

Bipolar Parçacık Sürü Optimizasyonu Algoritması

Mashar Cenk GENÇAL^{*1} ORCID 0000-0002-1317-3950

¹Ardahan Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü,
Ardahan

Geliş tarihi: 13.05.2022

Kabul tarihi: 23.09.2022

Atıf şekli/ How to cite: GENÇAL, M.C., (2022). Bipolar Parçacık Sürü Optimizasyonu Algoritması. Çukurova Üniversitesi, Mühendislik Fakültesi Dergisi, 37(3), 617-625.

Öz

Evrimsel Algoritmalar (EA'lar), ağ tasarımı problemleri, yol bulma problemleri, sosyal ve ekonomik planlama gibi karmaşık optimizasyon problemlerini çözmek için yaygın olarak kullanılan algoritmalar. Kullandıkları akıllı yöntemler sayesinde EA'lar, yeni çözümler üretmek için ilk çözümü, yinelemeli olarak, iyileştirirler. Sıklıkla kullanılan EA'lardan biri Parçacık Sürü Optimizasyonu (PSO) algoritmasıdır. PSO algoritması, sürünün sunduğu en iyi sonuç ile sürüdeki her bir bireyin en iyi sonuçlarını temel alarak, verilen bir fonksiyonun optimum değerine ulaşır. PSO başarılı bir algoritma olmasına rağmen, sadece en iyiyi temel alan yapısı sebebiyle, belli bir düzene sahip olmayan, aldatıcı fonksiyon türlerinde, lokal değere sıkışıp optimum değere ulaşmayabilmektedir. Bu makalede, daha önce yapmış olduğumuz bir çalışmadan ilham alarak, PSO'ya bipolar davranış eklenerek, yeni bir algoritma olan Bipolar Parçacık Sürü Optimizasyonu (BPSO) sunulmuştur. BPSO algoritmasında, PSO'da olduğu gibi sadece en iyi bireylerin değil, kötü bireylerin de algoritmanın işleyişine katılmasına olanak sağlanmıştır. BPSO algoritmasının performansını, standart PSO algoritmasının performansıyla kıyaslamak için on test fonksiyonu kullanılmıştır. Test sonuçlarına göre, BPSO, standart PSO'ya göre daha başarılı sonuçlar sunmuştur.

Anahtar Kelimeler: Bipolar eşleşme eğilimi, Evrimsel algoritmalar, Parçacık sürü optimizasyonu

Bipolar Particle Swarm Optimization Algorithm

Abstract

Evolutionary Algorithms (EAs) are commonly used algorithms to solve complex optimization problems such as network design problems, pathfinding problems, social and economic planning. Thanks to the intelligent methods they use, EAs iteratively refine the initial solution to generate new solutions. One of the frequently used EAs is the Particle Swarm Optimization (PSO) algorithm. PSO algorithm reaches the optimum value of a given function based on the best results offered by the swarm and the best results of each individual in the swarm. Although PSO is a successful algorithm, due to its structure based only on the best, it may not reach the optimum value by being stuck in the local value in deceptive function types that do not have a certain order. In this paper, by inspiring from a previous study, bipolar behavior was

*Sorumlu yazar (Corresponding author): Mashar Cenk GENÇAL, masharcenkgençal@ardahan.edu.tr

added to PSO and a new algorithm, Bipolar Particle Swarm Optimization (BPSO), is presented. In the BPSO algorithm, as in PSO, not only the best individuals but also the bad individuals are allowed to participate in the process of the algorithm. Ten test functions were used to compare the performance of the BPSO algorithm with the standard PSO algorithm. According to the test results, BPSO offered more successful results than the standard PSO.

Keywords: Bipolar mating tendency, Evolutionary algorithms, Particle swarm optimization

1. GİRİŞ

Evrimsel algoritmalar (EA'lar), biyolojik evrimden ilham alan ve genellikle doğadaki canlıların davranışlarını taklit eden meta-sezgisel algoritmalar. Doğayı taklit ederek modellenen EA'ların, optimizasyon problemlerinde sunduğu başarılı sonuçlar yeni düşünceleri doğurmuş ve böylelikle EA'lar, literatürde sıklıkla kullanılan algoritmalar olmaya başlamışlardır.

Rassallığı temel olan arama süreçleri, çözümlerden (bireyler) oluşan bir gruba (popülasyon) optimum değere(lere) ulaşmak için kullanılır. Bir EA, temel olarak, başlangıç popülasyonu adı verilen potansiyel çözümleri rastgele oluşturarak sürecine başlar. Başlangıç popülasyonu oluşturulduktan sonra, algoritma uygunluk fonksiyonunu kullanarak, popülasyondaki her bir bireyin uygunluk değerini belirler. Daha sonra, biyolojik evrimden ilham alan süreçleri başlar: seçim, çaprazlama ve mutasyon.

Bir sonraki neslin popülasyonu, oluşan yeni bireyler ile mevcut popülasyondaki bireylerden seçkin olanları seçilerek (Elitizm) oluşturulur. Algoritma, oluşan yeni nesli kullanarak seçim aşamasına döner ve sonlandırma kriterleri (ya optimum çözümler bulunur ya da maksimum nesil sayısına ulaşılır) karşılanana kadar adımları tekrarlar.

Parçacık sürüsü optimizasyonu (PSO) gibi birçok EA, optimal değere yakın bir çözüm bulduklarında arama alanını araştırmayı bırakma eğilimindedir. Keşfetmekten (*exploration*) vazgeçmelerinin nedeni, keşfedilen alanı kullanmanın optimal çözümü geliştirdiğine dair yaygın bir inançtır. Ancak, iyi bir çözüm tercih edildiğinde, ortalama veya zayıf çözümler (çoğunlukla) yeniden üretim sürecinden çıkarılır. Bu nedenle, daha az elverişli

bireylerde bulunan belirli miktardaki genetik materyal kaybedilmiş olur. Bu durum popülasyonun genetik çeşitliliğinin kaybolmasına ve algoritmanın yerel minimumlara takılıp kalmasına ya da erken yakınsamasına neden olabilmektedir.

PSO algoritması her ne kadar optimizasyon algoritmalarında başarılı sonuçlar sunsa da, belli bir düzene sahip olmayan ya da aldatıcı bir yapıya sahip olan fonksiyon türlerinde, beklenen ideal sonucu sunmayabilmektedir.

Bu makalenin amacı, standart bir PSO'nun karşılaşılabileceği yukarıda bahsedilen durumlar için, algoritmanın performansını iyileştirmektir. Bu sebeple, daha önce yapmış olduğumuz bir çalışmadan esinlenerek, standart PSO'ya bipolar davranış eklenmiş ve yeni bir algoritma olan Bipolar Parçacık Sürü Optimizasyonu (BPSO) tanıtılmıştır. BPSO algoritmasında sadece en iyi bireylerin değil, kötü bireylerin de eşleşmeye katılması sağlanmış ve böylece popülasyondaki genetik çeşitliliği artırmak ve algoritmanın keşfetme yeteneğini geliştirmek amaçlanmıştır.

Önerilen iyileştirmelerin performansa katkısını belirlemek için, BPSO, standart PSO ile test fonksiyonları üzerinde karşılaştırılmıştır.

Bu makale şu şekilde düzenlenmiştir: İkinci bölümde, literatürde yaygın olarak kullanılan ve PSO'yu temel alan algoritmalar ile ilgili bilgiler sunulmuş ve üçüncü bölümde ise, sunulan yönteme ilham kaynağı olan algoritmalar ilgili bilgiler verilmiştir. Dördüncü bölümde, bipolar PSO algoritması sunulurken, beşinci bölümde kullanılan test fonksiyonları ile ilgili bilgiler verilmiş, yapılan testler ve test sonuçları gösterilmiştir, makale 6. Bölüm ile sonlandırılmıştır.

2. ÖNCEKİ ÇALIŞMALAR

PSO'yu temel alan algoritmalar ilk örneklerden biri Evrimsel PSO (Evolutionary Particle Swarm Optimization) dur [1]. Evrimsel PSO algoritması, PSO algoritmasına mutasyon ve seçim yöntemi ekleyerek oluşturulmuştur. Popülasyondaki her bir bireye, bir ağırlık değeri verilmiş ve Gauss dağılımı kullanılarak bu değerler mutasyona uğratılmıştır [1]. Seçim yöntemi olarak ise standart turnuva yöntemi kullanılmıştır [2].

PSO'nun çıkışından sonraki ilk yıllarda yapılan bir diğer çalışma ise Uygunluk-Mesafe-Oran PSO (Fitness-Distance-Ratio Particle Swarm Optimization) algoritmasıdır [3]. Bu algortmada, her bir parçacık kendisinin ve sürünün en iyi değerini kaydetmekle birlikte, aynı zamanda çevresindeki parçacıklarla da irtibat halindedir. Parçacıklar, belirli bir alanda bulunan [3] komşularının uygunluk değerine de bakarak hareket etmektedirler. Bu çalışmanın sunulduğu sempozyumda, PSO'yu temel alan, dikkate değer bir diğer çalışma ise Gauss Mutasyonu ile PSO'dur (Particle Swarm Optimization with Gaussian Mutation) [4]. Bu çalışmada, yaygın olarak kullanılan bir diğer EA olan Genetik Algoritmaların (Genetic Algorithms, GAs) [5] mutasyon aşaması, PSO'ya eklenmiştir. Standart bir GA'dan farklı olarak, mutasyon için rastgele seçilen bireylerin konumları, Gauss dağılımı altındaki olasılıkla belirlenmiştir [4]. Esmine ve arkadaşları [6] da, GAs'nin mutasyon aşamasını PSO'ya eklemiş, ancak Gauss dağılımı kullanmak yerine, rassallığa dayalı bir formülle [6] mutasyonu gerçekleştirmişlerdir.

Juang, 2004 yılında, PSO ve GAs'yi birlikte kullanan hibrit bir yöntemi tanıtmıştır [7]. Algoritma, başlangıçta, bir GA olarak çalıştırılmış ve bir sonraki nesil için seçkin bireyler seçilmiştir (Elitism). Seçkin bireyler, PSO algoritması kullanılarak daha iyi bireylere geliştirilmiş ve yeni nesil geliştirilen bu bireyler kullanılarak oluşturulmuştur.

Bir diğer çalışmada, Zhan ve arkadaşları, standart PSO'dan daha iyi bir arama sundukları, Uyarlanabilir PSO (Adaptive Particle Swarm

Optimization, APSO) algoritmasını tanıtmışlardır [8]. Standart PSO'yu temel alan algoritma, iki temel adımdan oluşmaktadır: tahmin prosedürü, seçkin yaklaşım. Tahmin prosedüründe, ilk olarak, popülasyon dağılımı ve parçacık uygunluğu değerlendirilir. Her nesilde keşif (*exploration*), sömürü (*exploitation*), yakınsama (*convergence*) ve atlama (*jumping out*) dahil olmak üzere dört tanımlanmış evrimsel durumdan biri gerçekleştirilerek, algoritmanın kontrolü sağlanmaya çalışılır. Seçkin yaklaşımda ise, olası yerel optimumdan kurtulmak için, global en iyi parçacık üzerinde atlamalar gerçekleştirilir.

Pedersen ve arkadaşları, Yerel Tek Modlu Örnekleme (Local Unimodal Sampling) yöntemini, bir meta-iyileştirici olarak, PSO algoritmasında kullanmışlar ve bu yeni yöntemle basitleştirilmiş PSO adını vermişlerdir [9]. Her bir problem türü için, o probleme ait en uygun PSO parametrelerini belirlemiş ve yeni yöntemlerini yapay sinir ağı problemlerinde uygulamışlardır.

Son yıllarda yapılan bir çalışmada, standart PSO algoritmasındaki parçacık (*particle*) değerlerinin, çözüm değerlerinden ziyade, olasılık dağılımlarını temsil ettiği ve PSO güncellemesinin olasılık dağılımlarını değiştirdiği, Tamsayı ve Kategorik PSO (Integer and Categorical PSO, ICPSO) algoritması tanıtmıştır [10]. Bu algortmada, sürekli (continuous) değişkenler yerine, soyut (discrete) değişkenler kullanılmıştır.

Wang ve arkadaşları, 2018 yılında, uyarlanabilir öğrenme stratejisini (adaptive learning strategy) PSO'ya ekleyerek hibrit bir algoritma oluşturmuşlardır [11]. Uyarlanabilir Öğrenme tabanlı PSO (Adaptive Learning based PSO, ALPSO) adını verdikleri bu yeni yaklaşımla, sürü yapısı öğreniminin ve parçacıkların yerel arama stratejisi öğreniminin iyileştirilmesine odaklanmışlar, keşif ve sömürü arasındaki dengeyi sağlayabildikleri için standart PSO'dan daha başarılı sonuçlar elde ettiklerini iddia etmişlerdir.

2020 yılına gelindiğinde, Darwish ve arkadaşları, ortogonal öğrenme PSO (orthogonal learning particle swarm optimization, OLPSO) algoritmasını sunmuşlardır [12]. Bu yayında,

sağlıklı ve sağlıksız yaprak görüntülerini sınıflandırarak, görev bitki hastalığı teşhisi için, VGG16 ve VGG19 adlarında, önceden eğitilmiş iki evrişimli sinir ağının (Convolutional Neural Networks, CNN) bir topluluk modelini geliştirmişlerdir. Geliştirilen bu modelin parametrelerini optimize etmek için ise, sundukları yöntem olan, OLPSO'yu kullanmışlardır.

3. ESİNLENİLEN ALGORİTMALAR

3.1. Parçacık Sürü Optimizasyonu

Algoritma 1. Parçacık Sürü Optimizasyonu

N: Sürüdeki parçacık sayısı
X_j: *j*. parçacık konum vektörü
f(): Uygunluk fonksiyonu
gBest: Global en iyi konum vektörü
pBest_j: *j*. parçacığın şimdiye kadar ki bulunduğu en iyi konum vektörü
MaxIter: maksimum iterasyon sayısı
count=0;
Sürü içerisindeki her bir parçacığa rastgele konum belirle;
pBest_j ve *gBest* konumlarını belirle;
while *count* ≤ *MaxIter*
 for *j* = 1 : *N* (her bir parçacık için)
 v_j parçacık hız vektörünü hesapla;
 x_j parçacık yeni konum vektörünü hesapla;
 if $f(x_j) \leq f(pBest_j)$ *pBest_j* = *x_j*;
 end
 for *j* = 1 : *N*
 if $f(x_j) \leq f(gBest)$ *gBest* = *x_j*;
 end
 gBest değerini bulunan en iyi değer olarak döndür;
 count++;
end

Eberhart ve Kennedy tarafından sunulan Parçacık Sürü Optimizasyonu (Particle Swarm Optimization, PSO) algoritması, ilk olarak, kuş ve balık sürülerinin sosyal davranışlarını taklit etmesi için tasarlanmıştır [13]. Ancak, yapılan incelemeler sonucunda, PSO'nun aslında optimizasyon yaptığı anlaşılmıştır. Optimizasyon problemlerinde başarılı sonuçlar sunan algoritma,

araştırmacıların dikkatini çekmiş ve yeni birçok algoritmaya ilham vermiştir.

Başlangıçta PSO, rasgele çözümlerle ilk popülasyonu (sürü, *swarm*) oluşturur. Sürüde yer alan her parçacığın (*particle*) bir de hız değeri (*velocity*) bulunmaktadır. Böylelikle, parçacık bu hız değeri sayesinde arama uzayında dolaşabilmektedir.

Sürüde yer alan her bir parçacığın hareketini etkileyen diğer unsurlar ise parçacığın ulaşabildiği en iyi konum (*pBest*) ve sürünün bulabildiği en iyi konumdur (*gBest*). Parçacık, bu konum değerlerinden daha iyi bir konum keşfederse, keşfedilen konumun durumuna göre, *pbest*, *gbest* değerlerinden biri veya her ikisi de güncellenip, istenilen kritere ya da kriterlere ulaşana kadar arama işlemi devam ettirilir (Algoritma 1).

3.2. Bipolar Eşleşme Eğilimi

Bipolar Eşleşme Eğilimi (Bipolar Mating Tendency, BMT), standart bir GA'da kullanılan Standart Turnuva (ST) [2] seçim yöntemine dayanmaktadır [14].

Algoritma 2. Bipolar Eşleşme Eğilimi

İlk eş ST yöntemiyle belirle;
İkinci eş olmaya aday bireyleri rastgele seç ve uygunluk değerlerini hesapla;
Adaylardan uygunluk değeri en iyi ve en kötü olanları belirle;
if (rastgele değer ≤ iki kutupluluk olasılığı)
 İkinci eş uygunluk değeri en iyi olan birey olarak seç;
 Eşleri çaprazla;
else
 İkinci eş uygunluk değeri en kötü olan birey olarak seç;
 Eşleri çaprazla;
end

BMT, popülasyonda bulunan eş olmaya aday bireyleri iki gruba rastgele ayırır. İlk seçilen eş, ST de olduğu gibi, grup üyeleri arasından seçilen en uygun ya da diğer adıyla en iyi bireydir.

İlk eş seçiminin aksine, ikinci eşin seçimi için kullanılacak olan uygunluk kriteri belirsizdir. Birinci eşin *bipolar* (iki kutuplu) olan psikolojik durumuna göre birinci eş, grup üyelerinden en iyi bireyi ya da en kötü bireyi ikinci eş olarak seçebilmektedir (Algoritma 2).

4. ÖNERİLEN YÖNTEM

BMT yönteminin GA'da uygulanması, popülasyondaki çeşitliliği artırmış ve standart GA'ya göre daha iyi sonuçlar sunmasını sağlamıştır [14]. Elde edilen bu başarı, *bipolar* davranışın başka EA'larda da gözlenmesi gerektiği düşüncesini oluşturmuş ve böylelikle Bipolar Parçacık Sürü Optimizasyonu (BPSO) algoritmasının fikri doğmuştur.

BPSO algoritması tamamen PSO algoritmasına dayanmaktadır. PSO algoritmasından tek farkı hız vektörüdür (*velocity vector*).

Standart bir PSO'da, j 'ninci birey için hız vektörü Eşitlik 1'deki gibi hesaplanmaktadır [15]:

$$\vec{v}_j(t+1) = w \cdot \vec{v}_j(t) + c_1 \vec{r}_1 (\overrightarrow{pBest}_i - \vec{x}_j(t)) + c_2 \vec{r}_2 (\overrightarrow{gBest} - \vec{x}_j(t)) \quad (1)$$

Yukarıdaki eşitlikte (Eşitlik 1), c_1 ve c_2 değerleri ivme katsayıları, w ise eylemsizlik ağırlığı olarak adlandırılmaktadır. Eşitlik 1'de verilen \vec{r}_1 ve \vec{r}_2 vektörleri, $[0,1]$ aralığından rastgele seçilen reel değerlerin oluşturduğu vektörleri temsil etmektedir. Son olarak, $\vec{x}_j(t)$ vektörü ise, j 'ninci bireyin t anındaki konumunu göstermektedir.

BPSO algoritmasıyla, PSO'ya *bipolar* davranış eklenmiş, böylelikle popülasyondaki bireyin psikolojik durumuna bağlı olarak hız vektörü belirlenmiştir. Bireyin psikolojik durumu, $(0,1)$ aralığında rastgele belirlenen reel değerler ile [14]'de kullanılan iki kutupluluk olasılık değerine $(0,25)$ bağlıdır. Her birey için rastgele seçilen bu değerler, $0,75$ 'den küçükse Eşitlik 1'de verilen hız vektörü, değilse Eşitlik 2'de verilen hız vektörü kullanılır:

Algoritma 3. Bipolar PSO

N : Sürüdeki parçacık sayısı
 X_j : j . parçacık konum vektörü
 $f()$: Uygunluk fonksiyonu
 $gBest$: Global en iyi konum vektörü
 $gWorst$: Global en kötü konum vektörü
 $pBest_j$: j . parçacığın şimdiye kadar ki bulunduğu en iyi konum vektörü
 $pWorst_j$: j . parçacığın şimdiye kadar ki bulunduğu en kötü konum vektörü
 $MaxIter$: maksimum iterasyon sayısı
 $count=0$;
Sürü içerisindeki her bir parçacığa rastgele konum belirle;
 $pBest_i$, $gBest$, $pWorst_j$ ve $gWorst$ konumlarını belirle;
while $count \leq MaxIter$
 for $j = 1 : N$ (her bir parçacık için)
 if (rastgele değer \leq iki kutupluluk olasılığı)
 v_j hız vektörünü Eşitlik 1 ile hesapla;
 else
 v_j hız vektörünü Eşitlik 2 ile hesapla;
 end
 x_j parçacık yeni konum vektörünü hesapla;
 if $f(x_j) \leq f(pBest_j)$ $pBest_j = x_j$;
 if $f(x_j) > f(pWorst_j)$ $pWorst_j = x_j$;
 end
 for $j = 1 : N$
 if $f(x_j) \leq f(gBest)$ $gBest = x_j$;
 if $f(x_j) > f(gWorst)$ $gWorst = x_j$;
 end
 $gBest$ değerini bulunan en iyi değer olarak döndür;
 $count++$;
end

$$\vec{v}_j(t+1) = w \cdot \vec{v}_j(t) + c_1 \vec{r}_1 (\overrightarrow{pWorst}_j - \vec{x}_j(t)) + c_2 \vec{r}_2 (\overrightarrow{gWorst} - \vec{x}_j(t)) \quad (2)$$

Böylece, birey ya Eşitlik 1'i seçerek standart PSO'daki gibi davranmış ya da Eşitlik 2'deki gibi en iyiler yerine en kötülerini temel alan hız vektörüyle hareket etmiştir, bkz. Algoritma 3.

Eşitlik 1 ve Eşitlik 2'de verilen hız vektörlerinden birini kullanarak parçacık hızını hesaplayan algoritma, standart PSO'daki konum güncelleme

formülü (Eşitlik 3'te verilen) ile parçacığın yeni konumunu belirler.

$$\vec{x}_j(t+1) = \vec{v}_j(t+1) + \vec{x}_j(t) \quad (3)$$

5. TESTLER

BPSO algoritması, standart PSO algoritmasına dayandığından, BPSO algoritmasının performansı sadece standart PSO algoritmasının performansı ile kıyaslanmıştır. BMT algoritması bir seçim yöntemi olup, BPSO'ya fikri anlamda katkıda bulunduğundan, BMT yöntemi yapılan testlerde kıyaslama için kullanılmamıştır.

5.1. Test Fonksiyonları

BPSO algoritmasının performansını test etmek için, literatürde yaygın olarak kullanılan test fonksiyonlarından [16], Çizelge 1'de belirtilen on çok modlu fonksiyon kullanılmıştır.

Çizelge 1. Kullanılan test fonksiyonları

Fonksiyon no	Test fonksiyonları
f_1	Branins
f_2	Fifth function of de Jong
f_3	Drop wave
f_4	Goldstein-price
f_5	Langermann
f_6	Michalewicz
f_7	Rastrigin
f_8	Schubert
f_9	Schwefel
f_{10}	Six Hump Camel Back

Çok modlu fonksiyonların birden fazla yerel ekstremum değerine sahip olması, optimizasyon algoritmalarının bu ekstremum değerlerden birine yakalanma ihtimalini artırmaktadır [16]. Bu nedenle, algoritmaların performansını test etmek için çok modlu fonksiyonlar tercih edilmiştir.

Branins fonksiyonunda global minimum değeri $f_1 = 0,397887$ değeridir.

Fifth function of De Jong fonksiyonunda test alanı $-65,536 \leq x_i \leq 65,536$, $i = 1, 2$ bölgesinde sınırlandırılmıştır. Bu fonksiyonda yerel minimum değerleri düzensiz bir şekilde dağılmıştır.

Drop Wave fonksiyonunda test alanı genellikle $-5,12 \leq x_i \leq 5,12$ $i = 1, \dots, n$ bölgesinde sınırlandırılmıştır ve global minimum değeri $f_3 = -1$ dir.

Goldstein-Price fonksiyonunda test alanı $-2 \leq x_i \leq 2$, $i = 1, 2$ bölgesinde sınırlandırılmıştır. Global minimumu $f_4 = 0$ değeridir.

Langermann fonksiyonunun yerel minimum değerleri düzensiz bir şekilde dağılmıştır.

Michalewicz fonksiyonunda test alanı $0 \leq x_i \leq \pi$ $i=1, \dots, n$ bölgesinde sınırlandırılmıştır ve yerel minimum değerlerinden biri $f_6 = -1,8$ dir.

Rastrigin fonksiyonunda test alanı genellikle $-5,12 \leq x_i \leq 5,12$ $i = 1, \dots, n$ bölgesinde sınırlandırılmıştır. Global minimum değeri $f_7 = 0$ dir.

Schubert fonksiyonunda ise test alanı $-5,12 \leq x_i \leq 5,12$ $i=1, \dots, n$ bölgesinde sınırlandırılmıştır. Fonksiyonun yerel minimum değerleri düzensiz bir şekilde dağılmıştır.

Schwefel fonksiyonunda test alanı $-500 \leq x_i \leq 500$, $i = 1, \dots, n$ bölgesinde sınırlandırılmıştır. Global minimumu $f_9 = -418,9829 * n$ değeridir.

Six Hump Camel Back fonksiyonunda test alanı $-3 \leq x_1 \leq 3$, $-2 \leq x_2 \leq 2$ bölgesidir. $f_{10} = -1,0316$ değeri fonksiyonun global değeridir.

5.2. Test Sonuçları

Yapılan testlerde kullanılan algoritmalar, 20 farklı rastgele değerle çalıştırılmış ve böylece testlerin doğruluğunu ve güvenilirliğini sağlamak amaçlanmıştır. Çizelge 2'de sunulan sonuçlar bu 20 çalıştırmanın medyan ve standart sapma değerleridir.

Testler, popülasyon büyüklüğünün 20 ve tekrar sayısının (*iteration*) 100 olduğu durumda (düşük maliyetli bir durum altında) yapılmış, önerilen yöntem olan BPSO'nun, PSO'ya alternatif olup olmayacağı test edilmiştir. Ayrıca, testlerde standart PSO ayarları kullanılmıştır [13].

Yapılan tüm testlerde, kullanılan fonksiyonların minimum değerlerine ulaşmak amaçlanmıştır. Bu nedenle, Çizelge 2'de verilen sonuçlara bakıldığında, diğerine göre daha düşük değere ulaşabilen algoritma başarılı kabul edilmiş ve ulaştığı değer koyu olarak belirtilmiştir.

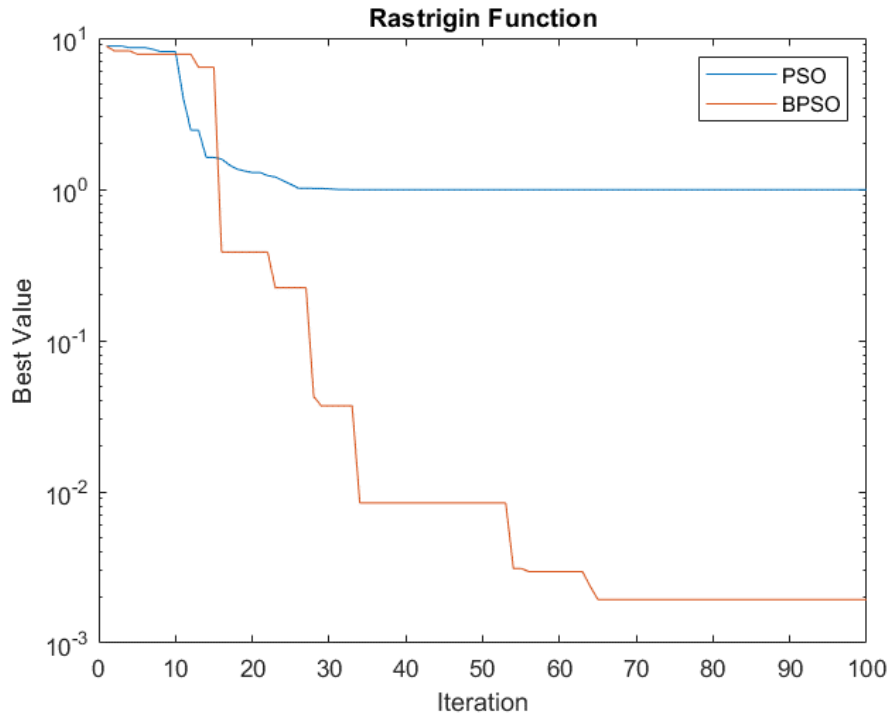
Çizelge 2 incelendiğinde, BPSO algoritması tüm test fonksiyonlarında da en iyi sonuçları vermiş, Fifth function of De Jong, Langermann, Michalewicz, Rastrigin, Schubert ve Schwefel fonksiyonlarında ise, standart PSO'ya kıyasla, çok daha başarılı sonuçlar sunmuştur.

Ayrıca, PSO ile BPSO'nun, Rastrigin fonksiyonundaki yakınsama yeteneklerinin

gösterildiği Şekil 1 incelenirse, PSO'nun yaklaşık 30. iterasyonda aramayı bıraktığı, buna rağmen, önerilen yöntem olan BPSO'nun yaklaşık 65. iterasyona kadar en iyiyi aramaya devam ettiği gözlemlenebilir. Yapılan bu gözlem neticesinde, BPSO'nun, belirtilen fonksiyon için, PSO'ya göre daha iyi bir arama sunduğu söylenebilir.

Çizelge 2. BPSO ve PSO'nun karşılaştırmalı test sonuçları

Fonksiyon no	PSO		BPSO	
	Medyan	Standart sapma	Medyan	Standart sapma
f_1	0,3979	0	0,3979	6,6831e-05
f_2	7,8721	6,2089	1,9920	2,1812
f_3	-0,9362	1,1102e-16	-0,9362	0,0314
f_4	3	11,3841	3	0,0013
f_5	-3,6775	0,7308	-3,8209	0,7527
f_6	-1,5076	0,3631	-1,8012	0,2533
f_7	0,9949	0,6633	0,0012	0,3459
f_8	-160,0571	78,3454	-210,4365	74,0412
f_9	-719,5274	111,8444	-837,9598	82,8065
f_{10}	-1,0316	2,3405e-16	-1,0316	2,8017e-05



Şekil 1. PSO ile BPSO'nun Rastrigin fonksiyonundaki yakınsama yetenekleri

6. SONUÇ

PSO algoritması, başarılı bir algoritma olmasına rağmen, zorlayıcı fonksiyon türlerinde, beklenen ideal sonucu sunmayabilmektedir. PSO'nun karşılaştığı bu sorunu çözmek için, standart PSO'ya bipolar davranış eklenmiş, sadece en iyi bireyleri değil, kötü bireyleri de algoritma işleyişine dahil eden yeni bir yöntem, BPSO, tanıtılmıştır.

BPSO'yu, standart PSO ile kıyaslamak için literatürde yaygın olarak kullanılan on test fonksiyonu kullanılmıştır. Yapılan testlerin sonucunda, geliştirilen yöntem olan BPSO'nun, on test fonksiyonunun altında en iyi sonucu sunduğu, kalan dört test fonksiyonun da ise PSO ile aynı sonuçları verdiği gözlemlenmiştir. Yapılan bu gözlem ışığında, BPSO'nun, test edilen durumlar altında, PSO'ya göre daha başarılı olduğu aşıkardır.

Alınan sonuçlar göz önünde bulundurularak, BPSO'nun yüksek maliyetli durumlar altında da test edilmesi, gelecek çalışmalar için, tavsiye edilmektedir.

7. KAYNAKLAR

1. Miranda, V., Fonseca, N., 2002. EPSO-Evolutionary Particle Swarm Optimization, a New Algorithm with Applications in Power Systems. In IEEE/PES Transmission and Distribution Conference, IEEE, USA, 2, 745-750.
2. Blicke, T., Thiele, L., 1996. A Comparison of Selection Schemes Used in Evolutionary Algorithms. *Evolutionary Computation*, 4(4), 361-394.
3. Peram, T., Veeramachaneni, K., Mohan, C.K., 2003. Fitness-distance-ratio Based Particle Swarm Optimization. In Proceedings of the 2003 IEEE Swarm Intelligence Symposium. SIS'03, pp. 174-181, 25-26 April 2003, Indiana, USA.
4. Higashi, N., Iba, H., 2003. Particle Swarm Optimization with Gaussian Mutation. In Proceedings of the 2003 IEEE Swarm Intelligence Symposium, SIS'03, 25-26 April 2003, Indiana, USA.
5. Holland, J.H., 1975. *Adaptation in Natural and Artificial Systems*. Ann Arbor, University of Michigan Press, Michigan, USA.
6. Esmine, A.A., Lambert-Torres, G., De Souza, A.Z., 2005. A Hybrid Particle Swarm Optimization Applied to Loss Power Minimization. *IEEE Transactions on Power Systems*, 20(2), 859-866.
7. Juang, C.F., 2004. A Hybrid of Genetic Algorithm and Particle Swarm Optimization for Recurrent Network Design. *IEEE Transactions on Systems, Man and Cybernetics*, 34(2), 997-1006.
8. Zhan, Z.H., Zhang, J., Li, Y., Chung, H.S.H., 2009. Adaptive Particle Swarm Optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(6), 1362-1381.
9. Pedersen, M.E.H., Chipperfield, A.J., 2010. Simplifying Particle Swarm Optimization. *Applied Soft Computing*, 10(2), 618-628.
10. Strasser, S., Goodman, R., Sheppard, J., Butcher, S., 2016. A New Discrete Particle Swarm Optimization Algorithm. In Proceedings of the Genetic and Evolutionary Computation Conference, GECCO'16, 20-24 July 2016, Colorado, 53-60, USA.
11. Wang, F., Zhang, H., Li, K., Lin, Z., Yang, J., Shen, X.L., 2018. A Hybrid Particle Swarm Optimization Algorithm Using Adaptive Learning Strategy. *Information Sciences*, 436, 162-177.
12. Darwish, A., Ezzat, D., Hassanien, A.E., 2020. An Optimized Model Based on Convolutional Neural Networks and Orthogonal Learning Particle Swarm Optimization Algorithm for Plant Diseases Diagnosis. *Swarm and Evolutionary Computation*, 52, 100616.
13. Eberhart, R., Kennedy, J., 1995. A New Optimizer Using Particle Swarm Theory. *Micro Machine and Human Science, 1995 MHS'95, Proceedings of the Sixth International Symposium*, 4-6 October 1995, Nagoya, Japan.
14. Gençal, M.C., Oral, M., 2022. Bipolar Mating Tendency: Harmony Between the Best and the

- Worst Individuals. *Arabian Journal for Science and Engineering*, 47, 1849–1871.
15. Tao, X., Li, X., Chen, W., Liang, T., Li, Y., Guo, J., Qi, L., 2021. Self-Adaptive Two Roles Hybrid Learning Strategies-based Particle Swarm Optimization. *Information Sciences*, 578, 457-481.
 16. Molga, M., Smutnicki, C., 2005. Test Functions for Optimization Needs. <http://www.robertmarks.org/Classes/ENGR5358/Papers/functions.pdf>, Erişim tarihi: Kasım 2021.

