



Makale / Research Paper

Diferansiyel Evrim Algoritması Temelli Global Arama Stratejisi ile Geliştirilmiş Yeni Bir Yapay Denizanası Arama Algoritması

Gülnur YILDIZDAN*

^a Selçuk Üniversitesi, Kulu Meslek Yüksekokulu, Bilgisayar Teknolojileri Bölümü, 42770, Konya / Türkiye

*gavsar@selcuk.edu.tr

Received/Geliş: 16.06.2022

Accepted/Kabul: 07.09.2022

Öz: Metasezgisel algoritmalar, bir problemi çözmek için olası çözümlerden daha etkili olanına karar vermek için kullanılan ve doğal fenomenlerden esinlenen algoritmalar. Her geçen gün sayıları artmakta olan bu algoritmalar, kesin çözümü garanti etmemesine rağmen kesin çözüm etrafındaki bir çözüme hızlı şekilde ulaşmayı vad eder. Yapay Denizanası Arama Algoritması(YDA) da 2021 yılında önerilmiş yeni bir metasezgisel algoritmadır. Bu çalışmada, YDA'nın global arama kabiliyetini geliştirmek amacıyla standart algoritmanın global arama bölümünde bir düzenleme yapılmıştır. Buna göre, Diferansiyel Evrim Algoritmasındaki başarılı mutasyon stratejilerinden biri olan "current-to-best" yaklaşımı, YDA'nın global arama yöntemine entegre edilmiştir. Bu düzenleme sonucu elde edilen gelişmiş algoritma(MYDA), yedi tanesi tek modlu, beş tanesi çok modlu özellikle olmak üzere toplam on iki kıyaslama fonksiyonu üzerinde 10,30,50,100,500 ve 1000 boyut için test edilmiştir. Ayrıca MYDA, literatürden seçilen algoritmalarla da karşılaştırılmıştır. Sonuçlar istatistik testler yardımıyla yorumlanmıştır. Elde edilen sonuçlar incelendiğinde, önerilen algoritmanın tüm fonksiyonlarda tüm boyutlar için standart algoritmadan daha iyi performans gösterdiği tespit edilmiştir. Literatürle yapılan karşılaştırmada ise algoritmanın başarılı ve yarışmacı sonuçlar ürettiği belirlenmiştir.

Anahtar Kelimeler: Diferansiyel evrim algoritması, metasezgisel algoritma, sürekli optimizasyon, yapay denizanası arama algoritması, klasik kıyaslama fonksiyonları.

A Novel Artificial Jellyfish Search Algorithm Improved with a Differential Evolution Algorithm-Based Global Search Strategy

Abstract: Metaheuristic algorithms are algorithms inspired by natural phenomena and that are used to decide which possible solution is more efficient to solve a problem. Although these algorithms, whose numbers are increasing day by day, do not guarantee the exact solution, they promise to reach a solution around the exact solution quickly. The Artificial Jellyfish Search Algorithm (YDA) is also a new metaheuristic algorithm proposed in 2021. In this study, a modification has been made to the global search part of the standard algorithm in order to improve the global search capability of YDA. Accordingly, the "current-to-best" approach, which is one of the successful mutation strategies in the Differential Evolution Algorithm, has been integrated into the global search method of YDA. The advanced algorithm (MYDA) obtained as a result of this modification has been tested for 10, 30, 50, 100, 500 and 1000 dimensions on a total of twelve benchmark functions, seven of which are uni-modal and five are multi-modal. In addition, MYDA has also been compared with algorithms selected from the literature. The results have been interpreted with the help of statistical tests. When the results obtained are examined, it has been determined that the proposed algorithm outperforms the standard algorithm for all dimensions in all functions. In the comparison with the literature, it has been determined that the algorithm produces successful and competitive results.

Keywords: Differential evolution algorithm, metaheuristic algorithm, continuous optimization, artificial jellyfish search algorithm, classic benchmark functions.

Bu makaleye atıf yapmak için

Yıldızdan, G., "Diferansiyel Evrim Algoritması Temelli Global Arama Stratejisi ile Geliştirilmiş Yeni Bir Yapay Denizanası Arama Algoritması" *El-Cezerî Fen ve Mühendislik Dergisi*, 9(4), 1178-1192.

How to cite this article

Yıldızdan, G., "A Novel Artificial Jellyfish Search Algorithm Improved with a Differential Evolution Algorithm-Based Global Search Strategy" *El-Cezerî Journal of Science and Engineering*, 9(4), 1178-1192.

ORCID ID: ^a 0000-0001-6252-9012

1. Giriş

Günlük hayatta karşılaştığımız problemlere farkında olarak ya da olmayarak en ideal çözümü bulmaya çalışırız. Bu çabanın amacı, kısıtlı kaynakları en ideal şekilde kullanılması olarak nitelendirilebilir. Sadece insanlarda değil, bir çok canlı türünde rastlanan bu davranış, belirli bir amacı gerçekleştirmek için belirli kısıtlar ve sınırlar dahilinde en uygun çözümün bulunması olarak tanımlanır ve optimizasyon olarak isimlendirilir. Optimizasyon problemleri, sadece günlük hayatta değil, matematik, finans, bilgisayar ve mühendislik gibi daha birçok alanda karşımıza çıkmaktadır. Optimizasyon problemlerinin çözülmesine ve bilgisayar ortamında modellenmesine olan ihtiyaç, farklı optimizasyon tekniklerinin ortaya çıkmasına neden olmuştur. Optimizasyon problemlerinin çözümünde, kesin çözümü bulmayı garanti eden klasik teknikler ve yaklaşık optimal bir çözüm bulmayı vadeden metasezgisel algoritmalar kullanılır. Klasik yöntemler, çoğunlukla tüm problem uzayını araştırarak çözüm bulma eğilimindedir ve diğer yöntemlere kıyasla maliyeti ve süresi fazladır. Bu yüzden problem uzayındaki tüm olası çözümler yerine, makul bir sürede belli sayıda çözümü değerlendiren ve doğal fenomenlerden esinlenen metasezgisel algoritmalar bu alanda sıklıkla kullanılmaktadır[1].

Her geçen gün sayıları hızla artan metasezgisel algoritmalarından biri de Yapay Denizanası Arama(YDA) algoritmasıdır[2]. YDA algoritması, denizanelerinin okyanustaki yiyecek arama davranışının modellenmesi ile oluşturulmuştur. YDA basit yapısı, kolay uygulanması ve ayarlanması gereken az sayıda parametreye sahip olması nedeniyle tercih edilen bir algoritmadır[3]. Ayrıca, YDA'nın popülasyon oluşturulurken rastgele yöntemler yerine kaotik bir harita kullanması, yakınsamayı hızlandırır ve çözümlerin arama alanı boyunca uygun bir şekilde dağıtılmasını sağlar. YDA'nın zayıf yönü ise, her denizanası için yeni konumun yönünün popülasyondaki tüm çözümlerin ortalamasına doğru olmasıdır. Bu yaklaşım, algoritmanın global arama kabiliyetini önemli ölçüde sınırlar[4]. Bu motivasyonla bu çalışmada, YDA'nın global arama kabiliyetini geliştirmek amacıyla standart algoritmanın global arama bölümüne, Diferansiyel Evrim Algoritmasındaki[5] başarılı mutasyon stratejilerinden biri olan "current-to-best" yaklaşımı entegre edilmiştir. Popülasyondaki en iyi bireyi ve rastgele seçilen iki bireyi kullanarak aday birey oluşturan bu stratejinin kullanılmasıyla, global arama sırasında popülasyon çeşitliliğinin daha uzun süre korunması desteklenmiştir. Böylece lokal minimuma takılma riski azalmıştır. Elde edilen gelişmiş algoritma(MYDA), klasik kıyaslama fonksiyonları üzerinde farklı boyutlar için test edilmiştir. Literatürden seçilen algoritmalarla da karşılaştırılan MYDA'nın performansı, istatistik testler yardımıyla yorumlanmıştır. Sonuçlar, MYDA algoritmasında yapılan düzenleme sayesinde, standart algoritmanın global arama yeteneğinin güçlenmesine ciddi katkı sağlandığını ve farklı optimizasyon problemleri için ümit vadedilen bir algoritma elde edildiğini göstermiştir.

Çalışmanın devamı şu şekilde organize edilmiştir: İkinci bölümde, YDA algoritmasının literatür özetine yer verilmiştir. Üçüncü bölümdeki materyal ve metot bölümünde, YDA ve önerilen MYDA algoritması anlatılmıştır. Dördüncü bölümde yapılan testlere ait bulgular sunulup, tartışılmıştır. Son olarak, beşinci bölümde sonuç ve önerilere yer verilmiştir.

2. Literatür Özeti

Son dönemde önerilmiş olan YDA algoritmasıyla yapılan çalışmalar iyi bilinen algoritmalara kıyasla daha sınırlıdır. Ancak, YDA algoritması kullanılarak farklı alanlarda yapılan çalışmaların sayısı hızla artmaktadır. Gouda ve arkadaşları, polimer değişim membranlı yakıt hücresi modelinin belirsiz parametrelerini bulma problemini çözmek için YDA algoritmasını kullanmıştır[6]. Shaheen ve arkadaşları, otomatik dağıtım sistemlerinde Volt/VAr koordinasyonu için daha verimli bir teknik sağlamak için YDA algoritmasını kullanmıştır[7]. Ginidi ve arkadaşları, ısı ve gücün ekonomik dağıtımına çözüm bulmak için yığın tabanlı denizanası arama algoritması adlı yeni bir hibrit algoritma önermiştir[8]. Yıldızdan ve Baykan, YDA algoritmasının yerel arama becerisini

geliştirmek için algoritmanın aktif hareket bölümünü düzenleyerek gelişmiş bir algoritma önerip, algoritmanın CEC 2017 test fonksiyonundaki performansını incelemiştir[9]. Kaveh ve arkadaşları, YDA'da üç modifikasyon yaparak yapısal optimizasyon problemleri için kuantum tabanlı bir YDA algoritması önermiştir[10]. Jiang ve arkadaşları, ekonomik yük dağıtım problemine YDA tabanlı bir çözüm önermiştir[3]. Chou ve Truong, mühendislik problemlerinin çözümü için Lévy uçuşu, elit popülasyon, kaotik harita ve karşıt tabanlı öğrenmeye dayalı çok amaçlı YDA algoritmasını önermiştir[11]. Bujok, YDA algoritmasında üç farklı mekanizma ile YDA varyantları oluşturmuştur. Önerilen bu varyantların performansını gerçek dünya problemleri üzerinde test etmiş ve varyantların performansını farklı yöntemlerle karşılaştırarak incelemiştir[12]. Abdel-Basset ve arkadaşları, YDA algoritması temelli yeni çok seviyeli eşikleme görüntü bölütleme tekniği önermiştir. Sıralamaya dayalı güncelleme ve uyarlamalı bu yöntemin YDA algoritmasına entegre edilmesiyle elde edilen teknik, MR görüntülerinin eşiklenmesi için kullanılmıştır[13]. Huang ve Lin, fotovoltaiik üretim sistemi için YDA tabanlı bir maksimum güç noktası izleme stratejisi önermiştir[14]. Youssef ve arkadaşları, tek fazlı bir güç transformatörünün eşdeğer devresinin parametrelerini tahmin etmek için YDA algoritmasını kullanmıştır[15]. Rajpurohit, YDA algoritmasının yakınsama hızını iyileştirmek için muhalefete dayalı öğrenme ve olasılık tabanlı değiştirme stratejilerini algoritmaya entegre etmiştir. Elde edilen algoritma, kıyaslama fonksiyonları ve gerçek dünya problemi üzerinde test edilmiştir[16]. Rajpurohit ve Sharma, sürü hareketi adımında kaotik haritalar kullanarak, YDA algoritmasının performansını iyileştiren gelişmiş bir algoritma önermiştir [17].

3. Materyal ve Metot

3.1. Yapay Denizanası Arama Algoritması (YDA)

Denizanaları kötü yüzücü olmalarına rağmen okyanusta hareket edebilmelerini sağlayan birtakım özelliklere sahiptir. Alt bölümleri bir şemsiye gibi açılıp kapanarak hareket etmelerini sağlar. Denizanaları hareket edebilmelerine rağmen genel olarak akıntı ve gelgitlere bağlı olarak sürüklenirler. Denizanaları sürü oluşturabilirler ve bu sürüye denizanası çiçeği denir. Denizanaları okyanus akıntısına bağlı olarak yaptıkları hareket ve sürü içerisinde yaptıkları hareket olmak üzere iki farklı hareket türüne sahiptir[2]. İşte bu özellikleri, YDA algoritmasının oluşturulmasında ilham kaynağı olmuştur. Algoritma adımları aşağıdaki gibi özetlenebilir:

- Algoritma popülasyonun oluşturulması ile başlatılır. Metasezgisel algoritmaların kullandığı tipik rastgele yöntemlerin yerine, YDA algoritmasında yerel optimuma takılmayı önlemek ve çözümleri problemin arama uzayı içine doğru bir şekilde dağıtmak için kaotik harita kullanılmıştır. Yapılan test işlemleri sonucu, Denklem 1'de verilen lojistik harita [18] formülü kullanılarak başlangıç popülasyonunun oluşturulmasının erken yakınsama olasılığını düşürdüğü tespit edilmiştir [2].

$$X_{i+1} = \eta X_i (1 - X_i), \quad 0 \leq X_0 \leq 1 \quad (1)$$

Denklem 1'de X_0 denizanalarının başlangıç popülasyonunu oluşturmak için kullanılır ve (0,1) aralığında rastgele bir sayıdır. X_i , i.denizanasının lojistik kaotik değeridir, $X_0 \notin \{0.0, 0.25, 0.5, 0.75\}$ dir ve η değeri 4'tür.

- Popülasyonun oluşturulduktan sonra, en iyi uygunluk değerine sahip olan denizanası (X^*) belirlenir. Sonrasında, her denizanasının konumu okyanus akıntısı ya da sürü içindeki hareketlere doğru güncellenir. Bu geçiş 'zaman kontrol mekanizmasına' göre gerçekleşir. Okyanus akıntısı Denklem 2'deki gibi formülize edilir.

$$X_i(t+1) = X_i(t) + r * (X^* - \beta * r1 * \mu) \quad (2)$$

Denklemden r ve $r1$ çarpanı $(0, 1)$ aralığında rastgele bir sayıyı, $\beta > 0$ olmak üzere dağılım katsayısını, μ popülasyondaki denizanalarının ortalamasını ifade eder. Yapılan test işlemleri sonucu, β değeri 3 olarak belirlenmiştir [2].

- Denizanalarının sürü içerisinde pasif ve aktif olmak üzere iki farklı hareketi vardır. Pasif harekette, denizanaları o anki konumları etrafında Denklem 3'e göre arama yaparlar.

$$X_i(t+1) = X_i(t) + r2 * \gamma * (U_b - L_b) \quad (3)$$

Denklem 3'de $r2$ çarpanı $(0,1)$ aralığında rastgele bir sayıyı, $\gamma > 0$ olmak üzere denizanalarının o anki konumları etrafındaki hareket miktarına ait hareket katsayısını, L_b arama uzayının alt sınır değerini, U_b ise arama uzayının üst sınır değerlerini ifade eder. Yapılan test işlemleri sonucu, γ değeri 0.1 olarak belirlenmiştir [2].

Aktif hareket ise Denklem 4'teki gibi formülize edilir. Denklemden $r3$ çarpanı $(0,1)$ aralığında rastgele bir sayıyı, Y ise denizanasının hareket yönünü gösterir. Y değeri, Denklem 5'e göre hesaplanır.

$$X_i(t+1) = X_i(t) + r3 * Y \quad (4)$$

$$Y = \begin{cases} X_i(t) - X_j(t), & \text{Eğer } f(X_i) < f(X_j) \\ X_j(t) - X_i(t), & \text{Diğer} \end{cases} \quad (5)$$

Denklem 5'te, f uygunluk fonksiyonunu, j rastgele seçilen bir denizanasının indeksini ifade eder.

- Algoritmada okyanus akıntısı ve sürü içerisindeki hareketler arasındaki geçiş 'zaman kontrol mekanizması' ile sağlanır. Zaman kontrol fonksiyonu, Denklem 6'daki gibi formülize edilir [2].

$$c(t) = \left| \left(1 - \frac{t}{t_{\max}} \right) * (2 * r4 - 1) \right| \quad (6)$$

Denklem 6'da, t o anki iterasyon sayısını, t_{\max} maksimum iterasyon sayısını, $r4$ ise $(0,1)$ aralığında rastgele bir sayıyı ifade eder. YDA'nın sözde kodu Şekil 1'de verilmiştir.

```

1. Hedef fonksiyonunu  $f(x)$  tanımla
2. Popülasyon boyutu ( $N$ ) ve maksimum iterasyon sayısını ( $Max\_It$ ) belirle
3. Başlangıç popülasyonunu oluştur.  $X_i$  ( $i = 1, 2, \dots, N$ )
4. Her  $X_i$  bireyi için uygunluk değerini  $f(X_i)$  hesapla
5. Popülasyondaki en iyi denizanasını ( $X^*$ ) bul
6. For  $S=1$  to  $Max\_It$ 
7.   For  $i=1$  to  $N$ 
8.     Denklem 6'ya göre  $c(t)$  hesapla
9.     If  $c(t) \geq c_0$  then           // okyanus akıntısına göre hareket
10.      Denklem 2'ye göre aday denizanasının konumunu ( $X_{aday}$ ) hesapla
11.     Else
12.       If  $rand(0,1) > (1 - c(t))$  then           // aktif hareket
13.        Denklem 5'e göre denizanasının yönünü belirle
14.        Denklem 4'e göre aday denizanasının konumunu ( $X_{aday}$ ) hesapla
15.       Else                                     // pasif hareket
16.        Denklem 3'e göre aday denizanasının konumu ( $X_{aday}$ ) hesapla
17.       End If
18.     End If
19.     If  $f(X_{aday}) < f(X_i)$  then
20.        $X_i = X_{aday}$ 
21.       If  $f(X_{aday}) < f(X^*)$  then
22.          $X^* = X_{aday}$ 
23.       End If
24.     End If
25.   End For
26. End For
27. Sonuçları görüntüle

```

Şekil 1. YDA sözde kodu

3.2. Önerilen Algoritma (MYDA)

Diferansiyel Evrim Algoritması, 1995'te önerilmiş popülasyon tabanlı bir algoritmadır[5]. Bu algorithmada, aday birey oluşturulurken mutasyon, çaprazlama ve seçim gibi işlemler uygulanır. Mutasyon, algoritma performansını geliştiren ve onu daha gürbüz hale getiren önemli bir işlemdir[19]. Algorithmada kullanılmak üzere önerilmiş çok sayıda mutasyon stratejisi mevcuttur[20, 21]. Bunlarda biri de current-to-best stratejisidir. Denklem 7'deki gibi formülize edilen bu strateji, popülasyondaki en iyi bireyi ve rastgele seçilen iki bireyi kullanarak aday birey oluşturur. Bu bakımdan, global aramaya yönelik çeşitliliği destekleyen bir strateji olduğu söylenebilir. Denklemde $F1$ ve $F2$, $[0,2]$ aralığında değer alan ölçekleme faktörünü, $r1, r2 \in \{0,1,\dots,N\}$ olmak üzere X_{r1} ve X_{r2} iki rastgele seçilmiş denizanasını, X_i o anki denizanasını, X^* popülasyondaki en iyi denizanasını ve V_i aday denizanasını ifade eder.

$$V_i = X_i + F1 * (X^* - X_i) + F2 * (X_{r1} - X_{r2}) \quad (7)$$

YDA algoritmasında global arama Denklem 2'ye göre gerçekleştirilir. Denklem incelendiğinde, aday denizanasının en iyi denizanası ve popülasyondaki denizanelerinin ortalaması arasındaki farktan faydalanılarak elde edildiği görülür. Denklem 2 ve Denklem 7 kıyaslandığında, Denklem 7'deki current-to-best yaklaşımının, ölçekleme faktörü ve rastgele bireyler sayesinde popülasyon çeşitliliğini daha uzun süre destekleyeceği öngörülmüştür. Buna göre, önerilen algoritmada Denklem 2'nin yerine global arama formülü olarak Denklem 7'de verilen current-to-best yaklaşımı kullanılmıştır. Elde edilen MYDA algoritmasının performansı, dördüncü bölümdeki test işlemleriyle incelenmiştir.

4. Bulgular ve Tartışma

4.1. Bulgular

Bu bölümde, önerilen MYDA algoritmasının klasik kıyaslama fonksiyonları üzerinde farklı boyutlardaki performansı incelenmiştir. Elde edilen sonuçlar, standart algoritma ve literatürden seçilen algoritmalarla karşılaştırılmıştır. Tablo 1'de test işleminde kullanılan on iki klasik kıyaslama fonksiyonu ve özellikleri verilmiştir. Bu fonksiyonların yedi tanesi tek modlu(T), kalan beş tanesi çok modlu(Ç) özelliğindedir. Test işleminde, algoritmalar her fonksiyon için bağımsız şekilde otuz defa yürütülmüştür. Elde edilen sonuçların en iyi, ortanca, en kötü, ortalama ve standart sapma değerleri hesaplanmıştır.

Tablo 1. Klasik kıyaslama fonksiyonları

Fonksiyon	Aralık	f_{min}	Özellik
$f1(x) = \sum_{i=1}^d x_i^2$	[-100,100]	0	T
$f2(x) = \sum_{i=1}^d x_i^2 + \prod_{i=1}^d x_i $	[-10,10]	0	T
$f3(x) = \sum_{i=1}^d (\sum_{j=1}^i x_j)^2$	[-100,100]	0	T
$f4(x) = \max_i \{ x_i , 1 \leq i \leq d\}$	[-100,100]	0	T
$f5(x) = \sum_{i=1}^{d-1} [100 (x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	[-30,30]	0	T
$f6(x) = \sum_{i=1}^d ([x_i + 0.5])^2$	[-100,100]	0	T
$f7(x) = \sum_{i=1}^d i * x_i^4 + rand[0,1)$	[-1.28,1.28]	0	T
$f8 = \sum_{i=1}^d -x_i \sin(\sqrt{ x_i })$	[-500,500]	-418.9829 ×d	Ç
$f9 = \sum_{i=1}^d [x_i^2 - 10 \cos(2\pi x_i) + 10]$	[-5.12,5.12]	0	Ç
$f10(x) = \sum_{i=1}^d -20 \exp\left(-0.2 \sqrt{\frac{1}{d} \sum_{i=1}^d x_i^2}\right) - \exp\left(\frac{1}{d} \sum_{i=1}^d \cos(2\pi x_i)\right) + 20 + e$	[-32,32]	0	Ç
$f11(x) = \frac{1}{4} \times 10^{-3} \sum_{i=1}^d x_i^2 - \prod_{i=1}^d \cos\left(\frac{x_i}{\sqrt{i}} + 1\right)$	[-600,600]	0	Ç

$$f_{12}(x) = \frac{\pi}{d} \left\{ 10 \sin(\pi y_1) + \sum_{i=1}^{d-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_n - 1)^2 \right\}$$

$$+ \sum_{i=1}^d u(x_i, 10, 100, 4) \quad [-50, 50] \quad 0 \quad \zeta$$

$$y_i = \frac{x_i + 5}{4} \quad u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & \text{Eğer } x_i > a \\ k(-x_i - a)^m & \text{Eğer } x_i < -a \\ 0 & \text{diğer} \end{cases}$$

Bu test işlemlerinde, YDA ve MYDA algoritmalarında kullanılan parametre değerleri Tablo 2’de gösterilmiştir. Devamında, YDA ve MYDA algoritmalarının 10,30,50,100,500 ve 1000 boyut için yürütülmesi ile elde edilen karşılaştırmalı sonuçlar sırasıyla Tablo 3, Tablo 4, Tablo 5, Tablo 6, Tablo 7 ve Tablo 8’de verilmiştir. Tablolarda her fonksiyon için elde edilen en iyi ortalama değer kalın font ile gösterilmiştir. Tablolarda verilen sonuçlar incelendiğinde, önerilen MYDA algoritmasının tüm boyutlarda, tüm fonksiyonlar için YDA algoritmasından daha iyi bir ortalama değer ürettiği tespit edilmiştir. Boyut artışı, genel olarak metasezgisel algoritmalarda performans düşüşlerine neden olur. Bu çalışmadaki test sonuçları incelendiğinde, standart YDA algoritmasında F8 fonksiyonu hariç diğer fonksiyonlarda boyut artışına bağlı performans düşüşü yaşandığı görülmüştür. MYDA algoritmasında ise F8-F12 arası çok modlu fonksiyonlarda performans düşüşü olmazken, kalan tek modlu fonksiyonlarda bir miktar performans düşüşü yaşanmıştır. Bu sonuç, önerilen algoritmanın çok modlu fonksiyonlarda daha gürbüz olduğunu ortaya koymaktadır.

Tablo 2. Parametre değerleri

Parametre	Değer	Parametre	Değer
Popülasyon boyutu (N)	50	F1	0.5
Maksimum iterasyon sayısı	500	F2	0.5
Yürütme(run) sayısı	30	Boyut(d)	10,30,50,100, 500,1000
C_0	0.5		

Tablo 3. YDA ve MYDA algoritmalarının 10 boyut için karşılaştırma sonuçları

		d=10				
		En iyi	Ortanca	En kötü	Ortalama	S. Sapma
F1	YDA	5,29E-20	1,21E-17	3,19E-15	2,01E-16	6,08E-16
	MYDA	5,90E-128	5,64E-121	2,86E-106	9,55E-108	5,23E-107
F2	YDA	1,56E-14	1,72E-13	9,51E-13	2,95E-13	2,74E-13
	MYDA	4,08E-67	1,82E-62	8,53E-57	3,06E-58	1,56E-57
F3	YDA	2,14E-08	8,02E-06	6,34E-03	2,58E-04	1,16E-03
	MYDA	1,39E-97	2,89E-64	1,36E-43	4,52E-45	2,48E-44
F4	YDA	3,98E-04	4,36E-03	7,52E-02	8,78E-03	1,42E-02
	MYDA	9,07E-59	6,16E-56	3,67E-52	1,30E-53	6,68E-53
F5	YDA	6,77E-01	4,70E+00	5,93E+00	4,41E+00	1,27E+00

	MYDA	0,00E+00	0,00E+00	5,64E+00	1,85E+00	2,48E+00
F6	YDA	2,10E-19	8,77E-17	4,56E-14	1,80E-15	8,30E-15
	MYDA	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00
F7	YDA	1,95E-04	4,74E-04	1,83E-03	6,71E-04	4,77E-04
	MYDA	1,58E-05	2,44E-04	5,78E-04	2,51E-04	1,43E-04
F8	YDA	-4,19E+03	-3,92E+03	-3,48E+03	-3,89E+03	2,20E+02
	MYDA	-4,19E+03	-4,19E+03	-3,97E+03	-4,18E+03	3,96E+01
F9	YDA	0,00E+00	6,91E-12	2,98E+00	3,62E-01	7,12E-01
	MYDA	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00
F10	YDA	1,25E-10	1,19E-09	8,75E-09	2,17E-09	2,14E-09
	MYDA	8,88E-16	4,44E-15	7,99E-15	4,20E-15	1,30E-15
F11	YDA	0,00E+00	9,99E-16	2,23E-02	3,29E-03	6,23E-03
	MYDA	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00
F12	YDA	8,15E-20	2,41E-18	1,90E-15	7,40E-17	3,46E-16
	MYDA	8,05E-22	2,94E-19	3,27E-18	7,80E-19	1,08E-18

Tablo 4. YDA ve MYDA algoritmalarının 30 boyut için karşılaştırma sonuçları

		d=30				
		En iyi	Ortanca	En kötü	Ortalama	S. Sapma
F1	YDA	7,75E-07	3,80E-05	5,10E-03	2,93E-04	9,34E-04
	MYDA	1,58E-121	4,52E-114	4,09E-98	1,36E-99	7,47E-99
F2	YDA	2,35E-05	3,52E-04	1,02E-02	1,32E-03	2,54E-03
	MYDA	7,94E-61	1,81E-57	1,17E-46	3,93E-48	2,14E-47
F3	YDA	4,09E+00	8,32E+01	4,35E+02	1,05E+02	1,06E+02
	MYDA	7,07E-70	1,41E-35	2,01E-16	6,72E-18	3,67E-17
F4	YDA	3,47E-02	1,13E-01	7,18E-01	1,60E-01	1,46E-01
	MYDA	3,50E-57	1,56E-54	6,05E-51	5,02E-52	1,31E-51
F5	YDA	5,56E-01	4,22E+00	2,74E+01	8,66E+00	8,87E+00
	MYDA	0,00E+00	0,00E+00	2,68E+01	4,43E+00	1,00E+01
F6	YDA	6,54E-06	5,14E-05	2,32E-03	1,94E-04	4,69E-04
	MYDA	0,00E+00	0,00E+00	4,93E-32	2,47E-33	9,92E-33
F7	YDA	3,96E-04	1,29E-03	3,35E-03	1,41E-03	6,66E-04
	MYDA	5,29E-05	3,13E-04	4,89E-03	6,18E-04	9,34E-04
F8	YDA	-1,08E+04	-8,27E+03	-5,64E+03	-8,15E+03	1,43E+03
	MYDA	-1,26E+04	-1,02E+04	-7,06E+03	-1,00E+04	2,09E+03
F9	YDA	5,59E-05	7,58E-03	3,00E+00	2,18E-01	6,65E-01
	MYDA	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00
F10	YDA	1,73E-04	1,84E-03	1,33E-02	2,61E-03	2,96E-03
	MYDA	8,88E-16	4,44E-15	4,44E-15	3,97E-15	1,23E-15
F11	YDA	2,47E-06	4,16E-05	1,53E-02	1,50E-03	3,74E-03
	MYDA	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00
F12	YDA	3,64E-08	5,87E-07	1,75E-05	1,81E-06	3,43E-06
	MYDA	1,57E-32	1,57E-32	1,64E-32	1,57E-32	1,40E-34

Tablo 5. YDA ve MYDA algoritmalarının 50 boyut için karşılaştırma sonuçları

		d=50				
		En iyi	Ortanca	En kötü	Ortalama	S. Sapma
F1	YDA	1,38E-04	1,74E-03	1,77E-01	9,91E-03	3,18E-02
	MYDA	3,91E-120	6,02E-114	5,25E-95	1,75E-96	9,59E-96
F2	YDA	2,18E-03	2,27E-02	1,45E-01	3,94E-02	4,18E-02
	MYDA	1,13E-61	2,23E-56	1,78E-46	5,94E-48	3,26E-47
F3	YDA	6,77E+01	6,39E+02	2,15E+03	8,51E+02	6,39E+02
	MYDA	4,99E-51	4,99E-51	4,99E-51	4,99E-51	4,99E-51
F4	YDA	3,20E-02	1,34E-01	7,13E-01	1,94E-01	1,80E-01
	MYDA	9,97E-58	5,09E-55	3,18E-51	1,37E-52	5,78E-52
F5	YDA	2,73E+01	1,24E+02	6,50E+02	1,58E+02	1,32E+02
	MYDA	0,00E+00	0,00E+00	4,73E+01	1,89E+00	8,68E+00
F6	YDA	2,23E-04	5,84E-03	5,32E-02	9,27E-03	1,08E-02
	MYDA	0,00E+00	0,00E+00	6,16E-32	3,70E-33	1,22E-32
F7	YDA	4,25E-04	1,50E-03	4,56E-03	1,67E-03	9,14E-04
	MYDA	4,04E-05	2,93E-04	3,14E-03	5,43E-04	6,50E-04
F8	YDA	-1,62E+04	-1,30E+04	-6,83E+03	-1,21E+04	2,72E+03
	MYDA	-2,09E+04	-2,09E+04	-1,15E+04	-1,76E+04	3,86E+03
F9	YDA	6,36E-03	1,10E-01	8,67E+00	7,54E-01	1,96E+00
	MYDA	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00
F10	YDA	1,94E-03	9,17E-03	4,65E-02	1,27E-02	1,05E-02
	MYDA	8,88E-16	4,44E-15	4,44E-15	3,97E-15	1,23E-15
F11	YDA	5,39E-05	4,37E-03	2,08E-01	1,77E-02	4,36E-02
	MYDA	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00
F12	YDA	4,01E-06	2,05E-05	6,78E-05	2,68E-05	1,75E-05
	MYDA	9,42E-33	9,42E-33	1,00E-32	9,53E-33	1,95E-34

Tablo 6. YDA ve MYDA algoritmalarının 100 boyut için karşılaştırma sonuçları

		d=100				
		En iyi	Ortanca	En kötü	Ortalama	S. Sapma
F1	YDA	1,76E-03	7,08E-02	2,15E+00	3,15E-01	5,70E-01
	MYDA	1,36E-119	2,98E-112	4,93E-96	2,13E-97	9,29E-97
F2	YDA	3,22E-02	1,29E-01	3,39E+00	2,90E-01	6,02E-01
	MYDA	9,32E-60	5,35E-55	1,02E-39	3,40E-41	1,86E-40
F3	YDA	8,72E+02	6,47E+03	1,17E+04	6,34E+03	3,09E+03
	MYDA	2,43E-57	2,16E-25	5,92E-02	1,97E-03	1,08E-02
F4	YDA	7,77E-02	2,84E-01	1,34E+00	3,64E-01	2,89E-01
	MYDA	3,13E-58	5,77E-55	4,87E-50	1,63E-51	8,88E-51
F5	YDA	9,66E-01	1,73E+01	9,77E+01	2,11E+01	2,13E+01
	MYDA	0,00E+00	0,00E+00	1,23E+01	8,22E-01	2,54E+00
F6	YDA	4,36E-03	9,79E-02	1,59E+00	2,26E-01	3,66E-01
	MYDA	0,00E+00	0,00E+00	3,70E-32	1,23E-33	6,75E-33
F7	YDA	7,33E-04	1,74E-03	1,63E-02	2,68E-03	3,46E-03
	MYDA	3,92E-05	3,62E-04	3,76E-03	8,03E-04	1,01E-03
F8	YDA	-2,83E+04	-1,93E+04	-1,03E+04	-1,91E+04	4,81E+03
	MYDA	-4,19E+04	-3,91E+04	-1,78E+04	-3,22E+04	1,04E+04

F9	YDA	1,12E-02	2,37E+00	4,10E+01	7,24E+00	1,15E+01
	MYDA	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00
F10	YDA	7,90E-03	3,52E-02	5,83E-01	7,59E-02	1,17E-01
	MYDA	8,88E-16	4,44E-15	4,44E-15	4,20E-15	9,01E-16
F11	YDA	1,76E-03	2,84E-02	1,26E+00	1,33E-01	2,81E-01
	MYDA	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00
F12	YDA	3,75E-05	1,81E-04	8,29E-04	2,16E-04	1,64E-04
	MYDA	4,71E-33	4,71E-33	5,68E-33	4,76E-33	1,85E-34

Tablo 7. YDA ve MYDA algoritmalarının 500 boyut için karşılaştırma sonuçları

		d=500				
		En iyi	Ortanca	En kötü	Ortalama	S. Sapma
F1	YDA	1,19E-01	9,24E-01	2,45E+02	1,33E+01	4,78E+01
	MYDA	7,25E-119	3,72E-110	2,40E-92	8,01E-94	4,39E-93
F2	YDA	5,12E-01	2,17E+00	2,40E+01	3,63E+00	4,48E+00
	MYDA	1,86E-59	6,91E-55	5,47E-43	1,82E-44	9,99E-44
F3	YDA	3,44E+04	2,24E+05	3,16E+05	2,13E+05	7,84E+04
	MYDA	9,06E-45	3,31E-09	1,07E+05	1,02E+04	2,93E+04
F4	YDA	1,71E-01	3,94E-01	1,50E+00	5,34E-01	3,77E-01
	MYDA	2,88E-57	2,19E-53	1,30E-50	1,29E-51	3,09E-51
F5	YDA	5,08E+00	2,20E+02	6,83E+02	2,77E+02	2,31E+02
	MYDA	0,00E+00	0,00E+00	3,56E+01	3,58E+00	7,96E+00
F6	YDA	1,65E-01	2,41E+00	6,66E+01	5,21E+00	1,20E+01
	MYDA	0,00E+00	0,00E+00	4,07E-31	3,57E-32	1,01E-31
F7	YDA	8,38E-04	2,58E-03	7,08E-03	2,81E-03	1,60E-03
	MYDA	3,04E-05	4,62E-04	2,78E-02	1,68E-03	5,07E-03
F8	YDA	-1,06E+05	-6,41E+04	-4,43E+04	-6,83E+04	1,29E+04
	MYDA	-2,09E+05	-2,09E+05	-6,04E+04	-1,69E+05	6,11E+04
F9	YDA	1,87E+00	3,25E+01	1,18E+03	1,20E+02	2,35E+02
	MYDA	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00
F10	YDA	2,18E-02	9,64E-02	2,27E+00	2,10E-01	4,06E-01
	MYDA	4,44E-15	4,44E-15	7,99E-15	4,56E-15	6,49E-16
F11	YDA	3,03E-02	2,30E-01	4,91E+00	6,50E-01	9,94E-01
	MYDA	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00
F12	YDA	2,02E-04	9,35E-04	9,62E-03	1,26E-03	1,70E-03
	MYDA	9,42E-34	9,42E-34	2,10E-33	1,05E-33	3,03E-34

Tablo 8. YDA ve MYDA algoritmalarının 1000 boyut için karşılaştırma sonuçları

		d=1000				
		En iyi	Ortanca	En kötü	Ortalama	S. Sapma
F1	YDA	1,67E-01	1,21E+00	2,69E+02	2,08E+01	6,59E+01
	MYDA	1,85E-116	2,38E-111	3,82E-100	1,28E-101	6,98E-101
F2	YDA	1,66E+01	2,45E+02	9,19E+02	2,93E+02	2,13E+02
	MYDA	2,29E-56	9,14E-51	6,84E-44	3,53E-45	1,40E-44
F3	YDA	3,46E+05	8,93E+05	1,63E+06	8,85E+05	2,74E+05
	MYDA	1,06E-29	6,76E+01	1,28E+06	1,20E+05	2,99E+05
F4	YDA	1,84E-01	6,22E-01	4,14E+00	7,75E-01	7,92E-01
	MYDA	3,55E-58	1,37E-53	1,33E-47	4,59E-49	2,43E-48
F5	YDA	2,10E+01	4,40E+02	2,72E+03	6,93E+02	7,08E+02
	MYDA	0,00E+00	7,41E-29	7,17E+01	5,99E+00	1,53E+01
F6	YDA	5,46E-01	7,00E+00	1,47E+02	2,02E+01	3,51E+01
	MYDA	0,00E+00	0,00E+00	2,34E-31	7,81E-33	4,28E-32
F7	YDA	8,57E-04	2,98E-03	1,16E-02	3,45E-03	2,17E-03
	MYDA	1,40E-04	6,47E-04	7,92E-03	1,40E-03	1,87E-03
F8	YDA	-1,76E+05	-1,16E+05	-9,11E+04	-1,21E+05	2,11E+04
	MYDA	-4,19E+05	-4,19E+05	-1,32E+05	-3,73E+05	8,89E+04
F9	YDA	2,30E+00	5,24E+01	2,65E+03	2,02E+02	4,83E+02
	MYDA	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00
F10	YDA	1,68E-02	8,25E-02	8,77E-01	1,20E-01	1,56E-01
	MYDA	4,44E-15	4,44E-15	4,44E-15	4,44E-15	0,00E+00
F11	YDA	2,51E-03	2,61E-01	2,50E+00	4,88E-01	5,71E-01
	MYDA	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00
F12	YDA	1,00E-04	8,16E-04	1,19E+00	4,30E-02	2,16E-01
	MYDA	4,71E-34	4,71E-34	2,19E-33	5,87E-34	3,49E-34

YDA ve MYDA algoritmaları literatürdeki LXWOA [22], DDSCA [23], HGWO [24] ve m-SCA [25] algoritmalarıyla da karşılaştırılmıştır. Bu karşılaştırma, otuz boyut için elde edilen sonuçlar kullanılarak yapılmıştır. Ortalama değere göre yapılan bu karşılaştırmaya ait sonuçlar Tablo 9’da verilmiştir. Tablodaki sonuçlar incelendiğinde, önerilen MYDA algoritmasının sekiz fonksiyonda daha iyi bir ortalama değer bulduğu tespit edilmiştir. Standart YDA algoritması ise hiçbir fonksiyonda daha iyi bir ortalama değer bulamamıştır. Ayrıca bu karşılaştırma sonuçları istatistik testler yardımıyla da yorumlanmıştır. Bunun için, Wilcoxon işaretli sıra testi ve Friedman testi kullanılmıştır[26]. Wilcoxon işaretli sıra testi, iki örnek arasındaki farkı belirler[27] ve bu çalışmada bir algoritmanın diğerinden daha iyi performans gösterip göstermediğini kontrol etme amaçlı kullanılmıştır. Friedman testi ise çoklu karşılaştırmalar yapmak için uygundur ve algoritmaların farklılıklarını istatistiksel olarak tespit etmek için algoritmaları en iyiden en kötüye doğru sıralar[28]. Bu testlerde anlamlılık değeri (p), $\alpha < 0.05$ olacak şekilde kullanılmıştır. Tablo 10’da MYDA’nın YDA ve literatürdeki diğer algoritmalarla olan Wilcoxon işaretli sıra testi sonuçları verilmiştir. Tablodaki İyi, Kötü ve Eşit sütunları sırasıyla MYDA’nın diğer algoritmayla olan ikili karşılaştırması sonucunda kaç fonksiyonda daha iyi, daha kötü ve eşit sonuç ürettiğinin bilgisini vermektedir. P sütunu anlamlılık değerini gösterir. Sembol sütununda ise +, \approx ya da - sembolleri yer alır ve sırasıyla MYDA’nın diğer algoritmadan anlamsal olarak daha iyi, eşit ve daha kötü olduğunu ifade eder. Tablo 10’daki sonuçlar incelendiğinde, MYDA algoritmasının YDA, HGWO ve m-SCA algoritmalarından anlamsal olarak daha iyi olduğu, LXWOA ve DDSCA algoritmalarıyla ise aralarında anlamlı bir fark bulunmadığı sonucuna varılmıştır.

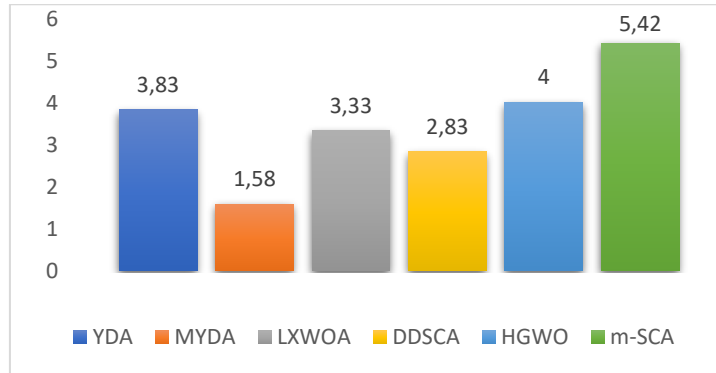
Tablo 9. Literatürdeki algoritmalarla karşılaştırma sonuçları (d=30)

	YDA	MYDA	LXWOA	DDSCA	HGWO	m-SCA
F1	2,93E-04	1,36E-99	6,54E-77	1,57E-125	1,21E-32	5,70E-03
F2	1,32E-03	3,93E-48	6,09E-53	1,25E-64	9,33E-20	9,11E-04
F3	1,05E+02	6,72E-18	1,71E+04	2,43E+03	3,18E-08	8,48E+02
F4	1,60E-01	5,02E-52	1,90E+01	2,70E-03	4,16E-08	7,07E-01
F5	8,66E+00	4,43E+00	2,75E+01	3,63E-01	2,64E+01	2,96E+01
F6	1,94E-04	2,47E-33	3,01E-01	1,82E-07	3,78E-01	1,24E+00
F7	1,41E-03	6,18E-04	2,29E-03	4,30E-02	1,49E-03	1,95E-02
F8	-8,15E+03	-1,00E+04	-1,05E+04	-9,85E+03	-6,40E+03	-4,27E+03
F9	2,18E-01	0,00E+00	0,00E+00	0,00E+00	2,27E-01	7,81E+01
F10	2,61E-03	3,97E-15	4,44E-15	3,29E-14	4,26E-14	3,36E-03
F11	1,50E-03	0,00E+00	0,00E+00	0,00E+00	1,37E-03	3,84E-02
F12	1,81E-06	1,57E-32	1,70E-02	7,04E-09	3,43E-01	1,45E-01

Tablo 10. Wilcoxon işaretli sıra testi sonuçları

Algoritmalar	İyi	Kötü	Eşit	p	Sembol
MYDA - YDA	12	0	0	0,002	+
MYDA - LXWOA	8	2	2	0,093	≈
MYDA - DDSCA	7	3	2	0,074	≈
MYDA - HGWO	12	0	0	0,002	+
MYDA - m-SCA	12	0	0	0,002	+

Şekil 2’de ise Friedman testinden elde edilen ortalama sıra değerlerine göre algoritmaların karşılaştırmalı bir grafiği verilmiştir. Grafiğe göre, MYDA algoritması 1,58 ortalama sıra değeri ile algoritmalar arasında ilk sırada yer almıştır. YDA algoritması ise 3,83 ortalama sıra değeri ile dördüncü sırada yer almıştır. Sonuçlar, YDA algoritmasında yapılan düzenlemenin algoritmanın sıralamasında önemli bir iyileşmeye neden olduğunu göstermektedir.

**Şekil 2.** Friedman testi sonuçları

4.2. Tartışma

Bulgular bölümünde, YDA ve MYDA algoritmalarının öncelikle küçük, orta ve büyük ölçekli optimizasyon problemlerindeki performansı incelenmiştir. Bunun için, on iki standart kıyaslama fonksiyonu üzerinde altı farklı boyut için algoritmalar test edilmiştir. Sonuçlar, yapılan toplam 72 kıyaslamaların tamamında, önerilen MYDA'nın standart algoritmadan daha iyi bir ortalama değer bulunduğunu göstermiştir. Elde edilen %100'lük bu başarı oranı, YDA'nın global arama kabiliyetinin, yapılan düzenleme sayesinde iyileştirildiğini ortaya koymuştur. Ancak, tüm metasezgisel algoritmalarda gözlenen boyut artışına bağlı performans düşüşleri, YDA ve MYDA algoritmalarında da gözlenmiştir. Bu düşüşler, YDA algoritmasında daha belirgin olmuştur. MYDA'nın tek modlu fonksiyonlara kıyasla, çok modlu fonksiyonlarda bu performans düşüşlerini daha az yaşadığı ve bu fonksiyonlarda daha gürbüz olduğu tespit edilmiştir. Bu sonucun elde edilmesinde, yapılan düzenlemeyle, birden fazla yerel optimuma sahip olan çok modlu fonksiyonlarda, algoritmanın lokal minimuma takılma olasılığının azaltılmış olması etkili olmuştur. Bunun yanında, MYDA'nın tek modlu fonksiyonlarda, boyut artışına bağlı daha belirgin düşüşler yaşaması, algoritmanın lokal arama bölümünün de düzenlenmesi gerekliliğini ortaya koymuştur.

Önerilen MYDA algoritması literatürle de karşılaştırılmıştır. Yapılan istatistik test sonuçlarına göre, MYDA karşılaştırıldığı algoritmalarla benzer ya da daha iyi performans göstermiştir. Algoritmalar arası sıralamada ise MYDA algoritması ilk sırada, YDA algoritması ise dördüncü sırada yer almıştır. Bu sonuç, yapılan düzenleme ile standart algoritma performansının ve sıralamasının ciddi şekilde iyileştiğini göstermiştir.

5. Sonuç ve Öneriler

Bu çalışmada, YDA algoritmasının arama kabiliyetini geliştirmek amacıyla algoritmanın global arama bölümünde bir düzenleme yapılmıştır. Diferansiyel Evrim algoritmasının başarılı mutasyon stratejilerinden biri olan current-to-best yaklaşımı YDA algoritmasının global arama bölümüne entegre edilmiştir. Bu yolla global arama sırasında çeşitliliğin artmasına katkı sağlanmıştır. Elde edilen gelişmiş MYDA algoritması, farklı özellikte klasik kıyaslama fonksiyonları üzerinde farklı boyutlar için test edilmiştir. Elde edilen sonuçlar, YDA ve literatürdeki algoritmalarla karşılaştırılmıştır. YDA ve MYDA algoritmaları arasındaki karşılaştırmaların tamamında, önerilen algoritma standart algoritmadan daha başarılı olmuştur. Literatürle yapılan karşılaştırma sonuçları istatistiksel testler yardımıyla yorumlanmıştır. Test sonuçları, önerilen algoritmanın literatürdeki algoritmalarla benzer yada daha iyi performans gösterdiğini ve yarışmacı bir algoritma olduğunu ortaya koymuştur.

Gelecek çalışma önerisi olarak, diferansiyel evrim algoritmasına ait farklı mutasyon stratejilerinin performansa etkisi incelenebilir. MYDA algoritmasının farklı test kümelerindeki ve gerçek dünya problemleri üzerindeki performansı için yeni test işlemleri yapılabilir.

Yazarın Katkıları

GY çalışma kapsamında gerçekleştirilen araştırma, kodlama, test etme, yazma gibi işlemlerin tümünü gerçekleştirmiştir. Yazar makalenin son halini okumuş ve onaylamıştır.

Çıkar Çatışması

Yazar, çıkar çatışması olmadığını beyan eder.

Kaynaklar

- [1]. Akay, B., Nümerik optimizasyon problemlerinde yapay arı kolonisi (artificial bee colony) algoritmasının performans analizi, Doktora Tezi, Erciyes Üniversitesi, Fen Bilimleri Enstitüsü, 2009.
- [2]. Chou, J.-S., Truong, D.-N., A novel metaheuristic optimizer inspired by behavior of jellyfish in ocean, *Applied Mathematics and Computation*, 2021, 389: 125535.
- [3]. Jiang, S.-J., Dao, T.-K., Vu, V.-D., Ngo, T.-G., A Power System Economic Load Dispatch Using Jellyfish Search Algorithm, in *Soft Computing for Problem Solving*, Springer, 321-331, 2021.
- [4]. Shaheen, A.M., El-Sehiemy, R.A., Alharthi, M.M., Ghoneim, S.S., Ginidi, A.R., Multi-objective jellyfish search optimizer for efficient power system operation based on multi-dimensional OPF framework, *Energy*, 2021, 237: 121478.
- [5]. Price, K.V., Storn, R.M., Lampinen, J.A., *The differential evolution algorithm*, *Differential evolution: a practical approach to global optimization*, 2005: 37-134.
- [6]. Gouda, E.A., Kotb, M.F., El-Fergany, A.A., Jellyfish search algorithm for extracting unknown parameters of PEM fuel cell models: Steady-state performance and analysis, *Energy*, 2021, 221: 119836.
- [7]. Shaheen, A.M., Elsayed, A.M., Ginidi, A.R., Elattar, E.E., El-Sehiemy, R.A., Effective automation of distribution systems with joint integration of DGs/SVCs considering reconfiguration capability by jellyfish search algorithm, *IEEE Access*, 2021, 9: 92053-92069.
- [8]. Ginidi, A., Elsayed, A., Shaheen, A., Elattar, E., El-Sehiemy, R., An Innovative Hybrid Heap-Based and Jellyfish Search Algorithm for Combined Heat and Power Economic Dispatch in Electrical Grids, *Mathematics*, 2021, 9(17): 2053.
- [9]. Yıldızdan, G., Baykan, Ö.K., A Novel Artificial Jellyfish Search Algorithm Improved with Detailed Local Search Strategy, in *2021 6th International Conference on Computer Science and Engineering (UBMK)*: 180-185, 2021.
- [10]. Kaveh, A., Biabani Hamedani, K., Kamalinejad, M., Joudaki, A., Quantum-based jellyfish search optimizer for structural optimization, *International Journal Of Optimization In Civil Engineering*, 2021, 11(2): 329-356.
- [11]. Chou, J.-S., Truong, D.-N., Multiobjective optimization inspired by behavior of jellyfish for solving structural design problems, *Chaos, Solitons & Fractals*, 2020, 135: 109738.
- [12]. Bujok, P., Three Steps to Improve Jellyfish Search Optimiser, in *MENDEL*: 29-40, 2021.
- [13]. Abdel-Basset, M., Mohamed, R., Abouhawwash, M., Chakraborty, R.K., Ryan, M.J., Nam, Y., An Improved Jellyfish Algorithm for Multilevel Thresholding of Magnetic Resonance Brain Image Segmentations, *Computers, Materials and Continua*, 2021, 68(3): 2961-2977.
- [14]. Huang, R., Lin, Y., A Maximum Power Point Tracking Strategy for Photovoltaic System Based on Improved Artificial Jellyfish Search Optimizer, in *2021 3rd International Academic Exchange Conference on Science and Technology Innovation (IAECST)*: 1918-1922, 2021.
- [15]. Youssef, H., Hassan, M.H., Kamel, S., Elsayed, S.K., Parameter Estimation of Single Phase Transformer Using Jellyfish Search Optimizer Algorithm, in *2021 IEEE International Conference on Automation/XXIV Congress of the Chilean Association of Automatic Control (ICA-ACCA)*: 1-4, 2021.
- [16]. Rajpurohit, J., A Modified Jellyfish Search Optimizer with Opposition Based Learning and Biased Passive Swarm Motion, *Ingénierie des Systèmes d'Information*, 2021, 26(6): 577-584.
- [17]. Rajpurohit, J., Sharma, T.K., Chaotic active swarm motion in jellyfish search optimizer, *International Journal of System Assurance Engineering and Management*, 2022: 1-17.
- [18]. Gandomi, A.H., Yang, X.-S., Talatahari, S., Alavi, A.H., Firefly algorithm with chaos, *Communications in Nonlinear Science and Numerical Simulation*, 2013, 18(1): 89-98.
- [19]. Keskintürk, T., Diferansiyel gelişim algoritması, *İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi*, 2006, 5(9): 85-99.

- [20]. Mallipeddi, R., Suganthan, P.N., Pan, Q.-K., Tasgetiren, M.F., Differential evolution algorithm with ensemble of parameters and mutation strategies, *Applied soft computing*, 2011, 11(2): 1679-1696.
- [21]. Opara, K., Arabas, J., Comparison of mutation strategies in differential evolution—a probabilistic perspective, *Swarm and Evolutionary Computation*, 2018, 39: 53-69.
- [22]. Singh, A., Laplacian whale optimization algorithm, *International Journal of System Assurance Engineering and Management*, 2019, 10(4): 713-730.
- [23]. Li, Y., Zhao, Y., Liu, J., Dimension by dimension dynamic sine cosine algorithm for global optimization problems, *Applied Soft Computing*, 2021, 98: 106933.
- [24]. Zhu, A., Xu, C., Li, Z., Wu, J., Liu, Z., Hybridizing grey wolf optimization with differential evolution for global optimization and test scheduling for 3D stacked SoC, *Journal of Systems Engineering and Electronics*, 2015, 26(2): 317-328.
- [25]. Gupta, S., Deep, K., A hybrid self-adaptive sine cosine algorithm with opposition based learning, *Expert Systems with Applications*, 2019, 119: 210-230.
- [26]. García, S., Molina, D., Lozano, M., Herrera, F., A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 special session on real parameter optimization, *Journal of Heuristics*, 2009, 15(6): 617-644.
- [27]. Derrac, J., García, S., Molina, D., Herrera, F., A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, *Swarm and Evolutionary Computation*, 2011, 1(1): 3-18.
- [28]. Demšar, J., Statistical comparisons of classifiers over multiple data sets, *The Journal of Machine Learning Research*, 2006, 7: 1-30.