



T.C.
KONYA TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ



POPÜLASYON ÇEŞİTLİLİĞİ KONTROLLÜ
ve GERİ BESLEMELİ GALAKTİK SÜRÜ
OPTİMİZASYON ALGORİTMASI

Oğuzhan UYMAZ

YÜKSEK LİSANS

Bilgisayar Mühendisliği Anabilim Dalı

Mayıs-2019
KONYA
Her Hakkı Saklıdır

TEZ KABUL VE ONAYI

Oğuzhan UYMAZ tarafından hazırlanan “Popülasyon Çeşitliliği Kontrollü ve Geri Beslemeli Galaktik Sürü Optimizasyon Algoritması” adlı tez çalışması 10/05/2019 tarihinde aşağıdaki jüri tarafından oy birliği ile Konya Teknik Üniversitesi Lisansüstü Eğitim Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı’nda YÜKSEK LİSANS olarak kabul edilmiştir.

Jüri Üyeleri

Başkan

Dr. Öğr. Üyesi Mehmet Akif ŞAHMAN

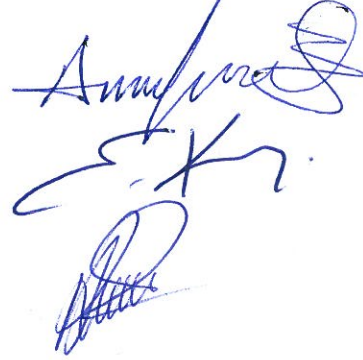
Danışman

Dr. Öğr. Üyesi Ersin KAYA

Üye

Doç. Dr. Barış KOÇER

İmza



Yukarıdaki sonucu onaylıyorum.

Prof. Dr. Yakup KARA
Enstitü Müdürü

Bu tez çalışması Selçuk Üniversitesi Bilimsel Araştırma Projeleri Koordinatörlüğü tarafından 18101013 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.



Oğuzhan UYMAZ

Tarih: 15.05.2019

ÖZET

YÜKSEK LİSANS

POPÜLASYON ÇEŞİTLİLİĞİ KONTROLLÜ ve GERİ BESLEMELİ GALAKTİK SÜRÜ OPTİMİZASYON ALGORİTMASI

Oğuzhan UYMAZ

Konya Teknik Üniversitesi
Lisansüstü Eğitim Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Dr. Öğr. Üyesi Ersin KAYA

2019,67 Sayfa

Jüri

Dr. Öğr. Üyesi Mehmet Akif ŞAHMAN
Dr. Öğr. Üyesi Ersin KAYA
Doç. Dr. Barış KOÇER

Optimizasyon, bir problem için en uygun çözümü makul bir sürede bulma işlemidir. Popülasyon tabanlı optimizasyon algoritmaları, çözüme ulaşmak için birden fazla çözüm adayını iyileştirerek daha iyi çözümler elde etmeyi amaçlamaktadır. Çözüm adaylarının keşif ve sömürü yeteneklerinin dengesi, optimizasyon yönteminin kaliteli çözümler elde etmesini sağlamaktadır. Keşif ve sömürü yeteneklerini gözlemlemek için, popülasyon çeşitliliği kontrolü yaygın olarak kullanılan bir araçtır. Popülasyon çeşitliliği, popülasyonu oluşturan bireylerin pozisyon, hız ve amaç fonksiyonunun değerlerinin gösterdiği çeşitliliğidir. Popülasyon çeşitliliğinin düşük olması, sömürü yeteneğini olumlu etkilerken, çeşitliliğin yüksek olması keşif yeteneğini olumlu olarak etkilemektedir. Galaktik sürü optimizasyonu yöntemi, gezegenlerin ve yıldızların hareketlerinden esinlenilerek ortaya konulmuş popülasyon tabanlı bir optimizasyon sistemidir. Galaktik sürü optimizasyonu yöntemi, doğrudan bir optimizasyon algoritması olmayıp, içerisinde parçacık sürü optimizasyonu algoritmasını kullanan bir üst çatıdır. Parçacık sürü optimizasyonu algoritması, sürü halinde gezen balıkların ve kuşların davranışlarından esinlenilerek ortaya konulmuş, popülasyon tabanlı bir optimizasyon algoritmasıdır. Galaktik sürü optimizasyonu yöntemi, iki aşamadan oluşmaktadır. İlk aşamanın amacı arama uzayını etkin bir şekilde taramak, ikinci aşamanın amacı ise birinci aşamadan elde edilen çözümlerin iyileştirilmesidir. Bu tez çalışmasında, ilk aşamada popülasyon çeşitliliği kontrolü ile, keşif yeteneği iyileştirilmiştir ve ikinci aşamadan elde edilen bireyler ilk aşamaya ilave edilerek, standart galaktik sürü optimizasyonu yönteminin çözüm yeteneği iyileştirilmiştir. Önerilen yöntemin performansı, literatürde sıklıkla kullanılan standart karşılaştırma fonksiyonları üzerinde test edilmiş ve sonuçları yakın zamanda önerilmiş optimizasyon algoritmaları ile karşılaştırılmıştır. Yapılan çalışmalar sonucunda önerilen yöntemin standart galaktik sürü optimizasyonu yönteminin performansını arttırdığı gözlemlenmiştir.

Anahtar Kelimeler: Galaktik Sürü Optimizasyonu, Optimizasyon, Parçacık Sürü Optimizasyonu, Popülasyon Çeşitliliği

ABSTRACT

MS THESIS

POPULATION DIVERSITY CONTROLLED and FEEDBACK GALACTIC SWARM OPTIMIZATION ALGORITHM

Oğuzhan UYMAZ

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

Advisor: Asst. Prof. Dr. Ersin KAYA

2019,67 Pages

Jury

**Asst. Prof. Dr. Mehmet Akif ŞAHMAN
Asst. Prof. Dr. Ersin KAYA
Assoc. Prof. Dr. Barış KOÇER**

Optimization is the process of finding the optimal solution for a problem in a reasonable time. Population-based optimization algorithms aim to achieve better solutions by improving more than one solution candidate to reach a solution. The balance of exploration and exploitation capabilities of the solution candidates enables the optimization method to achieve quality solutions. To improve exploration and exploitation capabilities, population diversity control is a widely used tool. Population diversity is the diversity of the values of the position, speed and objective function of the population. Low population diversity affects the exploitation ability positively while high diversity affects the exploration ability positively. The Galactic swarm optimization method is a population-based optimization system inspired by the movements of the planets and stars. The Galactic swarm optimization method is not a direct optimization algorithm, but a framework that uses the particle swarm optimization algorithm. The particle swarm optimization algorithm is a population-based optimization algorithm that is inspired by the behavior of fishes and birds. Galactic swarm optimization method consists of two stages. The aim of the first phase is to effectively scan the search space and the aim of the second stage is to improve the solutions obtained from the first stage. In this paper, with the control of population diversity at the first stage, the exploration ability was improved and the individuals obtained from the second stage were added to the first stage and the performance of the standard galactic swarm optimization method was improved. The performance of the proposed method has been tested on standard benchmark functions, which are frequently used in the literature, and the results have been compared with the recently proposed optimization algorithms. As a result of the studies, it has been observed that the proposed method improves the performance of the standard galactic swarm optimization method.

Keywords: Galactic Swarm Optimization, Optimization, Particle Swarm Optimization, Population Diversity

ÖNSÖZ

Bu çalışmayı hazırlarken geçirdiğim süreçte benden yardımlarını esirgemeyen danışmanım Dr. Öğr. Üyesi Ersin KAYA' ya ve çalışmalarım sırasında yardım ve desteklerini esirgemeyen iş arkadaşlarıma ve aileme teşekkürlerimi sunarım.

Oğuzhan UYMAZ
KONYA-2019



İÇİNDEKİLER

ÖZET	iv
ABSTRACT.....	v
ÖNSÖZ	vi
İÇİNDEKİLER.....	vii
SİMGELER VE KISALTMALAR.....	viii
1. GİRİŞ.....	1
2. KAYNAK ARAŞTIRMASI.....	5
2.1. Parçacık Sürü Optimizasyonu.....	5
2.2. Galaktik Sürü Optimizasyonu.....	8
2.3. Popülasyon Çeşitliliği.....	10
3. MATERYAL VE YÖNTEM.....	15
3.1. Optimizasyon.....	15
3.2. Parçacık Sürü Optimizasyonu.....	15
3.3. Galaktik Sürü Optimizasyonu.....	19
3.4. Popülasyon Çeşitliliği.....	20
3.4.1. Pozisyon tabanlı popülasyon çeşitliliği.....	21
3.4.2. Hız tabanlı popülasyon çeşitliliği.....	25
3.4.3. Kognitif tabanlı popülasyon çeşitliliği.....	27
3.5. Önerilen Yöntem.....	29
4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA.....	33
5. SONUÇLAR VE ÖNERİLER.....	60
5.1 Sonuçlar.....	60
5.2 Öneriler.....	61
KAYNAKLAR.....	62
ÖZGEÇMİŞ.....	67

SİMGELER VE KISALTMALAR

Kısaltmalar

GSO	: Galaktik Sürü Optimizasyonu (Galactic Swarm Optimization)
GSO_DFB	: Popülasyon Çeşitliliği Kontrollü ve Geri Beslemeli GSO
PSO	: Parçacık Sürü Optimizasyonu (Particle Swarm Optimization)
AAA	: Yapay Alg Algoritması (Artificial Algae Algorithm)
ABC	: Yapay Arı Algoritması (Artificial Bee Colony)
BAT	: Yarasa Algoritması (Bat Optimization Algorithm)
BEE	: Yapay Arı Kolonisi Algoritması (Bee Algorithm)
FPA	: Çiçek Tozlaşma Algoritması (Flower Pollination Optimization Algorithm)



1. GİRİŞ

Optimizasyon ya da matematiksel programlama, belirlenmiş bir problem için çözümler içerisinde en iyi çözümü saptama işlemidir. Bu işlemler yapılırken, kaynakları en az tutmak gerekmektedir. Ayrıca en az kaynakla en iyi çözümü bulmaya çalışılmalıdır (Uymaz ve ark., 2015). Optimizasyon, günlük hayatımızın birçok alanında kullanılmaktadır. Matematiksel işlemlerin yer aldığı bütün problemlerde kullanılabilir. İnşaat, planlama, kalite yönetimi, kar işlemleri gibi birçok alanda kullanılmaktadır. Bir fabrika ele alındığında, zamanı ve parayı en iyi şekilde kullanarak bu tesisten en iyi kazancı elde etmek isteyecektir. Böyle bir iş için, problem belirlenip parametreleri en uygun şekilde kullanıp en uygun çözüme gidilmelidir. Bunun için de optimizasyon yöntemlerinin kullanılması uygun olacaktır.

Doğa bilimi, kendi içinde türlerin evrimi gibi sayısız biyolojik olguya sahiptir. Biyolojik alandaki davranışlar, optimizasyon çözümleri için esinlenen bir alan olmuştur (Chen ve Mahfouf, 2009). Bu teknikler, büyümeye devam eden, geleneksel tekniklerin sınırlarını aşan ve hem akademik hem de profesyonel alanlarda yaygın olarak kabul gören biyolojik ilhamlı teknikler olarak bilinir (Rai ve Seth, 2013).

Doğa, farklı görevleri yerine getirmek için sosyal davranışlarla doludur. Bütün bireylerin ve ortak davranışların nihai amacı hayatta kalmak olsa da, canlı gruplar, sürüler, koloniler ve sürüler işbirliği yapmakta ve etkileşimde bulunmaktadır. Buna örnek olarak, avlanma, savunma, gezinme ve arama gibi durumlar söylenebilir (Mirjalili, 2016). Bu gibi algoritmalar popülasyon tabanlı çalışmaktadırlar. Popülasyon tabanlı algoritmalara örnek olarak; Genetik Algoritmalar, Parçacık Sürü Optimizasyonu, Yapay Arı Kolonisi, Ağaç Tohumu Algoritması, Yapay Alg Algoritması, Yarasa İlhamlı Optimizasyon Algoritması, The Bees Algoritması ve Çiçek Tozlaşma Algoritması verilebilir.

Genetik algoritma (GA), seçim yoluyla Darwin'in evrim ilkesi tarafından motive edilen bir sezgisel çözüm arama veya optimizasyon tekniğidir. Her GA bir yapay kromozom popülasyonunda çalışır. Her bir kromozom, bir problemin çözümünü temsil eder ve gerçek bir sayıya sahiptir. Rasgele oluşturulmuş bir kromozom popülasyonu ile başlayan bir GA, gelecek nesil olan bir popülasyon üretmek için uygunluk bazlı bir seçim ve rekombinasyon işlemi gerçekleştirir. Rekombinasyon sırasında, ana kromozomlar seçilir ve genetik materyalleri, çocuk kromozomları üretmek için yeniden birleştirilir. Bunlar daha sonraki popülasyona geçer. Bu işlem tekrarlandığından, art arda gelen

nesiller dizisi gelişir ve kromozomların ortalama uygunluğu, bir durma kriterine ulaşıncaya kadar artma eğilimindedir. Bu şekilde, bir GA belirli bir soruna en iyi çözümü geliştirir (McCall, 2005).

Yapay Arı Kolonisi (ABC), 2005 yılında Derviş Karaboga tarafından bal arılarının davranışıyla tanımlanmış bir algoritmadır. ABC, gıda pozisyonlarının yapay arılar tarafından zaman içinde değiştirildiği, popülasyona dayalı bir arama prosedürü sunar ve arının amacı, yüksek nektar miktarına sahip gıda kaynaklarının yerlerini bulmaktır. ABC sisteminde yapay arılar çok boyutlu bir arama alanında uçarlar ve bazıları, kendi deneyimlerine ve yuva arkadaşlarına göre gıda kaynaklarını seçer ve konumlarını ayarlar (Karaboga, 2005).

Ağaç tohum algoritması (TSA), 2015 yılında Mustafa Servet Kiran tarafından geliştirilen sezgisel ve popülasyon tabanlı bir optimizasyon algoritmasıdır. TSA' da ağaçlar ve tohumlar, optimizasyon problemi için olası çözümü temsil eder. Ağaçlardan bir veya daha fazla tohum üretilir ve daha iyi tohumların yerleri ağaçların yerleri ile değiştirilir. Tohumlar için yeni yerler üretilirken, en iyi çözüm ya da başka bir ağaç konumu dikkate alınır. Bu değerlendirme, arama eğilimi (ST) olarak adlandırılan bir kontrol parametresi kullanılarak gerçekleştirilir ve bu işlem önceden tanımlanmış sayıda yineleme için gerçekleştirilir (Kiran, 2015).

Yapay alg algoritması (AAA), 2015 yılında Sait Ali Uymaz tarafından geliştirilen sezgisel bir optimizasyon algoritmasıdır. Bu algoritma, evrimsel süreç, adaptasyon süreci ve mikro alglerin hareketlerine dayanan helisel hareket sürecinden oluşmaktadır. Adaptasyon ve evrimsel sürecin katkısı nedeniyle, yerel minimumdan kaçınmak için ışık kaynağını seçerken turnuva seçim kullanılır (Uymaz ve ark., 2015).

Yarasa ilhamlı optimizasyon algoritması (BAT), 2010 yılında Xin-She Yang tarafından geliştirilmiş, doğa esinli bir optimizasyon algoritmasıdır. Bu algoritma, yarasaların av ve yiyecek ararken sergilediği davranışlardan ilham almaktadır. Yarasalar avı yakalamak için rasgele bir uçuş davranışı sergilemektedir. Yayıdıkları seslerin yankılarını kullanarak avlarına olan mesafeleri hesaplayabilmektedirler. Algoritma bu davranışları kullanarak en iyiyi bulmayı hedeflemektedir (Yang, 2010).

The bees algoritması (BEE), 2005 yılında D. T. Pham ve arkadaşları tarafından geliştirilmiş bir doğa esinli optimizasyon algoritmasıdır. Bu algoritma, bal arıları sürülerinin yiyecek toplama davranışlarını taklit eder. Temel versiyonunda, rastgele aramayla birleştirilen bir tür komşuluk araması gerçekleştirir ve hem kombinasyonel optimizasyon hem de fonksiyonel optimizasyon için kullanılabilir (Pham ve ark., 2005).

Çiçek tozlaşma algoritması (FPA), 2012 yılında Xin-She Yang tarafından geliştirilen, popülasyon tabanlı bir optimizasyon algoritmasıdır. Bu algoritma, çiçeklerin tozlaşması ile etkileşiminden esinlenmiştir (nabil, 2016).

Parçacık Sürü Optimizasyonu (PSO), sürü halinde hareket eden balıklar ve kuşlardan esinlenerek, Kenedy ve Eberhart tarafından 1995 yılında geliştirilmiş bir optimizasyon yöntemidir. Temel olarak sürü zekâsına dayanan bir algoritmadır. Sürü halinde hareket eden hayvanların yiyecek ve güvenlik gibi durumlarda, çoğu zaman rastgele sergiledikleri hareketlerin, amaçlarına daha kolay ulaşmalarını sağladığı görülmüştür. PSO, bireyler arasındaki sosyal bilgi paylaşımını esas alır (Kennedy ve Eberhart, 1995; Shi ve Eberhart, 2008). Her birey parçacık olarak adlandırılır ve parçacıklardan oluşan popülasyona da sürü denir. Her bir parçacık kendi pozisyonunu, bir önceki tecrübesinden yararlanarak sürüdeki en iyi pozisyona doğru ayarlar. Her bir parçacık, mevcut küresel en iyi gbest pozisyonuna ve yereldeki en iyi pbest pozisyonuna doğru çekilirken, aynı zamanda rastgele hareket etme eğilimindedir (Yang ve Karamanoglu, 2013).

Galaktik Sürü Optimizasyonu (GSO), 2016 yılında Muthiah-Nakarajan ve arkadaşları tarafından geliştirilmiştir. GSO galaksiler, yıldızlar ve bunların birbirleri ile olan ilişkilerinden esinlenilerek ortaya konulmuş bir optimizasyon sistemidir (Muthiah-Nakarajan ve Noel, 2016). GSO iki aşamadan oluşan bir yapıya sahiptir. Birinci aşamada, birbirinden bağımsız popülasyon grupları belirlenen bir arama yöntemini kullanarak çözüm uzayında iyi çözüm adayları aramaktadır. İkinci aşamada ise birinci aşamadaki grupları elde ettikleri iyi sonuçlar kullanılarak yeni bir popülasyon oluşturulmaktadır. Bu popülasyon ikinci aşamanın başlangıç popülasyonudur. İkinci aşamada en iyi çözüm aranmaktadır. Bu işlem belirlenen adet kadar tekrar edilerek problem için en iyi çözüm elde edilmeye çalışılmaktadır. Orijinal olarak sunulan GSO yönteminde hem birinci aşamada hem de ikinci aşamada, arama yöntemi olarak PSO algoritması kullanılmıştır.

Çeşitlilik, algoritmanın çalışması sırasında bireylerin farklı varyasyon değerlerinin ölçülmesidir. PSO algoritmasının çalışması sırasında bireyler en iyi yolu izler ve bireyler en iyiye yaklaştıkça çeşitlilik azalır. Bireyler başlangıçta rastgele üretilir. Bu nedenle çeşitlilik başlangıçta yüksek iken, çalışma devam ederken düşer (Shi ve Eberhart, 2008).

Çeşitlilik, optimizasyonun keşif ve sömürü kabiliyetlerini belirlemektedir. Çeşitliliğin yüksek olması keşif yeteneğini arttırmaktadır. Çeşitliliğin düşük olması sömürü kabiliyetini arttırmaktadır.

Bir optimizasyon algoritmasının performansını etkileyen en önemli faktör, “keşif” ve “sömürme” yeteneğidir. Keşif, iyi gelecek vaat eden çözümler bulma olasılığının yüksek olması için, bir arama algoritmasının arama alanının farklı alanlarını keşfetme yeteneği olduğu anlamına gelir. Sömürü ise, aday bir çözüm bulmak için araştırmayı gelecek vaat eden bir bölgeye yoğunlaştırma yeteneği anlamına gelmektedir. İyi bir optimizasyon algoritması, tasarlanan iki hedefi en iyi şekilde dengelemelidir (Cheng ve ark., 2012).

Literatürde popülasyon tabanlı algoritmalar için pek çok çeşitlilik ölçütü ortaya konulmuştur. Genel olarak iki tip çeşitlilik ölçütü bulunmaktadır.

- Fitness Değer Tabanlı
- Pozisyon Tabanlı

Fitness Değer Tabanlı çeşitlilik ölçütlerinde bireylerin fitness değerleri arasındaki çeşitlilik ölçülmektedir.

Pozisyon Tabanlı çeşitlilik ölçütlerinde bireyin arama uzayındaki pozisyon çeşitliliği ölçülmektedir. Yaygın olarak pozisyon tabanlı çeşitlilik ölçütleri, keşif ve sömürü kabiliyetlerini gözlemlemekte kullanılmaktadır.

2. KAYNAK ARAŞTIRMASI

2.1. Parçacık Sürü Optimizasyonu

Parçacık sürü optimizasyonu, bilinen en popüler popülasyon tabanlı sezgisel optimizasyon algoritmalarından biridir. Bu algoritma ile ilgili olarak literatürde geçen bazı çalışmalar bu kısımda verilmiştir.

Bu algoritma ilk olarak 1995 yılında, Kenedy ve Eberhart tarafından, sürü halinde hareket eden balıklar ve kuşlardan esinlenerek geliştirilmiştir. Parçacık sürü optimizasyonunun iki temel metodolojide kökleri bulunmaktadır. Daha belirgin olanı, genel olarak yapay hayata, kuş ve balık sürülerine olan bağlarıdır. Bununla birlikte, aynı zamanda evrimsel hesaplamayla da ilgilidir ve hem genetik algoritma hem de evrimsel programlama ile bağları bulunmaktadır (Kennedy ve Eberhart, 1995).

PSO hakkında literatürde birçok çalışma yapılmış ve çeşitli versiyonları sunulmuştur. Yapılan bu çalışmalarda PSO üzerine yapılan geliştirmeler 5 ana grupta toplanmıştır (Zhang ve ark., 2015). Bunlar;

- PSO' nun modifikasyonları
- Popülasyon tabanlı,
- Hibridizasyon
- Teorik analiz
- Paralel uygulamalar

İlk grup olan PSO' nun modifikasyonları çalışmalarında kuantum davranışlı PSO (Jau ve ark., 2013), kaotik PSO (Chuang ve ark., 2011) ve Bulanık PSO (Juang ve ark., 2011) gibi PSO üzerine yapılan geliştirmeler bulunmaktadır. Popülasyon tabanlı yapılan çalışmalarda, halka, yıldız, rastgele vb. topolojiler kullanılarak PSO üzerinde değişiklikler yapılmıştır (Beheshti ve ark., 2014). PSO ile hibrit yapılan çalışmalarda; genetik algoritma (J. Kuo ve W. Hong, 2013), benzetilmiş tavlama (Chen ve Chien, 2011), tabu arama (Ktari ve Chabchoub, 2013), karınca koloni algoritması (Kiran ve ark., 2012) gibi algoritmalar ile hibrit yöntemler oluşturulmuştur. Teorik analiz grubu altında toplanan çalışmalarda PSO' nun parametrelerinin kontrolü üzerinde durulmuştur. Parametre seçimi ve parametre ayarları üzerine çalışmalar yapılmıştır. Bir diğer grup olan paralel uygulamalarda, çok işlemcili ve bulut bilişim formları gibi PSO' nun paralel programlama ile geliştirildiği çalışmalar yapılmıştır.

Shi ve Eberhart (1998) tarafından yapılan çalışmada, standart PSO algoritmasına atalet ağırlığı (inertia weight) kavramı eklenmiştir. Temel PSO algoritması, atalet ağırlığına sahip değildir. Araştırmacılar bu çalışmada, sabit atalet ağırlığını tanıtarak inertia weight kavramını sunmuştur. Ayrıca bu çalışmada, büyük bir atalet ağırlığının küresel bir aramayı kolaylaştırdığı, küçük bir atalet ağırlığının ise yerel bir aramayı kolaylaştırdığı belirtilmiştir (Shi ve Rc, 1998).

L.-Y. Chuang (2008) yaptığı çalışmasında, “Catfish PSO” adlı çalışmasında, bir yayın balığı parçacığının lineer olarak azalan ağırlık parçacığının optimizasyona dâhil edilmesine bağlı olan yeni bir yayın balığı PSO'yu önermiştir. Diğer sıradan parçacıkların aksine, yayın balığı parçacıkları, en iyi uygunluk değerinin belirli bir süre boyunca değiştirilmediğinde, arama alanının en uç noktalarından yeni bir arama başlatır ve bu durum, sürünün tamamı için rehberlik ederek daha iyi çözümler bulma fırsatlarıyla sonuçlanır. Bu durum sonucunda gelecek vaat eden yeni bölgelere arama alanı sağlanmakta ve yakınsama hızlanmaktadır (Chuang ve ark., 2008).

Xin ve ark. (2009) yaptıkları çalışmada, küresel ve yerel arama arasında daha iyi bir denge sağlamak için multi-stage linearly-decreasing inertia weight (MLDW) adında bir grup stratejisi önerilmiştir. Atalet ağırlığı, parçacık sürü optimize edicilerinin (PSO) küresel keşif ve yerel sömürü yeteneklerini kontrol etmek için sıklıkla kullanılmaktadır. Altı popüler benchmark fonksiyonu, PSO' nun performansındaki MLDW stratejilerini değerlendirmek için kullanılmıştır (Xin ve ark., 2009).

Shi ve Liu (2009) yaptıkları çalışmalarında, popülasyon çeşitliliğini artırmak için kaos başlatıldığı ve adaptif parametrelerin kontrol stratejisini belirli bir problemde bağımsız kılmak için kullanılan karma bir PSO önermiştir. Ayrıca, algoritmanın yaklaşmasını güvence altına almak için Metropolis kuralına dayalı yeni kabul politikası alınmıştır (Shi ve Liu, 2009).

Y. X. Shen, G. Y. Wang ve C. M. Tao (2011) yaptıkları çalışmalarında, kişisel deneyimi ve paylaşım deneyimini işlemek için yeni bir korelasyon stratejisinin kullanıldığı bir korelasyon PSO modeli sunmuştur. Korelasyon katsayısı ile popülasyon çeşitliliği arasındaki ilişki teorik analiz ile geliştirilmiştir. Pozitif doğrusal korelasyonlu işleme stratejisinin, popülasyon çeşitliliğinin korunmasında yardımcı olduğu saptanmıştır (Shen ve ark., 2011).

Ekolojik davranıştan ilham alan Lu ve ark. (2013) yaptıkları çalışmada, kafes yapıların tasarımını optimize etmek amacıyla iki yeni strateji olan, sınır kaydırma ve parçacık konumu sınırlama yöntemlerini kullanarak iki artırılmış PSO algoritması

geliştirmiştir. Sınır kaydırma yaklaşımı, parçacıkları, aramada yakınsama oranını arttırmak için uygulanabilmektedir ve olanaksız bölgeler arasındaki sınırlara hareket etmeye zorlamaktadır. Parçacık pozisyonunu sıfırlama yaklaşımı, parçacıkların çeşitliliğini arttırmak ve parçacıkların çözümlerinin yerel minimuma düşmesini önlemek için GA' daki mutasyon şemasıyla motive edilmiştir (Lu ve ark., 2013).

Wang ve ark. (2013) yaptıkları çalışmada, Nelder-Mead simpleks (SM) yöntemine ve değiştirilmiş bir the bare-bones PSO (MBBPSO)' ya dayalı olarak SM-MBBPSO adı verilen yeni bir karma algoritma önermiştir. MBBPSO' nun güçlü küresel arama özelliğini ve SM' nin yüksek hassas yerel arama özelliğini birleştirmek için k-means araç kümelemesine dayalı yeni bir strateji önerilmiştir. Bu, önerilen yöntem ile algoritmanın sömürü ve keşif kabiliyeti arasında iyi bir denge kurulması sağlanmıştır (Wang ve ark., 2013b).

Jiang ve ark. (2013) yaptıkları çalışmada, yeni bir yaşa dayalı PSO olan yaş grubu topolojisi (PSOAG) ile PSO' yu önermişlerdir. Her bir parçacığın yerel alandaki arama yeteneğini ölçmek için yeni bir yaş kavramı sunulmuştur. Arama sırasında popülasyon çeşitliliğini korumak için, parçacıklar yaşlarına göre farklı gruplara ayrılmıştır. Her yaş grubundaki parçacıklar, yalnızca daha genç gruplardakileri veya kendi gruplarını çevreleri olarak seçebilecek şekilde sınırlandırılmıştır. Yaşlanan parçacıklar düzenli olarak yeni ve rastgele oluşturulmuş olanlarla değiştirilmiştir (Jiang ve ark., 2013).

Wang ve ark. (2013) yaptıkları çalışmada, keşif ve sömürü yetenekleri arasında bir denge kurmak için çeşitlilik artırıcı mekanizma ve komşuluk arama stratejileri kullanan, “diversity enhanced PSO with neighborhood search” (DNSPSO) adında bir karma PSO algoritması önermiştir. Karşılaştırma sonuçları, DNSPSO' nun test problemlerinin çoğu üzerinde ümit verici bir performans elde ettiğini göstermiştir (Wang ve ark., 2013a).

Fu ve ark. (2013) yaptıkları çalışmada, keşif ve sömürüyü dengelemek için bir yerel ve bir küresel komşuluk arama stratejisinin kullanıldığı “neighborhood quantum-behaved PSO” (NQPSO) adlı yeni bir “quantum-behaved PSO” (QPSO) algoritması önermiştir. Ayrıca, popülasyonun başlatılması için muhalefet temelli bir öğrenme kavramı kullanılmıştır. Hesaplamalı sonuçlar, önerilen yaklaşımın bazı benzer QPSO algoritmalarını ve diğer beş durum PSO varyantını geride bıraktığını göstermiştir (Fu ve ark., 2013).

Liu ve ark. (2014) yaptıkları çalışmada, popülasyon çeşitliliğini arttırmak ve BBPSO' nun yakınsama oranını hızlandırmak amacıyla, arama sürecinde keşif ve sömürü

arasındaki yetenekleri kaydırmak için astrofizik kaynaklı yeni bir bozulma stratejisi önermiştir. Önerilen kesinti operatöründe dağıtım ve çeşitliliği araştırmışlar, orijinal ve kesintili konum arasındaki pozisyon ilişkisini göstermişlerdir (Liu ve ark., 2014).

Yu ve ark. (2014) yaptıkları çalışmalarında, PSO ve DE arasında dengeli bir parametre geliştirerek PSO ve Diferansiyel Gelişim (DE)' a (HPSO-DE) dayalı yeni bir uyarlanabilir hibrit algoritması ortaya çıkarmışlardır. Popülasyon, yerel optimum etrafında toplandığında, mevcut popülasyon üzerinde adaptif mutasyon yapılmıştır. HPSO-DE, PSO ve DE' nin avantajlarından yararlanılmış ve popülasyonun çeşitliliği korunmuştur. PSO, DE ve varyantları ile karşılaştırıldığında, HPSO-DE' nin performansının rekabetçi olduğu gözlemlenmiştir (Yu ve ark., 2014).

Vitorino ve ark. (2015) yaptıkları çalışmada, PSO' nun tüm parçacıklarının arama alanının tek bir noktasına yaklaştığında çeşitlilik üretmek için ABC' ye dayalı bir mekanizma ortaya koymuştur. Daha sonra, sürü varlıkları, bütün sürünün çeşitliliğine bağlı olarak, bulanık kurallar kullanarak, önceden tanımlanmış iki davranış arasında geçiş yapmıştır (Vitorino ve ark., 2015).

2.2. Galaktik Sürü Optimizasyonu

Galaktik sürü optimizasyonu, sürü tabanlı ve sezgisel bir optimizasyon yöntemidir. Bu yöntem ile ilgili olarak literatürde geçen bazı çalışmalar bu kısımda verilmiştir.

Muthiah-Nakarajan ve ark. (2016) yaptıkları çalışmalarında, yeni bir algoritma olarak galaktik sürü optimizasyonunu (GSO) geliştirmiştir. Bu algoritma, galaksiler, yıldızlar ve bunların birbirleri ile olan ilişkilerinden esinlenilerek ortaya konulmuştur. GSO iki aşamadan oluşan bir yapıya sahiptir. Birinci aşamada, birbirinden bağımsız popülasyon grupları belirlenen bir arama yöntemini kullanarak çözüm uzayındaki iyi çözüm adaylarını aramaktadır. İkinci aşamada ise, birinci aşamadaki grupları elde ettikleri iyi sonuçlar kullanılarak yeni bir popülasyon oluşturulmaktadır. Bu algoritma içerisinde PSO algoritmasını kullanmaktadır (Muthiah-Nakarajan ve Noel, 2016).

Emer Bernal ve ark. (2016) yaptıkları çalışmalarında, GSO yönteminde parametrelerin dinamik ayarlanması için, bulanık sistemlerin kullanılmasını önermiştir. Bu çalışmada, farklı boyutlarda 17 kriterde fonksiyonla algoritmanın performansını ölçmek için c3 ve c4 parametrelerinin dinamik adaptasyonu için farklı bulanık sistemler

önerilmiştir. Ayrıca optimizasyon problemlerinde yöntemin etkinliğini kanıtlamak için farklı değişkenler arasında bir karşılaştırma yapılmıştır (Bernal ve ark., 2016).

Ersin Kaya ve ark. (2017) yaptıkları çalışmalarında, GSO algoritması içerisinde yalnızca PSO algoritmasını değil, ABC algoritmasını da kullanmışlardır. ABC algoritmasının, PSO algoritmasına göre daha iyi keşif özelliklerinin olması nedeniyle, ABC algoritmasının birinci seviyede kullanımını ve PSO algoritmasının ikinci seviyede kullanımını kapsar. Önerilen yaklaşım, iyi bilinen 20 kıyaslama problemi üzerinde test edilmiş ve ön sonuçlar sunulmuştur. Deneysel sonuçlara göre, önerilen yaklaşım temel GSO yaklaşımından daha başarılı sonuçlar vermektedir (Kaya ve ark., 2017).

Nilay Nigam ve D. Geraldine Bessie Amali (2017) yaptıkları çalışmalarında, Galaktik sürü optimizasyon yöntemi ile eğitilmiş bir yapay sinir ağının (YSA) diğer optimizasyon algoritmalarına kıyasla daha doğru bir tahmin ürettiğini göstermişlerdir. Bu çalışmada, Delhi'deki kirletici madde seviyesini tahmin etmek için kullanılabilir çeşitli teknikler değerlendirilmiş ve analiz edilmiştir. Kirletici seviyesi öngörücü, belirli bir bölgedeki kirletici miktarını öngörmeye yardımcı olan bir sistemdir. Sistem, yeni girişin değerini tahmin etmek için geçmiş verileri kullanmaktadır. Tahmin sisteminde, kirlilik seviyesini birkaç sınıfa ayırmak için farklı optimizasyon algoritmaları ile eğitilmiş YSA kullanılmıştır. Bu aşamada YSA, GSO algoritması ile eğitilmiştir (Nigam ve Amali, 2017).

Nanda ve ark. (2018) yaptıkları çalışmada, hibrit bir k-means ve GSO algoritması önermişlerdir. Bu çalışmada beyin Manyetik Rezonans Görüntüleme (MRG) görüntülerinde görüntü segmentasyonu sorunu uygulanmıştır. Bu işlem, beş farklı tümör tipinin yerini, boyutunu ve şeklini tespit etmek, yakalamak için yapılmıştır. Görüntü bölümlenme, kümelenme problemi olarak ifade edilmiş ve etkili bir çözüm için yeni bir hibrit K-means GSO algoritması önerilmiştir. Otsus entropi ölçüsü, bölümleri üretmek için uygunluk işlevi olarak kullanılmıştır. Beş farklı beyin MRG görüntüsünde beş performans ölçümü ile yapılan kapsamlı simülasyon çalışmaları, önerilen yaklaşımın GSO, Real Kodlu Genetik Algoritma (RCGA) ve K-Means kümeleme algoritmaları üzerindeki üstün performansını ortaya koymaktadır (Nanda ve ark., 2018).

Bernal ve ark. (2018) yaptıkları çalışmada, bulanık mantık kullanarak GSO' nunda parametrelerin ayarlanması için bir değişken önermiştir. Bu çalışmada, FGS01 ile elde edilen sonuçların iyileştirilmesi amacıyla, bulanık GSO1 (FGSO1) adı verilen bulanık sistemde kullanılan üyelik fonksiyonlarının parametrelerinin optimizasyonunun yapılması önerilmiştir. Önerilen yöntem ve orijinal FGS01 bulanık sistemi, davranış

karakteristiklerini düşük boyutlardan yüksek boyutlara kadar değerlendirebilmek için CEC-2015 fonksiyonlarıyla test edilmiştir (Bernal ve ark., 2018).

2.3. Popülasyon Çeşitliliği

Çeşitlilik kavramı, popülasyon tabanlı optimizasyon algoritmalarında, popülasyonun çeşitliliği anlamına gelmektedir. Yani üzerinde işlem yapılan popülasyonun arama uzayına ne kadar yayıldığını anlatmaktadır. Bu kısımda bu çeşitlilik kavramı hakkında literatürde yer alan bazı çalışmalar yer almaktadır.

Zou ve ark. (2004) yaptıkları çalışmalarında, dinamik optimizasyon problemlerinde popülasyon çeşitliliği için iyi bir teknik sağlayan kaotik bir yaklaşımla birleştirilmiş bir evrimsel algoritma (CEA) önerilmiştir. Değişen ortamlara uyum problemlerine evrimsel algoritmaların (EA) uygulanmasında, popülasyonun çeşitliliğinin korunması şarttır. CEA' nın performansını diğer araştırmacıların kendinden uyarlamalı yaklaşımıyla karşılaştırmak için birçok sayısal deney rapor edilmiştir ve sayısal sonuçlar CEA' nın performansının iki dinamik kıyaslama problemi için yayınlanan diğer algoritmalarından daha üstün olduğunu göstermiştir (Xiufen ve ark., 2004).

Yuhui Shi ve Russell C. Eberhart (2008) yaptıkları çalışmalarında, PSO algoritmasının çeşitliliğini ölçmek için bir yöntem önermiştir. Evrimsel hesaplama alanında, popülasyonun önemli bir özelliği çeşitliliktir. Bu çalışma, popülasyonu oluşturan parçacıkların konum ve hız özelliklerinin ölçümünü içermektedir. Önerilen yöntem hesaplama açısından basittir ve diğer evrimsel algoritmalara uyarlanabilmektedir (Shi ve Eberhart, 2008).

Feng ve ark. (2009) yaptıkları çalışmalarında, Swarm Intelligence (SI) algoritmalarının önemli bir yönünün analiz edilmesini sağlayan olasılık teorisine dayanan, matematiksel bir çerçeve sunulmuştur. İlk olarak, SI algoritmalarındaki popülasyon için popülasyon yoğunluk derecesi tanımlanmıştır. Popülasyon çeşitliliği daha sonra istatistikteki normal dağılım açısından tartışılmaktadır. Bu çalışmada, dağıtım ve popülasyon çeşitliliği arasında yakın bir bağlantı olduğu ve popülasyon çeşitliliğinin temel yapısının dağılımdaki ile oldukça benzer olduğu gösterilmiştir. Olasılık dağılımlarına dayanan popülasyon çeşitliliği modelinin ayrıntılı bir inşası verilmiştir (Feng ve ark., 2009).

Li ve ark. (2010) arama sürecinin verimliliğini artırmak için bir çalışma yapmıştır. Bu çalışmada anlatılan algoritma, mevcut optimum bireyi bir arama başlangıç noktası

olarak görmektedir ve en uygun çözümleri aramak için tavllanmış benzetim ile etkin çaprazlama ve mutasyon operatörü tasarlanmaktadır. Evrimsel algoritma ile tek hedefli optimizasyon problemlerinin çözümünde, arama etkinliği ve popülasyon çeşitliliğinin korunması sık sık ortaya çıkmaktadır. Erken yakınsamanın üstesinden gelmek ve popülasyonun çeşitliliğini korumak amacıyla mevcut popülasyonu güncellemek için sıralama temelli bir seçim mekanizması kullanılmıştır. Seçim işlemi, popülasyon çeşitliliğini korumak için ve en iyisini seçmek için benzer bireyleri karşılaştırmaktadır. Sayısal sonuçları mevcut algoritmalar ile karşılaştırmak ve analiz etmek için çok sayıda tek modlu test fonksiyonu ile test edilmiştir (Li ve ark., 2010).

Jie Qi ve Shunan Pang (2010), PSO algoritmasının erken yakınsama problemini aşmak için, yeniden çeşitlendirilmiş PSO (RDPSO) adı verilen genişletilmiş bir PSO algoritması önermişlerdir. Popülasyon çeşitliliği düşük olduğunda, parçacıkların hızı sıfıra yaklaşır ve algoritma durur, çeşitlendirme mekanizmasını yeniden başlatma yaklaşımı çalışmaya başlar ve parçacıkları dağıtır, uzak konumlarda bırakmalarına izin verir. Sayısal deney sonuçları, RDPSO' nun küresel optimizasyonda, özellikle çözümü zor olan karmaşık algoritmalarda, multimodal fonksiyonlar için üstün performansa sahip olduğu saptanmıştır (Qi ve Pang, 2010).

Shi Cheng ve Yuhui Shi (2011) yaptıkları çalışmada, L1 normuna göre PSO'nun popülasyon çeşitliliğini vermiştir. PSO' nun popülasyon çeşitliliği, algoritmanın keşif veya sömürme yeteneğini ölçerken ve dinamik olarak ayarlanırken önemlidir. Çeşitlilik kontrolü için çeşitli yöntemler kıyaslama fonksiyonları üzerinde test edilmiştir ve mevcut konum ve mevcut hızların ortalamasını temel alan yöntem, en iyi performans gösteren yöntem olarak verilmiştir. Bu yöntem, PSO çeşitliliğini etkili bir şekilde kontrol etmiştir ve standart PSO' dan daha iyi performans elde etmiştir (Cheng ve Shi, 2011).

Zhan Zhongli ve Wang Qiang (2011) yaptıkları çalışmada, kaos teorisine dayanan iyileştirici bir bağışık genetik algoritma sunulmuştur. Kaosun aşırı yayılma karakteri ve rastgeleliği, popülasyonu başlatmak ve arama hızını arttırmak için kullanılmış ve kaosun başlangıçtaki değer duyarlılığı, arama alanını genişletmek için kullanılmıştır. Yerel optimumdan kaçınmak için algoritma, popülasyonu yeniler ve bağışıklık teorisinin yoğunluk hesaplamasını kullanır. Yeni kaos dizisini ayarlayarak popülasyon çeşitliliğini arttırmaktadır. Deneysel sonuçlar, kaos teorisine dayanan bağışık genetik algoritmanın, optimizasyonun sonucunu araştırabileceğini ve yakınsak hızını açıkça geliştirebileceğini göstermiştir (Zhongli ve Qiang, 2011).

Chi ve ark. (2012)' nin, yaygın olarak kullanılan birkaç popülasyon çeşitliliği ölçütüne ilişkin yapılan karşılaştırma çalışmasında, bu çeşitlilik ölçütlerinden elde edilen sonuçların birçok yerde kabul edilen bilgilerle uyuşmadığını ve popülasyon çeşitliliği ile PSO' nun evrimsel süreci arasındaki ilişkinin sorgulanabileceği saptanmıştır. Bu çalışmada, optimum noktaya dayalı yeni bir popülasyon çeşitliliği ölçütü önerilmiştir. Ayrıca, popülasyon çeşitliliğini kontrol etmek ve PSO algoritmasının optimizasyon performansını artırmak için optimum noktaya dayalı çeşitlilik (O-çeşitlilik) ölçüsü incelenmiştir. Çalışma sonuçlarına göre, o-çeşitliliği ölçütünün, popülasyon çeşitliliğini değerlendirmek ve kontrol etmek için etkili ve faydalı olduğunu ve gelişmiş PSO algoritmasının daha iyi optimizasyon performansına sahip olduğu saptanmıştır (Chi ve ark., 2012).

Li Feng Zhang ve Chen Xi Zhou (2012), genetik algoritmanın popülasyon çeşitliliği üzerine bir çalışma yapmıştır. Multimodal optimizasyonda, popülasyon çeşitliliğini korumak, genetik algoritma tasarımında en kritik konulardan biridir. Bu sorunla başa çıkabilmek için çok sayıda uygun yere yerleştirme tekniği geliştirilmiş ve başarıyla uygulanmıştır. Çoklu popülasyona dayalı paralel genetik algoritmalar için, bu yaklaşımlar açıkça uygulanamamaktadır. Çünkü paralel evrimleşme aşaması sırasında tüm popülasyon hakkında küresel bilgi edinmek çok zordur. Bu çalışmada, bu sorunun üstesinden gelmek için yeni bir ada modeli önerilmiştir. Küresel bir popülasyon çeşitlendirmesini sağlamak için sadece birkaç komşu alt popülasyondan elde edilen yerel bilgiler kullanılmıştır. Yeni ada modelinde, alt popülasyonlar otomatik olarak farklı arama alanlarına tahsis edilmiştir, böylece hem küresel optimum hem de yerel optimum dahil çoklu optimumu bulabilmişler, aynı zamanda bu optimumu adaların bağlantı topolojisine göre sıralamış ve evrimin sonuna kadar tutulabilmişlerdir. Ek olarak, önerilen yöntemi kullanarak, PGA' nin performansı da geliştirilmiştir ve gelişmiş bir küresel arama özelliği sergilenmiştir (Li Feng ve Chen Xi, 2012).

Nezami ve ark. (2013) yaptıkları çalışmalarında, çeşitliliğin daha iyi kontrol edilebilmesi için yeni bir fikir uygulamışlardır ve algoritmanın keşif ve sömürü olayları arasındaki erken yakınsamayı önlemeye yönelik davranış kontrol edilmeye çalışılmıştır. PSO' da yakınsama hızı çok yüksektir ve bu durum algoritmanın araştırma alanını gerçek anlamda araştıramaz. Popülasyon çeşitliliği azaldığında, tüm popülasyonlar birbirlerine aşırı yaklaşıma başlar ve algoritma yerel olarak en uygun olana hızla yaklaşır. Bu çalışmada, çeşitlendirilmiş yapay parçacıklar (DAP) üretilerek çeşitlilik kontrol altına alınmış ve çeşitlendirmenin azaldığı zaman, çeşitlendirilmiş yapay parçacıklara dayalı

PSO (PSO-DAP) algoritması olarak adlandırılan çeşitlilik azaltma sırasında popülasyona enjekte edilme işlemi yapılmıştır. Sayısal sonuçlar, önerilen algoritmanın, bu çalışmada alınan tüm test durumlarında temel PSO algoritmalarından daha iyi performans sergilediğini göstermiştir (Mohamad Nezami ve Bahrapour, 2013).

Cheng ve ark. (2014) yaptıkları çalışmada, the brain storm optimization (BSO) algoritmasındaki çeşitliliğin bir tanımını vermiştir. Sürü zekâsı, kısmen bir araya toplanan ve tekrar ayrılmayan çözümler nedeniyle gerçekleşen erken yakınsamayı yaşar. Sürü zekâsında yeni ve gelecek vaat eden bir algoritma olan BSO algoritması, insanın ortak beyin davranışına, yani beyin fırtınası sürecine dayanmaktadır. Erken yakınsama BSO algoritmasında gerçekleşebilmektedir. Çözümler birkaç yinelemeden sonra kümelenmekte ve bu da popülasyon çeşitliliğinin arama sırasında hızla azaldığını göstermektedir. Algoritmanın keşif ve sömürü kabiliyeti, popülasyon çeşitliliğinin değişmesine bağlı olarak ölçülebilmektedir. BSO algoritmasında popülasyon çeşitliliğini geliştirmek için iki tür kısmi yeniden başlatma stratejisi kullanılmıştır. Deneysel sonuçlar, BSO' nun performansının bu iki strateji ile geliştirildiğini göstermiştir (Cheng ve ark., 2014).

Huan Lian ve Yong Qin (2015) yaptıkları çalışmada, yeni tanımlanmış, popülasyon konum ve hız çeşitliliği ölçüsüne dayanan, yeni bir bulanık PSO (NFPSO) algoritması önermiştir. Önerilen NFPSO, bulanık optimizasyon problemleri için bulanık PSO algoritmalarının performansını arttırmayı amaçlayan bulanık değişken atalet ağırlığını ve üyelik fonksiyonunu uyarlamalı olarak ayarlamak için popülasyon çeşitlilik bilgisini kullanmaktadır. Önerilen NFPSO, standart PSO, bulanık PSO, ARPSO, DE ve ABC algoritmalarıyla, 25 sayısal karşılaştırma problemi üzerinde karşılaştırılmıştır. Sonuçlar ve istatistiksel analiz, önerilen NFPSO algoritmasının çok modlu sayısal problemlere uygulandığında iyi sonuçlar verdiğini göstermiştir (Lian ve Qin, 2015).

Ryouei Takahashi (2016) yaptığı çalışmasında, termo dinamik genetik algoritmanın (TDGA), popülasyon çeşitliliğinin sürdürülebilirliği ve çözüm arama etkinliği nedeniyle GA kullanılarak bir fonksiyon optimizasyon problemini çözmeye etkili olduğu deneysel olarak doğrulanmıştır. Çeşitliliği koruyarak, optimum çözümü ve çözümün doğruluğunu arama oranının arttırılabileceği hipotezini, deneysel ve niceliksel olarak doğrulanmıştır. Bu çalışmada, popülasyon çeşitliliği TDGA' da tanımlanan entropi ile ölçülmüştür. TDGA, termodinamikte minimum serbest enerji prensibine dayanan bir seçim stratejisidir. On adet test fonksiyonu kullanılarak, deneysel sonuçları bu çalışmada sunulmuştur (Takahashi, 2016).

Senkerik ve ark. (2018)' nın yaptıkları çalışma, kaotik dinamiklerin ve evrimsel hesaplamaların modern ve popüler hibridizasyonu ile ilgilidir. Kaos ve meta-niteliklerin bir araya getirilmesi üzerine yapılan pek çok araştırma çalışmasının aksine, bu çalışma, özellikle kaotik yaklaşımların popülasyon çeşitliliği üzerindeki etkisiyle, popülasyon dinamikleri hakkında daha derin bir kavrayışa odaklanmıştır. Deneyler, basit bir parametre uyarlamalı Diferansiyel Gelişim (DE) stratejisinde bireylerin seçimi için farklı randomizasyon şemalarının kapsamlı araştırılmasına odaklanmıştır (jDE). Popülasyon çeşitliliği ve jDE yakınsamaları, CEC 2015 kriterinden iki boyutlu ayarlarla (10D ve 30D) ve 15 test fonksiyonuyla test edilmiştir (Senkerik ve ark., 2018).



3. MATERYAL VE YÖNTEM

3.1. Optimizasyon

Optimizasyon, arama uzayında, amaç fonksiyonunu en iyi uygun yapacak parametrelerin belirlenmesi işlemidir. Arama uzayının büyüklüğü, parametre sayısının çok olması ve arama uzayındaki yerel minimum veya maksimum noktalarının sayısı gibi nedenlerden dolayı optimum çözümün bulunması bazen zor olabilmektedir. Bu durumun önüne geçebilmek için literatürde birçok metaheuristik yöntem önerilmiştir.

Optimizasyon problemlerinde, yerel minimum noktalarının sayısının fazlalığı, keşif ve sömürünün dengesizliğinden kaynaklanmaktadır. Bu iki kavramın dengede oluşu, çözümlerin uzaya iyi yayılması ve uygun sonuçları yakalayabilmek için uygun bir noktaya uygun zamanda sömürü yapabilmesini sağlamaktadır.

3.2. Parçacık Sürü Optimizasyonu

PSO, sürü halinde hareket eden balıklar ve böceklerden esinlenerek Kenedy ve Eberhart tarafından 1995 yılında geliştirilmiş bir optimizasyon yöntemidir. PSO algoritması, kuşların ve toplu olarak yiyecek arayan diğer türlerin davranışlarını taklit eder (Shi ve Eberhart, 2008). Kuşlar ve balıklar, yiyecek ve eşleri aramak, sıcaklık gibi çevresel parametreleri optimize etmek için fiziksel hareketlerini ayarlar.

PSO, çok basit bir kavramdan oluşur ve paradigmlar birkaç satır bilgisayar kodunda uygulanabilir. Sadece ilkel matematiksel operatörler gerektirir ve hem bellek gereksinimleri hem de hız açısından hesaplama açısından ucuzdur (Kennedy ve Eberhart, 1995).

PSO algoritması çok hızlı bir yakınsama sağlamaktadır. Bu hızlı yakınsamanın sonucu olarak popülasyondaki bireyler en iyi birey etrafında toplanır ve eğer ki en iyi birey optimum değerden uzaksa, popülasyon lokal bir değere takılabilir. Bu durumu düzeltmek için keşif ve sömürü özelliklerinin kontrolü gereklidir. Bu iki durumun bir dengede ilerlemesi sağlanabilirse, popülasyonun yerel bir yerde takılıp kalması engellenebilir. Keşif ve sömürünün dengesinin kontrolü için de popülasyon çeşitlilik değeri kullanılabilir. Çeşitlilik ne kadar düşük ise sömürü yüksek ve çeşitlilik ne kadar yüksek ise keşif yüksek olacaktır.

Bu erken yakınsama probleminden dolayı birçok PSO algoritması geliştirilmiştir. Bu algoritmaları örnek olarak; Catfish PSO (Chuang ve ark., 2008), MBBPSO (Wang ve ark., 2013b), NQPSO (Fu ve ark., 2013), HPSO-DE (Yu ve ark., 2014) vs. algoritmaları verilebilir. Bu algoritmalarda erken yakınsama sonucunda lokal bir değere takılıp kalmamak için keşif ve sömürü arasında denge sağlanmaya çalışılmıştır.

Algoritma temel olarak aşağıdaki basamaklardan oluşur;

- i. Rasgele üretilen başlangıç pozisyonları ve hızları ile başlangıç sürüsü oluşturulur.
- ii. Sürü içerisindeki tüm parçacıkların uygunluk değerleri hesaplanır.
- iii. Her bir parçacık için mevcut jenerasyondan yerel en iyi (pbest) bulunur. Sürü içerisinde en iyilerin sayısı parçacık sayısı kadardır.
- iv. Mevcut jenerasyondaki yerel en iyiler içerisinde küresel en iyi (gbest) seçilir.
- v. Pozisyon ve hızlar, Denklem 3.1 ve Denklem 3.2' deki gibi yenilenir.
- vi. v. adımdan sonra, kriter yerine getirilene kadar yada maksimum iterasyon sayısına kadar, ii. adım ile v. adım arasındaki işlemler tekrar edilir.

$$V_i(k+1) \leftarrow V_i(k) + C_1\phi_1(P_i(k) - X_i(k)) + C_2\phi_2(G(k) - X_i(k)) \quad (3.1)$$

$$X_i(k+1) \leftarrow X_i(k) + V_i(k) \quad (3.2)$$

$V_i = [V_{i1}, V_{i2}, \dots, V_{iD}]$, i parçacığının hız vektörü

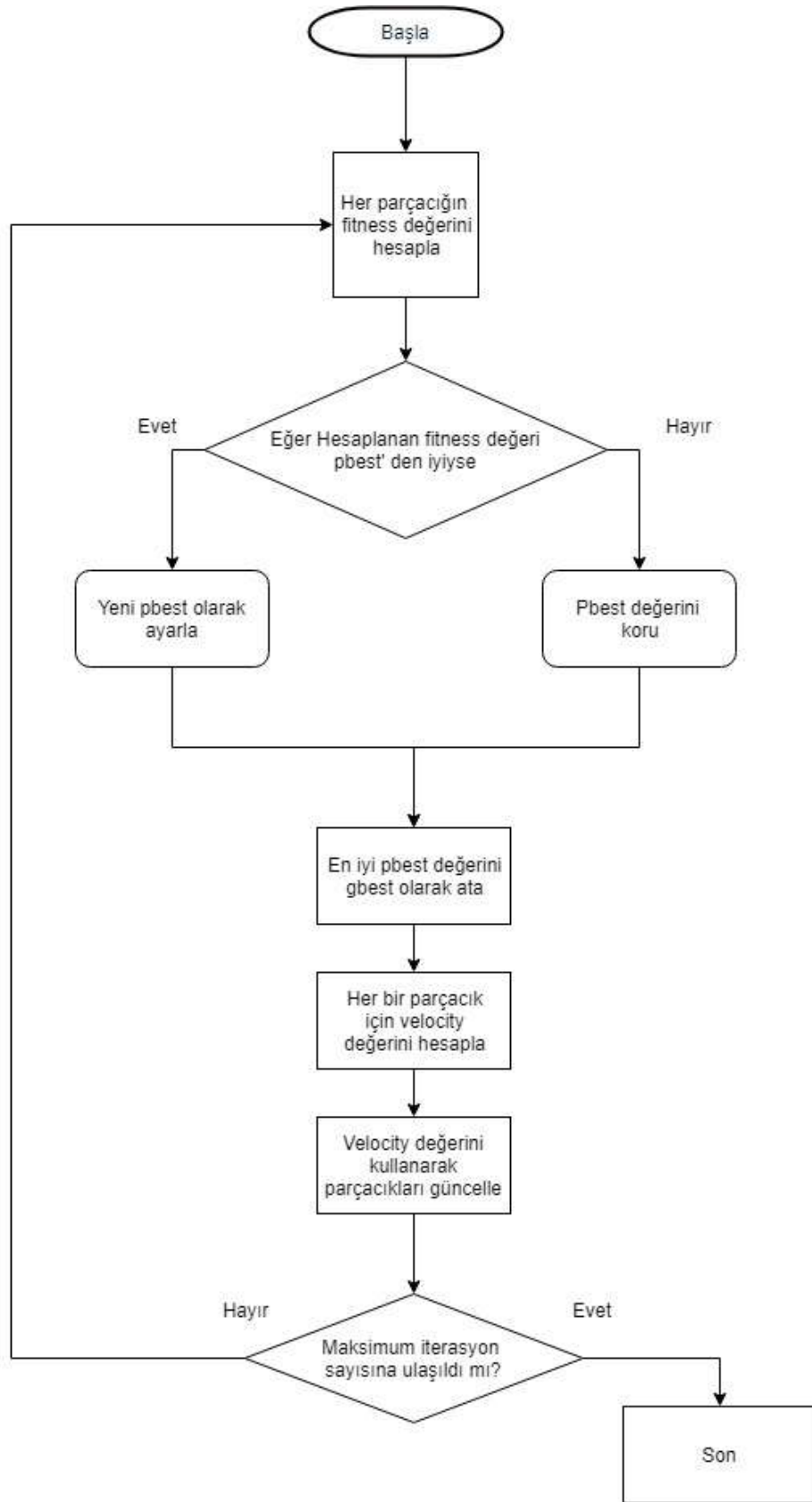
$X_i = [X_{i1}, X_{i2}, \dots, X_{iD}]$, i parçacığının pozisyon vektörü

ϕ_1 ve ϕ_2 : her bir boyut(d) için farklılık gösteren eşit şekilde dağıtılmış rastgele sayılardır.

Her bir parçacık X_i ile ilişkili olarak, geçmiş iterasyonlardan birinde en iyi fonksiyon değerini alan bir lokal en iyi (pbest) parçacığı vardır. Popülasyonun tamamıyla ilişkili olarak da küresel en iyi (gbest) bulunmaktadır.

C_1 ve C_2 , ilerleyiş yönünü pbest ya da gbest doğru çeken sabitlerdir (Muthiah-Nakarajan ve Noel, 2016).

PSO' ya ait akış şeması Şekil 3.1' de verilmiştir.



Şekil 3.1. PSO algoritması akış diyagramı

Yapılan tez çalışmasında, standart PSO algoritması değil de lineer azalan inertia weight PSO algoritması kullanılmıştır. Bu algoritmanın standart PSO algoritmasından farkı, bir atalet ağırlığı (inertia weight) sabitinin işlemlere dâhil edilmesidir. Atalet ağırlığı, keşif ve sömürü süreci arasında denge sağlama sürecinde kilit bir rol oynar. Atalet ağırlığı, bir parçacığın mevcut hızdaki önceki hızının katkısını belirler (Bansal ve ark., 2011). Büyük bir atalet ağırlığı küresel bir aramayı kolaylaştırırken, küçük bir atalet ağırlığının ise yerel bir aramayı kolaylaştırdığı söylenebilir. Atalet ağırlığının PSO' ya etkisi Denklem 3.3 ve Denklem 3.4 incelenerek gözlemlenebilir. Denklem 3.3' de hız hesaplanırken, önceki hıza bir w sabiti ile etki yapılarak, keşif ve sömürü arasındaki denge sağlanmış olacaktır.

$$V_i(k+1) \leftarrow w * V_i(k) + C_1 \phi_1 (P_i(k) - X_i(k)) + C_2 \phi_2 (G(k) - X_i(k)) \quad (3.3)$$

$$X_i(k+1) \leftarrow X_i(k) + V_i(k) \quad (3.4)$$

Denklem 3.3 ve Denklem 3.4 ile düzenlemiş PSO algoritmasının sözde kodu Şekil 3.2' gösterilmiştir.

inertia weight PSO	
<hr/>	
1: random bir popülasyon oluştur.	
2: for t=1:max iterasyon	
3: for i=1:popülasyon büyüklüğü	
4: if $f(X_{i,d}(t)) < f(p_i(t))$ then $p_i(t) = X_{i,d}(t)$	
5: $f(p_g(t)) = \min_i(f(p_i(t)))$	
6: end	
7: for d=1:boyut	
8: $V_{i,d}(t+1) \leftarrow w * V_{i,d}(t) + C_1 \phi_1 (p_i(t) - X_{i,d}(t)) + C_2 \phi_2 (p_g - X_{i,d}(t))$	
9: $X_{i,d} = X_{i,d}(t) + V_{i,d}(t+1)$	
10: if $V_{i,d}(t+1) > V_{max}$ then $V_{i,d}(t+1) = V_{max}$	
11: else if $V_{i,d}(t+1) < V_{min}$ then $V_{i,d}(t+1) = V_{min}$	
12: end	
13: if $X_{i,d}(t+1) > X_{max}$ then $X_{i,d}(t+1) = X_{max}$	
14: else if $X_{i,d}(t+1) < X_{min}$ then $X_{i,d}(t+1) = X_{min}$	
15: end	
16: end	
17: end	
18: end	

Şekil 3.2. inertia weight PSO sözde kodu

3.3. Galaktik Sürü Optimizasyonu

Galaktik Sürü Optimizasyonu (GSO), 2016 yılında Muthiah-Nakarajan ve arkadaşları tarafından geliştirilmiştir. GSO, galaksiler, yıldızlar ve bunların birbirleri ile olan ilişkilerinden esinlenilerek ortaya konulmuş bir optimizasyon yöntemidir.

Yıldızlar uzayda homojen bir şekilde dağılmamakta ve aynı zamanda homojen bir şekilde dağılmamış olan galaksiler halinde kümelenmektedir. Yeterince büyük ölçekteki galaksiler, nokta kitleler olarak görünür. Bir galaksinin içindeki yıldızların büyük kütlelere ve galaksilere kendilerini çekmesi, GSO algoritmasında şu şekilde uyarlanmıştır: İlk olarak, her alt popülasyondaki bireyler, PSO algoritmasına göre alt popülasyonda daha iyi çözümlere çekilir. İkincisi, her bir alt popülasyonun alt popülasyon tarafından bulunan en iyi çözümlerle temsil edildiği ve bir üst-perde olarak ele alındığı varsayılmaktadır. Her alt popülasyonda bulunan en iyi çözümlerden oluşan süper swarmdaki birey de PSO algoritmasına göre hareket eder (Muthiah-Nakarajan ve Noel, 2016).

GSO algoritmasında swarm, her biri N boyutunda, subswarm X_i olarak adlandırılan M parçalarından meydana gelen, $(X_j^{(i)} \in R^D)$ elemanlarını içeren bir set X dizisidir. X kümesinin elemanları, $[X_{min}, X_{max}]D$ arama uzayında rastgele oluşturulmaktadır.

Genel tanımlamalar Denklem 3.5, Denklem 3.6, Denklem 3.7 ve Denklem 3.8' de verilmiştir (Muthiah-Nakarajan ve Noel, 2016).

$$X_i \in X: i = 1, 2, \dots, M \quad (3.5)$$

$$X_j^{(i)} \in X_i: j = 1, 2, \dots, N \quad (3.6)$$

$$X_i \cap X_j = \emptyset: \text{if } i \neq j \quad (3.7)$$

$$\bigcup_{i=1}^M X_i = X \quad (3.8)$$

GSO algoritmasının çalışması ele alındığında; öncelikle rastgele şekilde bir popülasyon oluşturulmaktadır. Bu popülasyon, daha sonra 10 alt popülasyona bölünür. Bu alt popülasyonlar ayrı ayrı PSO algoritması altında çalıştırılır. Bu alt popülasyonlardan gelen gbest değerleri, level-2 deki popülasyonu oluşturur. Bu ikinci aşamada gelen popülasyon tekrar PSO algoritması altında çalıştırılır.

GSO algoritmasının sözde kodu Şekil 3.3' de verilmiştir.

GSO algoritması	
1:	Amaç Fonksiyonu: $f(x)$, $x = \{x_1, x_2, \dots, x_d\}$
2:	Level-1 popülasyon tanımları: $[X_{\min}, X_{\max}]^D$ kümesi içerisinde $X1$ oluşturulur.
3:	Level-1 değişkenler tanımları: $[X_{\min}, X_{\max}]^D$ kümesi içerisinde $V1_{ij}, P1_{ij}, G1_{ij}$ oluşturulur.
4:	Level-2 popülasyon tanımları: $[X_{\min}, X_{\max}]^D$ kümesi içerisinde $X2$ oluşturulur.
5:	Level-2 değişkenler tanımları: $[X_{\min}, X_{\max}]^D$ kümesi içerisinde $V2_i, P2_i, G2$ oluşturulur.
6:	for EP = 1 to EP _{max}
7:	for i = 1 to Parçacık Sayısı
8:	for k = 1 to Level-1 İterasyon Sayısı
9:	for j = 1 to Parçacığın Boyutu
10:	$V1_{ij} \leftarrow w_1 * V1_{ij} + C_1\phi_1(P1_{ij} - X1_{ij}) + C_2\phi_2(G1_i - X1_{ij})$
11:	$X1_{ij} \leftarrow X1_{ij} + V1_{ij}$
12:	if ($f(X1_{ij}) < f(P1_{ij})$) then
13:	$P1_{ij} \leftarrow X1_{ij}$
14:	if ($f(P1_{ij}) < f(G1_i)$) then
15:	$G1_i \leftarrow P1_{ij}$
16:	if ($f(G1_i) < f(G2)$) then
17:	$G2 \leftarrow G1_i$
18:	end
19:	end
20:	end
21:	end
22:	end
23:	end
24:	end
25:	Level-2 başlatılır
26:	for j = 1 to Parçacık Sayısı
27:	$X2_j \leftarrow G1_j$
28:	end
29:	for k = 1 to Level-2 İterasyon Sayısı
30:	for i = 1 to Parçacık Sayısı
31:	$V2_i \leftarrow w_2 * V2_i + C_3\phi_3(P2_i - X2_i) + C_4\phi_4(G2_i - X2_i)$
32:	$X2_i \leftarrow X2_i + V2_i$
33:	if ($f(X2) < f(P2)$) then
34:	$P2 \leftarrow X2$
35:	$G2 \leftarrow P2$
36:	end
37:	end
38:	end
39:	end
40:	Return G2

Şekil 3.3. GSO algoritması sözde kodu

3.4. Popülasyon Çeşitliliği

Popülasyon çeşitliliği, algoritmanın çalışması sırasında bireylerin farklı varyasyon değerlerinin ölçülmesidir. Popülasyon belirli bir alanda toplanmışsa, çeşitlilik düşük olacaktır ve sömürü yeteneği yüksek çıkacaktır. Ancak popülasyon geniş bir alana dağılmış ise çeşitlilik değeri ve keşif yeteneği yüksek olacaktır. Çeşitlilik kavramı için iki önemli ifade yer almaktadır. Bunlar keşif ve sömürü olarak literatürde yer almaktadır.

Keşif (Exploration): Arama uzayını derinlemesine araştırarak, farklı çözümler aranmasıdır. Popülasyon çeşitliliğinin yüksek olması, keşif yeteneğinin güçlü olmasını sağlamaktadır.

Sömürü (Exploitation): Daha yüksek uygunluk derecesine sahip bireylere yoğunlaşır ve bu çözümün çevresini araştırır. Sömürü, sezgisel olarak elimizdeki en iyi çözümden daha iyi bir çözümün yoğun olarak aranması işlemidir.

Optimizasyonun daha iyi sonuçlar verebilmesi için bu iki kavramın, dengede olması gerekmektedir. Diğer bir deyişle, ne popülasyon gereksiz yere uzaya yayılmalı, ne de tek bir noktada toplanmalıdır.

Popülasyon çeşitliliği hesaplanırken, birkaç farklı yöntem kullanılmaktadır. Bu yöntemler; pozisyon tabanlı çeşitlilik, hız tabanlı çeşitlilik ve kognitif (bilişsel) tabanlı çeşitlilik olarak gösterilebilir (Cheng, 2013).

3.4.1. Pozisyon tabanlı popülasyon çeşitliliği

Konum çeşitliliği mevcut parçacık dağılım bilgisini verir ve parçacıkların ayrışıp yakınsandığı, bu ölçümden algılanabilir. Pozisyon tabanlı çeşitlilik, parçacıkların mevcut konumlarının dağılımını ölçer, bu nedenle parçacıkların dinamiklerini yansıtabilir. Çeşitlilik ölçümlerinden, yararlı arama bilgileri elde edilebilir (Cheng, 2013).

Pozisyon tabanlı çeşitlilik için, bir bireyin pozisyonu ve tüm bireylerin pozisyon matrisi Denklem 3.9 ve Denklem 3.10' da verilmiştir.

$$x_i = \{x_{i1}, x_{i2}, \dots, x_{in}\}, i = 1, 2, \dots, m \quad (3.9)$$

$$X = (x_1, x_2, \dots, x_m)^T = \begin{pmatrix} x_{11} & \dots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \dots & x_{mn} \end{pmatrix} \quad (3.10)$$

Pozisyon tabanlı bir çeşitlilik hesabı için birkaç farklı yol izlenebilir. Öklid uzaklık tabanlı, boyut tabanlı ya da element tabanlı olarak bu hesaplamalar yapılabilir (Shi ve Eberhart, 2008).

3.4.1.1. Öklid tabanlı popülasyon çeşitliliği

Bu yöntemde, popülasyon üyeleri arasındaki Öklid mesafesine bağlı olarak, popülasyon çeşitliliği hesaplanmaktadır (Shi ve Eberhart, 2008). Öklid mesafesi ve popülasyon çeşitliliği hesabı, Denklem 3.11, Denklem 3.12 ve Denklem 3.13' e göre hesaplanmaktadır.

$$d^p(x_i, x_j) = \|x_i - x_j\| \quad (3.11)$$

$$\tilde{d}^p(x_i, x_j) = \frac{d^p(x_i, x_j)}{\|a - b\|} \quad (3.12)$$

$$D_{ED}^p = \frac{2}{n(n-1)} \sum_{i=1}^n \sum_{j=i+1}^n \tilde{d}^p(x_i, x_j) \quad (3.13)$$

Denklem 3.11' de, $d^p(x_i, x_j)$, x_i, x_j bireyleri arasındaki, Öklid mesafesini ifade etmektedir. x_i, x_j bireyleri arasındaki Öklid mesafesi Denklem 3.14' e göre hesaplanmaktadır.

$$d^p(x_i, x_j) = \sqrt{\sum_{k=1}^n (x_{ik} - x_{jk})^2} \quad (3.14)$$

Hesaplanan Öklid mesafesi, Denklem 3.12' e göre normalize edilmektedir. Belirtilen denklem içerisindeki [a,b] dinamik aralığı ifade etmektedir. Popülasyondaki bireylerin alabileceği maksimum ve minimum değerleri ifade etmektedir. Denklem 3.13' de bireyler arasındaki normalize edilmiş Öklid mesafelerinin ortalaması, popülasyon çeşitliliği (D_{ED}^p) olarak ifade edilmektedir.

3.4.1.2. Element tabanlı popülasyon çeşitliliği

Bu yöntemde, popülasyondaki bütün bireylerin aldıkları değerlerin ortalaması kullanılarak, popülasyon çeşitliliği hesaplanmaktadır. Popülasyondaki bireylerin ortalaması ve popülasyon çeşitliliği, Denklem 3.15 ve Denklem 3.16' ya göre hesaplanmaktadır.

$$\bar{x} = \frac{1}{n \times m} \sum_{i=1}^m \sum_{j=1}^n x_{ij} \quad (3.15)$$

$$D^p = \frac{1}{n \times m} \sum_{i=1}^m \sum_{j=1}^n [x_{ij} - \bar{x}]^2 \quad (3.16)$$

Bütün bireylerin ortalaması (\bar{x}), Denklem 3.15' de ifade edilmektedir. Hesaplanan bu değerin, popülasyondaki bütün bireylerin değerleri ile farkının ortalaması ise popülasyon çeşitliliğini (D^p) ifade etmektedir.

3.4.1.3. Boyut tabanlı popülasyon çeşitliliği

Bu yöntemde, popülasyonun çeşitliliği boyut tabanlı olarak hesaplanmaktadır. Popülasyondaki her boyutun çeşitlilik değerleri, Denklem 3.17 ve Denklem 3.18' e göre hesaplanmaktadır.

$$\bar{x}_j = \frac{1}{m} \sum_{i=1}^m x_{ij} \quad (3.17)$$

$$D_j^p = \frac{1}{m} \sum_{i=1}^m [x_{ij} - \bar{x}_j]^2 \quad (3.18)$$

Popülasyondaki j. boyutun ortalaması (\bar{x}_j), Denklem 3.17' ye göre hesaplanmaktadır. Bir boyutun çeşitliliği hesaplanırken, o boyuttaki bütün değerlerden, o boyutun ortalamasının (\bar{x}_j) farklarının karelerinin ortalaması hesaplanır. Bu şekilde, her bir boyuta ait boyut çeşitliliği değerleri elde edilir. Bu değerler kullanılarak, popülasyon çeşitliliğini hesaplamak için çeşitli yöntemler kullanılmaktadır. Bu yöntemler incelendiğinde:

- Ağırlıklı toplam ile boyut tabanlı popülasyon çeşitliliği yönteminde, Denklem 3.18 ile elde edilen boyutsal çeşitlilik değerlerinin her biri bir ağırlık sabitiyle çarpılarak ortalama alınmaktadır. Bu işlemdeki hesaplama Denklem 3.19' a göre yapılmaktadır.

$$D_{WS}^p = \sum_{j=1}^n w_j D_j^p \quad (3.19)$$

Seçilen her bir ağırlık değişkeni (w_j), 1'den küçük ya da 1'e eşit olmalıdır. Tüm boyutlar eşit şekilde ele alınırsa, tüm j değerleri için $w_j = 1/n$ olur (Shi ve Eberhart, 2008). $w_j = 1/n$ şeklinde bir düzenleme yapılarak, popülasyon çeşitliliği Denklem 3.20'ye göre yapılmaktadır.

$$D_{WS}^p = \frac{1}{n} \sum_{j=1}^n D_j^p \quad (3.20)$$

- Ağırlık ve maksimizasyon ile boyut tabanlı popülasyon çeşitliliği yönteminde, Denklem 3.18' e göre hesaplanan boyutsal çeşitlilik değerleri, bir ağırlık değişkeniyle genişletilerek, bu dizi içerisindeki en büyük değer seçimi sonucunda, popülasyon çeşitliliği hesaplanmaktadır. Bu hesaplama Denklem 3.21' e göre yapılmaktadır.

$$D_{WM}^p = \max\{w_j D_j^p\} \quad j = 1, \dots, n \quad (3.21)$$

- Vektör uzunluğu ile boyut tabanlı popülasyon çeşitliliği yönteminde popülasyon çeşitliliği, Denklem 3.18' e göre hesaplanan boyutsal çeşitlilik değerlerinin karelerinin toplamlarının karekökü alınarak Denklem 3.22' ye göre hesaplanmaktadır.

$$D_L^p = \sqrt{\sum_{j=1}^n D_j^{p^2}} \quad (3.22)$$

Boyutsal olarak popülasyon çeşitliliği, önce popülasyondaki değerleri normalize ederek hesaplanabilir. Normalizasyon yöntemiyle boyutsal popülasyon çeşitliliği Denklem 3.23, Denklem 3.24 ve Denklem 3.25' e göre hesaplanmaktadır.

$$\hat{x}_{ij} = \frac{x_{ij}}{|b_j - a_j|} \quad a_j \leq x_{ij} \leq b_j; \quad i = 1, \dots, m; \quad j = 1, \dots, n \quad (3.23)$$

$$\bar{x}_j = \frac{1}{m} \sum_{i=1}^m \hat{x}_{ij} \quad (3.24)$$

$$D_{jN}^p = \frac{1}{m} \sum_{i=1}^m [\hat{x}_{ij} - \bar{x}_j]^2 \quad (3.25)$$

Popülasyondaki her değer, Denklem 3.23' e göre normalize edilmektedir. a_j ve b_j , j . boyuttaki maksimum ve minimum değerlerdir. Bu işlem sonucunda işlem yapılan boyuttaki bütün elemanlar 0-1 aralığına taşınmış olacaktır. Normalize edilmiş değerlerin ortalaması (\bar{x}_j), Denklem 3.24' e göre hesaplanmaktadır. Denklem 3.25 sonucunda hesaplanan boyutsal çeşitlilik değerleri, Denklem 3.18 için uygulanan popülasyon çeşitliliği yöntemleri kullanılarak popülasyon çeşitliliği hesaplanmaktadır.

3.4.2. Hız tabanlı popülasyon çeşitliliği

Parçacıkların “hareket etme potansiyeli” çeşitliliğini temsil eden hız çeşitliliği, parçacıkların mevcut hızlarının dağılımını ölçmektedir. Başka bir deyişle, hız çeşitliliği parçacıkların faaliyet bilgisini ölçmektedir. Hız çeşitliliğinin ölçümüne dayanarak, parçacığın genişleme ve ya yakınsama eğilimi ortaya çıkarılabilir (Cheng, 2013).

Hız tabanlı çeşitlilik için bir bireyin hızı ve tüm bireylerin hız matrisi Denklem 3.26 ve Denklem 3.27' de verilmiştir.

$$v_i = \{v_{i1}, v_{i2}, \dots, v_{in}\}, i = 1, 2, \dots, m \quad (3.26)$$

$$V = (v_1, v_2, \dots, v_m)^T = \begin{pmatrix} v_{11} & \dots & v_{1n} \\ \vdots & \ddots & \vdots \\ v_{m1} & \dots & v_{mn} \end{pmatrix} \quad (3.27)$$

Hız tabanlı popülasyon çeşitliliğini hesaplamak için farklı yöntemler kullanılabilir. Bunlara örnek olarak element tabanlı popülasyon çeşitliliği ve boyut tabanlı popülasyon çeşitliliği yöntemleri söylenebilir.

3.4.2.1. Element tabanlı popülasyon çeşitliliği

Bu yöntemde, popülasyondaki bütün elementlerin hız değerlerinin ortalaması kullanılarak popülasyon çeşitliliği hesaplanmaktadır. Popülasyon çeşitliliği, Denklem 3.28 ve Denklem 3.29' a göre hesaplanmaktadır.

$$\bar{v} = \frac{1}{n \times m} \sum_{i=1}^m \sum_{j=1}^n v_{ij} \quad (3.28)$$

$$D_E^v = \frac{1}{n \times m} \sum_{i=1}^m \sum_{j=1}^n |v_{ij} - \bar{v}| \quad (3.29)$$

Elemen tabanlı çeşitliliğin hesaplanması için öncelikle popülasyondaki bütün elementlerin hız değerlerinin ortalaması (\bar{v}), Denklem 3.28' e göre hesaplanmaktadır. Hesaplanan bu ortalama değerinin tüm popülasyondaki hız değerlerine farklarının ortalaması da popülasyon çeşitlilik değerini vermektedir.

3.4.2.2. Boyut tabanlı popülasyon çeşitliliği

Bu yöntemde popülasyon çeşitliliği boyutsal olarak hesaplanmaktadır. Boyutsal olarak popülasyon çeşitliliği Denklem 3.30, Denklem 3.31 ve Denklem 3.32' ye göre hesaplanmaktadır.

$$\bar{v}_j = \frac{1}{m} \sum_{i=1}^m v_{ij} \quad (3.30)$$

$$D_j^v = \frac{1}{m} \sum_{i=1}^m |v_{ij} - \bar{v}_j| \quad (3.31)$$

$$D^v = \sum_{j=1}^n w_j D_j^v \quad (3.32)$$

Popülasyondaki hız değerleri matrisindeki her bir boyutun ortalaması (\bar{v}_j), Denklem 3.30' a göre hesaplanmaktadır. Bu ortalama değer, her bir boyut için boyuttaki hız değerlerinden farkının ortalaması alınarak, Denklem 3.31' e göre boyutsal olarak çeşitlilik değerlerini verir. Hesaplanan bu boyutsal çeşitlilik değerleri (D_j^v), bir ağırlık değişkeni (w_j) ile çarpılarak ortalamalarının alınması sonucunda, popülasyon çeşitliliği (D^v) hesaplanmaktadır.

Seçilen her bir ağırlık değişkeni (w_j), 1'den küçük ya da 1'e eşit olmalıdır. Tüm boyutlar eşit şekilde ele alınırsa, tüm j değerleri için $w_j = 1/n$ olur (Shi ve Eberhart,

2008). $w_j = 1/n$ şeklinde bir düzenleme yapılarak, popülasyon çeşitliliği Denklem 3.33' ye göre yapılmaktadır.

$$D^v = \sum_{j=1}^n \frac{1}{n} D_j^v = \frac{1}{n} \sum_{j=1}^n D_j^v \quad (3.33)$$

3.4.3. Kognitif tabanlı popülasyon çeşitliliği

Parçacıkların hareketli hedeflerinin dağılımını temsil eden kognitif (bilişsel) çeşitlilik, tüm parçacıklar için en iyi pozisyonların dağılımını ölçer. Kognitif çeşitliliğin ölçülmesi, her parçanın mevcut pozisyonu yerine mevcut en iyi kişisel pozisyonunu kullanması dışında pozisyon çeşitliliğiyle aynıdır (Cheng, 2013).

Kognitif tabanlı çeşitlilik için, bir bireyin en iyi durumu ve tüm bireylerin en iyi durum matrisi Denklem 3.34 ve Denklem 3.35' de verilmiştir.

$$p_i = \{p_{i1}, p_{i2}, \dots, p_{in}\}, i = 1, 2, \dots, m \quad (3.34)$$

$$P = (p_1, p_2, \dots, p_m)^T = \begin{pmatrix} p_{11} & \dots & p_{1n} \\ \vdots & \ddots & \vdots \\ p_{m1} & \dots & p_{mn} \end{pmatrix} \quad (3.35)$$

Kognitif tabanlı popülasyon çeşitliliği hesaplama yöntemlerine örnek olarak element tabanlı popülasyon çeşitliliği ve boyut tabanlı popülasyon çeşitliliği yöntemleri olarak söylenebilir.

3.4.3.1. Element tabanlı popülasyon çeşitliliği

Bu yöntemde popülasyon çeşitliliği, kognitif matristeki tüm elementlerin değerlerinin ortalaması kullanılarak Denklem 3.36 ve Denklem 3.37' ye göre hesaplanmaktadır.

$$\bar{p} = \frac{1}{n \times m} \sum_{i=1}^m \sum_{j=1}^n p_{ij} \quad (3.36)$$

$$D_E^c = \frac{1}{n \times m} \sum_{i=1}^m \sum_{j=1}^n |p_{ij} - \bar{p}| \quad (3.37)$$

Kognitif matrisin ortalaması (\bar{p}), Denklem 3.36' ya göre hesaplanmaktadır. Hesaplanan bu ortalama değerinin, tüm elementlerden farklarının ortalaması, kognitif tabanlı olarak popülasyon çeşitlilik (D_E^c) değerini vermektedir.

3.4.3.2. Boyut tabanlı popülasyon çeşitliliği

Bu yöntemde kognitif tabanlı çeşitlilik, boyutsal olarak hesaplanmaktadır. Bu hesaplamalar, Denklem 3.38, Denklem 3.39 ve Denklem 3.40' a göre yapılmaktadır.

$$\bar{p}_j = \frac{1}{m} \sum_{i=1}^m p_{ij} \quad (3.38)$$

$$D_j^c = \frac{1}{m} \sum_{i=1}^m |p_{ij} - \bar{p}_j| \quad (3.39)$$

$$D^c = \sum_{j=1}^n w_j D_j^c \quad (3.40)$$

Kognitif matrisindeki boyutların ortalama değerleri (\bar{p}_j), Denklem 3.38' e göre hesaplanmaktadır. Kognitif matristeki her bir boyutun çeşitliliği, ortalama değerlerinin (\bar{p}_j), j. boyuttaki tüm değerlerden farkının ortalaması ile Denklem 3.39' a göre yapılmaktadır. Hesaplanan bu boyutsal çeşitlilik değerlerinin bir ağırlık değişkeni (w_j) ile çarpılarak ortalamaları popülasyon çeşitliliğini (D^c) vermektedir.

Seçilen her bir ağırlık değişkeni (w_j), 1'den küçük ya da 1'e eşit olmalıdır. Tüm boyutlar eşit şekilde ele alınırsa, tüm j değerleri için $w_j = 1/n$ olur (Shi ve Eberhart, 2008). $w_j = 1/n$ şeklinde bir düzenleme yapılarak, popülasyon çeşitliliği Denklem 3.41' e göre yapılmaktadır.

$$D^c = \sum_{j=1}^n \frac{1}{n} D_j^c = \frac{1}{n} \sum_{j=1}^n D_j^c \quad (3.41)$$

3.5. Önerilen Yöntem

GSO algoritması, galaksideki yıldızlar ve yıldız kümelerinin hareketlerinden esinlenerek ortaya konulmuş bir optimizasyon yöntemidir. Yöntem diğer bilinen optimizasyon yöntemlerini kullanarak, iki aşamada en iyi çözümü bulmayı amaçlamaktadır. İlk aşama arama uzayında genel bir keşif yaparken, ikinci aşama elde edilen en iyi çözümü iyileştirmeye çalışmaktadır. GSO' da birinci aşamadan elde edilen iyi bireylerin oluşturduğu popülasyon ikinci aşamaya başlangıç popülasyonu olarak verilmektedir. Bu işlem belirlenen değer kadar tekrar edilmektedir. GSO da birinci aşamadan elde edilen bilgiler ikinci aşamaya aktarılırken, ikinci aşamadan birinci aşamaya herhangi bir bilgi aktarımı olmamaktadır. Bu tezde, ikinci aşamada elde edilen bilgi önerilen modifikasyon ile birinci aşamaya aktarılmaktadır. Orijinal çalışmada arama yöntemi olarak PSO yöntemi kullanılmıştır. PSO yöntemi hızlı yakınsama problemine sahip bir algoritmadır. Bundan dolayı ilk aşamada yeterince keşif yapamama sıkıntısı ortaya çıkmaktadır. Bu noktada popülasyon çeşitliliği ölçülerek, daha iyi bir keşif kabiliyetine sahip modifikasyon yapılmıştır.

GSO yönteminin ilk aşamasında birbirinden bağımsız popülasyon grupları PSO algoritması ile arama uzayını taramaktadırlar. Diğer bir deyişle, ilk aşamada algoritma keşif yapmaktadır. İlk aşamada elde edilen en iyi sonuçların oluşturduğu popülasyonla, ikinci aşamada ise sömürü yapılmaktadır. Bu noktada her bir epokta birinci aşamadan ikinci aşamaya bir bilgi aktarımı mevcuttur. Fakat yeni bir epok başladığında ikinci aşamadan elde edilen bilgi birikimi birinci aşamaya aktarılmamaktadır. Bu tezde, ikinci aşamadaki bilgi birikimini birinci aşamaya aktaracak bir modifikasyon sunulmuştur. İlk epok hariç diğer bütün epoklarda, ikinci aşamada yapılan arama sonucunda elde edilen bilgi birikimi, ikinci aşamadaki bireylerin bir sonraki epokta, birinci aşamaya transferi ile aktarılmaktadır.

PSO' da, algoritmanın doğası gereği, bireyler kendi en iyi ve küresel en iyi değeri takip etmektedirler. Bunun sonucunda, PSO' da erken yakınsama problemi ortaya çıkmaktadır. Erken yakınsama probleminde yöntem lokal minimum noktalara takılmakta ve arama uzayını iyi bir şekilde keşfedememektedir.

Popülasyon tabanlı yöntemlerde popülasyon çeşitliliği, yöntemin keşif ve sömürü kabiliyetleri hakkında bilgi vermektedir. Bununla birlikte, literatürde popülasyon çeşitliliği kullanılarak popülasyon tabanlı yöntemlerin keşif ve sömürü dengesini ayarlayan yöntemler mevcuttur. GSO algoritmasının ilk aşamasında keşif, ikinci

aşamasında sömürü amaçlanmaktadır. Bu nedenle, birinci aşamada popülasyon çeşitliliğinin yüksek olması hedeflenmektedir. Fakat PSO erken yakınsama eğiliminde olduğundan ilk epok sonunda birinci aşamadaki popülasyon gruplarının çeşitliliği oldukça düşmektedir. Bu tez çalışmasında, her epok sonunda birinci aşamadaki popülasyon gruplarının çeşitliliği ölçülerek belirli bir değerin altına düştüğünde, var olan bireyler yerine rastgele bireyler oluşturularak popülasyon çeşitliliği artırılmaktadır.

Literatürde bir popülasyonun çeşitliliğini ölçmek için farklı yöntemler bulunmaktadır. Bu çalışmada, pozisyon tabanlı çeşitlilik ölçütlerinden Öklid tabanlı pozisyon çeşitliliği (D_{ED}^p) kullanılmıştır. Öklid tabanlı pozisyon çeşitliliği ölçütünde, popülasyondaki her parçacık arasındaki Öklid mesafesi ($d^p(x_i, x_j)$) ölçülmektedir. Hesaplanan bu Öklid mesafeleri, normalize edilerek, normalize edilmiş Öklid mesafeleri ($\tilde{d}^p(x_i, x_j)$) hesaplanmaktadır. Son olarak, normalize edilmiş öklid mesafelerinin ortalaması hesaplanır ve popülasyon çeşitliliği (D_{ED}^p) olarak kullanılır. Öklid tabanlı popülasyon çeşitliliği işlemleri Denklem 3.11, Denklem 3.12 ve Denklem 3.13' e göre yapılmaktadır.

Önerilen yöntem iki modifikasyondan oluşmaktadır. İlk modifikasyon, ikinci fazda elde edilen çözümlerin, sonraki epok' un başlangıcında birinci fazdaki alt-popülasyonlara transferidir. Böylece ikinci fazdan birinci faza bir geri besleme yapılmaktadır. İkinci modifikasyonda ise, birinci fazdaki alt-popülasyon gruplarının çeşitlilikleri ölçülmekte ve belirlenen bir değerin altında ise rastgele oluşturulan yeni bireyler popülasyona eklenerek, çeşitlilik korunmaya çalışılmaktadır. Bu işlem sayesinde, algoritmanın ilk fazı daha etkin bir keşfetme gerçekleştirebilir ve ikinci aşamaya aktarılacak değerler de, istenilen optimum sonucu bulmak için daha uygun değerler olabilir.

Önerilen yönteme ait sözde kod Şekil 3.4' de verilmiştir.

Popülasyon çeşitliliği kontrollü geri beslemeli GSO algoritması	
1:	Amaç Fonksiyonu: $f(x)$, $x = \{x_1, x_2, \dots, x_d\}$
2:	Level-1 popülasyon tanımları: $[X_{\min}, X_{\max}]^D$ kümesi içerisinde X_1 oluşturulur.
3:	Level-1 değişkenler tanımları: $[X_{\min}, X_{\max}]^D$ kümesi içerisinde $V_{1ij}, P_{1ij}, G_{1ij}$ oluşturulur.
4:	Level-2 popülasyon tanımları: $[X_{\min}, X_{\max}]^D$ kümesi içerisinde X_2 oluşturulur.
5:	Level-2 değişkenler tanımları: $[X_{\min}, X_{\max}]^D$ kümesi içerisinde V_{2i}, P_{2i}, G_2 oluşturulur.
6:	for EP = 1 to EP _{max}
7:	for i = 1 to Parçacık Sayısı
8:	if (EP != 1) then
9:	if ($(D_{ED}^p(X_{1i}) < \theta)$) then
10:	$X_{1i1} \leftarrow P_{2i}$
11:	for p = 2 to Parçacığın Boyutu
12:	$X_{1ip} \leftarrow \text{Rasgele_Populasyon_Uret}()$
13:	if ($f(X_{1ip}) < f(P_{1ip})$) then
14:	$P_{1ip} \leftarrow X_{2ip}$
15:	end
16:	end
17:	else
18:	worst_id = Find_Worst_id(X_{1i})
19:	if ($f(P_{2i}) < f(P_{1i, \text{worst_id}})$) then
20:	$P_{1i, \text{worst_id}} \leftarrow P_{2i}$
21:	$X_{1i, \text{worst_id}} \leftarrow P_{2i}$
22:	else if ($f(P_{2i}) < f(X_{1i, \text{worst_id}})$) then
23:	$X_{1i, \text{worst_id}} \leftarrow P_{2i}$
24:	end
25:	end
26:	end
27:	for k = 1 to Level-1 İterasyon Sayısı
28:	for j = 1 to Parçacığın Boyutu
29:	$V_{1ij} \leftarrow w_1 * V_{1ij} + C_1 \phi_1 (P_{1ij} - X_{1ij}) + C_2 \phi_2 (G_{1i} - X_{1ij})$
30:	$X_{1ij} \leftarrow X_{1ij} + V_{1ij}$
31:	if ($f(X_{1ij}) < f(P_{1ij})$) then
32:	$P_{1ij} \leftarrow X_{1ij}$
33:	if ($f(P_{1ij}) < f(G_{1i})$) then
34:	$G_{1i} \leftarrow P_{1ij}$
35:	if ($f(G_{1i}) < f(G_2)$) then
36:	$G_2 \leftarrow G_{1i}$
37:	end
38:	end
39:	end
40:	end
41:	end
42:	end
43:	Level-2 başlatılır
44:	for j = 1 to Parçacık Sayısı
45:	$X_{2j} \leftarrow G_{1j}$
46:	end
47:	for k = 1 to Level-2 İterasyon Sayısı
48:	for i = 1 to Parçacık Sayısı
49:	$V_{2i} \leftarrow w_2 * V_{2i} + C_3 \phi_3 (P_{2i} - X_{2i}) + C_4 \phi_4 (G_2 - X_{2i})$
50:	$X_{2i} \leftarrow X_{2i} + V_{2i}$
51:	if ($f(X_2) < f(P_2)$) then
52:	$P_2 \leftarrow X_2$
53:	if ($f(P_2) < f(G_2)$) then
54:	$G_2 \leftarrow P_2$
55:	end
56:	end
57:	end
58:	end
59:	Return G ₂

Şekil 3.4. Popülasyon çeşitliliği kontrollü geri beslemeli GSO algoritması sözde kodu

Orijinal GSO algoritmasının Şekil 3.3' de belirtilen sahte kodu üzerinde yapılan modifikasyonlar, Şekil 3.4' de gösterilmiştir. Orijinal kod üzerinde iki adet modifikasyon yapılmıştır. Çeşitlilik kontrolü ve geri besleme modifikasyonları epok başlangıcında yapılmaktadır. Öncelikle, ilgili alt-popülasyonun çeşitliliği ölçülmektedir. Çeşitlilik, belirlenen eşik değerinin (θ) altına düşerse, alt-popülasyonun bireylerinden, bir tanesi hariç diğerleri, rastgele olarak yeniden oluşturulmaktadır. Rastgele oluşturulmayan birey, ikinci fazdan elde edilen bireyin pbest' inden kopyalanmaktadır. Çeşitlilik değeri, eşik değerinin altına inmemiş ise sadece geri besleme modifikasyonu devreye girmektedir. İlgili alt-popülasyondaki en kötü uygunluk değerine sahip çözüm bulunmakta ve ikinci fazdan alınan çözümün pbest değeri, birinci fazdaki en kötü çözümün pbest uygunluk değerleri karşılaştırılır. İkinci fazdan gelen çözümün pbest uygunluk değeri daha iyi ise çözüm, kötü çözümün yerine kopyalanmaktadır. Değilse ikinci fazda gelen çözümün pbest uygunluk değeri ile birinci fazdaki en kötü çözümün uygunluk değeri karşılaştırılmaktadır. Aktarılan çözümün pbest uygunluk değeri daha iyi ise birinci fazdaki çözümün pbest değeri değişmeden çözüm aktarılmaktadır.

4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA

Bu bölümde, önerilen popülasyon çeşitliliği kontrollü ve geri beslemeli GSO algoritmasının (GSO_DFB) popülasyon çeşitliliği üzerindeki etkisi gözlemlenmiş ve diğer optimizasyon algoritmaları ile performans karşılaştırılması yapılmıştır. Tüm testler Matlab (R2018b sürümü) programı üzerinde yapılmıştır.

GSO_DFB algoritmasının popülasyon çeşitliliği üzerindeki etkisi, 26 benchmark fonksiyonu üzerinde test edilmiştir. Kullanılan benchmark fonksiyonlarının kısıtları, özellikleri ve optimum noktaları Çizelge 4.1’ de sunulmuştur.

Çizelge 4.1. Benchmark test fonksiyonları

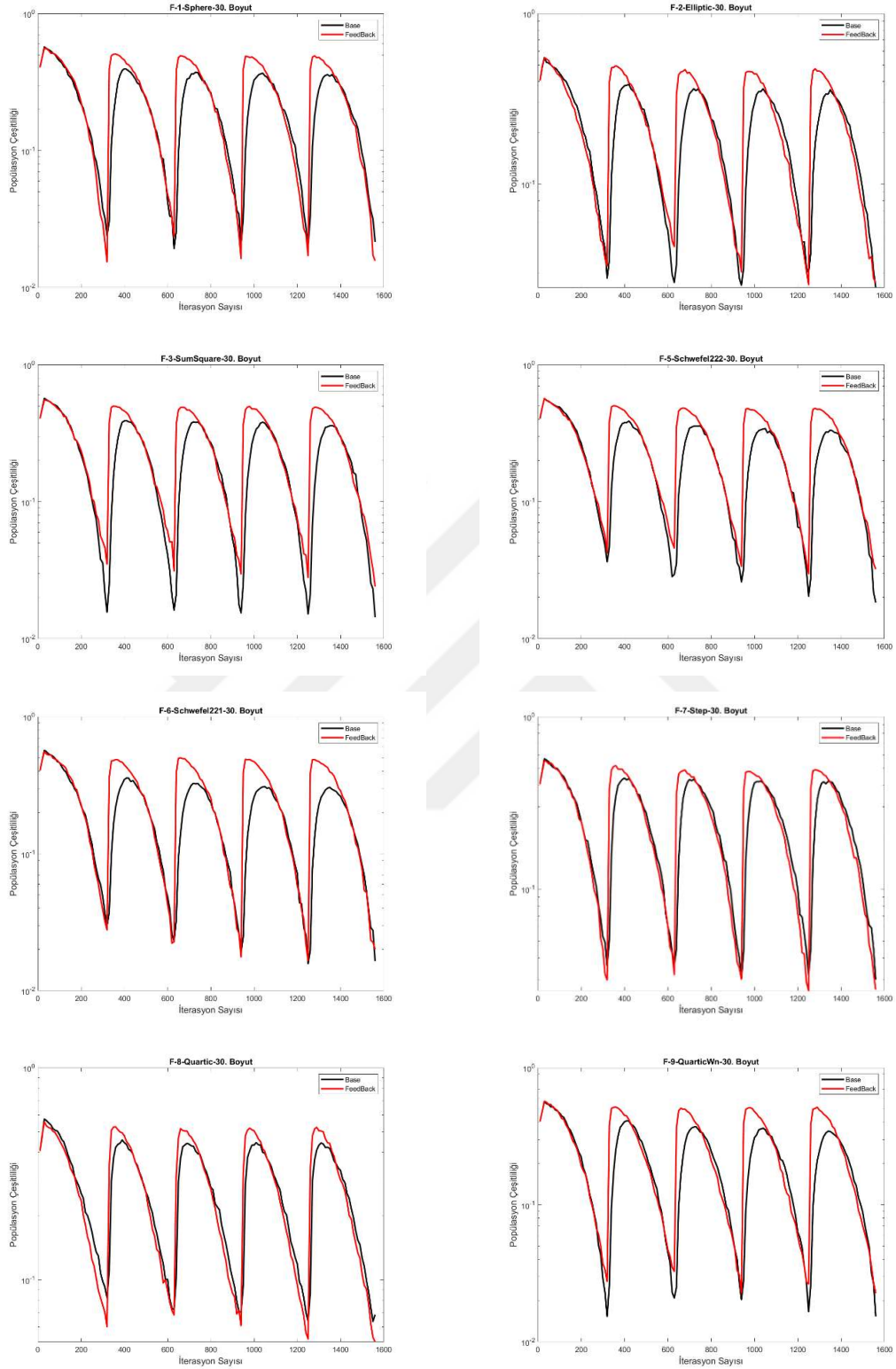
F. No	Fonk. Adı	Arama Alanı	F. Tip	Fonksiyon
F1	Sphere	[100,100]	US	$f_1(\bar{X}) = \sum_{i=1}^n x_i^2$
F2	Elliptic	[100,100]	UN	$f_2(\bar{X}) = \sum_{i=1}^n (10^6)^{(i-1)/(n-1)} x_i^2$
F3	SumSquares	[10,10]	US	$f_3(\bar{X}) = \sum_{i=1}^n ix_i^2$
F4	SumPower	[10,10]	MS	$f_4(\bar{X}) = \sum_{i=1}^n x_i ^{(i+1)}$
F5	Schwefel2.22	[10,10]	UN	$f_5(\bar{X}) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $
F6	Schwefel2.21	[100,100]	UN	$f_6(\bar{X}) = \max_i \{ x_i , 1 \leq i \leq n\}$
F7	Step	[100,100]	US	$f_7(\bar{X}) = \sum_{i=1}^n (\lfloor x_i + 0.5 \rfloor)^2$
F8	Quartic	[1.28,1.28]	US	$f_8(\bar{X}) = \sum_{i=1}^n ix_i^4$
F9	QuarticWN	[1.28,1.28]	US	$f_9(\bar{X}) = \sum_{i=1}^n ix_i^4 + random[0,1)$
F10	Rosenbrock	[10,10]	UN	$f_{10}(\bar{X}) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$
F11	Rastrigin	[5.12,5.12]	MS	$f_{11}(\bar{X}) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$ $f_{12}(\bar{X}) = \sum_{i=1}^n [y_i^2 - 10 \cos(2\pi y_i) + 10]$
F12	Non-Continuous Rastrigin	[5.12,5.12]	MS	$y_i = \begin{cases} x_i & x_i < \frac{1}{2} \\ \frac{round(2x_i)}{2} & x_i \geq \frac{1}{2} \end{cases}$

F13	Griewank	[600,600]	MN	$f_{13}(\bar{X}) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$
F14	Schwefel2.26	[500,500]	UN	$f_{14}(\bar{X}) = 418.98 * n - \sum_{i=1}^n x_i \sin\left(\sqrt{ x_i }\right)$
F15	Ackley	[32,32]	MN	$f_{15}(\bar{X}) = -20 \exp\left\{-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right\} - \exp\left\{\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right\} + 20 + e$
F16	Penalized1	[50,50]	MN	$f_{16}(\bar{X}) = \frac{\pi}{n} \{10 \sin^2(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_n - 1)^2\} + \sum_{i=1}^n u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{1}{4}(x_i + 1) u_{x,a,k,m} = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a \leq x_i \leq a \\ k(x_i - a)^m & x_i < -a \end{cases}$
F17	Penalized2	[50,50]	MN	$f_{17}(\bar{X}) = \frac{1}{10} \{ \sin^2(\pi x_1) + \sum_{i=1}^{n-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] + (x_n - 1)^2 [1 + \sin^2(2\pi x_{i+1})] \} + \sum_{i=1}^n u(x_i, 5, 100, 4)$
F18	Alpine	[10,10]	MS	$f_{18}(\bar{X}) = \sum_{i=1}^n x_i \cdot \sin(x_i) + 0.1 \cdot x_i $
F19	Levy	[10,10]	MN	$f_{19}(\bar{X}) = \sum_{i=1}^{n-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] + \sin^2(3\pi x_1) + x_n - 1 [1 + \sin^2(3\pi x_n)]$
F20	Weierstrass	[0.5,0.5]	MN	$f_{20}(\bar{X}) = \sum_{i=1}^D \left(\sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k (x_i + 0.5))] \right) - D \sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k 0.5)]$ $a = 0.5, b = 3, k_{\max} = 20$
F21	Schaffer	[100,100]	MN	$f_{21}(\bar{X}) = 0.5 + \frac{\sin^2\left(\sqrt{\sum_{i=1}^n x_i^2}\right) - 0.5}{\left(1 + 0.001 * \left[\sum_{i=1}^n x_i^2\right]\right)^2}$
F22	Shifted Sphere	[100,100]	US	$f_{24}(\bar{X}) = \sum_{i=1}^n z_i^2 \quad z = x - o$
F23	Shifted Rastrigin	[5.12,5.12]	MS	$f_{25}(\bar{X}) = \sum_{i=1}^n [z_i^2 - 10 \cos(2\pi z_i) + 10] \quad z = x - o$
F24	Shifted Griewank	[600,600]	MN	$f_{26}(\bar{X}) = \frac{1}{4000} \sum_{i=1}^n z_i^2 - \prod_{i=1}^n \cos\left(\frac{z_i}{\sqrt{i}}\right) + 1 \quad z = x - o$
F25	Shifted Ackley	[32,32]	MN	$f_{27}(\bar{X}) = -20 \exp\left\{-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n z_i^2}\right\} - \exp\left\{\frac{1}{n} \sum_{i=1}^n \cos(2\pi z_i)\right\}$ $z = x - o$
F26	Shifted Alpine	[10,10]	MN	$f_{28}(\bar{X}) = \sum_{i=1}^n z_i \cdot \sin(z_i) + 0.1 \cdot z_i \quad z = x - o$

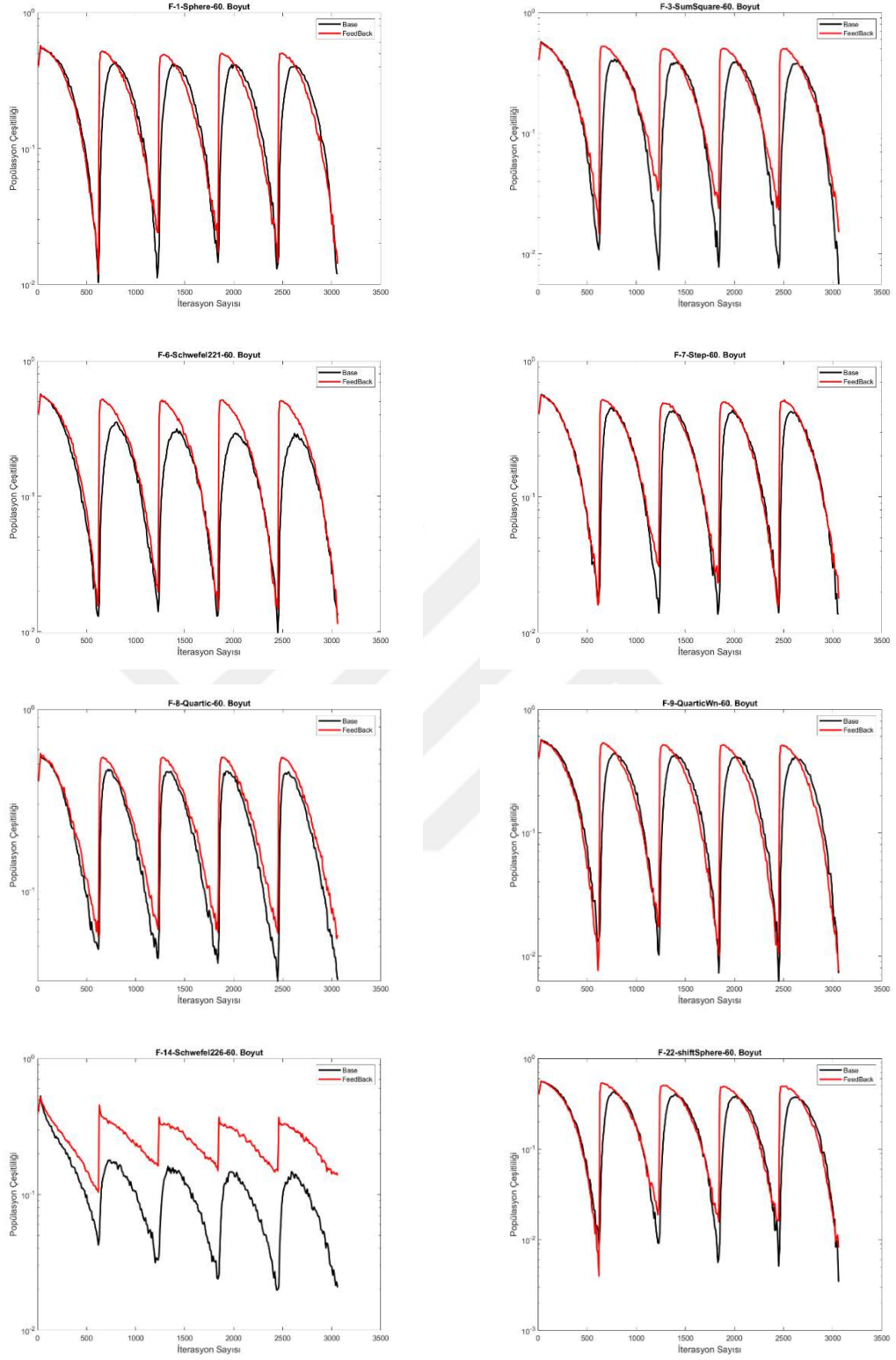
Benchmark fonksiyonları üzerinde yapılan testlerde, 30,60 ve 100 boyut üzerinde çalışmalar yapılmıştır. Benchmark fonksiyonları temel olarak tek modlu ve çok modlu özellikte olup, elde edilen sonuçlar bu özelliklere göre sınıflandırılmıştır. Fonksiyon numarası, fonksiyon adı, fonksiyon aralığı, fonksiyon tipi ve fonksiyon denklemi Çizelge 4.1’de verilmiştir.

Tek modlu fonksiyonlar için 30, 60 ve 100 boyutta yapılan çalışmalarda rastgele seçilen fonksiyonların popülasyon çeşitliliği değişim grafikleri Şekil 4.1, Şekil 4.2 ve Şekil 4.3’ de sunulmuştur. Çok modlu fonksiyonlar için 30, 60 ve 100 boyutta yapılan çalışmalarda rastgele seçilen fonksiyonların popülasyon çeşitliliği değişim grafikleri Şekil 4.4, Şekil 4.5 ve Şekil 4.6’ de sunulmuştur.

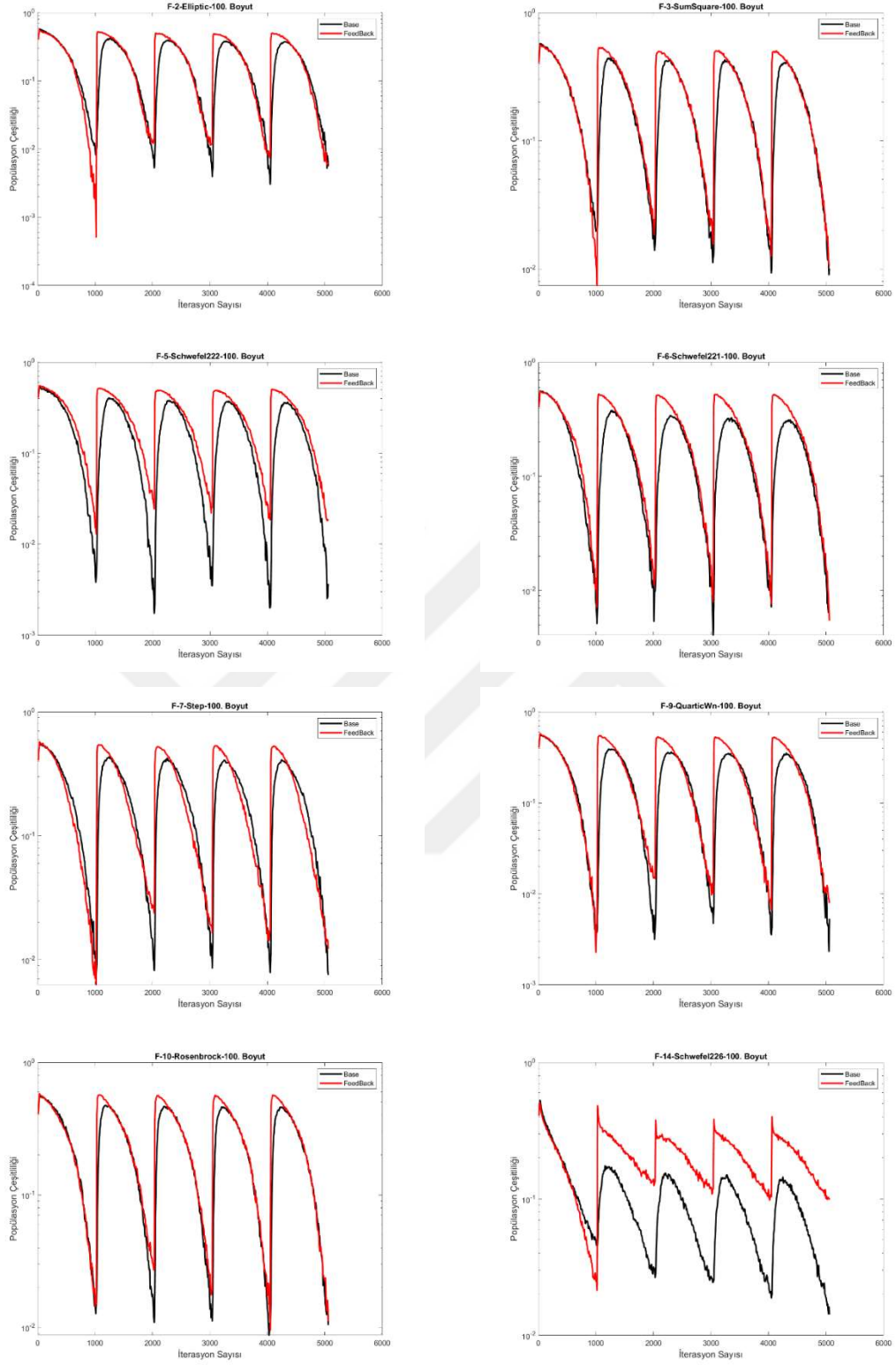




Şekil 4.1. 30 boyuttaki tek modlu fonksiyonların populasyon çeşitliliği grafiği



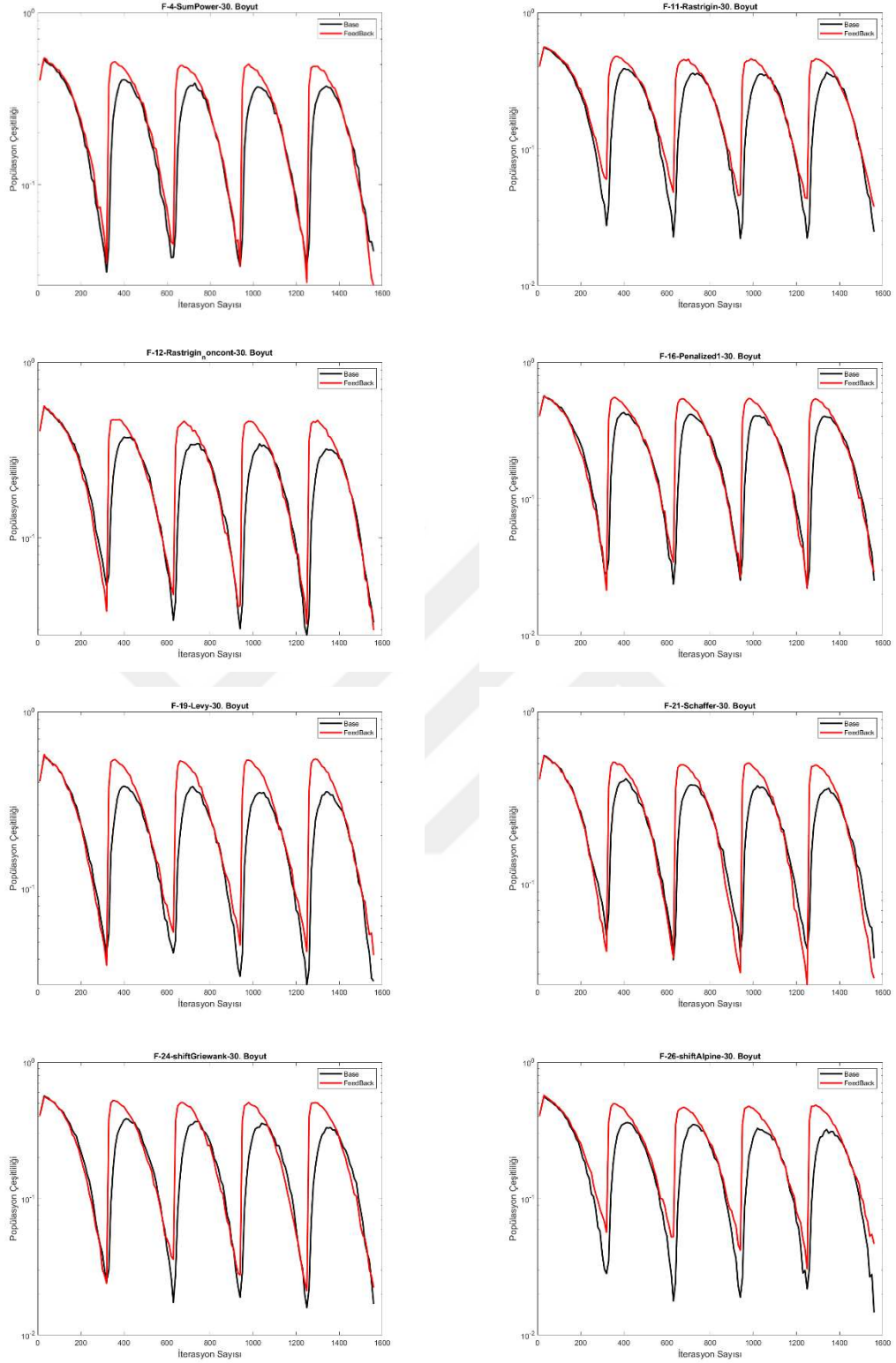
Şekil 4.2. 60 boyuttaki tek modlu fonksiyonların populasyon çeşitliliği grafiği



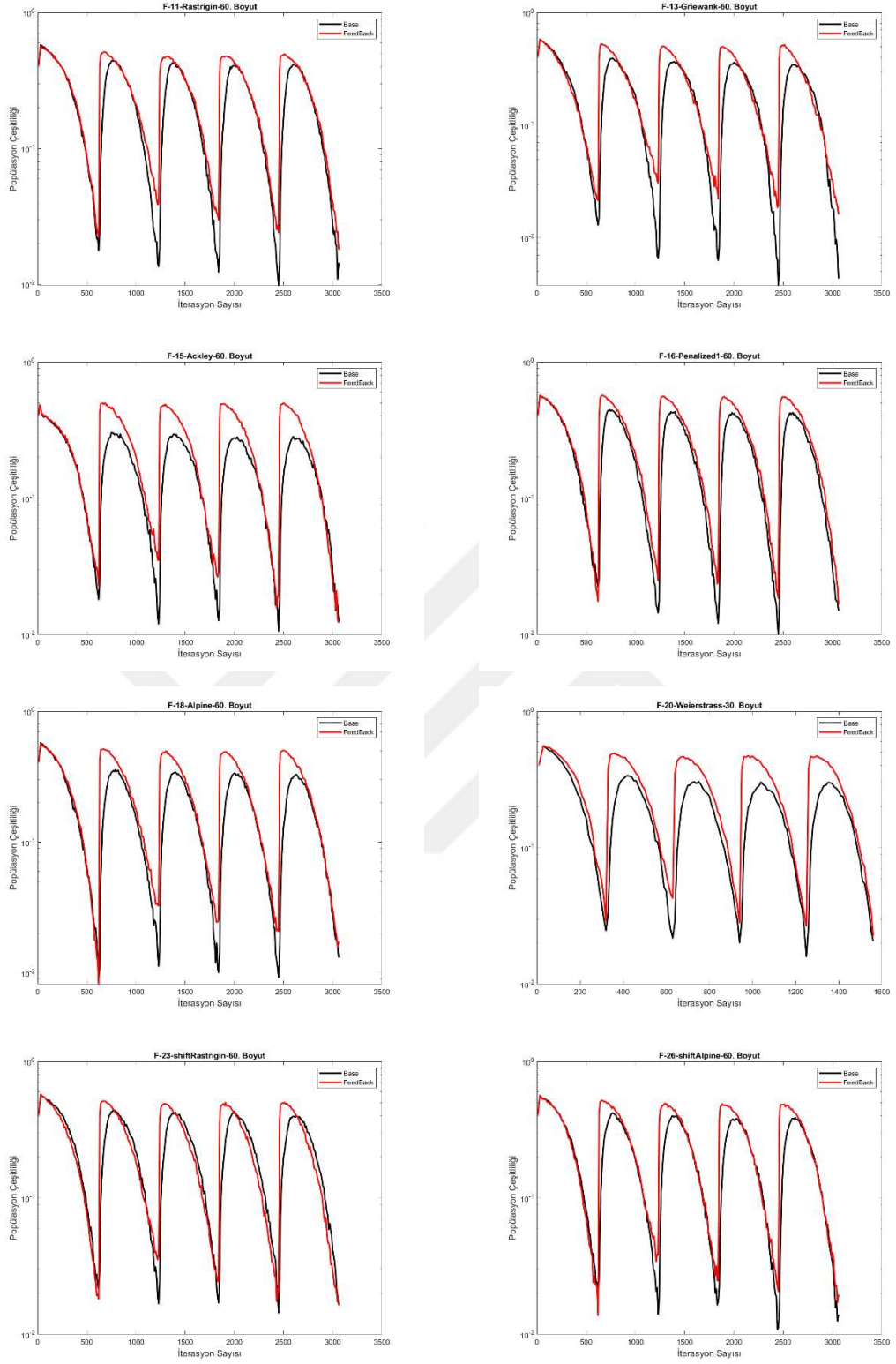
Şekil 4.3. 100 boyuttaki tek modlu fonksiyonların populasyon çeşitliliği grafiği

Tek modlu fonksiyonlar bir tane optimum noktasına sahip olan fonksiyonlardır. Bu tip fonksiyonlar daha geniş arama uzayına sahip fonksiyonlardır. Tek bir optimum noktaya sahip olmalarının sonucunda da, yerel minimum noktalarında takılma ihtimali bulunmamaktadır.

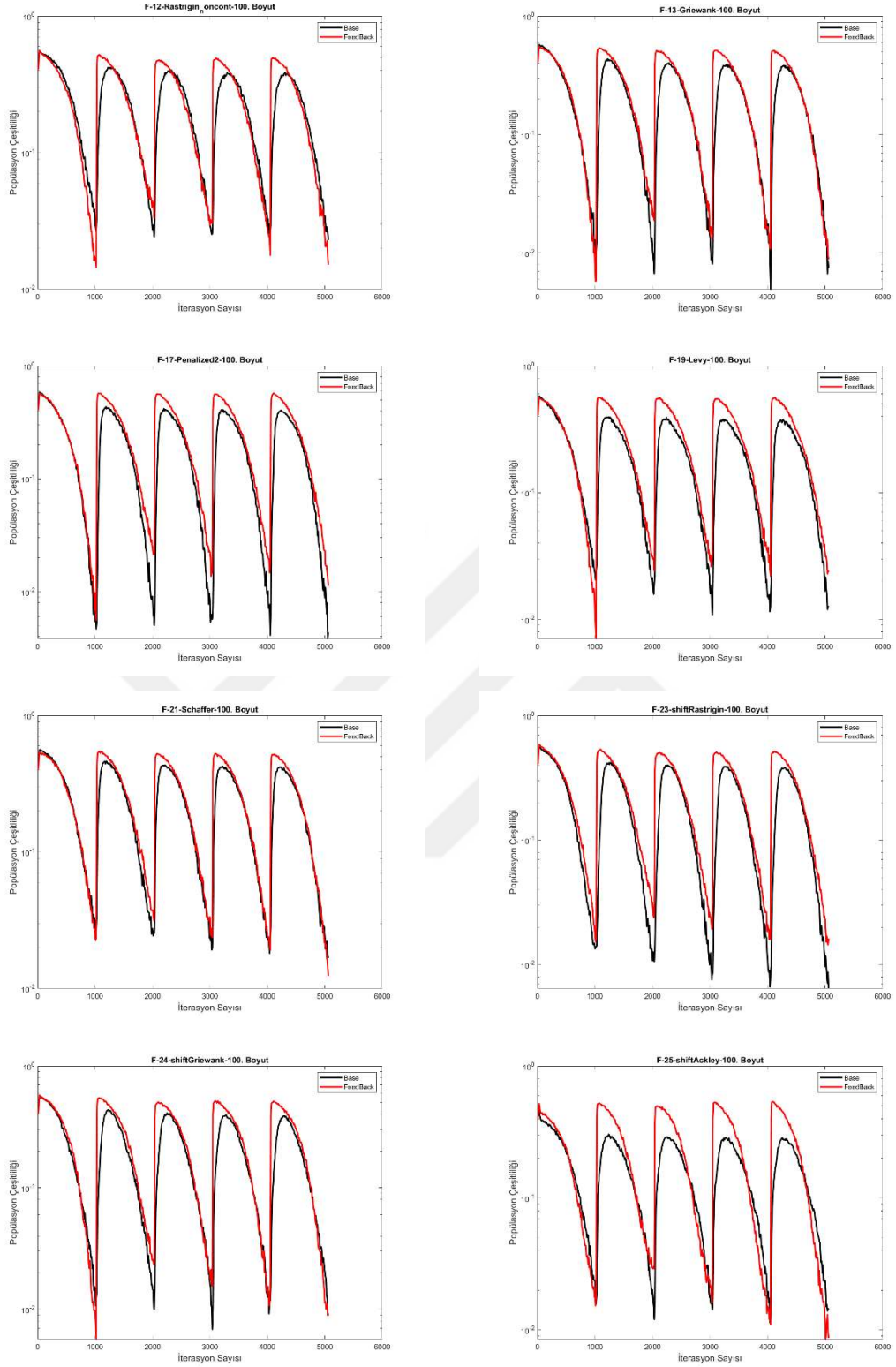
GSO_DFB algoritmasının ve temel GSO algoritmasının tek modlu fonksiyonlar için 30, 60 ve 100 boyuttaki popülasyon çeşitliliği değişikliği Şekil 4.1, Şekil 4.2 ve Şekil 4.3' de görülebilmektedir. Şekil 4.1' de görüldüğü gibi, 30 boyutta GSO_DFB algoritmasındaki popülasyon çeşitliliği kontrolü ile çeşitlilik değeri daha yüksek tutulabilmektedir. Popülasyon çeşitliliğinin yüksek olması sonucunda, keşfetme yeteneğini artırmaktadır. Keşfetme yeteneğinin artması sonucunda GSO algoritmasındaki level 2' ye daha iyi sonuçlar verilmektedir. Şekil 4.2 ve Şekil 4.3 incelemek olursa eğer, GSO_DFB algoritması tek modlu fonksiyonların farklı boyutlarında da, popülasyon çeşitliliğini yüksek tutabilme başarısını sürdürmektedir.



Şekil 4.4. 30 boyuttaki çok modlu fonksiyonların popülasyon çeşitliliği grafiği



Şekil 4.5. 60 boyutlu çok modlu fonksiyonların popülasyon çeşitliliği grafiği



Şekil 4.6. 100 boyuttaki çok modlu fonksiyonların popülasyon çeşitliliği grafiği

Çok modlu fonksiyonlar birden fazla yerel optimum noktasına sahiptir ve tek bir genel optimum noktasına sahiptirler. Yerel optimum noktalarının birden fazla olması sonucunda, bu noktalarda algoritmanın takılıp kalma ihtimali bulunmaktadır ve çözüm zorlaşmaktadır. Popülasyon çeşitliliğinin çok azalması, yerel optimum noktalarında takılma ihtimalini doğurabilmektedir. GSO_DFB algoritmasının etkisi, popülasyon çeşitliliği belli bir değerin altına düşmesi durumlarında ortaya çıkmaktadır. Bu özellik yerel optimum noktalarına takılma ihtimalini düşürmektedir ve algoritmanın başarısını artırmaktadır.

GSO_DFB algoritması ve temel GSO algoritması için çok modlu fonksiyonların 30, 60 ve 100 boyutlarının popülasyon çeşitliliği değişikliği Şekil 4.4, Şekil 4.5 ve Şekil 4.6' de görülebilmektedir. Genel olarak 30, 60 ve 100 boyutta, popülasyon çeşitliliğinin temel GSO algoritmasından daha yüksek olduğu görülebilmektedir. Özellikle çok modlu fonksiyonlar için popülasyon çeşitliliğinin yüksek olması, keşfetme özelliğini artırmaktadır ve yerel optimum değerlerine takılmalarını engellemektedir.

GSO_DFB algoritması, popülasyon çeşitliliği kontrolü dışında, level 2' den elde edilen bireylerin level 1' e geri beslenmesi işlemini de gerçekleştirmektedir. Level 1' de popülasyonlar seçilirken, level 2' den gelen bireyler sabit tutulup, geri kalan bireyler rasgele oluşturulmaktadır. Bu işlem sonucunda elde edilmiş pbest değerleri kaybedilmemiş olmaktadır.

Yapılan çalışmalardan elde edilen sonuçlar, temel GSO algoritması, AAA, ABC, BAT, BEE, FPA, PSO algoritmalarının 30, 60 ve 100 boyuttaki sonuçları ile karşılaştırılmıştır. Ortalama ve standart sapma değerleri dikkate alınarak, 30 boyutta yapılan karşılaştırma sonuçları, Çizelge 4.2' de, 60 boyutta yapılan karşılaştırma sonuçları Çizelge 4.3' de, 100 boyutta yapılan karşılaştırma sonuçları Çizelge 4.4' de sunulmuştur.

30 boyutta yapılan test sonuçlarını içeren Çizelge 4.2' de, önerilen algoritmanın, ortalama değerine bakılarak 12, standart sapma değerine bakılarak 13 fonksiyon için en iyi sonucu verdiği gözlemlenmiştir. AAA algoritmasının 30 boyutta, önerilen algoritmadan aynı sonuçlar elde ettiği görülmüştür. Bu algoritma dışındaki testte kullanılan diğer algoritmaları, önerilen GSO_DFB algoritmasının geride bıraktığı gözlemlenmiştir.

60 boyutta yapılan test sonuçlarını içeren Çizelge 4.3' de, önerilen algoritma, ortalama değerinde 16 fonksiyon için en iyi sonucu elde ederken, standart sapma değeri dikkate alındığında, 15 fonksiyon için en iyi değeri elde etmiştir. Hem ortalama değer için hem de standart sapma değerleri için önerilen GSO_DFB algoritması, karşılaştırma için kullanılan tüm algoritmaları geride bırakmıştır.

100 boyutta yapılan test sonuçlarını içeren Çizelge 4.4' de, önerilen algoritma, ortalama ve standart sapma değerlerine bakılarak, 13 fonksiyon için en iyi sonucu verdiği gözlemlenmiştir. Test için kullanılan algoritmalar içerisinde, en iyi ortalama ve standart sapma değerlerini elde eden algoritma, önerilen GSO_DFB algoritması olmuştur.

Yapılan test sonuçlarının özeti Çizelge 4.5' de sunulmuştur.

Çizelge 4.5. Test sonuçlarının özeti

	GSO_DFB	GSO	PSO	AAA	ABC	BAT	BEE	FPA
Ortalama								
D = 30	12	6	3	12	7	2	2	3
D = 60	16	4	2	10	7	2	2	3
D = 100	13	2	3	11	7	2	2	3
Std. Sapma								
D = 30	13	5	4	12	7	2	2	3
D = 60	15	5	4	10	6	2	3	3
D = 100	13	3	4	11	7	2	2	3

30,60 ve 100 boyutta, farklı algoritmalar üzerinde yapılan testlerin sonucunda Çizelge 4.5 'de yer alan değerler elde edilmiştir. 30 boyutta ortalama değerler dikkate alınarak, GSO_DFB ve AAA algoritmaları 12 fonksiyonda en iyi değeri elde etmiştir. 30 boyutta yapılan testlerin sonucunda önerilen GSO_DFB algoritması ve AAA algoritması en iyi algoritmalar olmuştur. GSO_B algoritması 6, ABC algoritması 7, PSO ve FPA algoritmaları 3 fonksiyon için en iyi değerleri elde etmiştir. BAT ve BEE algoritmaları 2 fonksiyon ile diğer fonksiyonların gerisinde kalmıştır. Standart sapma değerleri ele alındığında, önerilen GSO_DFB algoritması 13, AAA algoritması 12, ABC algoritması

7, GSO_B algoritması 5, PSO algoritması 4 fonksiyon için en iyi standart sapma değerine sahiptir. Karşılaştırılan diğer tüm algoritmalar, standart sapma değerlerinde ortalama değer ile aynı sonuçlara sahiptir.

60 boyutta yapılan testler için ortalama değer dikkate alındığında, GSO_DFB algoritması 16 fonksiyon için en iyi sonucu elde ederek diğer karşılaştırılan algoritmaları geride bırakmıştır. GSO_DFB algoritmasını takiben AAA algoritması 10 fonksiyon için en iyi sonucu elde etmiştir. ABC algoritması da 7 fonksiyon için en iyi sonucu elde etmiştir. GSO_B algoritması 4, FPA algoritması 3, PSO, BAT ve BEE algoritmaları 2 fonksiyon için en iyi sonucu elde etmiştir. 60 boyut için standart sapma değeri dikkate alındığında, GSO_DFB algoritması 15 fonksiyon için başarı göstermiştir. AAA algoritması 10 fonksiyon için en iyi standart sapma değerini elde ederken ABC algoritması 6 fonksiyon için en iyi standart sapma değerini elde etmiştir. Yine standart sapma değeri dikkate alındığında GSO_B algoritması 5, PSO algoritması 4, BEE ve FPA algoritmaları 3 ve BAT algoritması 2 fonksiyon için en iyi değeri elde etmiştir.

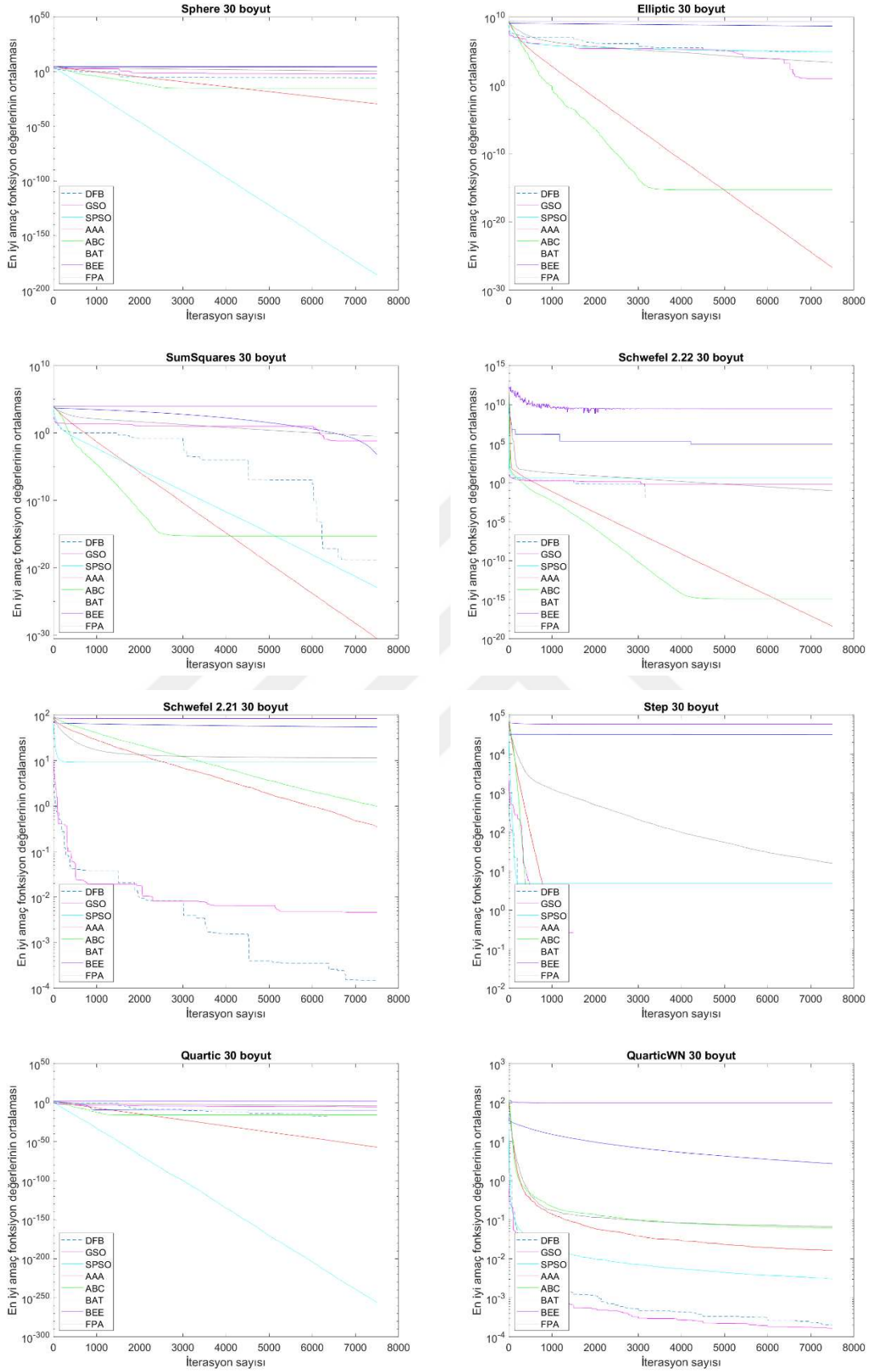
100 boyutta yapılan testlerden elde edilen ortalama değerleri dikkate alındığında, GSO_DFB algoritması 13 fonksiyonda en iyi sonucu elde etmiştir. AAA algoritması 11 fonksiyon için, ABC algoritması 7 fonksiyon için en iyi sonuçları elde eden diğer algoritmalar olmuştur. PSO ve FPA algoritmaları 3, GSO_B, BAT ve BEE algoritmaları 2 fonksiyon için en iyi ortalama değerini elde etmiştir. 100 boyutta yapılan testler için standart sapma değerleri dikkate alındığında, GSO_DFB algoritması 13 fonksiyon için en iyi standart sapma değerini elde ederek diğer algoritmaları geride bırakmıştır. Diğer karşılaştırılan algoritmalarından AAA algoritması 11, ABC algoritması 7, PSO algoritması 4, GSO_B ve FPA algoritmaları 3, BAT ve BEE algoritmaları 2 fonksiyon için en iyi standart sapma değerini elde etmiştir.

30, 60 ve 100 boyutta yapılan testler göz önüne alındığında, önerilen GSO_DFB algoritmasının 30 boyutta AAA algoritması ile aynı başarıyı gösterirken, rekabetçi bir yapı sergilemiştir ve diğer algoritmalarından daha iyi sonuçlar elde etmiştir. Boyut arttıkça GSO_B algoritmasının başarısı düşmesine karşın, GSO_DFB algoritmasının performansını koruduğu ve diğer algoritmalarla kıyaslandığında rekabetçi sonuçlar ortaya koyduğu gözlemlenmiştir.

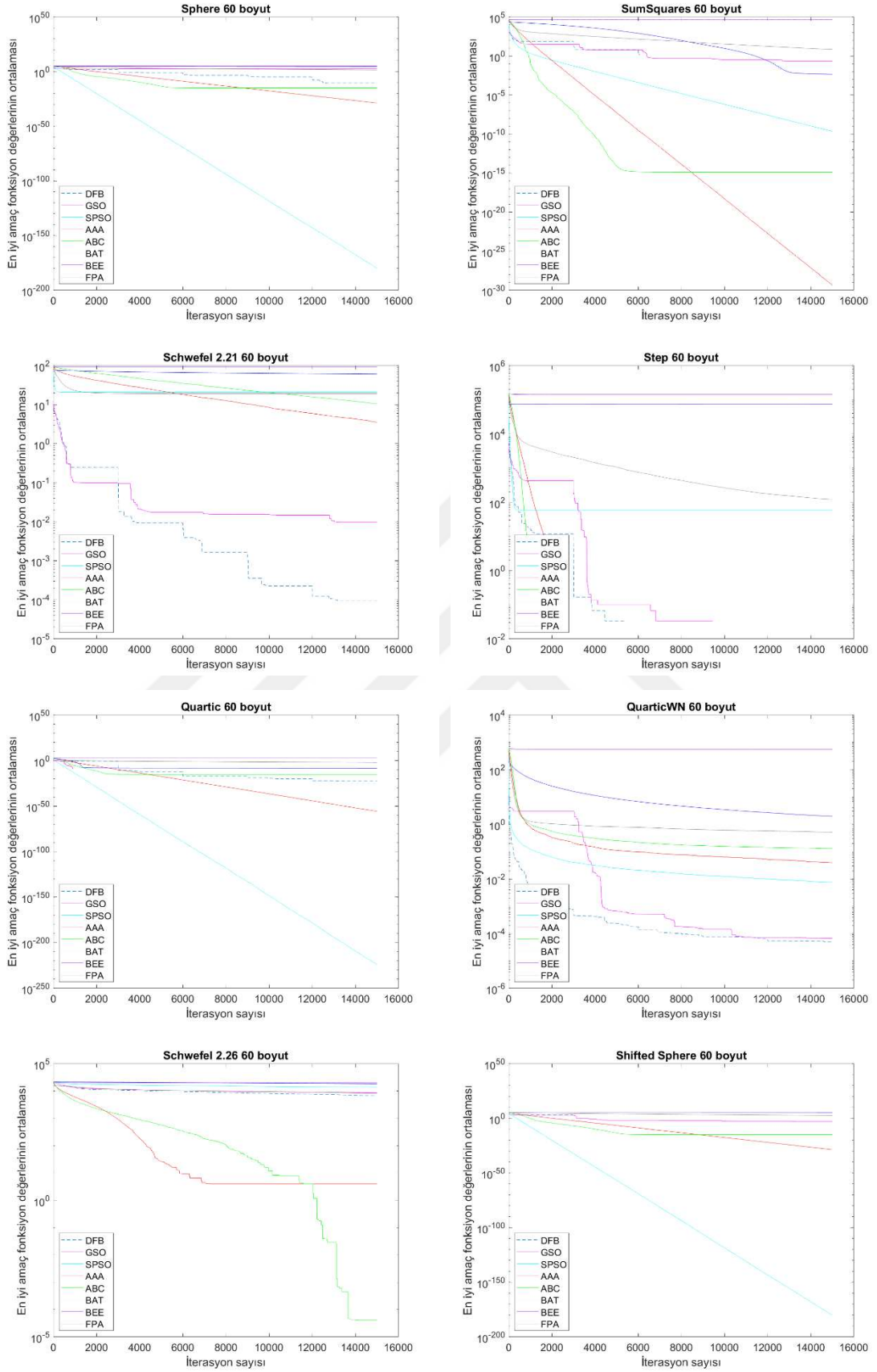
Şekil 4.7-4.12 arasında önerilen algoritmanın performansını değerlendirmek için tek modlu ve çok modlu fonksiyonların 30,60 ve 100 boyuttaki test sonuçlarının yakınsama eğrileri sunulmuştur. Şekil 4.7, Şekil 4.8 ve Şekil 4.9 tek modlu fonksiyonların yakınsama eğrilerini belirtmektedir. Şekil 4.10, Şekil 4.11 ve Şekil 4.12 çok modlu

fonksiyonların yakınsama eğrilerini belirtmektedir. Yapılan testlerin yakınsama eğrilerini oluşturmak için fonksiyon seçimleri yapılırken, popülasyon çeşitliliğini belirten grafiklerdeki fonksiyonlar seçilmiştir. Yakınsama eğrileri ile önerilen algoritma ve kıyaslanan diğer tüm algoritmalar arasındaki en uygun çözüme yakınsama hızları ile performans değerlendirilmesi yapılmıştır. Yakınsama hızları dikkate alındığında, önerilen algoritmanın kıyaslanan diğer algoritmalarından daha iyi sonuçlar elde ettiği ve rekabetçi yaklaşım sunduğu gözlemlenmiştir.

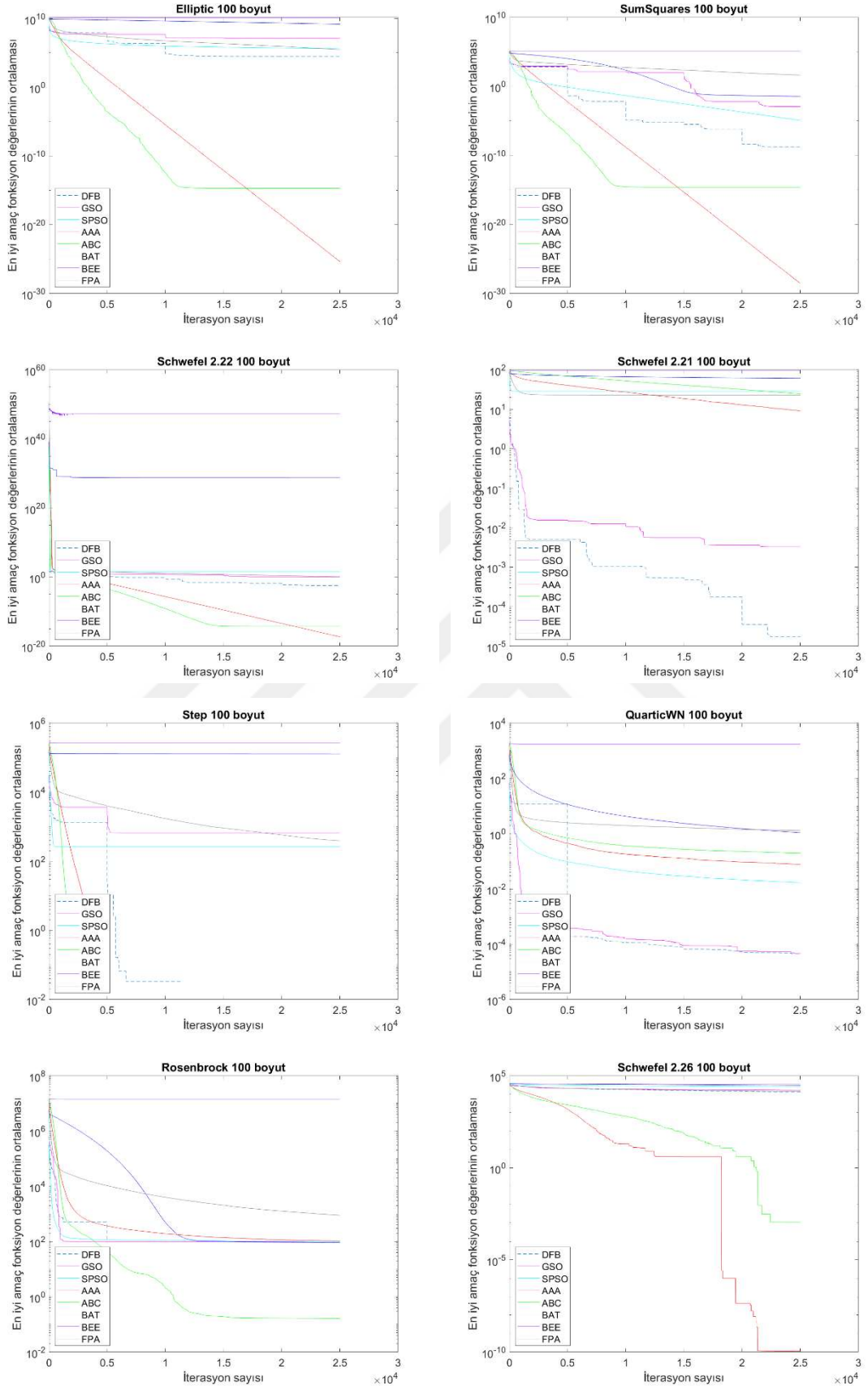




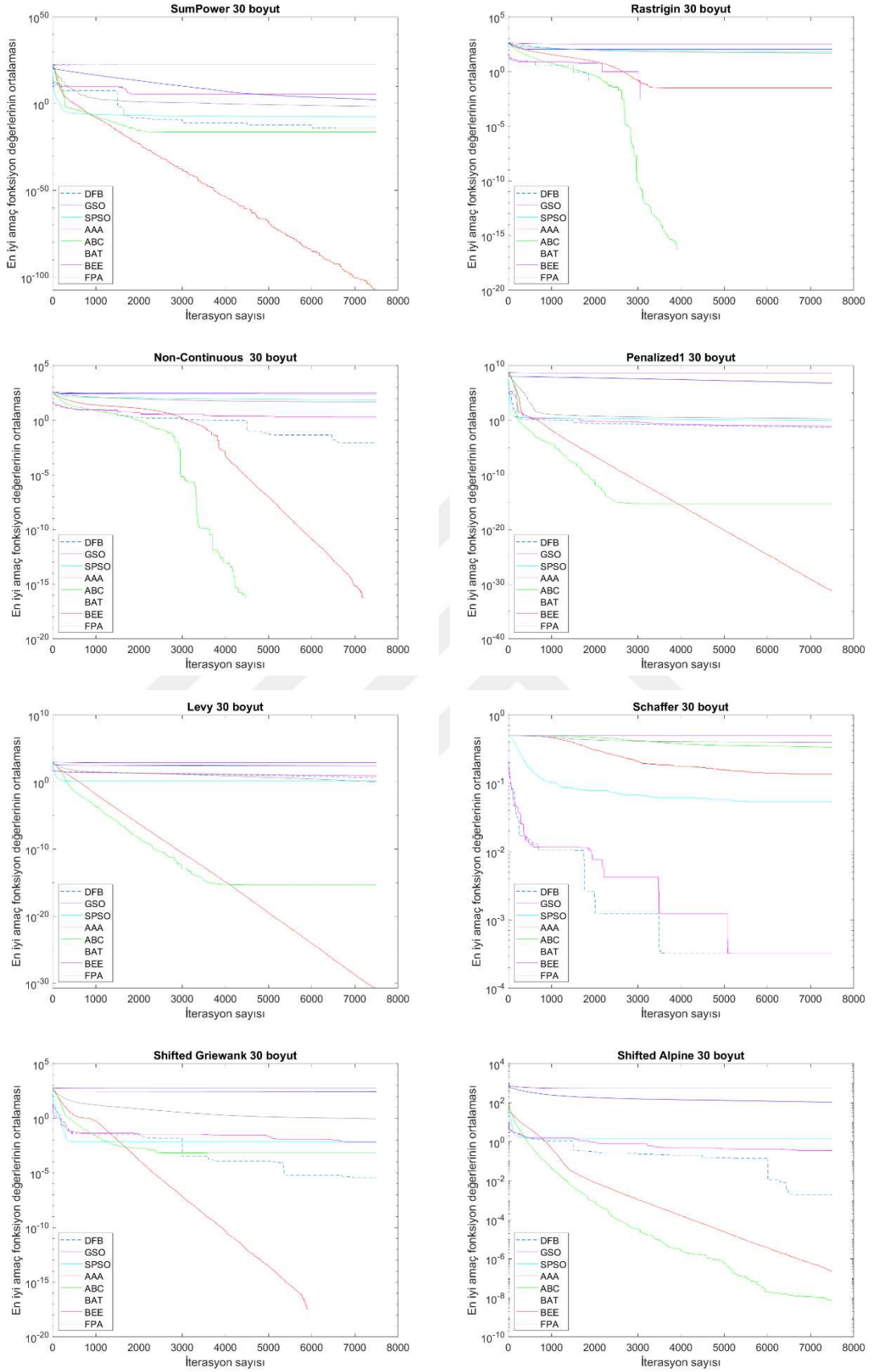
Şekil 4.7. 30 boyuttaki tek modlu fonksiyonların yakınsama grafiği



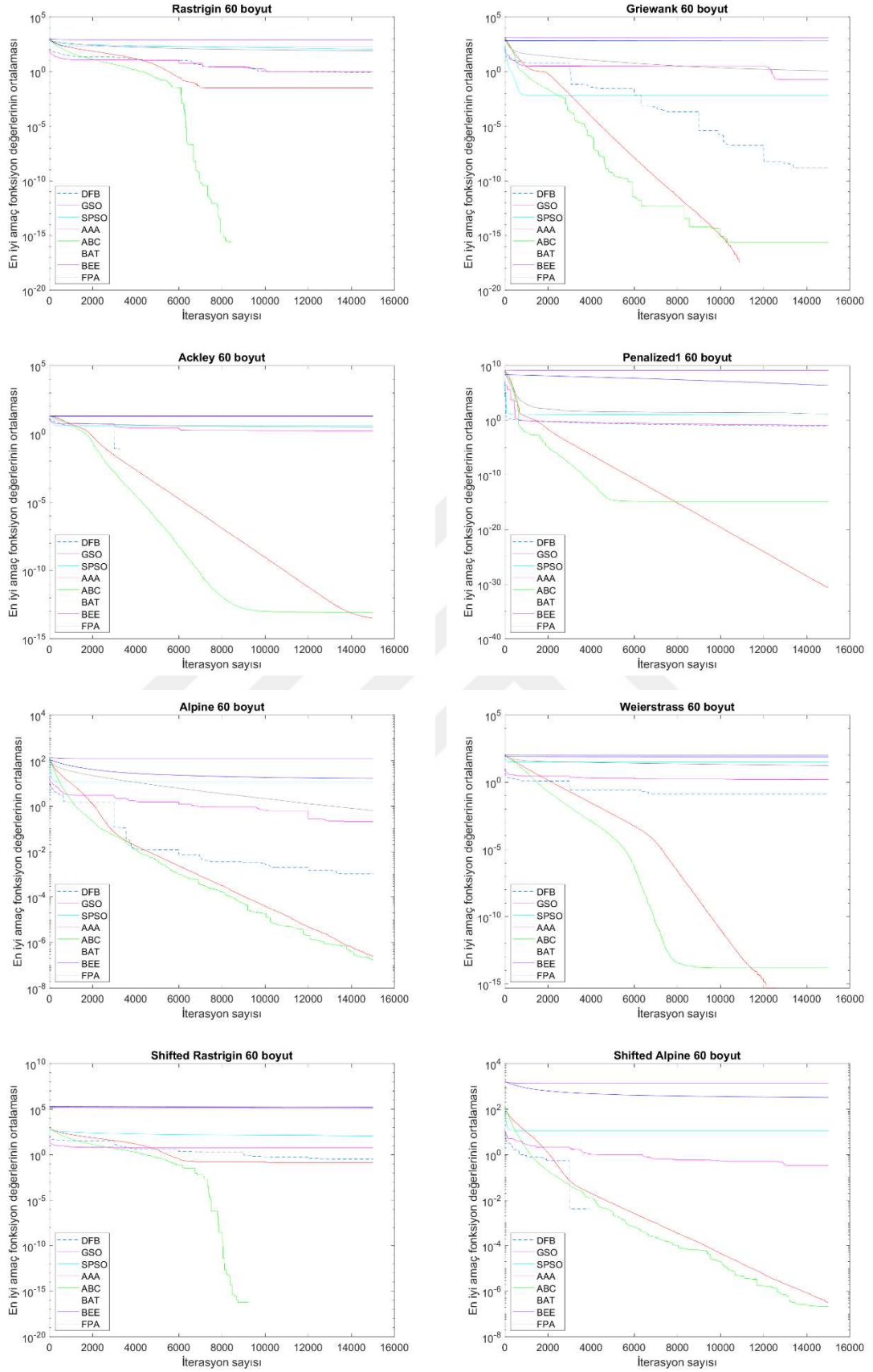
Şekil 4.8. 60 boyuttaki tek modlu fonksiyonların yakınsama grafiği



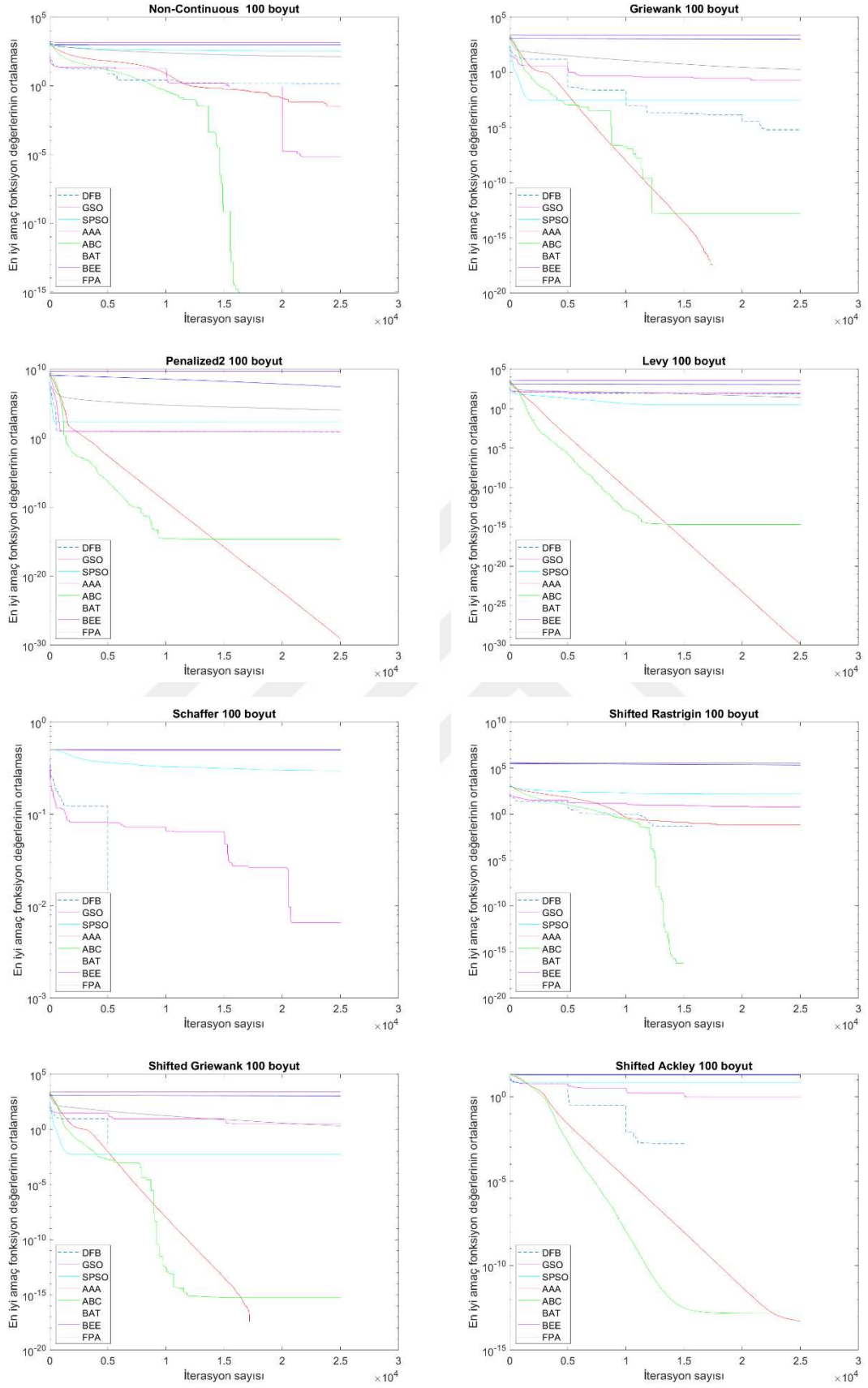
Şekil 4.9. 100 boyuttaki tek modlu fonksiyonların yakınsama grafiği



Şekil 4.10. 30 boyuttaki çok modlu fonksiyonların yakınsama grafiği



Şekil 4.11. 60 boyuttaki çok modlu fonksiyonların yakınsama grafiği



Şekil 4.12. 100 boyuttaki çok modlu fonksiyonların yakınsama grafiği

Önerilen algoritmanın kıyaslanan diğer algoritmalara göre performans üstünlüğünü gözlemleyebilmek için Wilcoxon işaretli-sıralama testi kullanılmıştır. Test işlemlerinde kullanılan tüm algoritmaların performansları ile önerilen algoritmanın performansı kıyaslanmıştır. Çizelge 4.6 30 boyutta, Çizelge 4.7 60 boyutta ve Çizelge 4.8 100 boyutta yapılan çalışmaların Wilcoxon işaretli-sıralama testi sonuçlarını göstermektedir. Bu Yapılan Wilcoxon testinde elde edilen işaret ve p-value değerleri Matlab programı kullanılarak elde edilmiştir. Verilen çizelgelerde + işareti karşılaştırma sonucunda anlamlı bir fark olduğunu belirtirken, - işareti karşılaştırma sonuçlarında anlamlı bir fark olmadığını belirtmektedir. Çizelgelerin altında bulunan toplam sayıları ise, algoritmanın Wilcoxon işaretli-sıralama testinde, + işarete sahip olduğu toplam fonksiyon sayısının - işarete sahip fonksiyon sayısına oranını belirtmektedir. Çizelge 4.6 - Çizelge 4.8 göz önüne alındığında, istatistik açısından önerilen algoritmanın kıyaslanan tüm algoritmalarından öne çıktığı gözlemlenmiştir.

Çizelge 4.6. 30 boyut wilcoxon test sonuçları

F.No	GSO_DFB-GSO_B		GSO_DFB-PSO		GSO_DFB-AAA		GSO_DFB-ABC		GSO_DFB-BAT		GSO_DFB-BEE		GSO_DFB-FPA	
	İşaret	P Value	İşaret	P Value	İşaret	P Value	İşaret	P Value	İşaret	P Value	İşaret	P Value	İşaret	P Value
F1	-	1.00E+00	+	3.11E+09	+	3.11E+09	+	3.11E+09	+	1.73E+08	+	1.73E+08	+	1.73E+08
F2	+	1.00E+00	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08
F3	-	2.50E-01	+	3.11E+09	+	3.11E+09	+	3.11E+09	+	1.73E+08	+	1.73E+08	+	1.73E+08
F4	+	5.00E-01	+	1.73E+08	+	3.11E+09	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08
F5	-	5.00E-01	+	1.73E+08	+	1.73E+08	+	1.71E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08
F6	-	1.00E+00	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08
F7	-	1.00E+00	+	2.40E+08	-	1.00E+00	-	1.00E+00	+	1.73E+08	+	1.73E+08	+	1.72E+08
F8	+	1.00E+00	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08
F9	-	2.71E-01	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08
F10	-	2.62E-01	+	8.22E-03	-	5.71E-02	+	6.89E+09	+	8.73E-03	+	1.73E+08	+	3.11E+09
F11	-	5.00E-01	+	1.73E+08	-	1.00E+00	-	5.00E-01	+	1.73E+08	+	1.73E+08	+	1.73E+08
F12	+	2.50E-01	+	1.73E+08	-	1.00E+00	-	1.00E+00	+	1.73E+08	+	1.73E+08	+	1.73E+08
F13	-	1.00E+00	+	1.70E+08	-	5.00E-01	+	3.60E-03	+	1.73E+08	+	1.73E+08	+	1.73E+08
F14	-	1.02E-01	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	7.69E+08
F15	-	1.00E+00	+	1.71E+08	+	1.14E+08	+	9.63E+07	+	1.73E+08	+	1.73E+08	+	1.73E+08
F16	+	8.22E-03	+	2.58E-03	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.92E+08
F17	+	2.96E-03	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08
F18	-	1.00E+00	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08
F19	+	1.96E-02	+	1.29E-03	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.15E-04
F20	-	4.69E-01	+	1.73E+08	-	2.50E-01	-	2.50E-01	+	1.73E+08	+	1.73E+08	+	1.73E+08
F21	-	1.00E+00	+	7.81E+07	+	9.44E+07	+	1.49E+08	+	1.72E+08	+	1.65E+08	+	1.61E+08
F22	-	1.00E+00	+	5.45E+08	-	5.00E-01	+	8.71E-03	+	1.73E+08	+	1.73E+08	+	1.73E+08
F23	+	6.25E-02	+	1.68E+08	+	1.47E+08	+	1.73E+08	+	1.68E+08	+	1.73E+08	+	1.73E+08
F24	-	5.00E-01	+	1.73E+08	+	3.11E+09	+	1.73E+08	+	1.73E+08	+	1.73E+08	-	1.00E+00
F25	-	1.00E+00	-	1.00E+00	-	1.00E+00	-	1.00E+00	-	1.00E+00	-	1.00E+00	-	1.00E+00
F26	-	1.00E+00	-	1.00E+00	-	1.00E+00	-	1.00E+00	-	1.00E+00	-	1.00E+00	-	1.00E+00
+/-		8/18		24/2		17/9		20/6		24/2		24/2		23/3

Çizelge 4.7. 60 boyut wilcoxon test sonuçları

F.No	GSO_DFB-GSO_B		GSO_DFB-PSO		GSO_DFB-AAA		GSO_DFB-ABC		GSO_DFB-BAT		GSO_DFB-BEE		GSO_DFB-FPA	
	İşaret	P Value	İşaret	P Value	İşaret	P Value	İşaret	P Value	İşaret	P Value	İşaret	P Value	İşaret	P Value
F1	-	1.00E+00	+	1.73E+08	+	1.73E+08	+	1.69E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08
F2	+	2.50E-01	+	1.73E+08	+	1.73E+08	+	1.70E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08
F3	-	1.25E-01	+	1.73E+08	+	1.73E+08	+	1.72E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08
F4	+	2.50E-01	+	1.73E+08	+	3.11E+09	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08
F5	-	1.25E-01	+	1.73E+08	+	1.73E+08	+	1.68E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08
F6	-	1.00E+00	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08
F7	-	1.00E+00	+	1.73E+08	-	1.00E+00	-	1.00E+00	+	1.73E+08	+	1.73E+08	+	1.73E+08
F8	-	1.00E+00	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08
F9	-	9.40E-01	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08
F10	+	7.71E-04	-	5.71E-02	+	1.48E-02	+	1.92E+08	+	7.27E-03	+	1.73E+08	+	1.73E+08
F11	-	8.75E-01	+	1.73E+08	-	5.00E-01	-	5.00E-01	+	1.73E+08	+	1.73E+08	+	1.73E+08
F12	+	5.00E-01	+	1.73E+08	-	7.50E-01	-	1.00E+00	+	1.73E+08	+	1.73E+08	+	1.73E+08
F13	-	6.25E-02	+	6.84E+08	-	1.00E+00	+	1.64E-04	+	1.73E+08	+	1.73E+08	+	1.73E+08
F14	+	7.27E-03	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.25E-04
F15	+	1.56E-02	+	1.73E+08	+	8.96E+07	+	1.44E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08
F16	+	8.22E-03	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08
F17	+	3.88E+08	+	2.35E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08
F18	-	2.50E-01	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08
F19	+	4.39E-03	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08
F20	+	3.13E-02	+	1.73E+08	-	1.00E+00	+	4.67E+09	+	1.73E+08	+	1.73E+08	+	1.73E+08
F21	-	6.25E-01	+	1.08E+08	+	1.51E+08	+	1.65E+08	+	1.72E+08	+	1.71E+08	+	1.65E+08
F22	-	1.00E+00	+	1.72E+08	-	1.00E+00	+	2.84E+09	+	1.73E+08	+	1.73E+08	+	1.73E+08
F23	+	6.25E-02	+	1.73E+08	+	1.29E+08	+	1.19E+08	+	1.68E+08	+	1.73E+08	+	1.73E+08
F24	+	1.56E-02	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08	-	1.00E+00
F25	-	1.00E+00	-	1.00E+00	-	1.00E+00	-	1.00E+00	-	1.00E+00	-	1.00E+00	-	1.00E+00
F26	-	1.00E+00	-	1.00E+00	-	1.00E+00	-	1.00E+00	-	1.00E+00	-	1.00E+00	-	1.00E+00
+/-		12/14		23/3		19/7		21/5		24/2		24/2		23/3

Çizelge 4.8. 100 boyut wilcoxon test sonuçları

F.No	GSO_DFB-GSO_B		GSO_DFB-PSO		GSO_DFB-AAA		GSO_DFB-ABC		GSO_DFB-BAT		GSO_DFB-BEE		GSO_DFB-FPA	
	İşaret	P Value	İşaret	P Value	İşaret	P Value	İşaret	P Value	İşaret	P Value	İşaret	P Value	İşaret	P Value
F1	-	6.25E-01	+	3.59E-04	+	3.59E-04	+	3.54E-04	+	1.73E+08	+	1.73E+08	+	1.73E+08
F2	+	1.56E-01	+	1.73E+08	+	3.11E+09	+	3.06E+09	+	1.73E+08	+	1.73E+08	+	1.73E+08
F3	-	5.00E-01	+	1.73E+08	+	1.73E+08	+	1.69E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08
F4	+	1.25E-01	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08
F5	-	6.25E-02	+	1.73E+08	+	1.73E+08	+	1.71E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08
F6	-	5.00E-01	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08
F7	-	1.00E+00	+	1.73E+08	-	1.00E+00	-	1.00E+00	+	1.73E+08	+	1.73E+08	+	1.73E+08
F8	-	2.50E-01	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08
F9	-	5.17E-01	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08
F10	+	5.32E-03	+	4.49E-02	-	1.78E-01	+	1.73E+08	+	2.05E-04	+	1.73E+08	+	1.73E+08
F11	-	4.38E-01	+	1.73E+08	-	1.25E-01	-	5.00E-01	+	1.73E+08	+	1.73E+08	+	1.73E+08
F12	+	1.00E+00	+	1.73E+08	-	6.25E-01	-	1.00E+00	+	1.73E+08	+	1.73E+08	+	1.73E+08
F13	-	6.25E-01	+	1.36E-04	-	5.00E-01	+	4.89E-04	+	1.73E+08	+	1.73E+08	+	1.73E+08
F14	+	1.48E-02	+	3.88E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08	-	2.71E-01
F15	-	2.50E-01	+	1.73E+08	+	1.44E+08	+	1.55E+08	+	1.72E+08	+	1.73E+08	+	1.73E+08
F16	+	3.88E-04	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08
F17	+	8.31E-04	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08
F18	+	3.13E-02	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08
F19	+	8.92E+09	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08
F20	-	7.81E-02	+	1.73E+08	-	5.00E-01	+	2.68E-04	+	1.73E+08	+	1.73E+08	+	1.73E+08
F21	-	2.50E-01	+	1.52E+08	+	1.70E+08	+	1.63E+08	+	1.72E+08	+	1.67E+08	+	1.66E+08
F22	-	1.00E+00	+	1.61E+08	-	1.00E+00	+	2.92E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08
F23	+	6.25E-02	+	1.73E+08	+	2.51E+09	+	1.56E+08	+	1.73E+08	+	1.73E+08	+	1.73E+08
F24	-	7.81E-02	+	1.73E+08	+	3.11E+09	+	1.73E+08	+	1.73E+08	+	1.73E+08	-	1.00E+00
F25	-	1.00E+00	-	1.00E+00	-	1.00E+00	-	1.00E+00	-	1.00E+00	-	1.00E+00	-	1.00E+00
F26	-	1.00E+00	-	1.00E+00	-	1.00E+00	-	1.00E+00	-	1.00E+00	-	1.00E+00	-	1.00E+00
+/-		10/16		24/2		17/9		21/5		24/2		24/2		23/3

5. SONUÇLAR VE ÖNERİLER

5.1 Sonuçlar

Optimizasyon, bir problem için, verilen kısıtlar sağlanarak en uygun çözümü bulma işlemidir. Gerçek dünya problemlerinin bir çoğu, optimizasyon yöntemleri kullanılarak çözülmektedir. Literatürde kullanılan birçok popülasyon tabanlı optimizasyon yöntemi bulunmaktadır.

Popülasyon tabanlı algoritmalar üzerinde yapılan çalışmalarda, araştırmacıların üzerinde durdukları konulardan biri, keşif ve sömürü yeteneklerinin dengede olmasıdır. Keşif yeteneği, popülasyon tabanlı bir algoritmadaki bireylerin arama uzayında ne kadar geniş bir alanda arama yapabildiğini ifade eder. Sömürü yeteneği ise, bu bireylerin en uygun çözüm etrafındaki yakınsama yeteneğidir. Keşif yeteneğinin gerekenden yüksek olması, bireylerin en uygun çözümden uzaklaşmasına sebep olurken, sömürü yeteneğinin gerekenden yüksek olması, erken yakınsama durumuna sebep olabilir. En uygun çözümün bulunması aşamasında, bireylerin keşif ve sömürü yeteneklerinin dengede olması gerekmektedir.

GSO algoritması, içerisinde PSO algoritmasını kullanan, popülasyon tabanlı bir yöntemdir ve iki aşamadan oluşmaktadır. Literatürde yapılan çalışmalar incelendiğinde, PSO algoritmasının erken yakınsama problemi üzerinde durulduğu görülmüştür. Erken yakınsama problemi, keşif ve sömürü yeteneğinin dengede olmadığını göstermektedir ve bu problemin gözlemlenmesinde, popülasyon çeşitliliği yaygınca kullanılmaktadır.

GSO algoritmasının ilk aşamasında, keşif yeteneğinin iyi olduğu ve ikinci aşamasında sömürü yeteneğinin iyi olduğu bilinmektedir. Bu tez çalışmasında, popülasyon çeşitliliği kontrolü yapılarak, GSO algoritmasının birinci seviyesinde, keşif yeteneği iyileştirilmiştir ve ikinci seviyedeki bireylerin geri besleme ile birinci seviyeye taşınması ile sömürü yeteneği geliştirilmiştir. Yapılan geliştirmeler sonucunda, GSO_DFB algoritması önerilmiştir.

Önerilen GSO_DFB algoritmasının performansını gözlemlemek için, literatürde yaygın olarak kullanılan benchmark test fonksiyonları üzerinde 30, 60 ve 100 boyutta çalışmalar yapılmıştır. Yapılan çalışmalardan elde edilen popülasyon çeşitliliği değerleri, temel GSO algoritmasından elde edilen popülasyon çeşitliliği değerleriyle karşılaştırılmıştır. Önerilen GSO_DFB algoritmasının popülasyon çeşitliliği bakımından temel GSO algoritmasını geride bıraktığı gözlemlenmiştir. Ayrıca GSO_DFB

algoritması, GSO_B, AAA, ABC, BAT, BEE, FPA ve PSO algoritmalarıyla karşılaştırılmıştır. GSO_DFB algoritmasının rekabetçi sonuçlar ortaya koyduğu ve boyut arttıkça, başarısını koruduğu gözlemlenmiştir.

Yapılan çalışmalar sonucunda GSO_DFB algoritmasının başarısı, yakınsama eğrisi grafikleri ve Wilcoxon testi ile sunulmuştur. GSO_DFB algoritmasının geliştirmeler sonucunda, GSO_B algoritmasından daha iyi sonuçlar elde ettiği ortaya konulmuştur.

5.2 Öneriler

Bu tez çalışmasında, GSO algoritmasının geliştirilmesi ile elde edilen GSO_DFB algoritmasının başarısı ortaya konulmuştur. Keşif ve sömürü yeteneklerinin dengelenmesinde kullanılan popülasyon çeşitliği kontrolü yöntemi yerine, keşif ve sömürü yeteneklerini kontrol edebilen farklı yöntemler kullanılabilir. Aynı zamanda, temel GSO algoritmasının içerisinde PSO algoritması yerine yakınsama problemi ile karşılaşılmayan farklı algoritmalar kullanılabilir.

KAYNAKLAR

- Bansal, J., Singh, P., Saraswat, M., Verma, A., Jadon, S. ve Abraham, A., 2011, Inertia Weight Strategies in Particle Swarm Optimization, p.
- Beheshti, Z., Shamsuddin, S. M. ve Sulaiman, S., 2014, Fusion Global-Local-Topology Particle Swarm Optimization for Global Optimization Problems, p.
- Bernal, E., Castillo, O. ve Soria, J., 2016, A fuzzy logic approach for dynamic adaptation of parameters in galactic swarm optimization, *Proceedings of 2016 Ieee Symposium Series on Computational Intelligence (Ssci)*.
- Bernal, E., Castillo, O., Soria, J., Valdez, F. ve Melin, P., 2018, A variant to the dynamic adaptation of parameters in galactic swarm optimization using a fuzzy logic augmentation, p.
- Chen, J. ve Mahfouf, M., 2009, Artificial Immune Systems as a Bio-Inspired Optimization Technique and Its Engineering Applications, p.
- Chen, S. M. ve Chien, C. Y., 2011, Solving the traveling salesman problem based on the genetic simulated annealing ant colony system with particle swarm optimization techniques, *Expert Systems with Applications*, 38 (12), 14439-14450.
- Cheng, S. ve Shi, Y., 2011, Diversity control in particle swarm optimization, *2011 IEEE Symposium on Swarm Intelligence*, 1-9.
- Cheng, S., Shi, Y. H. ve Qin, Q. D., 2012, Population Diversity Based Study on Search Information Propagation in Particle Swarm Optimization, *2012 Ieee Congress on Evolutionary Computation (Cec)*.
- Cheng, S., 2013, Population Diversity in Particle Swarm Optimization: Definition, Observation, Control, and Application.
- Cheng, S., Shi, Y. H., Qin, Q. D., Ting, T. O. ve Bai, R. B., 2014, Maintaining Population Diversity in Brain Storm Optimization Algorithm, *2014 Ieee Congress on Evolutionary Computation (Cec)*, 3230-3237.
- Chi, Y., Sun, F., Jiang, L. ve Yu, C., 2012, An efficient population diversity measure for improved particle swarm optimization algorithm, *2012 6th IEEE International Conference Intelligent Systems*, 361-367.
- Chuang, L. Y., Tsai, S. W. ve Yang, C. H., 2008, cns Catfish Particle Swarm Optimization, *2008 Ieee Swarm Intelligence Symposium*, 9-+.
- Chuang, L. Y., Tsai, S. W. ve Yang, C. H., 2011, Chaotic catfish particle swarm optimization for solving global numerical optimization problems, *Applied Mathematics and Computation*, 217 (16), 6900-6916.

- Feng, J., Xie, W. ve Xu, C., 2009, A New Population Diversity Model in Intelligence Optimization Algorithms, *2009 International Conference on Computational Intelligence and Security*, 80-83.
- Fu, X., Liu, W. S., Zhang, B. ve Deng, H., 2013, Quantum Behaved Particle Swarm Optimization with Neighborhood Search for Numerical Optimization, *Mathematical Problems in Engineering*.
- J. Kuo, R. ve W. Hong, C., 2013, Integration of Genetic Algorithm and Particle Swarm Optimization for Investment Portfolio Optimization, p.
- Jau, Y. M., Su, K. L., Wu, C. J. ve Jeng, J. T., 2013, Modified quantum-behaved particle swarm optimization for parameters estimation of generalized nonlinear multi-regressions model based on Choquet integral with outliers, *Applied Mathematics and Computation*, 221, 282-295.
- Jiang, B., Wang, N. ve Wang, L. P., 2013, Particle swarm optimization with age-group topology for multimodal functions and data clustering, *Communications in Nonlinear Science and Numerical Simulation*, 18 (11), 3134-3145.
- Juang, Y. T., Tung, S. L. ve Chiu, H. C., 2011, Adaptive fuzzy particle swarm optimization for global optimization of multimodal functions, *Information Sciences*, 181 (20), 4539-4549.
- Karaboga, D., 2005, An Idea Based on Honey Bee Swarm for Numerical Optimization, Technical Report - TR06, p.
- Kaya, E., Babaoglu, I. ve Kodaz, H., 2017, Galactic Swarm Optimization using Artificial Bee Colony Algorithm, *2017 15th International Conference on Ict and Knowledge Engineering (Ict&Ke)*, 23-28.
- Kennedy, J. ve Eberhart, R., 1995, Particle swarm optimization, *1995 Ieee International Conference on Neural Networks Proceedings, Vols 1-6*, 1942-1948.
- Kiran, M. S., Ozceylan, E., Gunduz, M. ve Paksoy, T., 2012, A novel hybrid approach based on Particle Swarm Optimization and Ant Colony Algorithm to forecast energy demand of Turkey, *Energy Conversion and Management*, 53 (1), 75-83.
- Kiran, M. S., 2015, TSA: Tree-seed algorithm for continuous optimization, *Expert Systems with Applications*, 42 (19), 6686-6698.
- Ktari, R. ve Chabchoub, H., 2013, Essential Particle Swarm Optimization queen with Tabu Search for MKP resolution, *Computing*, 95 (9), 897-921.
- Li Feng, Z. ve Chen Xi, Z., 2012, Self organized parallel genetic algorithm to automatically realize diversified convergence, *2012 IEEE Congress on Evolutionary Computation*, 1-9.

- Li, X., Hao, Z. ve Huang, H., 2010, An evolutionary algorithm with sorted race mechanism for global optimization, *2010 International Conference on Machine Learning and Cybernetics*, 1550-1555.
- Lian, H. ve Qin, Y., 2015, A new fuzzy particle swarm optimization based on population diversity, *Journal of Intelligent & Fuzzy Systems*, 29 (1), 135-147.
- Liu, H., Ding, G. Y. ve Wang, B., 2014, Bare-bones particle swarm optimization with disruption operator, *Applied Mathematics and Computation*, 238, 106-122.
- Lu, Y. C., Jan, J. C., Hung, S. L. ve Hung, G. H., 2013, Enhancing particle swarm optimization algorithm using two new strategies for optimizing design of truss structures, *Engineering Optimization*, 45 (10), 1251-1271.
- McCall, J., 2005, Genetic algorithms for modelling and optimisation, *Journal of Computational and Applied Mathematics*, 184 (1), 205-222.
- Mirjalili, S., 2016, Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems, *Neural Computing & Applications*, 27 (4), 1053-1073.
- Mohamad Nezami, O. ve Bahrampour, A., 2013, Particle Swarm Optimization algorithm based on Diversified Artificial Particles (PSO-DAP), p.
- Muthiah-Nakarajan, V. ve Noel, M. M., 2016, Galactic Swarm Optimization: A new global optimization metaheuristic inspired by galactic motion, *Applied Soft Computing*, 38, 771-787.
- nabil, e., 2016, A Modified Flower Pollination Algorithm for Global Optimization, p.
- Nanda, S., Gulati, I., Chauhan, R., Modi, R. ve Dhaked, U., 2018, A K-Means-Galactic Swarm Optimization-Based Clustering Algorithm with Otsu's Entropy for Brain Tumor Detection, p.
- Nigam, N. ve Amali, G., 2017, Pollution level predictor using artificial neural networks trained with galactic swarm optimization algorithms, p.
- Pham, D., Ghanbarzadeh, A., Koç, E., Otri, S., Rahim, S. ve Zaidi, M., 2005, The Bees Algorithm Technical Note, p.
- Qi, J. ve Pang, S. N., 2010, Re-diversified Particle Swarm Optimization, *Life System Modeling and Intelligent Computing, Pt Ii*, 6329, 30-39.
- Rai, D. ve Seth, 2013, Bio-inspired optimization techniques: a critical comparative study, p.
- Senkerik, R., Viktorin, A., Pluhacek, M. ve Kadavy, T., 2018, On the Population Diversity for the Chaotic Differential Evolution, *2018 IEEE Congress on Evolutionary Computation (CEC)*, 1-8.

- Shen, Y. X., Wang, G. Y. ve Tao, C. M., 2011, Particle Swarm Optimization with Novel Processing Strategy and Its Application, *International Journal of Computational Intelligence Systems*, 4 (1), 100-111.
- Shi, R. ve Liu, X., 2009, A Hybrid Improved Particle Swarm Optimization Based on Dynamic Parameters Control and Metropolis Accept Rule Strategy, *2009 Third International Conference on Genetic and Evolutionary Computing*, 649-653.
- Shi, Y. ve Rc, E., 1998, A Modified Particle Swarm Optimizer, p.
- Shi, Y. H. ve Eberhart, R. C., 2008, Population Diversity of Particle Swarms, *2008 Ieee Congress on Evolutionary Computation, Vols 1-8*, 1063-+.
- Takahashi, R., 2016, Verification of thermo-dynamical genetic algorithm to solve the function optimization problem through diversity measurement — Diversity measurement and its application to selection strategies in genetic algorithms, *2016 IEEE Congress on Evolutionary Computation (CEC)*, 168-177.
- Uymaz, S. A., Tezel, G. ve Yel, E., 2015, Artificial algae algorithm (AAA) for nonlinear global optimization, *Applied Soft Computing*, 31, 153-171.
- Vitorino, L. N., Ribeiro, S. F. ve Bastos, C. J. A., 2015, A mechanism based on Artificial Bee Colony to generate diversity in Particle Swarm Optimization, *Neurocomputing*, 148, 39-45.
- Wang, H., Sun, H., Li, C. H., Rahnamayan, S. ve Pan, J. S., 2013a, Diversity enhanced particle swarm optimization with neighborhood search, *Information Sciences*, 223, 119-135.
- Wang, P. P., Shi, L. P., Zhang, Y. ve Han, L., 2013b, A Hybrid Simplex Search and Modified Bare-bones Particle Swarm Optimization, *Chinese Journal of Electronics*, 22 (1), 104-108.
- Xin, J., Chen, G. ve Hai, Y., 2009, A Particle Swarm Optimizer with Multi-Stage Linearly-Decreasing Inertia Weight, p.
- Xiufen, Z., Wang, M., Anmin, Z. ve Mckay, B., 2004, Evolutionary optimization based on chaotic sequence in dynamic environments, *IEEE International Conference on Networking, Sensing and Control, 2004*, 1364-1369 Vol.1362.
- Yang, X.-S. ve Karamanoglu, M., 2013, Swarm Intelligence and Bio-Inspired Computation: An Overview, In, Eds, p. 3-23.
- Yang, X. S., 2010, A New Metaheuristic Bat-Inspired Algorithm, *Nicso 2010: Nature Inspired Cooperative Strategies for Optimization*, 284, 65-74.
- Yu, X. B., Cao, J., Shan, H. Y., Zhu, L. ve Guo, J., 2014, An Adaptive Hybrid Algorithm Based on Particle Swarm Optimization and Differential Evolution for Global Optimization, *Scientific World Journal*.

Zhang, Y. D., Wang, S. H. ve Ji, G. L., 2015, A Comprehensive Survey on Particle Swarm Optimization Algorithm and Its Applications, *Mathematical Problems in Engineering*.

Zhongli, Z. ve Qiang, W., 2011, An improved chaos immune genetic algorithm, *2011 International Conference on Mechatronic Science, Electric Engineering and Computer (MEC)*, 1147-1150.



ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Oğuzhan UYMAZ
Uyruğu : T.C.
Doğum Yeri ve Tarihi : Elazığ, 25.08.1991
Telefon : 0 (537) 293 62 43
E-Posta : ouymaz@selcuk.edu.tr

EĞİTİM

Derece	Adı	İlçe	İl	Bitirme Yılı
Lise	: Elazığ Anadolu Lisesi	Merkez	Elazığ	2009
Üniversite	: Selçuk Üniversitesi	Selçuklu	Konya	2014
Yüksek Lisans	: Konya Teknik Üniversitesi	Selçuklu	Konya	

İŞ DENEYİMLERİ

Yıl	Kurum	Görevi
2014-2017	Selçuk Üniversitesi Bilgi İşlem Daire Başk.	Bilgisayar Mühendisi
2017-	Selçuk Üniversitesi	Öğretim Görevlisi

UZMANLIK ALANI

Yazılım, Veritabanı, Web programlama.

YABANCI DİLLER

İngilizce Yökdil(53.75)

YAYINLAR

UYMAZ OĞUZHAN, KAYA ERSİN (2017). The Impact of Diversity on The Study of The Galactic Swarm Optimization Algorithm, II. INTERNATIONAL ACADEMIC RESEARC CONGRESS, 1110-1114.(Tam Metin/Bildiri/Sözlü Sunum) (Yüksek Lisans tezinden yapılmıştır)

KAYA ERSİN, UYMAZ OĞUZHAN, AYDİLEK İBRAHİM BERKAN (2017). Performance of Galactic Swarm Optimization using Artificial Algae Algorithm, II.INTERNATIONAL ACADEMIC RESEARC CONGRESS, .(Tam Metin/Bildiri/Sözlü Sunum)

KAYA ERSİN, ÇINAR AHMET CEVAHİR, UYMAZ OĞUZHAN, KORKMAZ SEDAT, KIRAN MUSTAFA SERVET (2018). The Binary Salp Swarm Algorithm with Using Transfer Function, INTERNATIONAL CONFERENCE ON ADVANCED TECHNOLOGİES, . (Tam Metin/Bildiri/Sözlü Sunum)