



T.C.
KONYA TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ



OPTİMİZASYON PROBLEMLERİNİN
ÇÖZÜMÜNDE MİKRO YAPAY ALG
ALGORİTMASININ GELİŞTİRİLMESİ

Hüseyin Samet CAN

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

Temmuz - 2023
KONYA
Her Hakkı Saklıdır

TEZ KABUL VE ONAYI

Hüseyin Samet CAN tarafından hazırlanan “**Optimizasyon Problemlerinin Çözümünde Mikro Yapay Alg Algoritmasının Geliştirilmesi**” adlı tez çalışması 05/07/2023 tarihinde aşağıdaki jüri tarafından oy birliği ile Konya Teknik Üniversitesi Lisansüstü Eğitim Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı’nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Jüri Üyeleri

İmza

Başkan

Prof. Dr. İsmail BABAOĞLU

.....

Danışman

Doç. Dr. Sait Ali UYMAZ

.....

Üye

Doç Dr. Mehmet Akif ŞAHMAN

.....

Yukarıdaki sonucu onaylarım.

Prof. Dr. Mevlüt UYAN
Enstitü Müdürü

Bu tez çalışması Bilimsel Araştırma Projeleri Koordinatörlüğü tarafından 191013022 nolu proje ile desteklenmiştir.

TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Hüseyin Samet CAN
Tarih: 05/07/2023

ÖZET

YÜKSEK LİSANS TEZİ

OPTİMİZASYON PROBLEMLERİNİN ÇÖZÜMÜNDE MİKRO YAPAY ALG ALGORİTMASININ GELİŞTİRİLMESİ

Hüseyin Samet CAN

Konya Teknik Üniversitesi
Lisansüstü Eğitim Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Sait Ali UYMAZ

2023, 39 Sayfa

Jüri

Doç. Dr. Sait Ali UYMAZ
Prof. Dr. İsmail BABAOĞLU
Doç. Dr. Mehmet Akif ŞAHMAN

Son yıllarda optimizasyon problemlerinin çözümünde metasezgisel algoritmalar sıklıkla kullanılmaktadır. Karmaşıklığı ve parametre sayısı yüksek problemlerde bu çözüm yöntemleri zaman ve maliyet açısından yeterli faydayı sağlayamamaktadır. Bu nedenle literatürde mikro metasezgisel optimizasyon yöntemleri önerilmiştir. Mikro algoritmalar daha küçük popülasyon boyutları ile büyük ölçekli problemlerin çözümünde başarılı sonuçlar vermektedir. Bununla birlikte, araştırmacılar tarafından yapılan çalışmalarda, popülasyon boyutunda küçülmenin yeterli olmadığı erken yakınsama ve yüksek durgunluk yaşayan mikro algoritmaların olduğu görülmüştür. Düşük popülasyonla birlikte algoritmanın yapısı bozulmadan eklenen metotlarla mikro metasezgisel algoritmalar literatüre sunulmuştur.

Bu tez projesinde, yapay alg algoritması üzerinde düşük popülasyon kullanarak ve algoritma çalışma yapısını bozmadan eklenecek yardımcı metotlar ile, optimizasyon problemleri çözümünde performansı yüksek bir mikro yapay alg algoritması üzerine bir çalışma yapılmıştır. Yapılan çalışma Mikro Parçacık Sürü Optimizasyonu (μ PSO), Mikro Bakteri Besin Arama Optimizasyonu Algoritması (μ BFOA) ve standart Yapay Alg Algoritması (AAA) ile karşılaştırılmış ve başarılı sonuçlar ortaya koyduğu gözlemlenmiştir. Önerilen yöntemle karşılaştırılan algoritmaların kullanıldığı fonksiyonlar dışında CEC2015 fonksiyonları ile denemelerde Lokal alanlara takıldığı gözlemlenmiş ve bu sorunu aşmak için araştırma ve çalışmalar yapılmıştır. Bu işlemlerle birlikte esas önerilen yöntem olan Levy ile Mikro Yapay Alg Algoritması (μ AAA_{levy}) sunulmuştur. Bu algoritma ise 3 ve 40 popülasyonlu AAA ile karşılaştırıldığında umut vadeden bir algoritma olduğunu göstermiştir.

Anahtar Kelimeler: Mikro Metasezgisel Algoritmalar, Optimizasyon, Yapay Alg Algoritması, Levy Uçuşu

ABSTRACT

MS THESIS

DEVELOPMENT OF MICRO-ARTIFICIAL ALGAE ALGORITHM FOR SOLUTION OF OPTIMIZATION PROBLEMS

Hüseyin Samet CAN

**Konya Technical University
Institute of Graduate Studies
Department of Computer Engineer**

Advisor: Assoc. Prof. Dr. Sait Ali UYMAZ

2023, 39 Pages

Jury

Assoc. Prof. Dr. Sait Ali UYMAZ

Prof. Dr. İsmail BABAOĞLU

Assoc. Prof. Dr. Mehmet Akif ŞAHMAN

In recent years, metaheuristic algorithms have been frequently used to solve optimization problems. These solution methods do not provide sufficient time and cost benefit in complex and high parameter optimization problems. However, due to the lighter hardware requirements and the possibility of working in embedded systems with memory saving approach, Micro-Metaheuristic Algorithm methods have been proposed in the literature. Many Micro-Metaheuristic Algorithms have been introduced by the researchers in the form of microstructures that enable them to produce solutions at a lower cost by developing solutions to accelerate problem solving such as creating and re-updating small-sized populations and protection of individuals.

In this thesis project, a study has been carried out on a micro artificial algae algorithm with high performance in solving optimization problems by using low population on the artificial algae algorithm and adding auxiliary methods without disturbing the algorithm's working structure. The study was compared with Micro Particle Swarm Optimization (μ PSO), Micro Bacteria Nutrient Foraging Optimization Algorithm (μ BFOA) and standard Artificial Algae Algorithm (AAA) and it was observed that it produced successful results. Apart from the functions using the algorithms compared with the proposed method, it was observed that CEC2015 functions were stuck in local areas in experiments with CEC2015 functions and research and studies were carried out to overcome this problem. With these procedures, the main proposed method, Micro Artificial Algorithm with Levy (μ AAAlevy) was presented. This algorithm has shown to be a promising algorithm compared to AAA with 3 and 40 populations.

Keywords: Artificial Algae Algorithm, Optimization, Micro Metaheuristic Algorithms, Levy Flight

ÖNSÖZ

Bu tez çalışmasında geçen süreçte sürekli desteklerini esirgemeyen danışmanım Doç. Dr. Sait Ali UYMAZ'a, çalışmalarım sırasında yardım ve desteklerini esirgemeyen anneme, kardeşime ve Konya Teknik Üniversitesi Bilgi İşlem Daire Başkanlığı'nda çalışan iş arkadaşlarıma saygı ve teşekkürlerimi sunarım.

Hüseyin Samet CAN
KONYA-2023



İÇİNDEKİLER

ÖZET	iv
ABSTRACT.....	v
ÖNSÖZ	vi
İÇİNDEKİLER	vii
1. GİRİŞ	1
2. KAYNAK ARAŞTIRMASI	3
3. MATERYAL VE YÖNTEM.....	6
3.1. Optimizasyon	6
3.2. Metasezgisel Algoritmalar	6
3.3. Mikro Metasezgisel Algoritmalar	8
3.3.1. Popülasyon Oluşturma ve Güncelleme	9
3.3.2. Bireyleri Koruma	10
3.3.3. Uyarlanabilir Popülasyon Büyüklüğü	11
3.3.4. Uyarlanabilir Lokal Arama	11
3.3.5. İşbirlikçi Alt-popülasyon/Alt-sürü	12
3.3.6. Hibritleştirme	13
3.3.7. Parametre Uyarlaması	14
3.4. Yapay Alg Algoritması	16
3.4.1. Helisel Hareket	16
3.4.2. Adaptasyon	17
3.4.3. Evrimsel Süreç	18
3.5. Levy Uçuşu	20
3.6. Mikro Algoritma Üzerine İlk Çalışma: Mikro Yapay Alg Algoritması (μ AAA)	22
3.7. Önerilen Yöntem: Levy ile Mikro Yapay Alg Algoritması (μ AAA _{levy})	23
4. ARAŞTIRMA BULGULARI VE TARTIŞMA	25
4.1. Temel Test Fonksiyonları	25
4.2. CEC2015 Fonksiyonları	26
4.3. Karşılaştırmalar	27
5. SONUÇLAR VE ÖNERİLER	32
5.1 Sonuçlar	32
5.2 Öneriler	32
KAYNAKLAR	33
ÖZGEÇMİŞ	38

SİMGELER VE KISALTMALAR

Simgeler

A_p	: AAA'da adaptasyon parametresi
E	: AAA'da enerji kaybı parametresi
F	: Amaç Fonksiyonu
μ	: Mikro

Kısaltmalar

EA	: Evrimsel Algoritma
AAA	: Yapay Alg Algoritması (Artificial Algae Algorithm)
ABC	: Yapay Arı Kolonisi (Artificial Bee Colony Algorithm)
PSO	: Parçacık Sürü Optimizasyonu (Particle Swarm Optimization)
μ PSO	: Mikro Parçacık Sürü Optimizasyonu (Mikro Particle Swarm Optimization)
BFOA	: Bakteri Besin Arama Optimizasyon Algoritması
TS	: Tabu Arama (Tabu Search)
GA	: Genetik Algoritma (Genetic Algorithm)
PSO	: Parçacık Sürü Optimizasyonu (Particle Swarm Optimization)
MOPSO	: Multi-Objektif PSO (Multi-Objective PSO)
MEPSO	: Multi-Elitist PSO
GCPSO	: Garantili Yakınsak PSO (Guaranteed Convergence PSO)
μ -algoritma	: Mikro Algoritma
μ GA	: Mikro Genetik Algoritma (Micro Genetic Algorithm)
MOO	: Çok Amaçlı Optimizasyon (Multi Objective Optimization)
NSGA-II	: Domine Edilmemiş Sınıflama Genetik Algoritması (Non-Dominated Sorted Genetic Algorithm)
SMES	: Basit Çok Üyeli Evrim Stratejisi
SR	: Stokastik Sıralama
DE	: Diferansiyel Evrim
μ DE	: Mikro Diferansiyel Evrim
μ ODE	: Mikro Muhalefet Tabanlı Diferansiyel Evrim
μ BFOA	: Mikro Bakteri Besin Arama Optimizasyonu Algoritması
Mikro-AIS	: Mikro Yapay Bağışıklık Sistemi
AMGA2	: Arşiv Tabanlı Mikro Genetik Algoritma
ANNs	: Yapay Sinir Ağları
$p\mu$ -CHC	: Paralel Mikro-Çapraz Jenerasyon Elitist Seçimi
HC	: Heterojen Hesaplama
PMG	: Paralel Mikro Genetik Algoritma
ED	: Rampa Oranı Kısıtlı Ekonomik Sevkiyatı
SVC	: Statik Değişken Kompansatör
MABC	: Mikro-Yapay Arı Kolonisi
CMABC	: Kooperatif Mikro-Yapay Arı Kolonisi
MOHGA	: Hiyerarşik GA
MSE	: Ortalama Kare Hatasını
MDEVIM	: Vektörize Edilmiş Rasgele Mutasyon Faktörü ile Mikro Diferansiyel Evrim

μ JADE	: Mikro Adaptif Diferansiyel Evrim
cDE	: Kompakt Diferansiyel Evrim
IEEE	: Elektrik ve Elektronik Mühendisleri Enstitüsü
CEC	: Evrimsel Hesaplama Kongresi
EEv	: Seçkin Elitist Evrim
CMBFOA	: Kaotik Mikro Bakteriyel Besin Arama Algoritması



1. GİRİŞ

Optimizasyon problemleri, özellikle gerçek dünya problemlerinin optimum şekilde çözülmesi için matematiksel ifadelerle tanımlanması gerekir. Bu problemler, amaç veya kısıtlama fonksiyonlarının sayısına, fonksiyonun yapısına, amaç fonksiyonunun karakteristiğine, tasarım değişkenlerinin tipine, değerlerin belirsizliğine ve hesaplama zorluğuna göre sınıflandırılır (Yang, 2010). Ayrıca objektif fonksiyonların sayısına göre; tek ve çok amaçlı olarak, kısıtlama fonksiyonlarına göre; sınırlandırılmamış, eşitlik kısıtlı ve eşitsiz kısıtlı, karakteristik fonksiyonlara göre; tek modlu, çok modlu ve hibrit olarak ayrılır. Optimizasyon problemlerinin çözümü için çeşitli yöntemler geliştirilmiştir. Bu yöntemler deterministik ve stokastik yöntemler olarak gruplandırılabilir. Klasik algoritmalar deterministiktir ve aynı başlangıç noktaları ile hep aynı yolu izlerler. Modern optimizasyon algoritmaları ise sezgiseldirler yani rastgeleliğe sahiptirler. Her zaman optimumu bulma garantisi vermezler ama kaliteli çözümler sunarlar. Bu nedenle daha yüksek oranda kaliteli çözümler bulabilmek için Metasezgisel Algoritmalar üzerine çalışmalar devam etmektedir.

Metasezgisel Algoritmalar çoğunlukla doğadan veya canlı yaşamından esinlenerek ele alınırlar ve optimizasyon problemlerine uygun çözümler getirmeyi amaçlamaktadırlar. Alan Turing'in ikinci dünya savaşında Bletchley Park'da Alman Enigma şifrelerini kırmış olduğu Turing'in çalışma yapısı Yang (2010) tarafından sezgisel algoritma olarak tanımlanmıştır. Metasezgisel kavramının ilk kullanımı ise 1986 yılında Fred Glover'in Tabu Arama (TS) algoritmasını sunduğu çalışma olarak kabul edilmektedir. Metasezgisel Algoritmalar genellikle büyük popülasyonlara ihtiyaç duyar. Bunun 2 nedeni vardır; daha çok birey arama uzayını daha iyi keşfetmeyi sağlar ve erken yakınsama ihtimalini düşürür (erken yakınsama popülasyondaki bütün bireylerin birbirine çok benzediği zaman gerçekleşir) (Viveros-Jiménez ve Mezura-Montes, 2012). Yüksek boyutlu problemlerde, büyük popülasyonlarla arama uzayının keşfedilmesi önemli miktarda zaman ve hesaplama maliyetini de beraberinde getirmektedir. Bu durumda araştırmacıları daha küçük popülasyonlarla çalışmaya teşvik etmiştir.

Mikro Algoritma terimi (μ -algoritma), popülasyon bazlı Metasezgisel Algoritmaların optimizasyon problemlerine çok küçük popülasyonlarla çözüm bulmaya çalışan algoritmaları ifade eder. μ -Algoritmalar, genellikle gömülü sistemlerde kullanılmaktadır. Bunun sebebi daha hafif donanım gereksinimleri ve bellek tasarrufu yaklaşımıyla gömülü sistemlerde çalışma olasılıkları nedeniyle, çeşitli uygulamalarda

kullanılmıştır (Caraffini ve ark., 2013). Küçük popülasyon kullanmak, işlev çağrım sayısını azaltır, ancak çeşitlilik eksikliği nedeniyle, durgunluk gibi erken yakınsama riskini de arttırmaktadır. Erken yakınsama problemi, popülasyonun optimal olmayan bir çözüme (lokal en iyi) yakınlaştığı durumu ifade eder (Lampinen ve Zelinka, 2000). Bu genellikle, popülasyon bireylerinin çeşitliliğini kaybettiğinde ve yerel olarak optimal çözümler üretemediğinde ortaya çıkar. Bu durumda, algoritma normalden daha yavaş ilerler ve daha iyi sonuç almayı engeller. μ -algoritma'nın bu olumsuz durumunu gidermek için araştırmacılar, çeşitli yardımcı özellikler kullanmışlardır.

Tezin içeriği aşağıdaki gibidir:

2. bölümde; Optimizasyon, Metasezgisel Algoritmalar, Mikro Metasezgisel Algoritmalar ve Levy Uçuşu hakkında yapılan kaynak araştırmasından bahsedilmiştir.

3. bölümde Optimizasyon Problemleri, Metasezgisel Algoritma ve Mikro Metasezgisel Algoritma kavramlarından bahsedilmiştir. Mikro Metasezgisel Yöntemlerden bahsedilmiştir. Sonrasında Yapay Alg Algoritması (AAA) ve önerilen algoritma olan Levy Uçuşu ile Mikro Yapay Alg Algoritması tanıtılmıştır.

4. bölümde öncelikle yapılan çalışmada kullanılan test fonksiyonları tanıtılmıştır. Daha sonra performans testleri, sonuçlar ve analizler yapılmıştır.

5. bölümde, yapılan çalışmaya ait sonuçlar değerlendirilmiş ve gelecek çalışmalar için öneriler sunulmuştur.

2. KAYNAK ARAŞTIRMASI

Metasezgisel algoritmalar, optimize edilmesi gereken karmaşık problemleri çözmek için kullanılan bir tür optimizasyon algoritmasıdır. Bu algoritmalar, doğal seleksiyon, genetik, taklit, sürü davranışı gibi doğadan veya diğer ilham kaynaklarından esinlenen yöntemleri kullanarak çözüm adaylarını ararlar.

Metasezgisel algoritmaların temel özellikleri şunlardır:

Popülasyon Tabanlı Yaklaşım: Metasezgisel algoritmalar, bir popülasyonu kullanarak çözüm adaylarını temsil eder. Popülasyon, olası çözümlerin bir arada bulunduğu bir grup bireyden oluşur.

Sezgisel Arama Stratejileri: Metasezgisel algoritmalar, sezgisel arama stratejilerini kullanarak çözüm alanında gezinmeyi sağlar. Bu stratejiler, bir bireyin çözüm alanında hareket etmesini, yeni çözüm adayları üretmesini ve mevcut çözümlerle karşılaştırmasını içerir.

Optimizasyon Hedefine Uygunluk Fonksiyonu: Metasezgisel algoritmalar, optimize edilmek istenen hedef fonksiyonunun değerini minimize etme veya maksimize etme amacıyla çalışır. Bu hedef fonksiyonu, çözüm adaylarının performansını ölçer.

Parametre Ayarlaması: Metasezgisel algoritmalar, farklı parametre değerleriyle deneme yaparak en iyi sonuçları elde etmek için parametrelerin ayarlanmasını gerektirir. Bu parametreler, popülasyon boyutu, iterasyon sayısı, mutasyon oranı gibi algoritmanın davranışını etkileyen değerlerdir.

Metasezgisel algoritmalar, birçok farklı probleme uygulanabilir ve genellikle karmaşık, çok boyutlu veya optimizasyon problemleri olarak adlandırılan problemlerin çözümünde etkilidir. Genetik algoritmalar, parçacık sürü optimizasyonu (PSO), taklit optimizasyonu (TO), küme optimizasyonu (CO), simüle edilen tavlama (SA), karınca kolonisi optimizasyonu (ACO) ve çiftçi algoritması (FA) gibi metasezgisel algoritma örnekleri vardır.

Bu algoritmalar, optimize edilen probleme ve kullanıcının gereksinimlerine göre seçilmelidir. Her bir algoritmanın kendi avantajları, dezavantajları ve uygunlukları vardır. Başarılı bir şekilde uygulanması, parametre ayarlamasının yanı sıra algoritmanın doğru şekilde yapılandırılmasını gerektirir.

Genel olarak, metasezgisel algoritmalar, karmaşık optimizasyon problemlerini çözmek için etkili ve esnek bir yaklaşım sağlar. Ancak, her problem için en uygun

algoritmayı belirlemek ve performansı optimize etmek için dikkatli bir analiz ve ayarlama süreci gereklidir.

Mikro Metasezgisel Algoritmalar, popülasyon tabanlı yaklaşımı benimserler ve bir grup çözüm adayını temsil eden bir popülasyon kullanırlar. Her bir birey, çözümün bir adayını temsil eder ve bir uygunluk fonksiyonu tarafından değerlendirilir.

Bu algoritmalar, sezgisel arama stratejilerini kullanarak çözüm alanında gezinmeyi sağlarlar. Bireyler, doğal seleksiyon, rekombinasyon, mutasyon gibi genetik operatörler veya taklit, iletişim, öğrenme gibi başka stratejiler kullanarak birbirleriyle etkileşime geçerler. Bu etkileşimler, yeni çözüm adaylarının üretilmesini ve daha iyi çözümlerin bulunmasını sağlar.

Mikro Metasezgisel Algoritmalar, şu avantajlara sahiptir:

Esneklik: Mikro Metasezgisel Algoritmalar, farklı problemlere uyarlanabilen genel bir çerçeve sunar. Bu algoritmalar, çeşitli optimizasyon problemlerinde başarılı bir şekilde uygulanabilir.

Küresel Arama: Mikro Metasezgisel algoritmalar, küresel optimuma yakınsamayı hedeflerler. Bu, çözüm alanında geniş bir arama yapabilme yetenekleri olduğu anlamına gelir ve daha iyi çözümlerin keşfedilmesini sağlar.

Paralel İşleme: Mikro Metasezgisel algoritmalar, popülasyon tabanlı yaklaşımları sayesinde paralel hesaplama için uygundur. Bu, çoklu işlemci veya hesaplama kaynağı kullanılarak algoritmanın hızlı bir şekilde çalıştırılabilmesini sağlar.

Optimizasyon problemleri çözümünde kullanımı yaygınlaşan μ -Algoritmalar'dan bazıları; Krishnakumar'ın (1990) çalışma ürünü olan mikro genetik algoritma (μ GA), mikro öğretme-öğrenme tabanlı optimizasyon (μ TLBO), Parsopoulos'un 2009 yılında sunmuş olduğu Mikro Diferansiyel Evrim (μ DE) ve yine Parsopoulos'aun sunmuş olduğu Mikro Parçacık Sürü Optimizasyonu (μ PSO) ve 2009'da Dasgupta'nın çalışma ürünü olan μ BFOA yöntemleri örnek gösterilebilir.

Bu örneklerden μ PSO'da popülasyon boyutu küçültülerek yüksek boyutta yaşanan zorluktan kurtulması sağlanmıştır. Ayrıca bireylerin elde ettiği optimuma uzak sonuçlar bir kara listede tutulmuş ve Coulomb yasası yardımı her yeni iterasyonda bu kara listedeki sonuçlardan kaçınılarak performanslı bir algoritma ortaya çıkarılmıştır (Huang ve Mohan, 2006).

μ BFOA'da ise popülasyon sayısını 3'e düşürmekle birlikte popülasyon güncelleme ve oluşturma yöntemi kullanılmıştır. Bu yöntemde, her iterasyon sonrasında en iyi birey saklanmaktadır. En iyi 2. birey 1. bireye yaklaştırılmakta ve 3. birey rastgele

yeniden oluşturulmaktadır. 3. Bireyin rastgele oluşturulmasındaki amaç ise çeşitliliği sağlamak olarak tanımlanmıştır (Dasgupta ve ark., 2009).



3. MATERYAL VE YÖNTEM

3.1. Optimizasyon

Optimizasyon, matematiksel bir yapıya dönüştürülmüş bir problemin maksimum veya minimum olması açısından optimal koşulların elde edilmesini sağlar. Birden fazla şekilde matematiksel olarak tanımlanabilecek optimizasyonlardan biri, aşağıda açıklanan temel yapıdır:

$$\text{Minimize veya Maksimize } f_i(x), \quad (i = 1, 2, \dots, M) \quad (3.1)$$

$$h_j(x) = 0, \quad (j = 1, 2, \dots, J) \quad (3.2)$$

$$g_k(x) \leq 0, \quad (k = 1, 2, \dots, K) \quad (3.3)$$

$$X = (x_1, x_2, x_3, \dots, x_d) \quad (3.4)$$

Burada f_i , h_j ve g_k genel doğrusal olmayan fonksiyonlar, integraller veya diferansiyel eşitlikler olabilir. Dizayn vektör X , d boyutlu uzayda sürekli, ayrık veya ikisinin karışımı olabilir. f_i fonksiyonu maliyet ya da amaç fonksiyonları olarak adlandırılır. Eğer $M > 1$ ise optimizasyon çok amaçlıdır.

Optimizasyon problemleri, amaç veya kısıtlama fonksiyonlarının sayısına, fonksiyonun yapısına, amaç fonksiyonunun karakteristiğine, tasarım değişkenlerinin tipine, değerlerin belirsizliğine ve hesaplama zorluğuna göre sınıflandırılır (Yang, 2010). Objektif fonksiyonların sayısına göre; tek ve çok amaçlı olarak, kısıtlama fonksiyonlarına göre; sınırlandırılmamış, eşitlik kısıtlı ve eşitsiz kısıtlamalı olarak, karakteristik fonksiyonlara göre; tek modlu, çok modlu, hibrit ve kompozit olarak ayrılır.

Optimizasyon problemlerinin çözümü için çeşitli yöntemler geliştirilmiştir. Bu yöntemler deterministik ve stokastik yöntemler olarak gruplandırılabilir. Klasik algoritmalar deterministiktir ve aynı başlangıç noktaları ile hep aynı yolu izlerler. Modern optimizasyon algoritmaları ise sezgiseldirler yani rastgeleliğe sahiptirler. Her zaman optimumu bulma garantisi vermezler ama kaliteli çözümler sunarlar. Bu nedenle daha yüksek oranda kaliteli çözümler bulabilmek için sezgisel algoritmalar üzerine çalışmalar devam etmektedir.

3.2. Metasezgisel Algoritmalar

Sezgisel veya Metasezgisel Algoritmalar doğada gerçekleşen bir olaydan esinlenerek ele alınan optimizasyon problemlerine uygun çözümler getirmeyi amaçlamaktadır. Bu yöntemler, ele aldığı farklı özellikler göz önünde bulundurularak çeşitli şekillerde sınıflandırılabilir. Temel olarak metasezgisel yöntemler, doğadan esinlenen ve doğadan esinlenmeyen yöntemler, dinamik ve statik amaç fonksiyonuna sahip yöntemler, bir komşuluk yapısına ve değişken komşuluk yapısına sahip yöntemler, hafıza kullanan ve kullanmayan yöntemler ve tek çözüme dayalı ya da toplum tabanlı metasezgisel yöntemler olarak sınıflandırılabilir (Blum ve Roli, 2003). Toplum tabanlı yöntemlerde kendi içinde evrimsel algoritmalar ve sürü zekâsı olmak üzere sınıflandırılmaktadır.

Metasezgisel yöntemlerin ilk kullanıldıkları zamanı kesin olarak belirtmek zor olsa da, Alan Turing'in ikinci dünya savaşında Bletchley Park'da Alman Enigma şifrelerini kırmış olduğu Turing'in çalışma yapısı Yang (2010) tarafından sezgisel algoritma olarak tanımlanmıştır. Metasezgisel kavramının ilk kullanımı ise 1986 yılında Fred Glover'in Tabu Arama (TS) algoritmasını sunduğu çalışma olmuştur (Glover, 1986).

1960 ve 1970'ler Evrimsel Algoritmaların (EAs) ortaya çıktığı ve geliştirildiği iki önemli 10 yıldır. İlk olarak John Holland ve arkadaşları tarafından Genetik Algoritma (GA) geliştirilmiş ve (Holland, 1975) çalışmasında özeti paylaşılmıştır. Bu çalışmada doğadaki evrimsel süreçten esinlenilmiştir. Bu algorithmada rastgele bir popülasyon oluşturulur ve uygunluk değerleri hesaplanır. Elde edilen uygunluk değerine göre bireylerden iyi olanların sonraki nesile aktarımını sağlamak için sırasıyla seleksiyon, çaprazlama ve mutasyon adımları gerçekleştirilir. Bu işlem programa değişken olarak verilen maksimum iterasyon sayısı tamamlanana kadar ya da en iyi sonuç elde edilene kadar döngü halinde işlenir. GA günümüzde farklı güncelleme ve versiyonlarıyla günümüzde de kullanılmaktadır.

Parçacık Sürü Optimizasyonu (PSO), sürü halinde hareket eden kuş ve balıkların sosyal davranışlarından esinlenerek Kennedy ve Eberhart (1995) tarafından geliştirilmiş bir optimizasyon yöntemidir (Kennedy ve Eberhart, 1995). Temel olarak sürü zekâsına dayanan bir algorithmadır. Sürü halinde hareket eden hayvanların yiyecek ve güvenlik gibi durumlarda, çoğu zaman rasgele sergiledikleri hareketlerin, amaçlarına daha kolay ulaşmalarını sağladığı görülmüştür. PSO bireyler arasındaki sosyal bilgi paylaşımını esas alır. Her bireye parçacık denir ve parçacıklardan oluşan popülasyona da sürü (swarm) denir. Her bir parçacık kendi pozisyonunu, bir önceki tecrübesinden yararlanarak

sürüdeki en iyi pozisyona doğru ayarlar. PSO çalışmasında kullanılan formüller Denklem 3.5 ve 3.6'da gösterilmiştir.

$$V_i(k+1) \leftarrow V_i(k) + C_1\phi_1(P_i(k) - X_i(k)) + C_2\phi_2(G(k) - X_i(k)) \quad (3.5)$$

$$X_i(k+1) \leftarrow X_i(k) + V_i(k) \quad (3.6)$$

Denklem 3.5'te k iterasyon sayısını, V_i bulunduğu iterasyondaki i . parçacığın hızını göstermektedir. C_1 ve C_2 öğrenme faktörleridir ve parçacıkları yerel ve global en iyi parçacıklara yönlendirir. Denklem 3.6'da ise X_i i . parçacığın konumu belirtir. PSO günümüzde Multi-Objektif Parçacık Sürü Optimizasyonu (MOPSO) (Mostaghim ve Teich, 2003), Multi-Elitist PSO (MEPSO) (Das ve ark., 2008a) ve Garantili Yakınsak PSO (GCP SO) (Peer ve ark., 2003) gibi çeşitli güncelleştirmelerle geliştirilerek kullanılmaya devam edilmektedir.

Metasezgisel Algoritmalar genellikle büyük popülasyonlara ihtiyaç duyar. Bunun 2 nedeni vardır; daha çok birey arama uzayını daha çok keşfetmeyi sağlar ve erken yakınsama ihtimalini düşürür (erken yakınsama bütün popülasyon bireyleri birbirine çok benzediği zaman gerçekleşir) (Viveros-Jiménez ve Mezura-Montes, 2012). Bu büyük popülasyonlarla arama uzayının keşfedilmesi önemli miktarda zaman ve hesaplama maliyetini de beraberinde getirmektedir. Bu durumda araştırmacıları daha küçük popülasyonlarla çalışmaya teşvik etmiştir.

3.3. Mikro Metasezgisel Algoritmalar

Mikro Algoritma (μ -algoritma) terimi, popülasyon bazlı Metasezgisel Algoritmaların optimizasyon problemlerine çok küçük popülasyonlarla çözüm bulmaya çalışan algoritmaları ifade eder. μ -algoritmalar, istisnai olarak daha hafif donanım gereksinimleri ve bellek tasarrufu yaklaşımıyla gömülü sistemlerde çalışma olasılıkları nedeniyle, çeşitli uygulamalarda kullanılmıştır (Caraffini ve ark., 2013). Küçük popülasyon kullanmak, işlev çağırım sayısını azaltır, ancak çeşitlilik eksikliği nedeniyle, durgunluk gibi erken yaklaşma riskini de arttırmaktadır.

Erken yakınsama problemi, popülasyonun çok modlu bir objektif fonksiyonun optimal olmayan bir çözümüne yakınlığı durumu ifade eder (Lampinen ve Zelinka, 2000). Bu durum çoğunlukla nüfus çeşitliliğini yitirdiğinde ve yerel tercihlerden

çıkamadığında ortaya çıkar. Bu durumda, algoritma normalden daha yavaş ilerler ve gelişmiş aday çözümlerinin daha da geliştirilmesini durdurabilir.

μ -algoritmalar üzerine yapılan çalışmaları mikro'nun etkisini arttırmasına yardımcı olan şu yöntemlere göre yapılabiliriz;

- Popülasyon oluşturma ve yeniden oluşturma
- Bireyleri Koruma
- Uyarlanabilir popülasyon büyüklüğü
- Uyarlanabilir lokal arama
- İşbirlikçi alt-popülasyon / alt-sürü
- Hibritleştirme
- Parametre Uyarlaması

3.3.1. Popülasyon Oluşturma ve Güncelleme

Mikro Algoritmalar için popülasyon yeniden başlatma fikri mikro Genetik Algoritma (μ GA) ile bu alanda yapılan çalışmalardan biri olmuştur (Goldberg, 1989). Bu yaklaşımda, önceden tanımlanmış sayıda jenerasyondan sonra her bir yakınsak popülasyonun en iyi bireylisi, bir sonraki yinelemenin popülasyonunda rastgele seçilmiş bir bireyle değiştirilir.

Mikro algoritmalar da çok amaçlı optimizasyonda (MOO) kullanılmıştır. Belirli bir popülasyon başlatma stratejisi ile Domine Edilmemiş Sınıflamalı Genetik Algoritmasının (NSGA-II) geliştirilmiş versiyonu, MOO sorunlarını çözmek için standart μ GA içine yerleştirilmiştir (Tan ve ark., 2013). Dört popülasyon büyüklüğüne ve yeniden başlatma stratejisine sahip bir μ GA pareto cephesinin büyük bir kısmını çok düşük bir hesaplama maliyetiyle üretebildiği (Coello ve Pulido, 2001) yapılan çalışmada sunulmuştur. İlk popülasyonu oluşturmak için üç elitizm biçimi ve bir bellek kullanılır (Coello ve Pulido, 2001).

PSO, küçük nüfuslu versiyonlarının son zamanlarda geliştirildiği bilinen sürüzeka algoritmalarından biridir (Huang ve Mohan, 2006; Parsopoulos, 2012). (Cabrera ve Coello, 2007)'de kısıtlı optimizasyon problemleriyle baş etmek için beş parçacıklı bir Mikro-PSO (μ PSO) kullanılır. Bu yöntem, başlatma işlemini kullanarak popülasyon çeşitliliğini korur ve algoritmanın keşif kabiliyetini geliştirmek için bir mutasyon operatörü içerir. Raporlanan sonuçlar Basit Çok Üyeli Evrim Stratejisi (SMES) ve

Stokastik Sıralama (SR) yöntemine karşı rekabetçi bir performans sergilemektedir (Cabrera ve Coello, 2007)

Mikro Muhalefet Tabanlı Diferansiyel Evrim (μ ODE), animasyon eşiklemesi vaka çalışması için önerilmiş ve değerlendirilmiştir (Rahnamayan ve Tizhoosh, 2008). Performansı Kittler algoritması ve Mikro Diferansiyel Evrim (μ DE) ile karşılaştırılır. μ ODE yöntemi bu algoritmaları 16 zorlu test görüntüsünde uygulamıştır ve daha hızlı yakınsama hızı göstermesinin nedeni karşıt tabanlı popülasyon başlatma şemasını gömmektir (Rahnamayan ve Tizhoosh, 2008).

Dasgupta ve arkadaşları 2009'da Mikro Bakteri Besin Arama Optimizasyonu Algoritması (μ BFOA) adı verilen Bakteriyel Besin Arama Optimizasyon Algoritmalarını (BFOAs) "mikro" bir sürümü önerilmiştir. Bu yöntem en iyi bakteriyi değiştirmeden tutmaktadır. Diğer popülasyon üyeleri ise yeniden başlatılmıştır (Dasgupta ve ark., 2009). Bu yaklaşım, standart BFOA'yı daha büyük bir popülasyon büyüklüğüyle gerçekleştirmiştir.

Herrera-Lozada ve arkadaşları 2011'de beş kişiden (antikorlar) oluşan ve sadece 15 klon elde edilen bir Mikro Yapay Bağışıklık Sistemi (Mikro-AIS) önerilmiştir. Bu yaklaşımda, yeniden başlatma işleminde birlikte çalışan, iki basit fakat hızlı mutasyon operatörü nominal bir yakınsama tarzında düşünülerek çeşitlilik korunur (Herrera-Lozada ve ark., 2011).

Centroid tabanlı DE (Rahnamayan ve ark., 2014a; Rahnamayan ve ark., 2014b)'da önerilmiştir. Bu yaklaşım, gelecek nesil nüfusun hesaplanmasında nüfus merkezini kullanır. Tek tip rastgele popülasyon üretime karşı üstün performans gösteren popülasyon başlatma için geliştirilmiştir.

3.3.2. Bireyleri Koruma

Popülasyona dayalı algoritmaların performansını ve özellikle de mikro sürümlerinin performansını artırmak için kullanılan genel bir teknik, gelecek nesil için mevcut nesilden bir veya daha fazla kişiyi korumaktır.

Daha önce yapılan araştırmalardan biri bu yöndeki çalışmalar, beş kromozomlu bir GA Krishnakumar (1990) tarafından önerilmiştir. Strateji, mevcut popülasyondan gelecek kuşağa en iyi bulunan kromozomu kopyalamaktır. Klasik GA ile karşılaştırıldığında daha hızlı bir yakınsama hızı ile sonuçlanan düşük boyutlu problemler üzerinde test edilmiştir.

Kısıtlı MOO için geliştirilmiş bir μ GA sürümü (Tiwari ve ark., 2011)'de, Arşiv Tabanlı Mikro GA (AMGA2) olarak önerilmiştir. Bu algoritma, en iyi ve yönlendirme adayı çözümlerinin dış arşivini koruyan sabit durumlu bir GA'ya dayanmaktadır. Bu küçük nüfus temelli yaklaşım, çalışan nüfusun ayrılmasını, dış arşivi ve optimizasyon işleminin sonucu olarak gereken çözüm sayısını kolaylaştırır.

Mikro-PSO, MOO için kullanılır. PSO yaklaşımı ile karşılaştırıldığında, Pareto'nun orta boyutlu problemlerin ön yüzünde makul derecede iyi yaklaşımlar üretiyor, az sayıdaki objektif fonksiyon değerlendirmeleri ile (çalışma başına sadece 3000 çağrı). Arşivlerden biri arama sırasında bulunan çözümü saklar. Diğer arşiv, elde edilen nihai çözümleri saklar (Cabrera ve Coello, 2010).

3.3.3. Uyarlanabilir Popülasyon Büyüklüğü

Popülasyon sayısını azaltarak DE tabanlı algoritmaların hesaplama maliyetini düşürmeye yönelik bazı yaklaşımlar önerilmiştir (Brest ve Maucec, 2008; Brest ve ark., 2008; Brest ve ark., 2009; Sotelo-Figueroa ve ark., 2013). Brest ve Maucec (2008)'de kademeli olarak azalan nüfus büyüklüğü yöntemi önerilmiştir. Bu yöntem, sonuçların temel DE'ye göre daha yüksek sağlamlık ve verimlilik gösterdiği 13 kıyaslama fonksiyonunda incelenmiştir.

Kendi kendine adapte olan popülasyon büyüklüğü fikri DE için mutlak ve göreceli kodlama yöntemlerini test etmek için kullanılır (Teng ve ark., 2009). 20 kıyaslama problemi üzerinde rapor edilen simülasyon sonuçları, ortalama performans ve stabilite açısından, göreceli kodlama kullanan kendinden uyarlamalı popülasyon büyüklüğünün mutlak kodlama yöntemini ve standart DE algoritmasını daha iyi oluşturduğunu göstermektedir. (Fajfar ve ark., 2012)'de kullanılan en küçük nüfus büyüklüğü NP = 10'dur. Bu yöntem, deneme vektörünü değiştirmek için adayları seçmek için üç farklı kural kullanarak popülasyon boyutunu azaltmaya çalışır.

3.3.4. Uyarlanabilir Lokal Arama

Mikro-GA, tekrarlayan yapay sinir ağlarının (ANNs) eğitimi için uyarlanabilir bir yerel arama yoğunluğu şeklinde yerel ince ayar için kullanılır (Ang ve ark., 2007). Bu yaklaşımın sistem tanımlama görevleri için yararlı olduğu bildirilmektedir.

Yerel bir arama prosedürü, yüksek boyutlu problemlerin üstesinden gelmek için (Olguin-Carbajal ve ark., 2013)'da μ DE algoritması ile hibritlenmiştir. Bununla birlikte, rapor edilen performans sonuçları diğer bazı yöntemlerle karşılaştırılabilir. μ DE

algoritmasının keşif yeteneğini arttırmak ve durgunluğu önlemek için, (Caraffini ve ark., 2013)'te μ DE algoritmasına fazladan bir arama hareketi dahil edilerek eksenler boyunca ilerletilir.

3.3.5. İşbirlikçi Alt-popülasyon/Alt-sürü

Karmaşık yüksek boyutlu problemlerle başa çıkmak için düşük boyutlu ve düşük taramalı alt kümelerden oluşan bir şirket kullanan (Parsopoulos, 2009)'da İşbirlikçi PSO yaklaşımı önerilmiştir. Bu yöntemler kullanılarak umut verici sonuçlar bildirilmiştir. (Parsopoulos, 2012)'de İşbirlikçi μ PSO'nun paralel bir usta-çırak modeli tanıtıldı. Orijinal arama alanı daha küçük boyutlara sahip alt alanlara ayrıştırılır. Ardından, alt-alt kısmi çözelti bileşenlerini tanımlamak için her alt-alanda beş kişi göz önünde bulundurulur. Performansı, standart PSO algoritmasına kıyasla, çözüm kalitesinde önemli gelişmelerle birlikte yaygın olarak kullanılan beş test problemi üzerinde değerlendirilir (Parsopoulos, 2012).

Bir μ DE yaklaşımında (Parsopoulos, 2009), asıl sorunun alt bileşenlerini eş zamanlı olarak bulmak için küçük boyutlu kooperatif alt popülasyonları kullanılır. Alt popülasyonların işbirliğinde sorunun tam çözümünü oluşturmak için alt bileşenler birleştirilir. Bu yöntemin performans değerlendirmesi, (Parsopoulos, K. E., 2009) 'de bildirilen sonuçları cesaretlendiren beş örnek test probleminin yüksek boyutlu örnekleri üzerinde yapılmıştır.

Nesmachnow ve ark., (2012)'de, Paralel Mikro-Çapraz Jenerasyon Elitist Seçimi ($p\mu$ -CHC), heterojen rekombinasyon ve atakistik mutasyona dayanan, Heterojen Hesaplama (HC) ve Grid Zamanlama için etkili bir programlayıcı önerilmiştir. Bu yöntem, paralel alt popülasyon modelini, bir mikro popülasyon ve rastgeleliğe dayalı bir yerel arama yöntemini arayan odaklanmış bir evrimsel model ile birleştirmektedir.

Paralel Mikro Genetik Algoritma (PMGA) olarak adlandırılan mikro-GA'nın paralel sürümü Tippayachai ve ark., (2002)'de rapor edilmiştir. Bu yöntem, monoton olmayan ve monoton olarak artan maliyet fonksiyonlarına sahip birimler üretmek için rampa oranı kısıtlı ekonomik sevkıyatı (ED) sorunlarını çözer. PMGA, 32 işlemcili bir Beowulf kümesinde uygulanır. Raporlanan sonuçlar, bu yaklaşımın çevrimiçi uygulamalar için uygulanabilirliğini göstermektedir.

3.3.6. Hibritleştirme

Bir μ PSO ile birlikte Yapay Bağışıklık Sistemi (AIS) ailesine ait olan bir Klonal Seçim Algoritması (CSA) (CS^2P^2SO) hibrit bir şema olarak (Mitra ve ark., 2008)'de sunulmuştur. Bu hibritlemede, standart PSO algoritmasının gücü artırılır; burada μ PSO, daha az bellek gereksinimi olan optimum çözümü bulmaya yardımcı olur ve CSA, yerel bir minimum noktaya yaklaşma şansını azaltırken keşif kabiliyetini artırır. Simülasyonlar, rekabetçi performansın rapor edildiği dört temel fonksiyon üzerinde gerçekleştirilmektedir.

μ PSO, hareket tahmini (Bakwad ve ark., 2011), güç sistemi stabilizatör tasarımı (Venayagamoorthy ve Das, 2006; Das ve ark., 2008b), statik değişken kompensatör (SVC) sönümlleme kontrol cihazlarının optimal tasarımı (Das ve ark., 2006), reaktif güç optimizasyonu (Han ve Sun, 2012), Kısa vadeli hidrotermal programlaması (Zhang ve ark., 2012), gemi güç sisteminin yeniden yapılandırılması (Wang ve ark., 2013) ve geçici stabilite kısıtlı optimal güç akışı (Wu ve ark., 2014) gibi birçok uygulama geliştirilmiştir.

(Hinojosa ve Araya, 2013)'de bir karma tam sayılı ikili küçük popülasyon PSO optimal güç akışı problemini çözmek için önerilmiştir. Bu algorithmada kullanılan kısıtlama tekniği, dört karar değişkenini uygulanabilir mesafeden buluşsal operatörlerde üretmek ve tutmak için bir stratejiye dayanmaktadır. Bu şekilde, algoritma daha iyi bir objektif değeri elde etmek için araştırma prosedürünü uygulanabilir çözüm alanına odaklar. Bu teknik, son çözümün niteliklerini yakınsama hızı kadar iyileştirir (Hinojosa ve Araya, 2013).

Büyük ölçekli optimizasyon için bir Kooperatif Mikro-Yapay Arı Kolonisi (CMABC) yaklaşımı (Rajasekhar ve Das, 2013)'de sunulmuştur. Bu yaklaşım, kooperatif algoritmaların bölme ve fethetme özelliğini ve Mikro-Yapay Arı Kolonisinin (MABC) düşük hesaplama maliyeti yöntemini birleştirmiştir.

μ DE'nin bir uygulaması olarak, bir global çözüm bulmak için popülasyon büyüklüğü beş olan bir Hibrit Diferansiyel Evrim (HDE) kullanılmıştır (Tsai ve Wang, 2005). μ DE, çöp kutusu probleminin kabul edilebilir bir performansla dolaylı bir temsilini geliştirmek için de kullanılır (Sotelo-Figueroa ve ark., 2013).

Modüler Sinir Ağları (MNNs) optimizasyonu için μ GA yaklaşımına dayanan Hiyerarşik GA (MOHGA) için bir MOO modeli (Sanchez ve ark., 2013)'te önerilmiştir. Bu yaklaşım iris tanıma uygulaması için kullanılır. MOHGA, giriş verilerini granüllere ve alt modüllere böler ve daha sonra verileri işleme ve test aşamalarını bölmeye karar

verir. Bu tekniğin daha az veri kullanmaya dayalı iyi sonuçlar elde edebileceği bildirilmektedir (Sanchez ve ark., 2013)

Lahoz ve ark., (2013)'de, eğitim aşamasının Ortalama Kare Hatasını (MSE) en aza indiren problemin çözümü için makinede uygun sayıda gizli düğüm sağlayan çok amaçlı bir mikro genetik aşırı öğrenme makinesi önerilmiştir. μ GA, dielektrik merceklerle (Itoh ve ark., 2012) dalga kılavuzu yuvası anteni tasarımı, kusurlu kompozitlerin tespiti (Xu ve Liu, 2002) ve daha iyi performansların karşılaştırıldığı gerçek dünya boru hattı ağının (Ribas ve ark., 2013) programlanması gibi birçok uygulama için başarıyla uygulandığı standart GA ile karşılaştırılarak rapor edilmiştir.

3.3.7. Parametre Uyarlaması

Mikro popülasyona dayalı algoritmaların uyarlanması, bu tür algoritmaların performansının arttırılması için umut vaat eden yaklaşımlardan biridir. Literatürde DE algoritmasının sağlamlığını ve güvenilirliğini arttırmak için uyarlamalı veya kendinden uyarlamalı yaklaşımlarla birçok yöntem önerilmiştir (Neri ve Tirronen, 2010; Neri ve Mininno, 2010; Mininno ve ark., 2011). Bu özellikle hiper-parametrelerin ayarlanması için önemlidir. Mutasyon faktörü, genellikle sabit bir değere ayarlanan parametrelerden biridir (Segura ve ark., 2015). Bununla birlikte, mutasyon faktörünün randomize edilmesinin, potansiyel olarak yeni bir arama hareketi sunabileceği ve standart bir DE algoritmasının aşırı belirleyici arama yapısını telafi edebileceği gösterilmiştir (Das ve Konar, 2005; Salehinejad ve ark., 2014). Çalışmalar, rastgele mutasyon faktörü oluşturmak için Gaussian, Log-normal ve Cauchy gibi çeşitli dağılımlar kullanmıştır. Bununla birlikte, hiçbirisi diğerlerine göre üstün değildir (Segura ve ark., 2015).

Das ve Konar (2005) ve Das ve ark. (2005) çalışmalarında önerilen yöntemler popülasyonun çeşitliliğini arttırmak için her nesilde rastgele mutasyon faktörü kullanır, bunun hem gürültü hem de durağan sorunlar için etkili olduğu bildirilir (Price ve ark., 2006). Bu yöntemler standart popülasyon boyutunu kullanırken, F mutasyon faktörü kullanılır. (0,5, 1) aralığından rastgele seçilen, ortalama değeri 0,75'te kalacak şekilde kontrol edilir. Weber ve ark. (2011)'de dört farklı mutasyon ölçeği faktör şeması önerilmiştir. Popülasyon büyüklüğü $N_P = 200$ olarak ayarlanmıştır. Çalışma, performansın kullanılan dağılım tipine bağlı olduğundan, hiçbir yöntemin tüm problemler için ümit verici sonuçlar gösteremediğini göstermektedir (Weber ve ark., 2011). Brest ve ark. (2008)'de, mutasyon faktörü ve çapraz hız oranı parametreleri için kendinden uyarlanmış bir DE algoritması sunulmaktadır. Deneylerde kullanılan en küçük

popülasyon büyüklüğü $N_p=25$ 'tir. (Brest ve ark., 2009)'da mutasyon faktörü (F)'yi ve kuşaklar arası geçiş hızını (C_r) değiştirmek için kendinden uyarlamalı bir kontrol mekanizması kullanılır. Bu yöntemde, yaşlanma mekanizmasına sahip çok popülasyonlu bir yöntem için sadece “rand/1/bin” mutasyon vektörü kullanılır. DE yöntemi, standart popülasyon büyüklüğündeki her bir birey için belirli bir oranla F üretildiği için umut vaat eden yöntemlerden biridir (Brest ve ark., 2006). (Salehinejad ve ark., 2014)'de en düşük düzeyde rastgele F üretme fikri (her popülasyon birey için ve sorunun boyutu için). Bu teknik standart MDE algoritmalarının arama performansını arttırmak için kullanılır. Bu yöntem, sonuçların üstün keşif performansı gösterdiği CEC-2013 yarışması için bir dizi karşılaştırmalı işlev üzerinde değerlendirilir. Vektörize Edilmiş Rastgele Mutasyon Faktörü ile Mikro Diferansiyel Evrim (MDEV) olarak adlandırılan bu algoritma, mJADE algoritmasında önerilen “by-rand-to-best” adı verilen yeni bir mutasyon faktörünün performansını karşılaştırmak için bir ölçüt olarak kullanılmaktadır (Brown ve ark., 2015). Bu yaklaşım kısıtlanmamış optimizasyon problemleri için bir DE algoritmasıdır, dikkate alınan en küçük popülasyon büyüklüğü $N_p = 8$ 'dir (Brown ve ark., 2015). Mutasyon faktörü (F) ve geçme hızı (C_r), her neslin başlangıcında rastgele oluşturulur, buradaki dağılım ortalaması her nesilde güncellenir. (Brown ve ark., 2015)'de akla gelebilecek en iyiye kadar önerilen önerilen mutasyon faktörü, 13 klasik kıyaslama fonksiyonunun bir takımında hem büyük hem de küçük popülasyon boyutları için test edilmiştir. (Salehinejad ve ark., 2014)'deki karşılaştırmalı sonuçlar MDEV ve Mikro Adaptif Diferansiyel Evrim (μ JADE) algoritmaları arasında rekabetçi bir performans göstermektedir. Kompakt Diferansiyel Evrim (cDE) metotları, problemi çözme boyutuna bakılmaksızın, dört kişiden oluşan popülasyonlara benzer bellek gereksinimine sahip nüfusun istatistiksel bir sunumunu kullanır (Mininno ve ark., 2011; Brown ve ark., 2015). Bu çalışmada, sanal olmayan küçük toplulukları tartışmaya odaklandığımız için, bu DE algoritmaları sınıfı diğer çalışmalarda daha fazla araştırma yapmak için ayrılmıştır.

Çevresel ekonomik sevk vaka çalışması için (Pandit ve ark., 2011)'de zamana göre değişen kemotaktik adım boyutuna sahip bir Kaotik Mikro Bakteriyel Besin Arama Algoritması (CMBFOA) önerilmiştir. Bu yöntemin yakınsama karakteristiği, hızı ve çözüm kalitesinin, 3 üniteli bir sistem ve standart Elektrik ve Elektronik Mühendisleri Enstitüsü (IEEE) 30 veri yolu test sistemi için klasik BFOA'dan daha iyi olduğu bildirilmektedir.

Seçkin Elitist Evrim (EEv) adı verilen diğer bir EA türü, Hessian veya kovaryans matrisi gibi karmaşık mekanizmalar kullanmadan çalışan (Viveros-Jimenez ve ark.,

2009)'da yüksek boyutlu problemleri optimize etmek için önerilmiştir. Bu yaklaşım, tek bir uyarlamalı parametrenin evrimsel operatörleri makul yerel ve global arama yetenekleri sağlamak için kontrol ettiği adaptif ve seçkin davranışlardan yararlanır (Viveros-Jimenez ve ark., 2009).

Coulomb yasası, μ PSO yönteminde yüksek boyutlu problemler için kullanılmaktadır (Huang ve Mohan, 2006). Bu yaklaşımın ilk başarısı, kara listeye alınmış çözeltileri içine almak için gereken uygun alan büyüklüğünü belirlemek için gereken yükü ve parçacıkları püskürtmek için gereken itme miktarını ortadan kaldırmaktır, çünkü bu parametrelerin yüksek boyutlu problemleri tespit etmek oldukça zordur. Diğer bir başarı, parçacıklar üzerinde itilmeyi, belirli bir pozisyonda parçacıklar tarafından yaşanan itme miktarını kontrol eden bir parametre kullanarak kontrol etme esnekliğidir. Beş yüksek boyutlu kıyaslama fonksiyonundaki simülasyon sonuçları, geniş popülasyon büyüklüğüne sahip standart PSO'ya karşı μ -PSO'nun üstün performansını göstermektedir.

3.4. Yapay Alg Algoritması (AAA)

AAA, yapay alglerin davranışlarından esinlenilerek 2015 yılında ortaya konmuş doğa esinli bir optimizasyon algoritmasıdır. Yapay alg, gerçek yaşamdaki bir alg gibi, fotosentez yapabilmek için ışık kaynağına doğru hareket eder ve hareketi helisel yüzme şeklindedir. Ortama uyum sağlayıp, baskın türü değiştirebilir ve mitoz bölünme ile çoğalabilir. Problem uzayındaki her bir çözüme bir yapay alg kolonisi karşılık gelmektedir. Her bir alg kolonisindeki alg hücresi sayısı, problem boyutuna eşittir. Bir alg kolonisi ideal çözüme ulaştığında optimum elde edilmiş olur. Yapay alg algoritması 3 temel adımdan oluşur. Bunlar; helisel hareket, adaptasyon ve evrimsel süreçtir. (Uymaz ve ark., 2015). AAA'ya ait akış diyagramı Şekil 3.1.'de verilmiştir (Uymaz, 2015).

3.4.1. Helisel Hareket

Yapay alg hücreleri, ışığa doğru helisel olarak hareket etmektedir. Her bir helisel hareket sonucunda koloninin uzayda yer değiştirip değiştirmeyeceğini enerjileri belirler. Her döngünün başlangıcında, enerji koloni büyüklüğüne doğru orantılı olarak hesaplanır ve bu enerji çözümün kalitesini göstermektedir. Alglerin bu hareketinin birinci boyuttaki hareketi Denklem 3.7'de diğer 2 boyuttaki hareketleri ise Denklem 3.8 ve Denklem 3.9'te gösterilmiştir (Zhang ve ark., 2016).

$$x_{im}^{t+1} = x_{im}^t + (x_{jm}^t - x_{im}^t)(sf - \omega_i)p \quad (3.7)$$

$$x_{ik}^{t+1} = x_{ik}^t + (x_{jk}^t - x_{ik}^t)(sf - \omega_i) \cos \alpha \quad (3.8)$$

$$x_{il}^{t+1} = x_{il}^t + (x_{jl}^t - x_{il}^t)(sf - \omega_i) \sin \beta \quad (3.9)$$

Burada, m, k ve $l \in [1, d]$ arasından seçilen rastgele sayılardır. x_{im}, x_{ik} ve x_{il} , i . alg kolonisinin sırasıyla x, y ve z koordinatlarını temsil eder. j , turnuva seçimi ile elde edilen komşu bir alg kolonisinin indeksidir. p , $[-1, 1]$ aralığında seçilen reel bir sayıdır. α ve β , $[0, 2\pi]$ aralığında rastgele seçilen derecelerdir. sf , viskoz hareketten kaynaklanan kesme kuvvetidir. ω_i , alg kolonisinin büyüklüğü ile orantılı olan alg kolonisinin sürtünme yüzey alanıdır. Sürtünme yüzeyi, alg kolonisinin küresel şeklinden dolayı alg kolonisini saran yarımkürenin yüzey alanı olarak hesaplanır. Denklem 3.11'de ise sürtünme yüzeyi verilmiştir.

$$\omega_i = 2\pi r_i^2 \quad (3.10)$$

$$r_i = \sqrt[3]{\frac{3S_i}{4\pi}} \quad (3.11)$$

Burada ise, r_i , i . alg kolonisine ait yarımkürenin yarıçapı ve S_i 'de büyüklüğüdür.

3.4.2. Adaptasyon

Adaptasyon, yeterince büyüemeyen bir alg kolonisi tarafından çevrede bulunan en büyük alg kolonisine benzemeye çalıştığı süreçtir. Helisel hareket sonucunda belirlenen açlık seviyesi kullanılır. Daha iyi çözüme giden kolonide açlık seviyesi değişmezken, kötüleşen koloninin açlık seviyesi artar. Her bir helisel hareketten sonra, en yüksek açlık değerine sahip olan koloni adaptasyona uğrar (Denklem 3.12-13). Adaptasyonun olup olmayacağı Adaptasyon parametresi (Ap) ile belirlenir. $Ap \in [0, 1]$ aralığında sabit bir değerdir ve bu aralıkta rastgele bir sayı ile kıyaslanır eğer sayı Ap parametresinden küçük ise adaptasyon süreci işletilir (Zhang ve ark., 2016).

$$starving^t = \arg \max \{starvation(x_i)\}, \quad i = 1, 2, \dots, N \quad (3.12)$$

$$starving^{t+1} = starving^t + (biggest^t - starving^t) \times \text{rand} \quad (3.13)$$

Burada, $starvation(x_i)$ i . alg kolonisinin açlık seviyelerini, $starving^t$ t anındaki açlık değeri en yüksek alg kolonisini, $biggest^t$ t anındaki en büyük alg kolonisini ve $\text{rand} \in [0, 1]$ aralığında rastgele üretilen reel bir sayıyı ifade eder.

3.4.3. Evrimsel Süreç

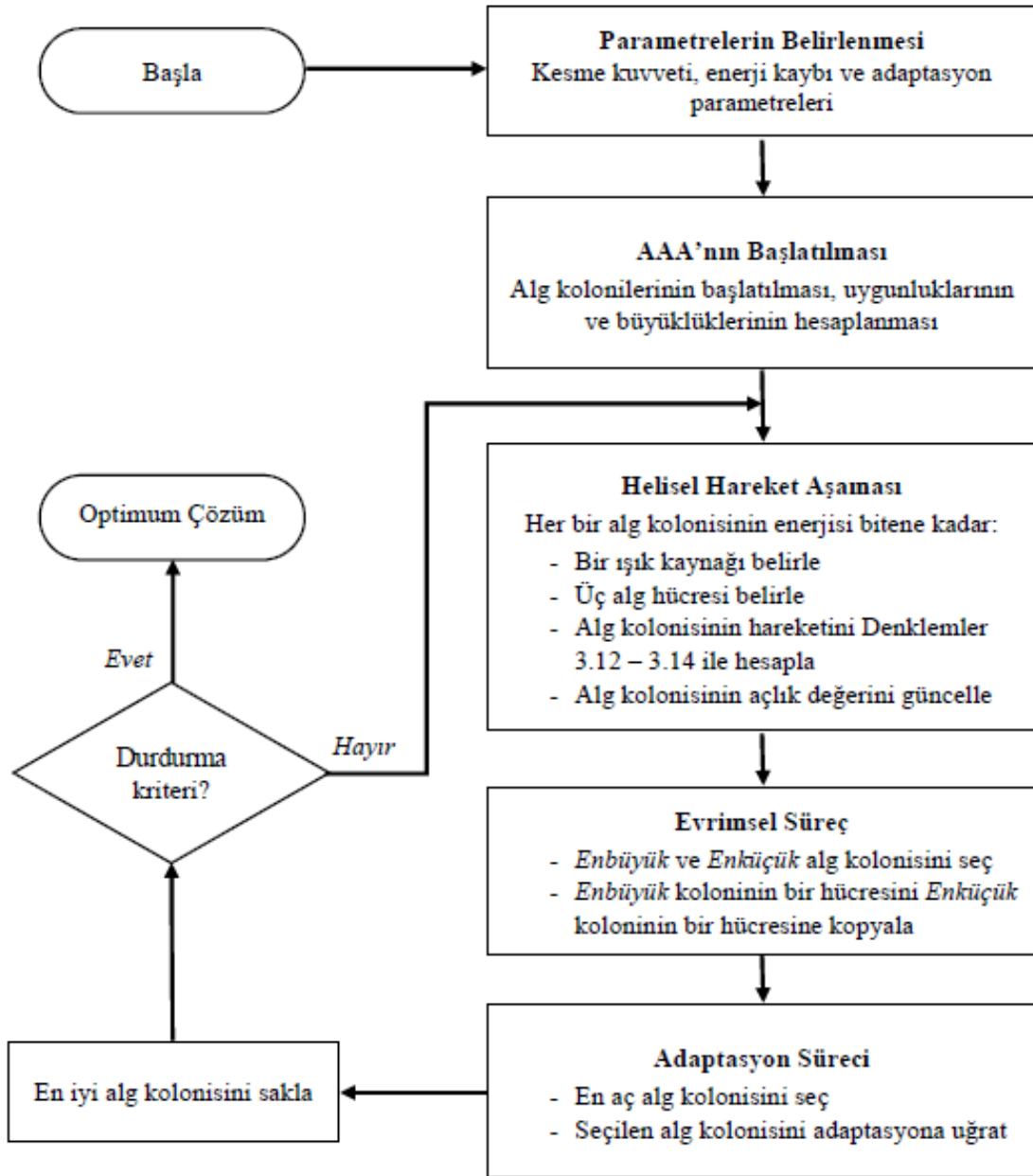
Yapay alg hücresi yeterli ışık aldığında büyür, gelişir ve bölünme ile iki yapay alg hücresi oluşur. Yeterli ışık alamayan alg hücresi ise bir süre sonra ölür. Evrimsel süreç; arama sürecinde en küçük alg kolonisinin (bulduğu çözümlerin uygunluk fonksiyon değerleri diğer kolonilerden daha kötü olan) ölen her bir hücresi yerine en büyük alg kolonisinin (bulduğu çözümlerin uygunluk fonksiyon değerleri diğer kolonilerden daha iyi olan) bir alg hücresinin kopyalandığı aşamadır. Bu süreç Denklem 3.14 - 3.16'daki gibi işletilir.

$$biggest^t = \arg \max size(x_i^t), \quad i = 1, 2, \dots, N \quad (3.14)$$

$$smallest^t = \arg \max size(x_i^t), \quad i = 1, 2, \dots, N \quad (3.15)$$

$$smallest_m^{t+1} = biggest_m^{t+1}, \quad m = 1, 2, \dots, D \quad (3.16)$$

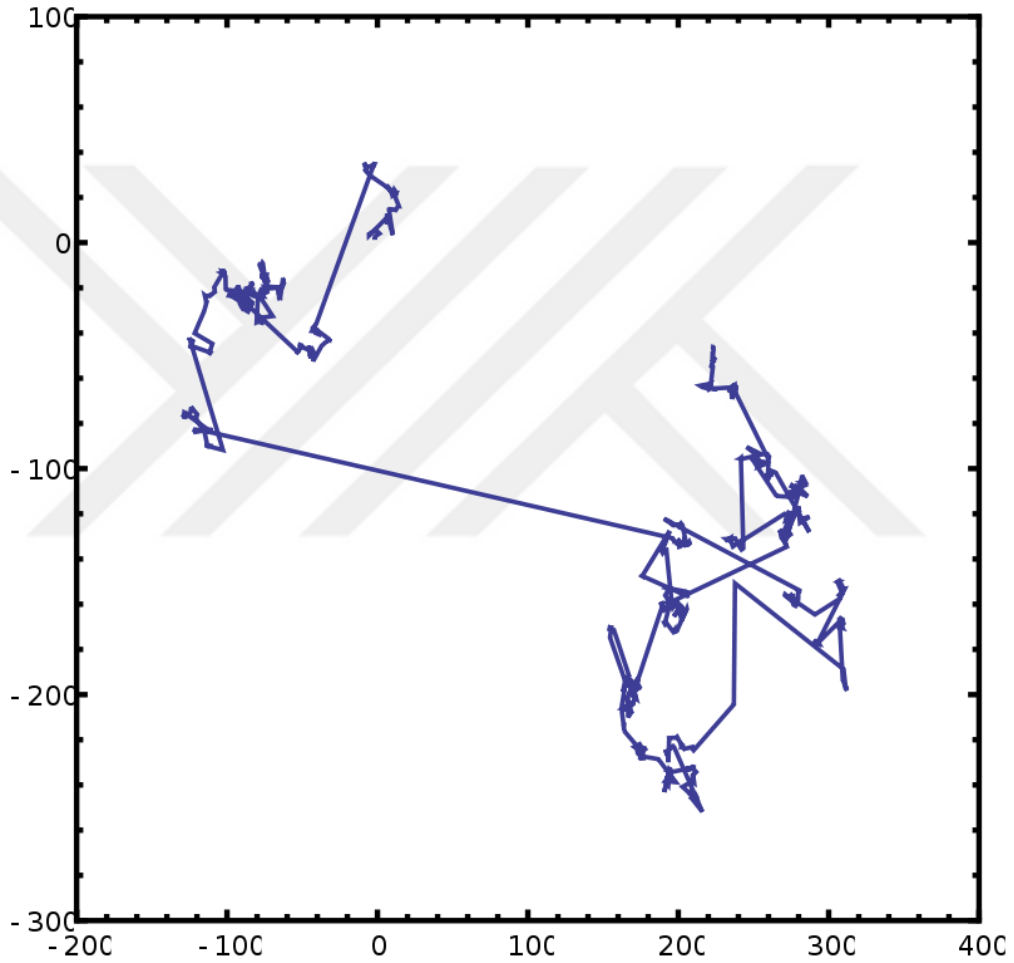
Burada, *smallest* en küçük alg kolonisi, *biggest* en büyük alg kolonisi ve *D* problem boyutunu ifade eder.



Şekil 3.1. AAA Akış Diyagramı (Uymaz ve ark., 2015)

3.5. Levy Uçuşu

Levy Uçuş algoritması, optimize edilmesi gereken problemleri çözmek için kullanılan bir metasezgisel algoritmadır. Bu algoritma, Benoît Mandelbrot'ın ortaya sunduğu doğada gözlemlenen bir fenomen olan "Levy Uçuşu"ndan esinlenir (Mandelbrot, 1982). Levy Uçuşu, rastgele hareket eden bir organizmanın bazen uzun mesafeler kat etmesini, bazen ise kısa mesafelerde kalmasını tanımlar. Şekil 3.2'de 2 Boyutlu düzlemde 1000 adıma sahip Levy Uçuş örneği verilmiştir.



Şekil 3.2 Boyutlu düzlemde 1000 adıma sahip Levy Uçuş Örneği

Levy Uçuş algoritması, çözüm adaylarının çözüm alanında rastgele hareket ettiği bir arama stratejisini kullanır. Bu hareketler, Levy dağılımına göre belirlenir. Levy dağılımı, uzun mesafelerdeki hareketlerin daha olası olduğu bir olasılık dağılımıdır.

Algoritmanın genel işleyişi şu adımlardan oluşur:

Başlangıç Popülasyonu Oluşturma: Algoritma, çözüm adaylarını temsil eden bir başlangıç popülasyonu oluşturur. Bu popülasyon, çözüm alanında rastgele noktalarda yerleştirilen bireylerden oluşur.

Yeni Çözüm Adayları Üretme: Her bir birey, Levy dağılımına göre belirlenen bir adım boyutuyla çözüm alanında rastgele bir yöne hareket eder. Bu hareket sonucunda yeni bir çözüm adayı üretilir.

Uygunluk Değerlerinin Hesaplanması: Üretilen yeni çözüm adaylarının uygunluk değerleri, optimize edilmek istenen hedef fonksiyonuna göre hesaplanır.

En İyi Çözümün Güncellenmesi: Elde edilen yeni çözüm adayları arasından en iyi uygunluğa sahip olan çözüm adayı, en iyi çözüm olarak seçilir ve kaydedilir.

Durdurma Kriterine Ulaşıncaya Kadar Yineleme: Belirlenen bir durdurma kriteri sağlanana kadar adımlar 2-4 tekrarlanır. Durdurma kriteri genellikle belirli bir iterasyon sayısı veya belirli bir uygunluk değeri elde edildiğinde sağlanır. Çizelge 3.1’de algoritmaya ait pseudo kodu paylaşılmıştır.

Çizelge 3.1. Levy Uçuşu Psuedo Kodu

Levy Uçuşu Pseudo Kodu
Input function $Min f(X)$ and β
Select x_i in swarm that will modify the position
Initialize $\tau = 1$ and $\sigma_h = 1$
While ($\tau < \epsilon$)
Compute step_size
Generate new solution X_i
Calculate $f(X_i)$
If $f(X'_i) < f(X_i)$ then
$X_i = X'_i$
End If
$\tau = \tau + 1$
End

Levy Uçuş algoritması, özellikle karmaşık ve çok boyutlu problemlerin çözümünde etkilidir. Levy dağılımının uzun mesafelerdeki hareketleri teşvik etmesi,

algoritmanın çözüm alanında daha geniş bir keşif yapmasını sağlar. Buna karşın, algoritma lokal optimumlarda sıkışma riski taşır.

Levy Uçuş algoritması, farklı parametre ayarları ve problem tipleri için değişebilirlik gösterebilir. Bu nedenle, algoritmanın performansını optimize etmek için parametrelerin dikkatli bir şekilde ayarlanması önemlidir.

Levy Uçuş algoritması, optimizasyon problemlerinin çözümünde kullanılan güçlü bir araç olmasına rağmen, her problem için en uygun algoritmayı belirlemek için problemi ve gereksinimleri iyi anlamak önemlidir.

3.6. Mikro Algoritma Üzerine İlk Çalışma: Mikro Yapay Alg Algoritması (μ AAA)

Mikro Algoritma geliştirme sürecine başlarken yapılan literatür çalışmaları göz önüne alındığında öncelikle popülasyon sayısını en aza indirmek temel hedef olmuştur. Mikro Algoritma geliştirirken ilk adım olarak popülasyon sayısı, literatürde yoğun olarak kullanılan, 3 birey olarak düşünülmüştür. Bununla birlikte popülasyondaki birey sayısı aza indirildiğinde olabildiğince tüm uzayı gezebilecek bir yapı geliştirmek hedeflenmiştir.

Yapısal olarak uzayı gezdiğini belirten Dasgupta ve arkadaşlarının (2009) çalışmalarında sunmuş oldukları μ BFOA örnek alınmış olup öncelikle Yapay Alg Algoritmasına ait olan helisel hareket, evrimsel süreç ve adaptasyon adımları işletilerek elde edilen popülasyonda gerekli mikro adımları bir sonraki adımda işlenmiş ve Yapay Alg Algoritmasına ait olan parametreler Çizelge 3.1’de verilmiştir.

Geliştirilen algorithmada ilk olarak elde edilen popülasyondan en iyi 1. birey korunarak devam ettirilmiştir. Popülasyonda sonraki birey olan 2. Birey içinse bir Epsilon (ϵ) değeri ile 1. Bireye yaklaştırılmaya çalışılmıştır (Dasgupta ve ark, 2009). Burada ϵ bir uzaklık parametresi olarak tanımlanır ve Denklem 3.17’de aralık (range) çıkarım denklemi verilmiş olup, Denklem 3.18’de ise ϵ ’a ait hesaplama gösterilmiştir. Son birey olan 3. Birey ise uzaydan rastgele değerlerle oluşturularak çeşitlilik oluşması sağlanması amaçlanmıştır.

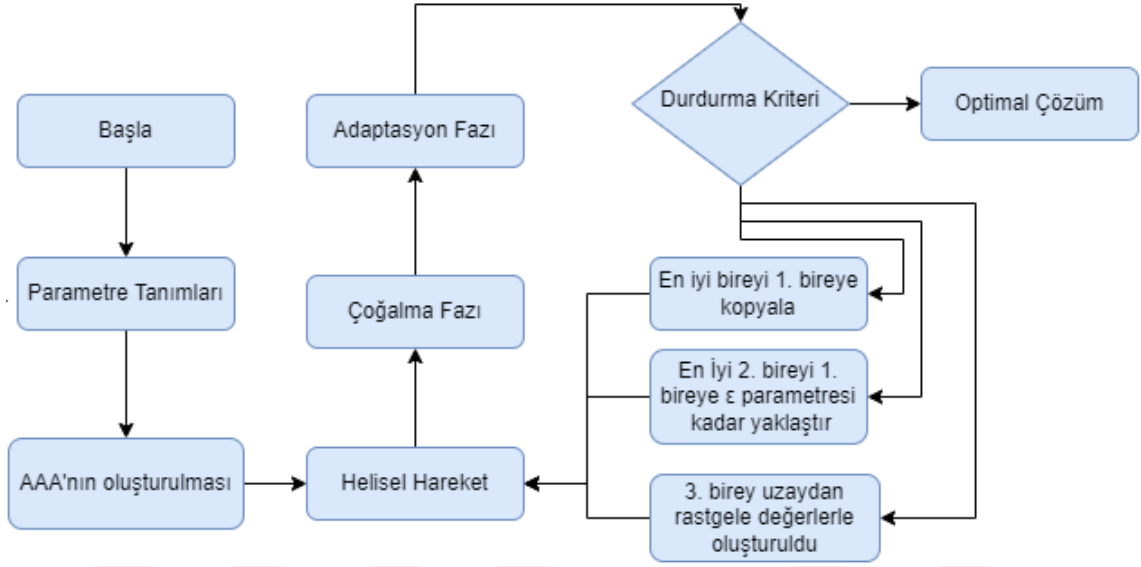
Önerilen μ AAA’ya ait akış diyagramı Şekil 3.3’de verilmiştir.

$$range = max - min \quad (3.17)$$

$$\epsilon = 0.06 \times range \quad (3.18)$$

Çizelge 3.1. AAA ve μ AAA'ya ait parametreler

Parametreler	Değerler (μ AAA)
Boyut	500
Kesme Kuvveti	2
Enerji Kaybı	0.3
A_p	0.5

Şekil 3.3. μ AAA Akış Diyagramı

3.7. Önerilen Yöntem: Levy ile Mikro Yapay Alg Algoritması (μ AAA_{Levy})

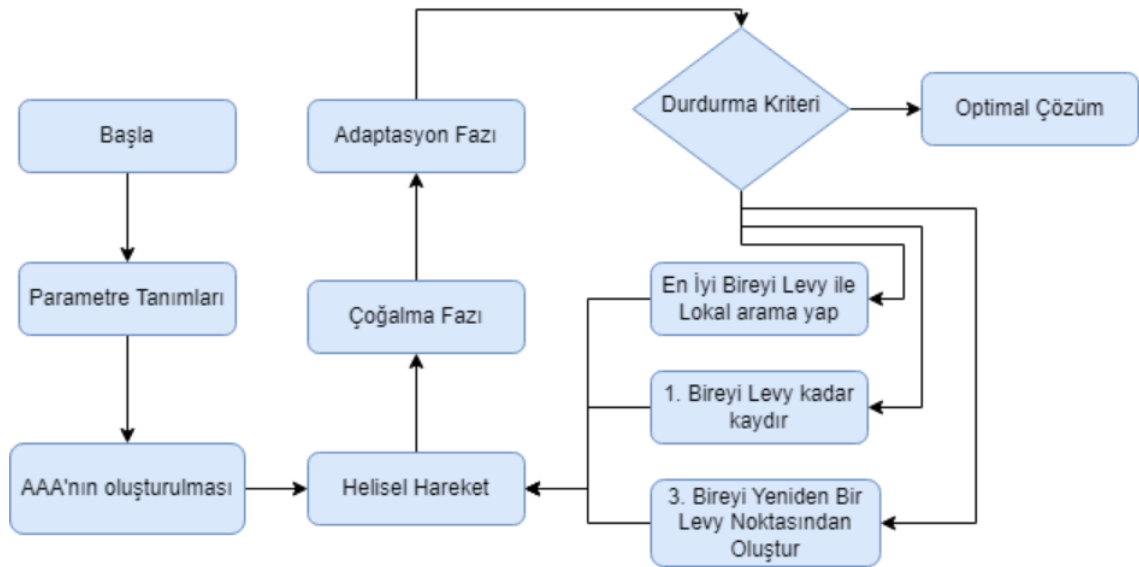
Bu çalışmada öncelikle popülasyon sayısı, herhangi bir mikro algoritmalarındaki gibi birey sayısı az tutulacak şekilde ve bir önceki çalışma olan μ AAA'da olduğu gibi 3 birey olarak kullanılmıştır. μ AAA'nın helisel hareket, evrimsel süreç ve adaptasyon adımları tamamlandıktan sonra oluşan popülasyonun en iyi sonuca sahip bireyi bir sonraki adımın ilk aday bireyi olmak üzere saklanmaktadır. Seçilen bu en iyi bireyden Levy Uçuşu yardımı ile yeni bireyler üretilerek Lokal Arama işlemi yapılmaktadır (Denklem 3.19). Lokal arama sürecinde yeni en iyi birey problemin boyut sayısı kadar Levy Uçuşu ile genişleyip daraltılmaktadır. Birey daha iyi çözüme ulaştığında konumu güncellenmektedir ve yeni Levy Uçuşu bu konumdan gerçekleşmektedir.

$$x_m^{t+1} = x_m^t \pm Levy(\beta) \quad (3.19)$$

Bir sonraki popülasyonun 2. bireyi için ise 1. birey yine Levy Uçuşu yardımı ile yeniden oluşturulur (Denklemler 3.20). Yeni popülasyonun ilk iki bireyi lokal aramayı geliştirmek amacı ile bir önceki popülasyondaki en iyi bireyden faydalanılarak oluşturulmaktadır.

$$x_m^{t+1} = x_m^t + Levy(\beta) \quad (3.20)$$

Algoritmanın global arama yeteneğini geliştirmek ve hızlı yakınsamanın önüne geçmek amacı ile 3. birey, Levy Uçuşu metodu ile rastgele olarak uzay sınırlarına oranlanarak üretilir. Algoritma her çevrimde helisel hareket, evrimsel süreç, adaptasyon süreci ve Levy Uçuşu ile popülasyon güncelleme adımlarını durdurma kriterini sağlayana kadar tekrarlar. Önerilen yeni Levy Uçuşu ile Yapay Alg Algoritmasına ait akış diyagramı Şekil 3.4’de verilmiştir.



Şekil 3.4. μAAA_{levy} Akış Diyagramı

4. ARAŞTIRMA BULGULARI VE TARTIŞMA

Tez kapsamında önerilen yöntemin performansını doğrulamak ve kıyaslamak için yapılan çalışmalar iki aşamada gerçekleştirilmiştir. İlk olarak önerilen μ AAA yöntemi 6 test fonksiyonu (Rastrigin, Griewank, Ackley, Rosenbrock, Schwefel ve Schwefel v2) üzerinde çalıştırıldı ve literatürde önerilmiş mikro algoritmalar olan μ PSO, μ BFOA yöntemleri ile AAA ve AAA'nın popülasyonu indirgenmiş (3 birey) hali ile kıyaslandı.

İkinci aşamada, önerilen μ AAA_{levy} yöntemi CEC2015 fonksiyon seti üzerinde çalıştırıldı. Sonuçlar tüm parametreleri yöntemin önerildiği Uymaz ve ark. (2016) çalışmasındaki değerler ile aynı bırakılarak AAA (40 koloni) ve AAA'nın popülasyonu indirgenmiş (3 koloni) hali ile kıyaslandı.

4.1. Temel Test Fonksiyonları

Yapılan Çalışmaların test edileceği literatürde yaygın olarak kullanılan fonksiyonlar Çizelge 4.1.'de verilmiştir. Bu test fonksiyonlarından F1, F2, F3, F4 ve F6 ise Huang ve Mohan'ın (2006) yılında yine yüksek boyutlu problemler için geliştirdiği μ PSO araştırmasında kullanılmıştır. F1, F2, F3, F4 ve F5 fonksiyonları ise Dasgupta ve arkadaşlarının 2009 yılında yine yüksek boyutlu problemler için ortaya koyduğu μ BFOA araştırmasında kullanılmıştır. Çizelge 4.1. de test fonksiyonlarına ait isimleri, matematiksel denklemleri, çözüm uzaylarının alt ve üst sınırları ve global optimum değerleri verilmiştir. Schwefel ve Schwefel v2 test fonksiyonları diğer fonksiyonlara göre daha geniş aralıkta bir çözüm uzayına sahiptir.

Çizelge 4.1. Karşılaştırma için kullanılan fonksiyonlar

No.	Fonksiyon Adı	Matematiksel Gösterim	Arama Uzayı	Teorik Minimum
F1	Rastrigin	$\sum_{n=1}^N [x_n^2 - 10 \cos(2\pi x_n) + 10]$	[-2,2]	0
F2	Griewank	$1 + \frac{1}{4000} \sum_{n=1}^N x_n^2 - \prod_{n=1}^N \cos\left(\frac{x_n}{\sqrt{n}}\right)$	[-10,10]	0
F3	Ackley	$20 + e - 20 \exp\left(-0.2 \sqrt{\frac{1}{N} \sum_{n=1}^N x_n^2}\right) - \exp\left(\frac{1}{N} \sum_{n=1}^N \cos(2\pi x_n)\right)$	[-2,2]	0
F4	Rosenbrock	$\sum_{n=1}^{N-1} [100(x_{n+1} - x_n^2)^2 + (x_n - 1)^2]$	[-2,2]	0
F5	Schwefel	$-\sum_{n=1}^N x_n \sin\left(\sqrt{ x_n }\right)$	[-500,500]	-416.99x500
F6	Schwefel v2	$418.9829n + \sum_{n=1}^N x_n \sin\left(\sqrt{ x_n }\right)$	[-500,500]	0

4.2. CEC2015 Fonksiyonları

Tez kapsamında yapılan deneysel çalışmaların ikinci aşamasında bir diğer test seti ise CEC 2015 konferansında optimizasyon yöntemlerinin kıyaslanması için sunulan ve fonksiyonların döndürülmüş ve kaydırılmış versiyonlarını barındıran fonksiyon setidir. Bu test seti tekli ve çoklu global optimum değere sahip ve global optimum değerleri sıfırdan farklı fonksiyonlardan oluşmaktadır. Test seti, Chen ve arkadaşları (2014) tarafından yapılan çalışmadan alınmıştır. Test setine ait fonksiyonlar ve özellikleri Çizelge 4.2’de verilmiştir.

Çizelge 4.2. CEC2015 Fonksiyonları

Kategori	No	Fonksiyonlar	İlgili Temel Fonksiyonlar	Teorik Minimum
Tek Tepeli Fonksiyonlar (Unimodal Functions)	1	Rotated Bent Cigar	Bent Cigar	100
	2	Rotated Discus	Discus	200
	3	Shifted and Rotated Weierstrass	Weierstrass	300
	4	Shifted and Rotated Schwefel's	Schwefel's	400
	5	Shifted and Rotated Katsuura	Katsuura	500
Basit Çok Tepeli Fonksiyonlar (Simple Multimodal Functions)	6	Shifted and Rotated HappyCat	HappyCat	600
	7	Shifted and Rotated HGBat	HGBat	700
	8	Shifted and Rotated Expanded Griewank's plus Rosenbrock's	Griewank's Rosenbrock's	800
	9	Shifted and Rotated Expanded Scaffer's F6	Expanded Scaffer's F6	900

4.3. Karşılaştırmalar

4.3.1. Temel Test Fonksiyonu Karşılaştırmaları

Önerilen μ AAA ilk olarak popülasyondaki koloni sayısı 3 ve 40 olarak alınan temel AAA algoritması ile Çizelge 4.1'de verilen test fonksiyonları üzerinde test edilmiştir. Kıyaslama sonuçları 10 bağımsız çalıştırma üzerinden hesaplanmıştır. Her bağımsız çalıştırma 5000 fonksiyon hesaplama ve 500 boyutta gerçekleştirilmiştir (Çizelge 4.3.).

Çizelge 4.3. μ AAA, AAA (pop:40) ve AAA (pop:3) test fonksiyonları karşılaştırması

Fonksiyonlar		μ AAA	AAA (Pop:40)	AAA (Pop:3)
F1	En iyi	0,0000e+00	3,6080e+03	3,7219e+03
	Ortalama	1,1771e+03	3,7535e+03	4,6558e+03
	Std. Sapma	8,2884e+02	7,5214e+01	4,8873e+02
F2	En iyi	0,0000e+00	2,3505e+00	2,9789e+00
	Ortalama	0,0000e+00	2,5794e+00	3,2286e+00
	Std. Sapma	0,0000e+00	1,1466e-01	2,2556e-01
F3	En iyi	7,9936e-15	4,1126e+00	4,6372e+00
	Ortalama	8,7041e-15	4,3729e+00	4,8335e+00
	Std. Sapma	2,2469e-15	1,7517e-01	1,5033e-01
F4	En iyi	4,9865e+02	4,8545e+04	7,22450e+04
	Ortalama	4,9888e+02	5,8222e+04	1,01348e+05
	Std. Sapma	1,2989e-01	7,2285e+03	1,94408e+04
F5	En iyi	-1,0866e+05	-4,2899e+04	-3,8757e+04
	Ortalama	-9,6893e+04	-4,0837e+04	-1,6940e+04
	Std. Sapma	2,8195e+04	1,7316e+03	9,3647e+03
F6	En iyi	1,17167e+05	2,00768e+05	2,04351e+05
	Ortalama	1,20650e+05	2,01441e+05	2,06593e+05
	Std. Sapma	2,26638e+03	4,04179e+02	1,35221e+03

Çizelge 4.3.'te verilen sonuçlar karşılaştırıldığında tüm test fonksiyonları için en iyi ve ortalama değerlerde μ AAA'nın başarılı sonuçlar verdiği görülmektedir. F1, F5 ve F6 test fonksiyonlarının standart sapmalarının yüksek olduğu gözlenmektedir.

Çizelge 4.4. μ AAA ve μ BFOA test fonksiyonları karşılaştırması

Fonksiyonlar		μ AAA	μ BFOA
F1	En iyi	0,00000e+00	2,15120e+03
	Ortalama	1,17712e+03	2,2850e+03
	Std. Sapma	8,28849e+02	3,65299e+01
F2	En iyi	0,00000e+00	2,5401e+00
	Ortalama	0,00000e+00	2,7074e+00
	Std. Sapma	0,00000e+00	0,0394e+00
F3	En iyi	7,99360e-15	3,5924e+00
	Ortalama	8,70414e-15	3,8175e+00
	Std. Sapma	2,24693e-15	0,1128e+00
F4	En iyi	4,98658e+02	4,57482e+04
	Ortalama	4,98883e+02	4,90355e+04
	Std. Sapma	1,29899e-01	1,7455e+03
F5	En iyi	-1,08662e+05	-9,65585e+04
	Ortalama	-9,68937e+04	-9,30796e+04
	Std. Sapma	2,81954e+04	2,0679e+03

Çizelge 4.4'te verilen μ AAA ve μ BFOA karşılaştırma tablosuna bakıldığında F1 ve F5 test fonksiyonlarına ait standart sapma değerleri dışında kalan değerlerde başarılı olduğu gözlenmektedir.

μ AAA'nın temel test fonksiyonları üzerinde elde ettiği sonuçlar μ PSO (Huang ve Mohan, 2006) yöntemi ile karşılaştırılması Çizelge 4.5'de yapılmıştır. μ PSO 'ya ait sonuçlar Huang ve Mohan (2006) çalışmalarından alınmıştır ve bu çalışmada testler 3000 fonksiyon hesaplama üzerinden gerçekleştirilmiştir (Huang ve Mohan, 2006). Kıyaslamaların adil olması için μ AAA'da diğer parametreler değiştirilmeden 3000 fonksiyon hesaplama ile çalıştırılarak sonuçlar elde edilmiştir.

Çizelge 4.5. μ AAA ve μ PSO test fonksiyonları karşılaştırması

Fonksiyonlar		μ AAA	μ PSO
F1	Ortalama	1,35328e+03	1,60981e+03
	Std. Sapma	1,19304e+03	3,0144e+02
F2	Ortalama	0,00000e+00	1,40e+00
	Std. Sapma	0,00000e+00	0,11e+00
F3	Ortalama	7,99360e-15	4,13e+00
	Std. Sapma	0,00000e+00	0,14e+00
F4	Ortalama	4,98917e+02	1,41658e+04
	Std. Sapma	9,87111e-02	4,56785e+03
F6	Ortalama	1,20650e+05	1,23536e+05
	Std. Sapma	2,26638e+03	5,52960e+03

Çizelge 4.5'te verilen μ AAA ve μ PSO yöntemlerine ait sonuçlar incelendiğinde, (Huang ve Mohan, 2006) yapılan araştırmada en iyi değerler paylaşılmadığından kıyaslamalar ortalama değerler ve standart sapmalar üzerinden gerçekleştirilmiştir. μ AAA ortalama değerlere göre μ PSO yöntemine tüm fonksiyonlarda daha iyi sonuçlar vermektedir. Standart sapma değerleri incelendiğinde F1 test fonksiyonu hariç tüm fonksiyonlarda μ AAA yönteminin daha başarılı olduğu gözlenmektedir.

4.3.2. CEC2015 Test Fonksiyonu Karşılaştırmaları

CEC2015 test fonksiyon setinde bulunan 9 fonksiyon üzerine yapılan deneysel çalışmaların sonuçları Çizelge 4.6'da gösterilmiştir. Kıyaslamalarda kullanılan her yöntem (μ AAA_{levy}, AAA (pop=40) ve AAA (pop=3)) için 10 bağımsız çalıştırmanın En İyi, En Kötü, Ortalama, Orta Değer, Standart Sapma ve Ortalama Zaman değerleri elde edilmiş ve Çizelge 4.6'da sonuçlar verilmiştir.

Çizelge 4.6. $\mu\text{AAA}_{\text{levy}}$, AAA ve AAA (pop:3) CEC2015 test fonksiyonları karşılaştırması

Fonksiyonlar		$\mu\text{AAA}_{\text{levy}}$	AAA (pop:40)	AAA (pop:3)
F1	En İyi	1,9425226038e+08	5,8267577491e+08	6,5789381457e+09
	En Kötü	3,3372508167e+08	2,1033018689e+09	1,6687843978e+10
	Ortalama	3,3372508167e+08	1,2432223775e+09	1,1679441204e+10
	Orta Değer	3,3267564791e+08	1,2395130309e+09	1,1900617893e+10
	Std. Sapma	1,1168664786e+08	4,9126355557e+08	3,2268423404e+09
	Ort. Zaman	1,30200e+00	4,01134e+00	7,99331e+00
F2	En İyi	3,6835847293e+07	1,3149560611e+10	2,9313985342e+11
	En Kötü	6,1593644446e+08	4,3548422196e+10	7,2630938390e+11
	Ortalama	2,3507559486e+08	2,7923885630e+10	4,6510465159e+11
	Orta Değer	1,6498897268e+08	2,7687596485e+10	4,1977576351e+11
	Std. Sapma	1,8550993695e+08	9,0321549768e+09	1,4229097603e+11
	Ort. Zaman	1,05677e+00	3,67262e+00	7,77429e+00
F3	En İyi	3,2000019905e+02	3,2041668684e+02	3,2134497751e+02
	En Kötü	3,2000565977e+02	3,2110077931e+02	3,2144131191e+02
	Ortalama	3,2000019905e+02	3,2060034287e+02	3,2138159860e+02
	Orta Değer	3,2000029584e+02	3,2054968826e+02	3,2137890074e+02
	Std. Sapma	1,6895540700e-03	1,9714973442e-01	2,5819929623e-02
	Ort. Zaman	1,10746e+00	3,83912e+00	8,22581e+00
F4	En İyi	2,0516458441e+03	1,4769188053e+03	2,1757147890e+03
	En Kötü	2,4416240491e+03	1,7216961218e+03	2,8970109759e+03
	Ortalama	2,2473581739e+03	1,6330860019e+03	2,4322075980e+03
	Orta Değer	2,2356798638e+03	1,6238646315e+03	2,3567670186e+03
	Std. Sapma	1,4091458703e+02	7,0953038850e+01	2,4783052323e+02
	Ort. Zaman	1,09886e+00	3,79544e+00	7,91486e+00
F5	En İyi	1,5834659491e+04	1,6732461241e+04	3,1560870148e+04
	En Kötü	1,9848420030e+04	2,8247391267e+04	3,3030553597e+04
	Ortalama	1,7575042919e+04	2,0090900878e+04	3,2338916608e+04
	Orta Değer	1,7201680077e+04	1,9300131387e+04	4,8852445925e+02
	Std. Sapma	1,2571221845e+03	3,0965877473e+03	3,2279855076e+04
	Ort. Zaman	1,63554e+00	4,42920e+00	8,72116e+00
F6	En İyi	1,1032623807e+08	7,1052589834e+07	4,9844400258e+08
	En Kötü	2,7275120065e+08	1,8938530596e+08	1,5936623608e+09
	Ortalama	1,7323914617e+08	1,2608119882e+08	1,0241725640e+09
	Orta Değer	1,7064932643e+08	1,2228694400e+08	9,5547998373e+08
	Std. Sapma	5,1323148524e+07	3,6564347144e+07	3,5209512176e+08
	Ort. Zaman	1,37062e+00	4,05019e+00	8,20835e+00
F7	En İyi	9,0786607631e+02	8,7150751675e+02	3,5629101084e+03
	En Kötü	1,0164564053e+03	9,0814721053e+02	1,0178398085e+04
	Ortalama	9,5346128829e+02	8,8931330211e+02	7,5419406482e+03
	Orta Değer	9,4675938028e+02	8,8821615697e+02	7,5872017369e+03
	Std. Sapma	3,4441825007e+01	1,0301160286e+01	2,3606562316e+03
	Ort. Zaman	6,88511e+00	9,28443e+00	1,45159e+01
F8	En İyi	3,3346506033e+07	2,6866097914e+07	3,7142177462e+08
	En Kötü	1,6622029726e+08	7,4088322793e+07	9,8612118856e+08
	Ortalama	9,2603536733e+07	4,6600243775e+07	6,9427471544e+08
	Orta Değer	8,4610910273e+07	1,7627031668e+07	7,1942733020e+08
	Std. Sapma	4,6468422274e+07	4,6701155440e+07	2,3057803297e+08
	Ort. Zaman	1,29348e+00	3,96265e+00	8,04874e+00
F9	En İyi	1,0132233739e+03	1,0530141563e+03	2,1640540336e+03
	En Kötü	4,0304491899e+03	1,1284959320e+03	3,9204635092e+03
	Ortalama	2,8226692490e+03	1,0823969197e+03	2,9013317089e+03
	Orta Değer	3,2480760440e+03	1,0761775037e+03	2,8741827870e+03
	Std. Sapma	1,1335564239e+03	2,4249781758e+01	4,9093133675e+02
	Ort. Zaman	3,32496e+00	5,93410e+00	1,09636e+0

Çizelge 4.6’da ortalama deęerler incelendięinde μAAA_{levy} ’nin 4 test fonksiyonda, En İyi deęerlere g re ise 5 test fonksiyonunda dięer y ntemlere g re  st nl k saęladığı g r lmektedir.  alıřma zamanları incelendięinde μAAA_{levy} ’nin dięer y ntemlere g re daha hızlı řekilde iřlemine tamamladığı g zlemlenmektedir. Tek tepeli fonksiyonlardaki bařarısı ve hızlı sonu   retmesi a ısından μAAA_{levy} umut vadeden bir Mikro Algoritma yaklařımı olduęu g r lmektedir.



5. SONUÇLAR VE ÖNERİLER

5.1 Sonuçlar

Optimizasyon problemleri doğaları gereği boyutları büyüdükçe karmaşıklıkları artar ve çözümleri için daha fazla hesaplama kaynağına ihtiyaç duyulur. Mikro algoritmalar küçük popülasyonlar sayesinde hesaplama maliyetlerini düşürerek düşük hesaplama gücüne sahip donanımlar üzerinde optimizasyon problemlerine çözüm sağlamayı amaçlamaktadır. Bu tez çalışmasında optimizasyon problemlerine kaliteli çözümler sunan Yapay Alg Algoritması üzerinde geliştirmeler yapılarak Mikro Yapay Alg Algoritması önerilmiştir. Önerilen algoritmanın performansını artırmak için Levy Uçuşu yaklaşımından faydalanılmıştır. Levy Uçuşu bireylerin konum güncellenmesinde ve lokal arama aşamasında kullanılarak çözüm uzayının daha iyi keşfedilmesini ve lokal çözümlerin çeşitliliğini sağlamıştır.

Önerilen algoritmanın performansı öncelikle temel Yapay Alg Algoritmasının birey sayısı 40 ve 3 olarak alınarak kıyaslanmıştır. Sonuçlar, yapılan geliştirmeler sonucunda önerilen mikro AAA yönteminin daha başarılı sonuçlara ulaştığını göstermektedir. Önerilen mikro yöntemin başarısını doğrulamak için literatürde önerilmiş ve başarısı ortaya koyulmuş mikro (μ BFOA ve μ PSO) algoritmalar ile kıyaslanmıştır. Deneysel sonuçlar Levy Uçuşu ile μ AAA algoritmasının kıyaslanan yöntemlere göre daha başarılı sonuçlar verdiğini göstermiştir.

5.2 Öneriler

Yapılan çalışmaların ışığında, gelecek çalışmalarda; literatür araştırması göz önüne alınarak daha karmaşık optimizasyon problemlerinde başarılı performans gerçekleştirmesini sağlamak üzere geliştirilecek mikro çalışmalar yapılması planlanmış ve devam edilmektedir. Düşük hesaplama kapasitesi olan cihazlar üzerinde optimizasyon problemlerinin çözümünde Levy Uçuşu ile μ AAA yöntemi bir alternatif olarak kullanılabilir.

KAYNAKLAR

- Ang, J. H., Goh, C. K., Teoh, E. J. ve Mamun, A. A., 2007, Multi-objective evolutionary recurrent neural networks for system identification, *2007 Ieee Congress on Evolutionary Computation, Vols 1-10, Proceedings*, 1586-1592.
- Coello Coello Coello, C. A., & Toscano Pulido, G. (2001, March). A micro-genetic algorithm for multiobjective optimization. In *International conference on evolutionary multi-criterion optimization* (pp. 126-140). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Bakwad, K. M., Pattnaik, S. S., Sohi, B., Devi, S., Gollapudi, S. V., Sagar, C. V. ve Patra, P., 2011, Fast motion estimation using small population-based modified parallel particle swarm optimisation, *International Journal of Parallel, Emergent and Distributed Systems*, 26 (6), 457-476.
- Blum, C. ve Roli, A., 2003, Metaheuristics in combinatorial optimization: Overview and conceptual comparison, *Acm Computing Surveys*, 35 (3), 268-308.
- Brest, J., Greiner, S., Boskovic, B., Mernik, M. ve Zumer, V., 2006, Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems, *IEEE Transactions on Evolutionary Computation*, 10 (6), 646-657.
- Brest, J. ve Maucec, M. S., 2008, Population size reduction for the differential evolution algorithm, *Applied Intelligence*, 29 (3), 228-247.
- Brest, J., Zamuda, A., Boskovic, B., Maucec, M. S. ve Zumer, V., 2008, High-Dimensional Real-Parameter Optimization using Self-Adaptive Differential Evolution Algorithm with Population Size Reduction, *2008 Ieee Congress on Evolutionary Computation, Vols 1-8*, 2032-2039.
- Brest, J., Zamuda, A., Boskovic, B., Maucec, M. S. ve Zumer, V., 2009, Dynamic Optimization using Self-Adaptive Differential Evolution, *2009 Ieee Congress on Evolutionary Computation, Vols 1-5*, 415-422.
- Brown, C. L., Smith, K., Wall, D. M. ve Walker, D., 2015, Activity of Species-specific Antibiotics Against Crohn's Disease-Associated Adherent-invasive Escherichia coli, *Inflammatory bowel diseases*, 21 (10), 2372-2382.
- Cabrera, J. C. F. ve Coello, C. A. C., 2007, Handling constraints in particle swarm optimization using a small population size, *Micai 2007: Advances in Artificial Intelligence*, 4827, 41-+.
- Cabrera, J. C. F. ve Coello, C. A. C., 2010, Micro-MOPSO: A multi-objective particle swarm optimizer that uses a very small population size, In: *Multi-objective swarm intelligent systems*, Eds: Springer, p. 83-104.

- Caraffini, F., Neri, F. ve Poikolainen, I., 2013, Micro-Differential Evolution with Extra Moves Along the Axes, *Proceedings of the 2013 Ieee Symposium on Differential Evolution (Sde)*.
- Coello, C. A. ve Pulido, G. T., 2001, Multiobjective optimization using a micro-genetic algorithm, *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation*, 274-282.
- Das, S. ve Konar, A., 2005, An improved differential evolution scheme for noisy optimization problems, *Pattern Recognition and Machine Intelligence, Proceedings*, 3776, 417-421.
- Das, S., Konar, A. ve Chakraborty, U. K., 2005, Improved differential evolution algorithms for handling noisy optimization problems, *2005 Ieee Congress on Evolutionary Computation, Vols 1-3, Proceedings*, 1691-1698.
- Das, S., Abraham, A. ve Konar, A., 2008a, Automatic kernel clustering with a Multi-Elitist Particle Swarm Optimization Algorithm, *Pattern Recognition Letters*, 29 (5), 688-699.
- Das, T. K., Jetti, S. R. ve Venayagamoorthy, G. K., 2006, Optimal design of SVC damping controllers with wide area measurements using small population based PSO, *2006 Ieee International Joint Conference on Neural Network Proceedings, Vols 1-10*, 2255-+.
- Das, T. K., Venayagamoorthy, G. K. ve Aliyu, U. O., 2008b, Bio-inspired algorithms for the design of multiple optimal power system stabilizers: SPPSO and BFA, *IEEE Transactions on Industry Applications*, 44 (5), 1445-1457.
- Dasgupta, S., Biswas, A., Das, S., Panigrahi, B. K. ve Abraham, A., 2009, A Micro-Bacterial Foraging Algorithm for High-Dimensional Optimization, *2009 Ieee Congress on Evolutionary Computation, Vols 1-5*, 785-+.
- Fajfar, I., Tuma, T., Puhari, J., Olensek, J. ve Burmen, A., 2012, Towards Smaller Populations in Differential Evolution, *Informacije Midem-Journal of Microelectronics Electronic Components and Materials*, 42 (3), 152-163.
- Glover, F., 1986, Future paths for integer programming and links to artificial intelligence, *Computers & operations research*, 13 (5), 533-549.
- Goldberg, D. E., 1989, Sizing Populations for Serial and Parallel Genetic Algorithms, *Proceedings of the Third International Conference on Genetic Algorithms*, 70-79.
- Han, W. H. ve Sun, J. P., 2012, Improved MICROPSON Algorithm and Its Application on Reactive Power Optimization, *2012 Asia-Pacific Power and Energy Engineering Conference (Appeec)*.
- Herrera-Lozada, J. C., Calvo, H. ve Taud, H., 2011, A Micro artificial immune system, *Polibits* (43), 107-111.

- Hinojosa, V. H. ve Araya, R., 2013, Modeling a mixed-integer-binary small-population evolutionary particle swarm algorithm for solving the optimal power flow problem in electric power systems, *Applied Soft Computing*, 13 (9), 3839-3852.
- Holland, J. H., 1975, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*, University of Michigan Press Ann Arbor.
- Huang, T. ve Mohan, A. S., 2006, Micro-particle swarm optimizer for solving high dimensional optimization problems (mu PSO for high dimensional optimization problems), *Applied Mathematics and Computation*, 181 (2), 1148-1154.
- Itoh, K., Miyata, K. ve Igarashi, H., 2012, Evolutional Design of Waveguide Slot Antenna With Dielectric Lenses, *Ieee Transactions on Magnetics*, 48 (2), 779-782.
- Kennedy, J. ve Eberhart, R., 1995, Particle swarm optimization, *1995 Ieee International Conference on Neural Networks Proceedings, Vols 1-6*, 1942-1948.
- Krishnakumar, K., 1990, Micro-genetic algorithms for stationary and non-stationary function optimization, *Intelligent Control and Adaptive Systems*, 289-297.
- Kumar, M., Thakur, A. ve Singh, S., Optimization of Some Standard Functions using Artificial Algae Algorithm.
- Lahoz, D., Lacruz, B. ve Mateo, P. M., 2013, A multi-objective micro genetic ELM algorithm, *Neurocomputing*, 111, 90-103.
- Lampinen, J. ve Zelinka, I., 2000, On stagnation of the differential evolution algorithm, *Proceedings of MENDEL*, 76-83.
- Mininno, E., Neri, F., Cupertino, F. ve Naso, D., 2011, Compact differential evolution, *IEEE Transactions on Evolutionary Computation*, 15 (1), 32-54.
- Mitra, P., Chowdhury, S. P., Chowdhury, S., Pal, K. ve Crossley, P. A., 2008, Intelligent AVR and PSS with Adaptive Hybrid Learning Algorithm, *2008 Ieee Power & Energy Society General Meeting, Vols 1-11*, 3719-3725.
- Mostaghim, S. ve Teich, J., 2003, Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO), *Proceedings of the 2003 Ieee Swarm Intelligence Symposium (Sis 03)*, 26-33.
- Neri, F. ve Mininno, E., 2010, Memetic Compact Differential Evolution for Cartesian Robot Control, *Ieee Computational Intelligence Magazine*, 5 (2), 54-65.
- Neri, F. ve Tirronen, V., 2010, Recent advances in differential evolution: a survey and experimental analysis, *Artificial Intelligence Review*, 33 (1-2), 61-106.
- Nesmachnow, S., Cancela, H. ve Alba, E., 2012, A parallel micro evolutionary algorithm for heterogeneous computing and grid scheduling, *Applied Soft Computing*, 12 (2), 626-639.

- Olguin-Carbajal, M., Alba, E. ve Arellano-Verdejo, J., 2013, Micro-Differential Evolution with Local Search for High Dimensional Problems, *2013 Ieee Congress on Evolutionary Computation (Cec)*, 48-54.
- Pandit, N., Tripathi, A., Tapaswi, S. ve Pandit, M., 2011, Static/Dynamic Environmental Economic Dispatch Employing Chaotic Micro Bacterial Foraging Algorithm, *Swarm, Evolutionary, and Memetic Computing, Pt I*, 7076, 585-+.
- Parsopoulos, K. E., 2009, Cooperative micro-differential evolution for high-dimensional problems, *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, 531-538.
- Parsopoulos, K. E., 2012, Parallel cooperative micro-particle swarm optimization: a master-slave model, *Applied Soft Computing*, 12 (11), 3552-3579.
- Peer, E. S., Van den Bergh, F. ve Engelbrecht, A. P., 2003, Using neighbourhoods with the guaranteed convergence PSO, *Proceedings of the 2003 Ieee Swarm Intelligence Symposium (Sis 03)*, 235-242.
- Price, K., Storn, R. M. ve Lampinen, J. A., 2006, Differential evolution: a practical approach to global optimization, Springer Science & Business Media, p.
- Rahnamayan, S. ve Tizhoosh, H. R., 2008, Image Thresholding Using Micro Opposition-Based Differential Evolution (Micro-ODE), *2008 Ieee Congress on Evolutionary Computation, Vols 1-8*, 1409-1416.
- Rahnamayan, S., Jesuthasan, J., Bourennani, F., Naterer, G. F. ve Salehinejad, H., 2014a, Centroid opposition-based differential evolution, *International Journal of Applied Metaheuristic Computing (IJAMC)*, 5 (4), 1-25.
- Rahnamayan, S., Jesuthasan, J., Bourennani, F., Salehinejad, H. ve Naterer, G. F., 2014b, Computing opposition by involving entire population, *Evolutionary Computation (CEC), 2014 IEEE Congress on*, 1800-1807.
- Rajasekhar, A. ve Das, S., 2013, Cooperative Micro Artificial Bee Colony Algorithm for Large Scale Global Optimization Problems, *Swarm, Evolutionary, and Memetic Computing, Pt I (Semcco 2013)*, 8297, 469-480.
- Ribas, P. C., Yamamoto, L., Polli, H. L., Arruda, L. V. R. ve Neves, F., 2013, A micro-genetic algorithm for multi-objective scheduling of a real world pipeline network, *Engineering Applications of Artificial Intelligence*, 26 (1), 302-313.
- Salehinejad, H., Rahnamayan, S., Tizhoosh, H. R. ve Chen, S. Y., 2014, Micro-Differential Evolution with Vectorized Random Mutation Factor, *2014 Ieee Congress on Evolutionary Computation (Cec)*, 2055-2062.
- Sanchez, D., Melin, P., Castillo, O. ve Valdez, F., 2013, Modular Granular Neural Networks Optimization with Multi-Objective Hierarchical Genetic Algorithm for

- human recognition based on iris biometric, *2013 Ieee Congress on Evolutionary Computation (Cec)*, 772-778.
- Segura, C., Coello, C. A. C. ve Hernández-Díaz, A. G., 2015, Improving the vector generation strategy of differential evolution for large-scale optimization, *Information Sciences*, 323, 106-129.
- Sotelo-Figueroa, M. A., Soberanes, H. J. P., Carpio, J. M., Huacuja, H. J. F., Reyes, L. C. ve Alcaraz, J. A. S., 2013, Evolving bin packing heuristic using micro-differential evolution with indirect representation, In: *Recent Advances on Hybrid Intelligent Systems*, Eds: Springer, p. 349-359.
- Tan, C. J., Lim, C. P. ve Cheah, Y. N., 2013, A Modified micro Genetic Algorithm for undertaking Multi-Objective Optimization Problems, *Journal of Intelligent & Fuzzy Systems*, 24 (3), 483-495.
- Teng, N. S., Teo, J. ve Hijazi, M. H. A., 2009, Self-adaptive population sizing for a tune-free differential evolution, *Soft Computing*, 13 (7), 709-724.
- Tippayachai, J., Ongsakul, W. ve Ngamroo, I., 2002, Parallel micro genetic algorithm for constrained economic dispatch, *Ieee Transactions on Power Systems*, 17 (3), 790-797.
- Tiwari, S., Fadel, G. ve Deb, K., 2011, AMGA2: improving the performance of the archive-based micro-genetic algorithm for multi-objective optimization, *Engineering Optimization*, 43 (4), 377-401.
- Tsai, K. Y. ve Wang, F. S., 2005, Evolutionary optimization with data collocation for reverse engineering of biological networks, *Bioinformatics*, 21 (7), 1180-1188.
- Uymaz, S. A., 2015, Yeni bir biyolojik ilhamlı metasezgisel optimizasyon metodu: Yapay alg algoritması, *Selçuk Üniversitesi Fen Bilimleri Enstitüsü*.
- Uymaz, S. A., Tezel, G. ve Yel, E., 2015, Artificial algae algorithm (AAA) for nonlinear global optimization, *Applied Soft Computing*, 31, 153-171.
- Venayagamoorthy, G. K. ve Das, T. K., 2006, Optimal design of power system stabilizers using a small population based PSO.
- Viveros-Jimenez, F., Mezura-Montes, E. ve Gelbukh, A., 2009, Elitistic Evolution: a Novel Micro-Population Approach for global optimization problems, *2009 Eighth Mexican International Conference on Artificial Intelligence, Proceedings*, 15-+.
- Viveros-Jiménez, F. ve Mezura-Montes, E., 2012, Empirical analysis of a micro-evolutionary algorithm for numerical optimization, *International Journal of Physical Sciences*, 7 (8), 1235-1258.
- Wang, C., Liu, Y. C. ve Zhao, Y. T., 2013, Application of dynamic neighborhood small population particle swarm optimization for reconfiguration of shipboard power system, *Engineering Applications of Artificial Intelligence*, 26 (4), 1255-1262.

- Weber, M., Neri, F. ve Tirronen, V., 2011, A study on scale factor in distributed differential evolution, *Information Sciences*, 181 (12), 2488-2511.
- Wu, D., Gan, D. ve Jiang, J. N., 2014, An improved micro-particle swarm optimization algorithm and its application in transient stability constrained optimal power flow, *International Transactions on Electrical Energy Systems*, 24 (3), 395-411.
- Xu, Y. ve Liu, G., 2002, Detection of flaws in composites from scattered elastic-wave field using an improved μ GA and a local optimizer, *Computer methods in applied mechanics and engineering*, 191 (36), 3929-3946.
- Yang, X.-S., 2010, Nature-inspired metaheuristic algorithms, Luniver press, p.
- Zhang, J. R., Wang, J. ve Yue, C. Y., 2012, Small Population-Based Particle Swarm Optimization for Short-Term Hydrothermal Scheduling, *Ieee Transactions on Power Systems*, 27 (1), 142-152.
- Zhang, X. D., Wu, C. Z., Li, J., Wang, X. Y., Yang, Z. J., Lee, J. M. ve Jung, K. H., 2016, Binary artificial algae algorithm for multidimensional knapsack problems, *Applied Soft Computing*, 43, 583-595.
- Mandelbrot, B. B., & Mandelbrot, B. B. (1982). The fractal geometry of nature (Vol. 1). New York: WH freeman.
- Yang, X. S., & Deb, S. (2013). Multiobjective cuckoo search for design optimization. *Computers & Operations Research*, 40(6), 1616-1624.