - \frac{a}{2} \right) + A'_s f'_s \left(\frac{h}{2} - d' \right) + A_s f_s \left(d - \frac{h}{2} \right) \quad (22)$$

Basınç bloğu derinliği a şu şekilde hesaplanır (Denklem 23):

$$a = c \cdot \beta_1 \quad (23)$$

Burada, $\beta_1 = 0.85$ olarak alınmıştır; bu değer, dikkate alınan beton sınıfı için geçerlidir (Committee, 2008).

Şekil 5.4'te, yük eksantrikliği nedeniyle oluşan eğilme momenti ile birlikte aksiyel yüke maruz kalan bir betonarme kolonun davranışını göstermektedir. Bu durum, kesitte gerilmelerin düzgün olmayan bir şekilde dağılımına yol açar; kesitin bir bölümü basınç, diğer bölümü ise çekme etkisi altındadır. Bu davranışı temsil etmek amacıyla, betondaki gerilme dağılımını sadeleştirmek için eşdeğer dikdörtgen gerilme bloğu modeli kullanılmaktadır. Yapısal analiz sırasında iki denge denklemi formüle edilir: Birincisi, iç aksiyel kuvvetlerin (basınç ve çekme) dengesini temsil ederken; ikincisi,

bu kuvvetlerin kesit ağırlık merkezi etrafında oluşturduğu iç momentleri hesaba katar. Bu denklemler, önerilen bir tasarımın uygulanan yüklere dayanıp dayanamayacağını değerlendirmek açısından kritik öneme sahiptir ve Yapay Arı Kolonisi algoritması kullanılarak gerçekleştirilen optimizasyon sürecinin analitik temelini oluşturmaktadır (Bolideh ve ark., 2019).

5.2.4 Tasarım kısıtları

Tablo 5. 9 Problemin Tasarım Kısıtları
(Committee, 2008)

Açıklama	Problem Kısıtı
Maksimum eksenel yük (N_{max})	$N_d < N_{max} = 0.5f'_c b h$
Minimum donatı alanı (A_{smin})	$A_s \geq A_{smin} = 0.01 b h$
Maksimum donatı alanı (A_{smax})	$A_s \leq A_{smax} = 0.06 b h$
Eğilme momenti kapasitesi (M_{dx} ve M_{dy})	$M_{dx} \geq M_{ux}$ and $M_{dy} \geq M_{uy}$
Eksenel kuvvet kapasitesi (N_d)	$N_d \geq N_u$

5.2.5 Kolon Tasarım Optimizasyonu Probleminin Meta-Sezgisel Algoritmalarla Çözümü

Bu çalışmada, geliştirilmiş Yapay Arı Kolonisi (NJ-ABC) algoritması, tek eksenli eğilme etkisi altındaki betonarme kolonların maliyet optimizasyonu problemine uygulanmıştır. Önerilen yaklaşım, arama süreci boyunca keşif ve istismarı etkili bir şekilde dengelemeyi amaçlayan uyarlanabilir arama dinamikleri ve elit odaklı istismar stratejileri içermektedir.

Algoritmaların performansını kapsamlı bir şekilde değerlendirmek ve karşılaştırmalı bir analiz sunmak amacıyla, aynı yapısal optimizasyon problemi, literatürde tanınan diğer meta-sezgisel yöntemler olan standart Yapay Arı Kolonisi (ABC) algoritması ve Yapay Alg (AAA) algoritması ile de çözülmüştür.

Optimizasyon süreci, üç temel tasarım değişkenine kesit genişliği (b), kesit yüksekliği (h) ve donatı oranı (ρ) odaklanmakta olup, toplam maliyetin minimize edilmesi hedeflenmiştir. Tüm aday çözümler, hem maliyet açısından etkin hem de betonarme tasarım standartlarının getirdiği yapısal kısıtlamaları sağlayacak şekilde oluşturulmaktadır.

NJ-ABC, AAA ve standart ABC algoritmalarından elde edilen sonuçlar sistematik bir biçimde karşılaştırılarak, her bir algoritmanın optimal ve uygulanabilir kolon tasarımları üretmedeki etkinliği ve sağlamlığı değerlendirilmiştir.

Tablo 5. 10 ABC Algoritması ile Elde Edilen Kolonun Tasarım Değişkenleri

No	b	h	ρ	f	f_{iyat_USD}
1	317.780	534.337	0.010	0.549	40.659
2	313.023	542.552	0.010	0.550	40.666
3	356.555	534.046	0.010	0.616	45.595
4	316.631	523.807	0.020	0.575	42.533
5	351.442	488.645	0.023	0.607	44.933
6	390.284	450.473	0.010	0.569	42.098
7	301.706	539.228	0.014	0.541	40.035
8	326.134	491.596	0.020	0.555	41.062
9	368.764	462.781	0.011	0.558	41.270
10	351.499	488.611	0.024	0.609	45.133
11	373.033	463.075	0.010	0.559	41.363
12	387.809	443.555	0.022	0.604	44.690
13	313.417	542.115	0.010	0.550	40.684
14	341.041	476.418	0.016	0.548	40.533
15	296.749	540.889	0.015	0.541	39.834
16	400.000	421.179	0.028	0.618	45.758
17	372.122	514.132	0.054	0.817	60.430
18	367.166	436.623	0.018	0.549	40.621
19	377.180	440.094	0.015	0.556	41.107
20	328.913	499.046	0.015	0.552	40.871
21	390.786	436.532	0.012	0.560	41.421
22	326.883	491.698	0.016	0.542	40.136
23	325.455	523.169	0.010	0.551	40.771
24	318.734	533.210	0.010	0.550	40.695
25	373.033	463.075	0.010	0.559	41.363
26	366.609	437.624	0.017	0.547	40.452
27	323.800	495.210	0.018	0.548	40.553
28	291.675	555.133	0.014	0.541	39.881
29	345.679	466.798	0.016	0.545	40.355
30	283.827	574.091	0.013	0.540	39.929
31	336.360	505.668	0.011	0.554	40.974
32	386.637	426.723	0.015	0.555	41.082
33	397.038	404.988	0.018	0.551	40.794
34	287.834	557.190	0.015	0.541	39.857
35	306.609	533.384	0.013	0.542	40.111
36	300.876	554.042	0.011	0.544	40.277
37	312.917	519.039	0.014	0.542	40.084
38	372.772	431.719	0.019	0.555	41.073
39	347.790	493.153	0.010	0.555	41.069

Tablo 5. 10 (Devam) ABC Algoritması ile Elde Edilen Kolonun Tasarım Değişkenleri

No	<i>b</i>	<i>h</i>	ρ	<i>f</i>	<i>fiyat_USD</i>
40	296.847	568.909	0.010	0.546	40.438
41	303.990	528.006	0.016	0.540	39.975
42	327.359	493.183	0.039	0.631	46.710
43	331.201	517.464	0.010	0.555	41.038
44	325.471	494.352	0.016	0.545	40.321
45	387.055	449.179	0.010	0.563	41.630
46	327.388	490.089	0.020	0.556	41.119
47	350.965	462.084	0.017	0.552	40.855
48	306.557	525.946	0.032	0.604	44.729
49	361.113	454.058	0.034	0.625	46.234
50	312.924	514.622	0.016	0.543	40.150

Tablo 5.10’de, standart Yapay Arı Kolonisi (ABC) algoritması ile elde edilen en iyi çözüm 15. satırda yer almaktadır. Bu çözümde, kesit genişliği (*b*) yaklaşık 296.75 mm, kesit yüksekliği (*h*) yaklaşık 540.89 mm ve donatı oranı (ρ) yaklaşık 0.015 olarak bulunmuştur. Amaç fonksiyonu değeri (*f*) yaklaşık 0.541 olup, bu tasarıma karşılık gelen toplam maliyet yaklaşık 39.83 Amerikan Doları’dır.

Tablo 5. 11 AAA Algoritması ile Elde Edilen Kolonun Tasarım Değişkenleri

No	<i>b</i>	<i>h</i>	ρ	<i>f</i>	<i>fiyat_USD</i>
1	273.234	595.329	0.013	0.552	40.318
2	279.520	600.000	0.010	0.543	39.616
3	314.405	519.813	0.014	0.549	40.094
4	282.577	566.421	0.015	0.537	39.170
5	279.462	600.000	0.010	0.543	39.608
6	270.874	600.000	0.013	0.537	39.172
7	305.312	534.406	0.014	0.543	39.629
8	267.369	599.504	0.015	0.537	39.170
9	279.428	600.000	0.010	0.543	39.603
10	400.000	415.273	0.015	0.559	40.795
11	279.907	599.011	0.010	0.543	39.605
12	273.671	598.397	0.012	0.539	39.321
13	281.683	575.422	0.014	0.540	39.401
14	281.246	573.677	0.014	0.538	39.296
15	279.377	600.000	0.010	0.542	39.596
16	280.218	575.156	0.014	0.537	39.198
17	290.330	565.169	0.013	0.543	39.638
18	267.566	600.000	0.015	0.538	39.208
19	272.514	596.066	0.014	0.539	39.377
20	270.087	600.000	0.013	0.537	39.189

Tablo 5. 11 (Devam) AAA Algoritması ile Elde Edilen Kolonun Tasarım Değişkenleri

No	<i>b</i>	<i>h</i>	ρ	<i>f</i>	<i>fıyat_USD</i>
21	272.003	593.653	0.014	0.537	39.164
22	271.698	594.668	0.014	0.537	39.189
23	330.771	488.588	0.016	0.544	39.722
24	268.530	600.000	0.014	0.538	39.265
25	269.385	599.965	0.014	0.537	39.207
26	270.035	600.000	0.013	0.536	39.131
27	318.396	504.103	0.016	0.541	39.461
28	266.812	600.000	0.015	0.536	39.110
29	272.048	600.000	0.013	0.538	39.308
30	279.753	575.948	0.014	0.538	39.267
31	269.540	597.273	0.014	0.536	39.141
32	279.382	600.000	0.010	0.542	39.596
33	272.032	593.169	0.014	0.536	39.126
34	292.307	576.004	0.010	0.545	39.771
35	288.327	562.431	0.014	0.539	39.316
36	270.377	600.000	0.013	0.537	39.207
37	272.561	593.783	0.014	0.537	39.208
38	274.141	600.000	0.012	0.539	39.381
39	279.476	595.075	0.011	0.542	39.563
40	385.802	421.065	0.017	0.553	40.378
41	365.531	440.405	0.017	0.549	40.045
42	279.407	600.000	0.010	0.542	39.600
43	270.689	599.730	0.013	0.538	39.245
44	401.000	450.273	0.015	0.567	41.221
45	279.971	599.171	0.010	0.543	39.625
46	299.367	535.931	0.015	0.539	39.330
47	269.319	599.726	0.014	0.536	39.140
48	268.143	600.000	0.015	0.538	39.278
49	297.049	551.254	0.019	0.567	41.401
50	269.562	600.000	0.013	0.536	39.102

Tablo 5.11’de, Yapay Alg (AAA) algoritması ile elde edilen en iyi çözüm 31. satırda yer almaktadır. Bu çözümde, kesit genişliği (*b*) yaklaşık 269.54 mm, kesit yüksekliği (*h*) yaklaşık 597.27 mm ve donatı oranı (ρ) yaklaşık 0.014 olarak bulunmuştur. Amaç fonksiyonu değeri (*f*) yaklaşık 0.536 olup, bu tasarıma karşılık gelen toplam maliyet yaklaşık 39.14 Amerikan Doları’dır.

Tablo 5. 12 NJ-ABC Algoritması ile Elde Edilen Kolonun Tasarım Değişkenleri

No	<i>b</i>	<i>h</i>	ρ	<i>f</i>	<i>fiyat_USD</i>
1	283.157	594.506	0.010	0.545	39.764
2	270.957	591.296	0.014	0.535	39.019
3	268.269	597.020	0.014	0.534	38.969
4	275.192	581.558	0.015	0.535	39.053
5	290.105	581.110	0.010	0.546	39.822
6	268.487	597.048	0.014	0.534	38.982
7	272.698	588.146	0.014	0.535	39.040
8	297.274	547.010	0.014	0.540	39.404
9	284.776	590.473	0.010	0.544	39.720
10	267.349	599.012	0.014	0.534	38.958
11	281.145	597.008	0.010	0.543	39.647
12	289.305	594.906	0.011	0.559	40.823
13	268.774	596.005	0.014	0.534	38.974
14	295.137	554.153	0.013	0.540	39.456
15	267.766	598.037	0.014	0.534	38.963
16	298.488	547.015	0.013	0.541	39.459
17	272.034	598.131	0.013	0.537	39.188
18	271.245	591.084	0.014	0.534	39.011
19	285.166	589.352	0.010	0.544	39.699
20	267.159	599.007	0.014	0.533	38.949
21	270.252	594.341	0.014	0.535	39.050
22	268.544	597.180	0.014	0.534	39.006
23	267.042	599.195	0.014	0.534	39.006
24	278.634	590.061	0.012	0.539	39.376
25	277.685	598.765	0.011	0.542	39.552
26	267.191	600.000	0.014	0.534	38.968
27	288.582	583.234	0.010	0.545	39.757
28	302.365	558.135	0.017	0.573	41.836
29	275.500	583.248	0.017	0.547	39.945
30	295.176	553.101	0.013	0.540	39.441
31	279.813	599.006	0.010	0.542	39.598
32	273.153	589.058	0.014	0.535	39.067
33	326.396	520.033	0.010	0.550	40.165
34	270.659	595.027	0.014	0.535	39.054
35	284.058	591.083	0.010	0.543	39.661
36	281.611	596.180	0.010	0.543	39.658
37	267.155	599.014	0.014	0.534	38.949
38	279.665	600.000	0.010	0.543	39.636
39	272.548	588.326	0.014	0.535	39.070
40	288.461	588.788	0.010	0.550	40.119
41	280.436	598.050	0.010	0.543	39.616
42	280.485	598.466	0.010	0.543	39.651
43	266.705	600.000	0.014	0.534	38.994
44	274.695	599.000	0.012	0.539	39.351

Tablo 5.12 (Devam) NJ-ABC Algoritması ile Elde Edilen Kolonun Tasarım Değişkenleri

No	b	h	ρ	f	f_{iyat_USD}
45	294.049	575.500	0.010	0.548	39.973
46	299.722	548.044	0.013	0.542	39.545
47	276.379	579.615	0.015	0.535	39.086
48	266.780	599.889	0.014	0.534	38.991
49	266.766	600.000	0.014	0.534	38.952
50	270.822	599.011	0.013	0.536	39.143

Tablo 5.12’de, tek eksenli eğilme etkisi altındaki betonarme bir kesitin maliyet etkin tasarımı için önerilen NJ-ABC algoritmasıyla elde edilen optimizasyon sonuçlarını özetlemektedir. Tasarım değişkenleri; kesit genişliği (b), kesit yüksekliği (h) ve donatı oranı (ρ) olarak tanımlanmıştır. Ana amaç fonksiyonu f , kullanılan toplam malzeme hacmini metreküp (m^3) cinsinden ifade eder ve beton ile donatının katkılarını dolaylı olarak yansıtır.

Hazır betonlar, C25’ten C50’ye kadar kalite sınıflarına göre sınıflandırılmaktadır. C35 beton sınıfı, günümüzde en dayanıklı betonlardan biri olarak kabul edilmektedir. C35 sınıfı betonun 1 m^3 birim fiyatı 3.120,00 Türk lirası (73 Amerikan doları) dır (Anonymous, 2025).

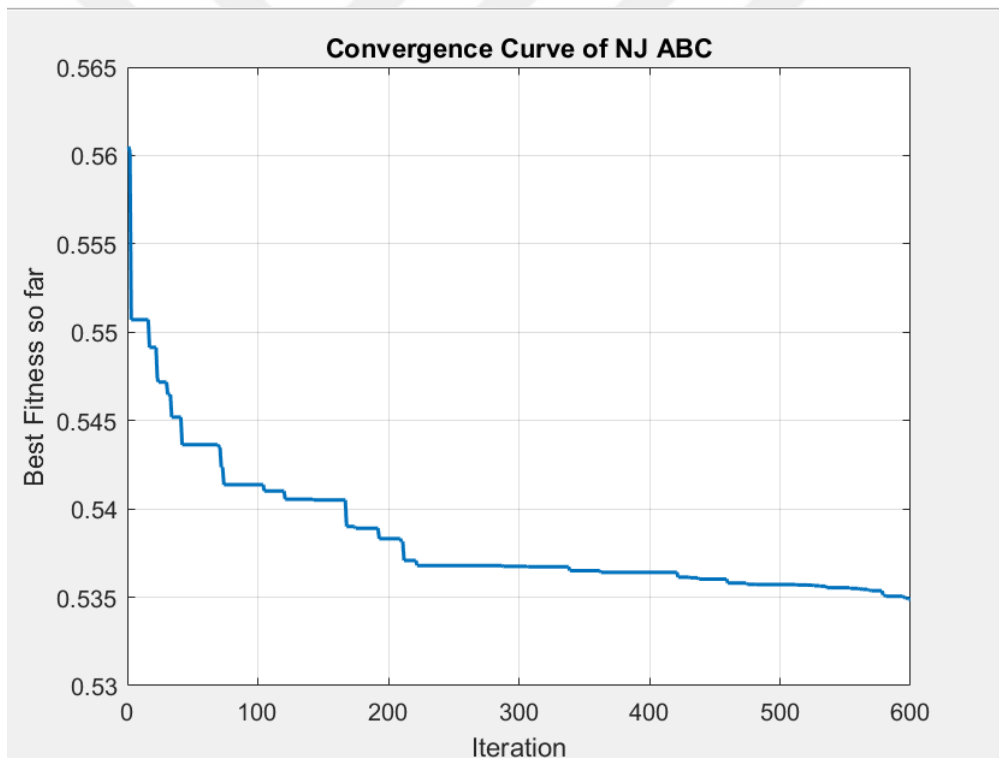
Sonuçlar, algoritmanın yüksek kaliteli ve maliyet açısından verimli çözümlere yakınsamadaki yetkinliğini açıkça ortaya koymaktadır. Elde edilen çözümlerin çoğu, dar bir optimal aralık içinde yer almakta olup, ilgili maliyetler 38.94 USD ile 41.83 USD arasında değişmektedir. Bu durum, algoritmanın istikrarlı yakınsama davranışını ve yapısal kısıtları sağlarken maliyeti azaltma konusundaki etkinliğini göstermektedir.

Tablo 5.10, 5.11 ve 5.12’de sunulan sonuçlar karşılaştırıldığında, önerilen NJ-ABC algoritmasının hem standart ABC algoritmasına hem de Yapay Alg (AAA) algoritmasına kıyasla daha düşük maliyetli ve daha uygun tasarım çözümleri ürettiği açıkça görülmektedir. Bu bulgu, önerilen yöntemin optimize edilen kolon tasarımında üstün performans sergilediğini ve problem kapsamında diğer popüler meta-sezgisel algoritmalara göre daha başarılı sonuçlar elde ettiğini göstermektedir.

Özellikle, algoritma tarafından tanımlanan en iyi çözüm, 20. satırda yer almaktadır. Bu çözümde tasarım parametreleri $b = 267.15$ mm, $h = 599.00$ mm ve $\rho = 0.01$ olarak belirlenmiş olup, amaç fonksiyonu değeri $f = 0.53354$ m^3 ve toplam maliyet = 38.94

USD olarak elde edilmiştir. Bu tasarım yalnızca tüm yapısal ve güvenlik kısıtlarını karşılamakla kalmayıp, aynı zamanda beton ve donatı miktarları arasında optimal bir denge sağlayarak algoritmanın çelişkili hedefler arasındaki dengeyi sağlama becerisini ortaya koymaktadır.

f fonksiyonunun, malzeme kullanımına doğrudan bağlı hacimsel bir ifade olarak yorumlanması, pratik uygulamalar açısından kritik bir içgörü sunmaktadır. Optimizasyon sürecini yönetmelik gerekliliklerine uyum sağlayarak minimum malzeme hacmine yönlendiren NJ-ABC algoritması, gerçek dünya koşullarında ekonomik olarak uygulanabilir yapısal tasarımlar sunma konusundaki yetkinliğini kanıtlamaktadır. Bu bulgular, algoritmanın, malzeme verimliliği ve maliyet azaltımının ön planda olduğu karmaşık mühendislik problemlerini çözme potansiyelini açıkça ortaya koymaktadır.



Şekil 5. 5 Gerçek Dünya Optimizasyonu İçin NJ-ABC Algoritmasının Yakınsama Eğrisi

Şekil 5.5'te, betonarme kesit tasarımının optimizasyonu sırasında NJ-ABC algoritmasının yakınsama davranışını göstermektedir. Eğri, ilk iterasyonlarda amaç fonksiyonu değerinde hızlı ve belirgin bir azalma olduğunu ortaya koymakta olup, bu durum algoritmanın erken arama aşamalarında güçlü bir küresel keşif yeteneğine sahip olduğunu göstermektedir. Yaklaşık 220. iterasyonda, eğri kademeli olarak stabil bir

hâle gelmeye başlamakta ve yaklaşık 600. iterasyona kadar yavaş bir azalma ile yakındurağan bir duruma ulaşmaktadır. Bu davranış, algoritmanın erken yakınsamayı önleyerek, umut vadeden bölgelerde aramayı yoğunlaştırma konusundaki etkinliğini yansıtmaktadır.

Bu tür bir yakınsama dinamiği, algoritmaya entegre edilen adaptif kontrol mekanizmaları, yerel arama entegrasyonu ve çözüm kümesinin hiyerarşik kümeleme yapısı gibi iyileştirmelerin etkinliğini vurgulamaktadır. Elde edilen sonuçlar, NJ-ABC algoritmasının yalnızca doğru çözümler üretmekle kalmayıp, aynı zamanda yüksek düzeyde sağlamlık sergilediğini ve bu sayede doğruluk ve tutarlılık gerektiren gerçek dünya yapısal optimizasyon problemleri için güvenilir bir yöntem olduğunu kanıtlamaktadır.

6 SONUÇ VE GELECEK ÇALIŞMALAR

Bu tez kapsamında, orijinal ABC algoritmasının temel sınırlılıkları olan yavaş yakınsama, erken yakınsama ve zayıf sömürü yeteneği gibi problemlere çözüm sağlamak amacıyla NJ-ABC isimli geliştirilmiş bir algoritma önerilmiştir. Zor ya da yüksek boyutlu optimizasyon problemlerinde bu sınırlılıklar genellikle ABC algoritmasının verimliliğini azaltmaktadır.

NJ-ABC algoritmasının temel yeniliği, komşu birleştirme tekniğinden esinlenerek geliştirilen kümeleme tabanlı bir arama yaklaşımının entegrasyonudur. Bu mekanizma, aday çözümleri karşılıklı uzaklıklarına göre mekânsal olarak tutarlı kümeler halinde düzenleyerek, arama sürecini daha verimli alanlara yönlendirmektedir. Aramanın yerel kümeler içinde odaklanması, algoritmanın yüksek kaliteli bölgeleri daha etkili bir şekilde değerlendirmesine ve uygun seviyede çeşitliliği korumasına yardımcı olmaktadır. Bu önemli gelişmeye ek olarak, başlangıç popülasyonunun oluşturulmasında Latin Hiperküp Örnekleme (Latin Hypercube Sampling - LHS) yöntemi kullanılarak, arama uzayının başlangıçtan itibaren düzenli ve kapsamlı şekilde kapsanması garanti edilmektedir. Ayrıca, adaptif kontrol mekanizması sayesinde keşif ve sömürü arasındaki denge, optimizasyon süreci boyunca bozunum temelli bir adaptasyon faktörü kullanılarak dinamik biçimde ayarlanmaktadır. Parçacık Sürüşü Optimizasyonu (PSO) algoritmasından esinlenen güncelleme dinamikleri ile gözcü arı aşaması değiştirilerek arama sağlamlığı daha da iyileştirilmekte ve durgun çözümlerin daha başarılı bir şekilde yenilenmesi sağlanmaktadır. Son aşamalarda, hassasiyeti artırmak ve en iyi çözümleri geliştirmek amacıyla gradyan tabanlı yerel bir arama yöntemi de algoritmaya eklenmiştir.

NJ-ABC algoritmasının performansı, CEC 2022 test fonksiyonları üzerinde yapılan deneysel analizlerle kapsamlı bir şekilde doğrulanmıştır. Elde edilen sonuçlar, NJ-ABC'nin PSO, TSA, GWO gibi güncel algoritmaların yanı sıra, MNG-ABC, IGAL-ABC ve OCG-ABC gibi gelişmiş ABC varyantları ve standart ABC algoritmasından da belirgin şekilde üstün olduğunu göstermiştir. NJ-ABC algoritması, geniş bir uygulama alanında daha iyi yerel optimum direnci, gelişmiş yakınsama hızı ve daha kaliteli çözümler sunmuştur.

Bunların yanı sıra, NJ-ABC algoritması, yapısal tasarım yönetmelik kısıtlarını dikkate alarak tek eksenli eğilme altındaki betonarme kolonların maliyet optimizasyonunu

içeren pratik bir mühendislik problemine başarıyla uygulanmıştır. Algoritmanın sınırlı tasarım koşullarına sahip zorlu mühendislik problemlerinde uygulanabilirliği, makul ve ekonomik çözümler üretmesiyle kanıtlanmıştır. Sonuç olarak NJ-ABC algoritması, klasik ABC paradigmasına önemli ölçüde katkıda bulunmuştur. Kümeleme tabanlı yönlendirme, adaptif arama dinamikleri ve yerel iyileştirme tekniklerini içeren algoritma, küresel keşif ve yerel sömürü arasında daha verimli bir denge sağlamıştır. Bu gelişmeler, algoritmanın teorik performansını artırmakla kalmayıp, hem test problemlerinde hem de gerçek mühendislik uygulamalarında güvenilirliğini ortaya koymuştur.

NJ-ABC algoritmasının üstün performansı ve diğer optimizasyon yöntemlerine karşı sağladığı belirgin avantajlara rağmen, gelecekte yapılabilecek çalışmalar için hâlen pek çok potansiyel alan bulunmaktadır. Algoritmanın dinamik sistemler, çok amaçlı optimizasyon ve ileri düzey kısıtlamalar içeren daha karmaşık gerçek dünya problemlerine uygulanması mantıklı bir ilerleme olacaktır. Arama sürecinde popülasyondaki değişimlere göre kendini ayarlayabilen dinamik kümeleme mekanizmalarının oluşturulması, sonuçları daha da geliştirebilir ve algoritmanın keşif-sömürü dengesini güçlendirebilir. Ayrıca NJ-ABC'nin diğer meta-sezgisel yöntemlerle, örneğin Genetik Algoritmalar veya Diferansiyel Evrim ile hibritlenmesi, algoritmanın yakınsama hızını ve çözüm çeşitliliğini artırabilir. Son olarak, NJ-ABC algoritmasının kendini yapılandıran ya da adaptif bir versiyonunun geliştirilmesi ve parametre duyarlılık analizleri ile yakınsama ispatları gibi kapsamlı teorik analizlerin yapılması, algoritmanın davranışını daha derinlemesine anlamayı sağlayarak, daha geniş bir problem kümesine uygulanabilirliğini artıracaktır.

7. KAYNAKLAR

- Ahirwal, M. K., Kumar, A., & Singh, G. K. (2014). Adaptive filtering of EEG/ERP through bounded range artificial bee colony (BR-ABC) algorithm. *Digital Signal Processing*, 25, 164-172.
- Akay, B., & Karaboga, D. (2012). A modified artificial bee colony algorithm for real-parameter optimization. *Information Sciences*, 192, 120-142.
- Anonymous. (2025). *karakod: The price of one cubic meter of concrete*. <https://www.karekod.org/blog/hazir-beton-fiyaatlari-2023>
- Babaeizadeh, S., & Ahmad, R. (2017). Enhanced constrained artificial bee colony algorithm for optimization problems. *Int. Arab J. Inf. Technol.*, 14(2), 246-253.
- Bai, Y., Zhang, C., & Bai, W. (2023). A two-level parallel decomposition-based artificial bee colony method for dynamic multi-objective optimization problems. *Applied Soft Computing*, 147, 110741.
- Banitalebi, A., Abd Aziz, M. I., Bahar, A., & Aziz, Z. A. (2015). Enhanced compact artificial bee colony. *Information Sciences*, 298, 491-511.
- Bansal, J. C., Sharma, H., & Jadon, S. S. (2013). Artificial bee colony algorithm: a survey. *International Journal of Advanced Intelligence Paradigms*, 5(1-2), 123-159.
- Bekdas, G., & Nigdeli, S. M. (2016). Bat algorithm for optimization of reinforced concrete columns. *PAMM*, 16(1), 681-682.
- Bekdaş, G., Nigdeli, S. M., Yücel, M., & Kayabekir, A. (2021). Yapay zeka optimizasyon algoritmaları ve mühendislik uygulamaları. *Seçkin Yayıncılık, Ankara*.
- Biswas, S., Chatterjee, A., & Goswami, S. K. (2013). An artificial bee colony-least square algorithm for solving harmonic estimation problems. *Applied Soft Computing*, 13(5), 2343-2355.
- Blum, C., & Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM computing surveys (CSUR)*, 35(3), 268-308.
- Bolideh, A., Arab, H. G., & Ghasemi, M. (2019). OPTIMAL DESIGN OF COLUMNS FOR AN INTERMEDIATE MOMENT FRAME UNDER UNIAXIAL MOMENT AND AXIAL LOADS. *Int. J. Optim. Civil Eng*, 9(4), 689-708.
- Chang, W.-D. (2013). Nonlinear CSTR control system design using an artificial bee colony algorithm. *Simulation Modelling Practice and Theory*, 31, 1-9.
- Committee, A. (2008). Building code requirements for structural concrete (ACI 318-08) and commentary.
- Das, S., Biswas, S., & Kundu, S. (2013). Synergizing fitness learning with proximity-based food source selection in artificial bee colony algorithm for numerical optimization. *Applied Soft Computing*, 13(12), 4676-4694.
- Dorigo, M., & Stutzle, T. (2004). *Ant Colony Optimization*. MIT Press, Cambridge, MA.
- Etminaniefahani, A., Gu, H., & Salehipour, A. (2022). ABFIA: A hybrid algorithm based on artificial bee colony and Fibonacci indicator algorithm. *Journal of Computational Science*, 61, 101651.
- Gao, W., & Liu, S. (2011). Improved artificial bee colony algorithm for global optimization. *Information Processing Letters*, 111(17), 871-882.
- Gao, W., Liu, S., & Huang, L. (2012). A global best artificial bee colony algorithm for global optimization. *Journal of Computational and Applied Mathematics*, 236(11), 2741-2753.

- Holland, J. H. (1975). Adaptation in natural and artificial systems. *University of Michigan Press google schola*, 2, 29-41.
- Imanian, N., Shiri, M. E., & Moradi, P. (2014). Velocity based artificial bee colony algorithm for high dimensional continuous optimization problems. *Engineering Applications of Artificial Intelligence*, 36, 148-163.
- Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3), 264-323.
- Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization.
- Karaboga, D., & Akay, B. (2009). A comparative study of artificial bee colony algorithm. *Applied mathematics and computation*, 214(1), 108-132.
- Karaboga, D., & Basturk, B. (2008). On the performance of artificial bee colony (ABC) algorithm. *Applied Soft Computing*, 8(1), 687-697.
- Karaboga, D., Gorkemli, B., Ozturk, C., & Karaboga, N. (2014). A comprehensive survey: artificial bee colony (ABC) algorithm and applications. *Artificial intelligence review*, 42, 21-57.
- Karaboga, N., & Latifoglu, F. (2013). Elimination of noise on transcranial Doppler signal using IIR filters designed with artificial bee colony-ABC-algorithm. *Digital Signal Processing*, 23(3), 1051-1058.
- Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. Proceedings of ICNN'95-international conference on neural networks,
- Kumar, A., Price, K. V., Mohamed, A. W., Hadi, A. A., & Suganthan, P. N. (2022). *Problem Definitions and Evaluation Criteria for the CEC 2022 Special Session and Competition on Single Objective Bound Constrained Numerical Optimization* [Technical Report].
- Kumar, M., Thakur, A., & Singh, S. (2016). Optimization of Some Standard Functions using Artificial Algae Algorithm. *Int. J. Eng. Res. Technol.(IJERT)*, 4(10.17577).
- Li, J.-Q., Pan, Q.-K., & Tasgetiren, M. F. (2014). A discrete artificial bee colony algorithm for the multi-objective flexible job-shop scheduling problem with maintenance activities. *Applied Mathematical Modelling*, 38(3), 1111-1132.
- Li, X., Zhang, S., & Shao, P. (2024). Discrete artificial bee colony algorithm with fixed neighborhood search for traveling salesman problem. *Engineering Applications of Artificial Intelligence*, 131, 107816.
- Ma, L., Zhu, Y., Zhang, D., & Niu, B. (2016). A hybrid approach to artificial bee colony algorithm. *Neural Computing and Applications*, 27, 387-409.
- Ma, M., Liang, J., Guo, M., Fan, Y., & Yin, Y. (2011). SAR image segmentation based on artificial bee colony algorithm. *Applied Soft Computing*, 11(8), 5205-5214.
- McKay, M. D., Beckman, R. J., & Conover, W. J. (2000). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 42(1), 55-61.
- Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in engineering software*, 69, 46-61.
- Moehle, J. P. (2019). Key changes in the 2019 edition of the ACI Building Code (ACI 318-19). *Concrete International*, 41(8), 21-27.
- Nematollahi, A. F., Rahiminejad, A., & Vahidi, B. (2019). A novel multi-objective optimization algorithm based on Lightning Attachment Procedure Optimization algorithm. *Applied Soft Computing*, 75, 404-427.
- Ni, X., Hu, W., Fan, Q., Cui, Y., & Qi, C. (2024). A Q-learning based multi-strategy integrated artificial bee colony algorithm with application in unmanned vehicle path planning. *Expert systems with applications*, 236, 121303.

- Nigdeli, S. M., Bekdaş, G., & Yang, X.-S. (2015). Application of the flower pollination algorithm in structural engineering. In *Metaheuristics and optimization in civil engineering* (pp. 25-42). Springer.
- Niu, W., She, W., Zhong, L., Wang, Y., Tian, Z., & Liu, W. (2023). Elite-centered artificial bee colony algorithm with extended solution boundary. *Applied Soft Computing*, 148, 110906.
- Ozturk, C., Hancer, E., & Karaboga, D. (2014). Automatic clustering with global best artificial bee colony algorithm. *Journal of the Faculty of Engineering and Architecture of Gazi University*, 29(4), 677-687.
- Pan, X., Wang, Y., Lu, Y., & Sun, N. (2024). Improved artificial bee colony algorithm based on two-dimensional queue structure for complex optimization problems. *Alexandria Engineering Journal*, 86, 669-679.
- Pian, J., Wang, G., & Li, B. (2018). An improved ABC algorithm based on initial population and neighborhood search. *IFAC-PapersOnLine*, 51(18), 251-256.
- Rajasekhar, A., Lynn, N., Das, S., & Suganthan, P. N. (2017). Computing with the collective intelligence of honey bees—a survey. *Swarm and Evolutionary computation*, 32, 25-48.
- Sabat, S. L., Udgata, S. K., & Abraham, A. (2010). Artificial bee colony algorithm for small signal model parameter extraction of MESFET. *Engineering Applications of Artificial Intelligence*, 23(5), 689-694.
- Sahin, O., Akay, B., & Karaboga, D. (2021). Archive-based multi-criteria Artificial Bee Colony algorithm for whole test suite generation. *Engineering Science and Technology, an International Journal*, 24(3), 806-817.
- Saitou, N., & Nei, M. (1987). The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular biology and evolution*, 4(4), 406-425.
- Sharma, S., Kumar, S., & Sharma, K. (2021). Improved Gbest artificial bee colony algorithm for the constraints optimization problems. *Evolutionary Intelligence*, 14, 1271-1277.
- Song, S., Tang, C., Song, Z., Qu, J., & Chen, X. (2024). Improving the artificial bee colony algorithm with a proprietary estimation of distribution mechanism for protein–ligand docking. *Applied Soft Computing*, 161, 111732.
- Sun, H., Luş, H., & Betti, R. (2013). Identification of structural models using a modified Artificial Bee Colony algorithm. *Computers & Structures*, 116, 59-74.
- Szeto, W. Y., & Jiang, Y. (2014). Transit route and frequency design: Bi-level modeling and hybrid artificial bee colony algorithm approach. *Transportation Research Part B: Methodological*, 67, 235-263.
- Tsai, H.-C. (2014). Integrating the artificial bee colony and bees algorithm to face constrained optimization problems. *Information Sciences*, 258, 80-93.
- Ustun, D., Toktas, A., Erkan, U., & Akdagli, A. (2022). Modified artificial bee colony algorithm with differential evolution to enhance precision and convergence performance. *Expert systems with applications*, 198, 116930.
- Uzlu, E., Akpınar, A., Özturk, H. T., Nacar, S., & Kankal, M. (2014). Estimates of hydroelectric generation using neural networks with the artificial bee colony algorithm for Turkey. *Energy*, 69, 638-647.
- Wang, C., Shang, P., & Shen, P. (2022). An improved artificial bee colony algorithm based on Bayesian estimation. *Complex & Intelligent Systems*, 8(6), 4971-4991.
- Xiang, W.-l., & An, M.-q. (2013). An efficient and robust artificial bee colony algorithm for numerical optimization. *Computers & Operations Research*, 40(5), 1256-1265.

- Xiang, Y., Peng, Y., Zhong, Y., Chen, Z., Lu, X., & Zhong, X. (2014). A particle swarm inspired multi-elitist artificial bee colony algorithm for real-parameter optimization. *Computational Optimization and Applications*, 57, 493-516.
- Yancang, L., Jie, W., Libo, L., & Jie, Z. (2017). Improved artificial bee algorithm for reliability-based optimization of truss structures. *The Open Civil Engineering Journal*, 11(1).
- Yang, X.-S. (2020). *Nature-inspired optimization algorithms*. Academic Press.
- Yavuz, G., Durmuş, B., & Aydın, D. (2022). Artificial bee colony algorithm with distant savants for constrained optimization. *Applied Soft Computing*, 116, 108343.
- Yildizdan, G., & Baykan, Ö. K. (2020). A new hybrid BA_ABC algorithm for global optimization problems. *Mathematics*, 8(10), 1749.
- Zhang, D.-L., Ying-Gan, T., & Xin-Ping, G. (2014). Optimum design of fractional order PID controller for an AVR system using an improved artificial bee colony algorithm. *Acta Automatica Sinica*, 40(5), 973-979.
- Zhang, S., Lee, C., Choy, K., Ho, W., & Ip, W. (2014). Design and development of a hybrid artificial bee colony algorithm for the environmental vehicle routing problem. *Transportation Research Part D: Transport and Environment*, 31, 85-99.
- Zhao, M., Song, X., & Xing, S. (2022). Improved artificial bee colony algorithm with adaptive parameter for numerical optimization. *Applied Artificial Intelligence*, 36(1), 2008147.
- Zhao, S., Lin, Q., Feng, X., Chen, H., & Zhang, D. (2024). Hybrid Hierarchical Particle Swarm Optimization with Evolutionary Artificial Bee Colony Algorithm for Task Scheduling in Cloud Computing. *Computers, Materials & Continua*, 78(1).
- Zhou, X., Lu, J., Huang, J., Zhong, M., & Wang, M. (2021). Enhancing artificial bee colony algorithm with multi-elite guidance. *Information Sciences*, 543, 242-258.
- Zhou, X., Tan, G., Wang, H., Ma, Y., & Wu, S. (2025). Artificial bee colony algorithm based on multi-neighbor guidance. *Expert systems with applications*, 259, 125283.
- Zhu, G., & Kwong, S. (2010). Gbest-guided artificial bee colony algorithm for numerical function optimization. *Applied mathematics and computation*, 217(7), 3166-3173.

ÖZGEÇMİŞ

