



Kaotik Denizatı Optimizasyon Algoritması

Feyza Altunbey Özbay^{1*}, Erdal Özbay²

^{1*} Fırat Üniversitesi, Mühendislik Fakültesi, Yazılım Mühendisliği Bölümü, Elazığ, Türkiye, (ORCID: 0000-0003-0629-6888), faltunbey@firat.edu.tr

² Fırat Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Elazığ, Türkiye (ORCID: 0000-0002-9004-4802), erdalozbay@firat.edu.tr

(6th International Symposium on Innovative Approaches in Smart Technologies (ISAS) 2022 – 8-10 December 2022)

(DOI: 10.31590/ejosat.1216396)

ATIF/REFERENCE: Altunbey Özbay, F. & Özbay, E. (2022). Kaotik Denizatı Optimizasyon Algoritması. *Avrupa Bilim ve Teknoloji Dergisi*, (44), 51-58.

Öz

Global optimizasyon teknikleri olarak bilinen metasezgisel algoritmalar, çeşitli karmaşık ve gerçek optimizasyon problemlerini çözmek için başarıyla kullanılmaktadır. Metasezgisel yöntemler, fizik, sürü zekâsı ve biyolojinin farklı ilkelerinden ilham almaktadır. Denizatı Optimizasyon Algoritması (DOA), denizatlarının doğadaki hareket, avlanma ve üreme davranışlarından esinlenerek önerilmiş sürü zekâsına tabanlı metasezgisel bir optimizasyon algoritmasıdır. Sürü zekâsına dayalı metasezgisel optimizasyon algoritmalarından daha hızlı ve yüksek doğrulukta yakınsama elde etmek için farklı yöntemler önerilmiştir. Bu çalışmada, DOA'nın yakınsama hızını artırmak ve yerel çözümlerde takılıp kalmasını engellemek için rastgele değerler yerine Chebyshev, Circle, Gauss, Iterative, Logistic, Piecewise ve Sine olmak üzere yedi farklı kaotik harita uygulanmıştır. İlk kez bu çalışmada önerilen Kaotik Denizatı Optimizasyon Algoritması (KDOA), tek modlu, çok modlu ve sabit boyutlu çok modlu olmak üzere yedi farklı kıyaslama fonksiyonuna uygulanmıştır. Önerilen KDOA'nın performansını değerlendirmek için klasik DOA karşılaştırılmıştır. Deneysel sonuçlara göre, KDOA'nın yedi farklı kıyaslama fonksiyonunda klasik DOA'ya göre daha iyi sonuçlar elde ettiği gözlemlenmiştir.

Anahtar Kelimeler: Denizatı optimizasyon algoritması, Kaotik haritalar, Metasezgisel algoritmalar, Optimizasyon.

Chaotic Seahorse Optimization Algorithm

Abstract

Metaheuristic algorithms, known as global optimization techniques, have been successfully used to solve a variety of complex and real optimization problems. Metaheuristic methods are inspired by different principles of physics, swarm intelligence, and biology. The Seahorse Optimization Algorithm (SOA) is a suggested swarm intelligence-based metaheuristic optimization algorithm inspired by the movement, hunting, and breeding behavior of seahorses in nature. Different methods have been proposed to achieve faster and higher accuracy convergence than metaheuristic optimization algorithms based on swarm intelligence. In this study, seven different chaotic maps, namely Chebyshev, Circle, Gauss, Iterative, Logistic, Piecewise, and Sine, were applied instead of random values in order to increase the convergence speed of SOA and to prevent it from getting stuck in local solutions. The Chaotic Seahorse Optimization Algorithm (CSOA), proposed for the first time in this study, has been applied to seven different benchmarking functions. Classic SOA was compared to evaluate the performance of the proposed CSOA. According to the experimental results, it was observed that CSOA achieved better results than classical SOA in seven different comparison functions.

Keywords: Seahorse optimization algorithm, Chaotic maps, Metaheuristic algorithms, Optimization.

* Sorumlu Yazar: faltunbey@firat.edu.tr

1. Giriş

Son yıllarda, mühendislik alanında çeşitli optimizasyon problemlerinin üstesinden gelmek için çok sayıda algoritma önerilmiştir (Onay & Aydemir, 2022). Bu optimizasyon problemleri, amaç fonksiyonunu optimize etmek için belirli koşullar altında birkaç parametrenin değerini belirlemektedir. Genel olarak amaç fonksiyonu, probleme dayalı olarak minimum veya maksimum bir çözüm sağlayan belirli bir özelliktir. Optimum çözüm elde edilirken çeşitli optimizasyon problemleri ortaya çıkmaktadır. Bu optimizasyon problemleri dinamik veya statik, sürekli veya ayırık, tek amaçlı veya çok amaçlı, kısıtlı veya kısıtsız olmak üzere çeşitli tiplere ayrılmaktadır. Bu nedenle, bu tür optimizasyon problemlerinin doğruluğunu ve verimliliğini artırmak için, birçok araştırmacı, kolay uygulanabilen, gradyan bilgisi gerektirmeyen ve yerel çözümlerden kaçınmak veya atlamak için metasezgisel algoritmalar önermiştir (Hassan, 2021).

Metasezgisel optimizasyon algoritmaları (MOA), gerçek dünya problemlerini çözmek için sıklıkla kullanılan optimizasyon yöntemleridir. Optimizasyon süreci ile, amaç fonksiyonunu minimize veya maksimize ederek bir problemin optimal karar değişkenlerini bulunmaktadır. Bu algoritmalar yüksek verimlilikleri ve düşük hesaplama karmaşıklıkları nedeniyle hem matematik hem mühendislik problemlerinin çözümünde de başarıyla kullanılmaktadır. MOA'lar keşif ve sömürü olmak üzere iki ana aşamadan oluşmaktadır. MOA'da en önemli durum bu iki aşama arasındaki dengedir (Kaveh & Mahdavi, 2014). Son 10 yılda, keşif ve sömürü aşamalarını dengelemek için fizik, sürü zekası ve biyolojiden esinlenerek birçok yeni MOA önerilmiştir. Örneğin; genetik algoritma, Darwin'in evrim teorisinden esinlenerek önerilmiştir (Holland, 1992). Diferansiyel evrim algoritması, genetik algoritmadaki mutasyon ve çaprazlama ile aynı operatörleri farklı bir yaklaşım ile kullanmaktadır (Storn & Price, 1997). Parçacık sürüsü optimizasyonu, kuşların ve balıkların yiyecek yakalamak için sergiledikleri sosyal davranışlarından ilham almıştır (Eberhart & Kennedy, 1995). Yapay arı kolonisi algoritması, bal arılarının bilgi paylaşım kabiliyetini ve yiyecek arama davranışını taklit etmektedir (Karaboga & Basturk, 2007). Gri kurt optimizasyon algoritması, doğadaki gri kurtların avlanma davranışından ilham alarak önerilmiştir (Mirjalili vd., 2014). Sinüs ve kosinüs algoritması, sinüs ve kosinüs fonksiyonları temelinde matematiksel bir fonksiyon kullanmaktadır (Mirjalili, 2016). Yerçekimi arama algoritması, yerçekimi ve hareket yasalarını kullanmaktadır (Rashedi vd., 2009). Yarasa algoritması, yarasalarda yer alan ekolokasyon davranışını taklit etmektedir (Yang, 2010). Atom arama optimizasyonu (ASO), atom dinamiklerine dayalı olarak doğadaki atomların hareketini taklit etmektedir (Zhao vd., 2019).

Bu çalışmada, Denizati Optimizasyon Algoritmasındaki (DOA) rastgele bir değişken yedi farklı kaotik harita ile kontrol edilerek yeni bir Kaotik Denizati Optimizasyon Algoritması (KDOA) geliştirilmiştir. Önerilen KDOA tek modlu, çok modlu ve sabit boyutlu çok modlu fonksiyonlar olmak üzere farklı kıyaslama problemleri üzerinde test edilmiştir. KDOA, kullanılan kıyaslama problemlerinde klasik DOA'dan daha iyi sonuçlar vermiştir.

Bu çalışmanın geri kalanı şu şekilde planlanmıştır: DOA hakkında bilgiler ikinci bölümde verilmiştir. Üçüncü bölümde kaos teorisine dayalı kaotik haritalar incelenmiştir ve önerilen KDOA açıklanmıştır. Dördüncü bölümde ise KDOA performansı

değerlendirilmiştir. Son bölümde, sonuçlar ve gelecek çalışmalar incelenmiştir.

2. Denizati Optimizasyon Algoritması

Denizati optimizasyon algoritması (DOA), temel olarak hareket, avlanma ve üreme olmak üzere üç önemli bileşenden meydana gelmektedir. DOA'nın keşfini ve kullanımını dengelemek adına geliştirilen yerel ve küresel arama stratejileri sırasıyla hareket ve yırtıcılığın sosyal davranışları için tasarlanmıştır. İlk iki bileşene ait davranışların tamamlanması halinde üreme davranışı gerçekleştirilmektedir. Bu bölümde DOA'nın matematiksel modelleri ayrıntılı olarak açıklanmaktadır (Zhao vd., 2022).

2.1. Başlatma

Çoğu MOA'da olduğu gibi DOA'da da bir başlangıç popülasyonu bulunmaktadır. Her bir denizatının, problemlerin arama uzayında bir aday çözümü temsil ettiklerini varsayarsak denizatlarının tüm popülasyonu Denklem 1'de verildiği şekilde ifade edilebilir (Zhao vd., 2022):

$$\text{Denizatları} = \begin{bmatrix} x_1^1 & \dots & x_1^{Dim} \\ \vdots & \ddots & \vdots \\ x_{pop}^1 & \dots & x_{pop}^{Dim} \end{bmatrix} \quad (1)$$

burada Dim değişkenin boyutunu, pop ise popülasyon boyutunu göstermektedir. Her bir çözüm, belirli bir problemin alt sınırı ve üst sınırı arasında sırasıyla AS ve US ile gösterildiği gibi rastgele oluşturulmaktadır. X_i^j 'nin i 'inci bireysel ifadesinin temsil edildiği arama uzayındaki $[AS, US]$ Denklem 2'de verildiği gibi hesaplanmaktadır:

$$X_i = [x_i^1, \dots, x_i^{Dim}] \\ x_i^j = \text{rast} \times (US^j - AS^j) + AS^j \quad (2)$$

burada $\text{rast} [0, 1]$ aralığında rastgele bir değeri göstermektedir. x_i^j , i . bireydeki j . boyutu ifade etmektedir. i , 1 ile pop arasında değişen pozitif bir tamsayıdır. j , $[1, Dim]$ aralığındaki pozitif bir tamsayı ile ifade edilmektedir. AS^j ve US^j , optimize edilmiş problemin j . değişkeninin alt sınırını ve üst sınırını ifade etmektedir. Minimum optimizasyon probleminin örnek alındığı minimum uygunluğa sahip olan birey, X_{elit} ile gösterilen elit birey olarak kabul edilmektedir. X_{elit} Denklem 3'te verildiği üzere elde edilmektedir (Zhao vd., 2022):

$$X_{elit} = \text{argmin} (f(X_i)) \quad (3)$$

burada $f(\cdot)$, verilen bir problemin amaç fonksiyonunun değerini temsil etmektedir.

2.2. Denizatinın Hareket Davranışları

İlk davranış için, denizatlarının farklı hareket kalıpları yaklaşık olarak rast (0, 1) normal dağılımını takip etmektedir. Keşif ve sömürü performansından ödün vermemek adına, kesme noktası olarak belirlenen $r_1 = 0$ ifadesinin yarısı yerel arama ve diğer yarısı küresel arama için alınmaktadır. Dolayısıyla hareketler iki duruma ayrılarak tanımlanmıştır (Zhao vd., 2022):

Durum-1: Denizdeki girdap olayı ile birlikte denizatının yaptığı sarmal hareket olarak tanımlanmıştır. Rastgele normal değer r_1 kesme noktasının sağ tarafında bulunduğu, esas olarak DOA'nın yerel kullanımını gerçekleştirmektedir. Denizatları spiral hareketi takip ederek elite X_{elit} doğru ilerlemektedir. Özellikle, Lévy uçuşu, denizatlarının erken iterasyonlarda diğer pozisyonlara geçme olasılığı yüksek olan ve DOA'nın aşırı yerel

sömürsünden kaçınan denizatlarının hareket adım boyutunu simüle etmek için kullanılmaktadır (Mantegna, 1994). Aynı zamanda, denizatının bu spiral hareket modu, mevcut yerel çözümlerin komşuluklarını genişletmek için dönme açısını sürekli olarak değiştirmektedir. Bu durumda denizatının yeni konumu matematiksel olarak şöyle ifade edilmiştir:

$$X_{yeni}^1(t+1) = X_i(t) + Levy(\lambda)((X_{elit}(t) - X_i(t)) \times x \times y \times z + X_{elit}(t)) \quad (4)$$

burada $x=\rho \times \cos(\theta)$, $y=\rho \times \sin(\theta)$ ve $z=\rho \times \theta$, arama ajanlarının pozisyonlarının güncellenmesine yardımcı olan sırasıyla sarmal hareket altındaki koordinatların (x, y, z) üç boyutlu bileşenlerini göstermektedir. $\rho=u \times e^{\theta v}$ logaritmik spiral sabitleri u ve v tarafından tanımlanan gövdelerin uzunluğunu temsil etmektedir ($u=0,05$ ve $v=0,05$ olarak ayarlanmıştır). θ , $[0, 2\pi]$ aralığında rastgele bir değer olarak tanımlıdır. Lévy(z), Lévy uçuş dağılım fonksiyonudur ve Denklem 5'te verildiği gibi hesaplanmaktadır (Mantegna, 1994):

$$Levy(z) = s \times \frac{k \times \sigma}{|w|^{\frac{1}{\lambda}}} \quad (5)$$

burada λ , $[0, 2]$ aralığında tanımlı rastgele bir sayıdır (çalışmada $\lambda = 1,5$ olarak belirlenmiştir). s , 0,01 olarak tanımlanmış sabit bir değerdir. k ve w $[0, 1]$ aralığında rastgele sayılardır. σ Denklem 6'da verildiği şekilde hesaplanmaktadır:

$$\sigma = \frac{\Gamma(1 + \lambda) \times \sin\left(\frac{\pi \lambda}{2}\right)}{\Gamma\left(\frac{1+\lambda}{2}\right) \times \lambda \times 2^{\left(\frac{\lambda-1}{2}\right)}} \quad (6)$$

Durum-2: Denizatının deniz dalgaları ile birlikte yaptığı Brownian hareketi olarak tanımlanmıştır. Sürüklenme eylemi altında, r_1 kesme noktasının sol tarafında bulunduğu DOA'nın keşfi gerçekleştirilmektedir. Bu durumda, arama işlemi DOA'nın yerel ekstremumdan kaçınması için önemlidir. Arama uzayında daha iyi bir keşfi sağlamak adına denizatının başka bir hareketli uzunluğunu taklit etmek için Brownian hareketi uygulanmaktadır. Bu durumun matematiksel ifadesi Denklem 7'de verilmiştir:

$$X_{yeni}^1(t+1) = X_i(t) + rast * l * \beta_t * (X_i(t) - \beta_t * X_{elit}) \quad (7)$$

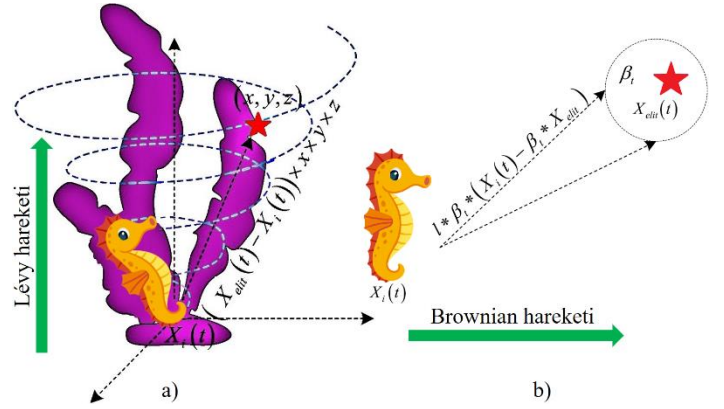
burada l sabit katsayıdır (çalışmada $l=0,05$ olarak belirlenmiştir). β_t , rastgele bir değer olarak standart normal dağılıma uyan Brownian hareketinin rastgele yürüyüş katsayısıdır ve Denklem 8'de verildiği gibi hesaplanmaktadır (Einstein, 1956):

$$\beta_t = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) \quad (8)$$

Genel olarak bu iki durumun toplamında, denizatının t iterasyonundaki yeni konumunun elde edilmesinde Denklem 9'da verilen formül uygulanmaktadır:

$$= \begin{cases} X_{yeni}^1(t+1) \\ X_i(t) + Levy(\lambda) \\ \left((X_{elit}(t) - X_i(t)) \times x \times y \times z + X_{elit}(t) \right) r_1 > 0 \\ X_i(t) + rast * l * \beta_t * (X_i(t) - \beta_t * X_{elit}) r_1 \leq 0 \end{cases} \quad (9)$$

burada $r_1 = \text{rast}()$ rastgele normal bir sayıdır. Denizatının iki farklı türdeki yani spiral veya Brownian hareket modunu takip ederek konum güncelleme diyagramı Şekil 1'de gösterilmiştir. Lévy ve Brownian hareket kalıplarının her ikisi de denizdeki belirsiz ortama dayalı olarak denizatının hareket eden rastgeleliğini yansıtmaktadır.



Şekil 1. Denizatının denizdeki a) Lévy, b) Brownian hareketleri

2.3. Denizatının Yırtıcı Davranışları

Denizatının zooplankton ve küçük kabukluları avlama çabası iki neticeyle sonuçlanır, bunlar; başarı ve başarısızlıktır. Denizatının yiyecek yakalamayı başarma olasılığının %90'ın üzerinde olduğu göz önünde bulundurulduğunda, DOA'nın bu iki sonucu ayırt etmek için tasarlanmış rasgele sayısı r_2 , 0,1 gibi kritik bir değere ayarlanmıştır. Elitin, belirli bir dereceye kadar, avın yaklaşık konumunu gösterdiği varsayıldığından, avlanma başarısı DOA'nın sömürü yeteneğini vurgulamaktadır. $r_2 > 0,1$ ise denizatının yırtıcı hareketinin başarılı olduğu kabul edilmektedir yani denizatı avına (elit) gizlice yaklaşır, avdan daha hızlı hareket ederek onu yakalamaktadır. Bunun aksine, avlanma başarısız olduğunda, her ikisinin de tepki hızı öncekinin tersi olarak kabul edilir, bu da denizatının arama alanını keşfetme eğilimlerini ima etmektedir. Bu yırtıcı davranışın matematiksel ifadesi Denklem 10'da verilmiştir:

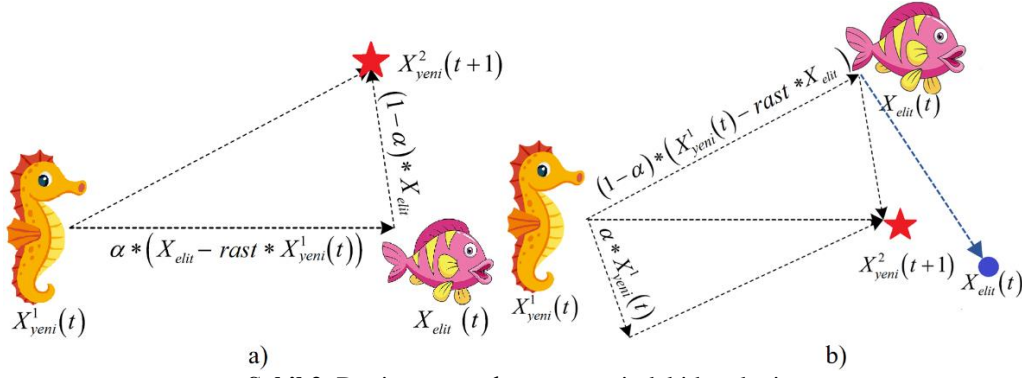
$$X_{yeni}^2(t+1) = \begin{cases} \alpha * (X_{elit} - rast * X_{yeni}^1(t)) + \\ (1 - \alpha) * X_{elit} & r_2 > 0,1 \\ (1 - \alpha) * (X_{yeni}^1(t) - rast * X_{elit}) + \\ \alpha * X_{yeni}^1(t) & r_2 \leq 0,1 \end{cases} \quad (10)$$

burada $X_{yeni}^1(t)$, t yinelemesindeki hareketten sonra denizatının yeni konumunu belirtmektedir. r_2 , $[0, 1]$ aralığındaki rastgele bir değer olarak tanımlanmıştır. α , denizatının avlanacak ava doğru hareket edilen adım boyutunun ayarlanması için yinelemelerle doğrusal olarak azaltılmaktadır ve Denklem 11'de verildiği gibi hesaplanmaktadır:

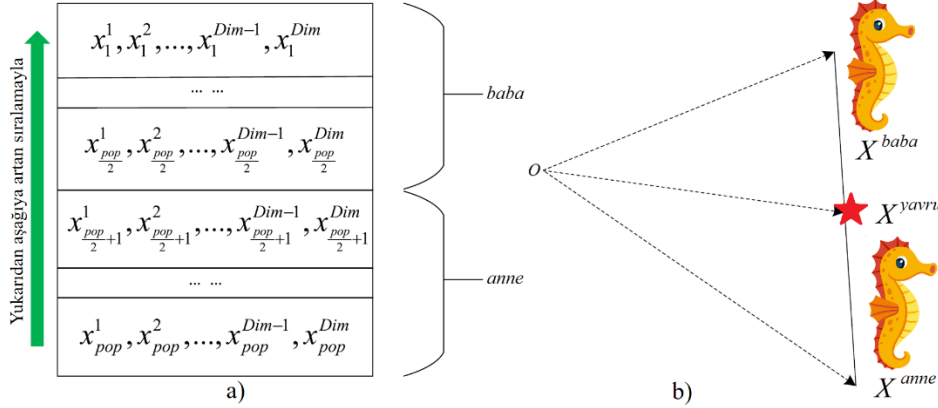
$$\alpha = \left(1 - \frac{t}{N}\right)^{\frac{2t}{N}} \quad (11)$$

burada N , maksimum yineleme sayısını göstermektedir.

Denizatının yırtıcı davranışının iki olası sonucu Şekil 2'de gösterilmektedir. Buna göre, kırmızı yıldız konumu denizatının güncellenmiş konumunu göstermektedir ve avın yaklaşık konumu mavi nokta ile işaretlenmiştir. Denizati başarılı bir şekilde avlandığı taktirde, denizatının elit konuma geçtiği Şekil 2a'da görülmektedir. Bu durumda α parametresinin kontrolü altında, artan iterasyonlarla birlikte kademeli olarak global optimal bireye yakınsayacaktır. Şekil 2b'de, av yakalanmadığı için global arama yapılmaktadır. $1-\alpha$ parametresi, mevcut birey ile elit arasındaki vektöre uygulanmaktadır ve α , güncel olarak güncellenen birey üzerinde hareket etmektedir. Bu yaklaşım, denizatlarının ilk yinelemelerde küresel olarak arama yapmasına ve sonraki yinelemelerde aşırı kullanımdan kaçınmasına izin vermek için tasarlanmıştır.



Şekil 2. Denizatının avlanma sürecindeki hareketi



Şekil 3. Denizatının üreme sürecinde, a) bireylerin sıralanması, b) yavrunun konumu

2.4. Denizatının Üreme Davranışları

Popülasyon, uygunluk değerlerine göre erkek veya kadın gruplarına ayrılmaktadır. DOA, erkek denizatlarının üremeden sorumlu olduklarından dolayı, en iyi uygunluk değerlerine sahip bireylerin yarısının baba, diğer yarısının ise anne olarak alındığını vurgulamak gerekmektedir. Bu bölünme, gelecek neslin devamlılığı için anneler ve babalar arasındaki iyi özelliklerin mirasını kolaylaştıracak ve yeni çözümlerin aşırı yerleştirilmesini önleyecektir. Denizatlarının üremedeki rol atamasının matematiksel ifadesi Denklem 12’de verilmiştir:

$$\begin{aligned} baba &= X_{sıra}^2(1 : pop/2) \\ anne &= X_{sıra}^2(pop/2 + 1 : pop) \end{aligned} \quad (12)$$

burada $X_{sıra}^2$, artan uygunluk değerleri sırasına göre tüm X_{yeni}^2 ’leri göstermektedir. Babalar ve anneler sırasıyla erkek ve kadın popülasyonlarını göstermektedir. Erkekler ve dişiler, yeni yavrular üretmek için rastgele çiftleştirilmektedir. Önerilen DOA’yı kolayca yürütmek için, her bir denizati çiftinin sadece bir çocuk dünyaya getirdiği varsayılmıştır. Denklem 13’te i . yavrunun matematiksel ifadesi verilmiştir:

$$X_i^{yavru} = r_3 X_i^{baba} + (1 - r_3) X_i^{anne} \quad (13)$$

burada r_3 [0, 1] aralığında rastgele bir sayıdır. i , [1, $pop/2$] aralığında pozitif bir tamsayıdır. X_i^{baba} ve X_i^{anne} , sırasıyla erkek ve kadın popülasyonlarından rastgele seçilen bireyleri temsil etmektedir.

Denizatlarının üreme süreci Şekil 3’te gösterilmektedir. Şekil 3a’da görüldüğü üzere, her birey uygunluk değerlerine göre artan düzende sıralanmıştır. Şekil 3b, yeni oluşturulmuş bir yavrunun konumunu yaklaşık olarak göstermektedir. İki alt popülasyon arasında genetik bilgiyi etkili bir şekilde ileten ebeveynler arasındaki çizgide rastgele oluşturulmaktadır.

2.5. DOA’nın Uygulama Süreci

Algoritma-1

- Girdi:** Popülasyon boyutu pop , değişken boyutu Dim , Maksimum yineleme sayısı N
- Çıktı:** Optimum arama ajanı X_{best} , onun uygunluk değeri f_{best}
- 1: Denizati başlatılır $X_i (i = 1, \dots, N)$
 - 2: Her bir denizati için uygunluk değeri hesaplanır
 - 3: En iyi denizati X_{elit} belirlenir
 - 4: **while** ($t < N$) **do**
 - 5: **if** ($r_1 = rastn > 0$) **do** //Hareket davranışull
 - 6: Sabit $u=0,05$ $v=0,05$ parametreleri ayarlanır
 - 7: Dönüş açısı θ , $Rast[0, 2\pi]$
 - 8: Denklem 5 ile Lévy katsayıları oluşturulur
 - 9: Denklem 4 ile Denizatının pozisyonu güncellenir
 - 10: **else if** **do**
 - 11: $l = 0,05$ sabit parametre ayarlanır
 - 12: Denklem 7 ile Denizatının pozisyonu güncellenir
 - 13: **end if**
 - 14: Denklem 10 ile Denizatının pozisyonu güncellenir //Yırtıcı davranışull
 - 15: Sınır aralığının dışındaki değişkenler işleme alınır
 - 16: Her bir denizati için uygunluk değeri hesaplanır
 - 17: Denklem 12 ile $baba$ ve $anne$ seçilir //Üreme davranışull
 - 18: Denklem 13 ile yavru dünyaya getirilir
 - 19: Sınır aralığının dışındaki değişkenler işleme alınır
 - 20: Her bir yavru için uygunluk değeri hesaplanır
 - 21: Yavrulardan bir sonraki yineleme popülasyonu seçilir ve uygunluk değerlerinde en iyi ebeveynler pop sıralanır
 - 22: Elit pozisyonu X_{elit} güncellenir
 - 23: $t = t + 1$
 - 24: **end while**

Önerilen DOA'nın sözde kodu Algoritma-1'de gösterilmektedir. DOA'nın uygulama süreci, bir dizi rastgele çözüm oluşturularak popülasyonun başlatılmasıyla ilerlemektedir. Denizati popülasyonu Denklem 9 ve 10 ile güncellendikten sonra, yavruları yetiştirmek için Denklem 13 kullanılmaktadır. Yeni bir popülasyon, yavrular ve daha önce güncellenen denizatlarından oluşmaktadır. Ancak, bu yeni popülasyonun boyutu 1,5 *pop*'tur. Popülasyonun sınırsız genişlemesini önlemek için yeni popülasyondaki her bir birey tahmin edilmektedir. Bireyler uygunluk değerlerine göre yukarıdan aşağıya doğru artan düzende sıralanmaktadır ve ilk pop denizatları bir sonraki evrimsel süreç için yeni popülasyon olarak iteratif olarak seçilmektedir.

3. Kaotik Denizati Optimizasyon Algoritması

Matematiğin bir alt dalı olan kaos teorisi, doğrusal olmayan ve karmaşık dinamik sistemlere dayanan deterministik bir rastgelelik durumudur. Optimizasyon algoritmalarında çözüm uzayında aramalar genellikle rastgele yapılmaktadır. Metasezgisel optimizasyon algoritmalarında kullanılan rastgele değerler yerine kaotik haritalar kullanılarak tekrarlamaya ve rastgelelikten kaynaklanan ergodik olmayan problemler kontrol edilir ve bu da algoritmaların performansına katkıda bulunmaktadır. Kaotik sistemler bağımsızdır ve kaotik haritalama ile yerel minimum ve maksimum problemlerden kaçınılabilir (Arora & Anand, 2019).

Tablo 1. Kaotik haritalar ve matematiksel tanımları

Harita adı	Matematiksel ifade
Chebyshev	$x_{k+1} = \cos(k \cos^{-1}(x_k))$
Circle	$x_{k+1} = \text{mod}(x_k + b - \frac{a}{2\pi} \sin(2\pi x_k), 1)$
Gauss	$x_{k+1} = \begin{cases} 1, & x_k = 0 \\ \frac{1}{\text{mod}(x_k, 1)}, & \text{Diğer} \end{cases}$
Iterative	$x_{k+1} = \sin(\frac{a\pi}{x_k})$
Logistic	$x_{k+1} = \alpha x_k (1 - x_k)$
Piecewise	$x_{k+1} = \begin{cases} \frac{x_k}{P}, & 0 \leq x_k < P \\ \frac{x_k - P}{0.5 - P}, & P \leq x_k < 0.5 \\ 1 - P - x_k, & 0.5 \leq x_k < 1 - P \\ \frac{0.5 - P}{1 - x_k}, & 1 - P \leq x_k < 1 \end{cases}$
Sine map	$x_{k+1} = \frac{a}{4} \sin(\pi x_k)$

İkinci bölümde de belirttiği gibi denizatları denizdeki girdap olayı ile birlikte sarmal olarak hareket etmektedir. Arama ajanlarının sarmal hareket altındaki *x*, *y* ve *z* koordinatları dönme açısı θ 'ya bağlı olarak değişmektedir. Klasik DOA'da θ değeri $[0, 2\pi]$ arasındaki rastgele olarak değerler almaktadır. Bu çalışmada, klasik DOA'nın rastgele θ parametresi için yedi farklı kaotik harita uygulanmıştır. Bu haritalar Tablo 1'de açıklanmıştır.

4. Deneysel Sonuçlar

Tablo 2. Kıyaslama fonksiyonları

Fonksiyon	Fonksiyon Denklemi	Arama Uzayı []	Dim	f _{min}
Sphere	$f(x) = \sum_{i=1}^n x_i^2$	-100,100	30	0
Rosenbrock	$f(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	-30,30	30	0
Schwefel	$f(x) = -x_i \sin(\sqrt{ x_i })$	-500,500	30	-418.9829×D
Penalized2	$f(x) = 0.1 \left\{ \sin^2(3\pi x_i) + \sum_{i=1}^n (x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2 + [1 + \sin^2(2\pi x_n)] \right\} + \sum_{i=1}^n u(x_i, 5, 100, 4)$	-50,50	30	0
Kowalik	$f(x) = \sum_{i=1}^{11} [a_i - \frac{x_i(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}]^2$	-5,5	4	0.00030
Hartman6	$f(x) = -\sum_{i=1}^4 c_i \exp(-\sum_{j=1}^6 a_{ij}) - p_{ij}^2)$	0,1	6	-3.32
Shekel10	$f(x) = -\sum_{i=1}^{10} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	0,10	4	-10.5363

Önerilen KDOA'nın performansını değerlendirmek için yedi farklı kıyaslama fonksiyonu kullanılmıştır. Bu fonksiyonlardan Sphere ve Rosenbrock tek modlu, Schwefel ve Penalized2 çok modlu ve Kowalik, Hartman6 ve Shekel10 sabit boyutlu çok modlu kıyaslama fonksiyonlarıdır. Bu kıyaslama fonksiyonlarına ilişkin bilgiler Tablo 2'de verilmiştir. Bu çalışmada Klasik DOA ve önerilen KDOA'lar için popülasyon sayısı 30, maksimum iterasyon sayısı 500 olarak belirlenmiş ve 30 kez çalıştırılmıştır. Tüm deneyler Intel(R) Core(TM) i7-7700, 4.0 GHz CPU, 32 GB RAM, GTX1070 8GB GDDR5 Grafik Kartı, Win 11 ve Matlab 2021b ile gerçekleştirilmiştir.

Tablo 3. Kıyaslama fonksiyonlarının karşılaştırılmalı sonuçları

		Klasik	Chebyshev	Circle	Gauss/ mouse	Iterative	Logistic	Piecewise	Sine
Sphere	Min	1.12E-141	2.81E-147	3.84E-147	2.08E-148	3.17E-149	1.61E-148	1.42E-148	8.71E-150
	Maks	6.92E-141	8.13E-143	6.88E-143	1.38E-142	5.93E-143	3.66E-143	8.71E-144	1.80E-144
	Ort	3.13E-141	7.50E-144	6.13E-144	6.40E-144	6.43E-144	1.74E-144	1.20E-144	1.87E-145
	Std	1.7E-141	1.67E-143	1.4E-143	2.5E-143	1.5E-143	6.7E-144	2E-144	3.6E-145
Rosenbrock	Min	2.81E+01	2.72E+01	2.70E+01	2.72E+01	2.72E+01	2.70E+01	2.71E+01	2.72E+01
	Maks	2.89E+01	2.81E+01	2.81E+01	2.81E+01	2.80E+01	2.79E+01	2.79E+01	2.79E+01
	Ort	2.86E+01	2.76E+01	2.76E+01	2.76E+01	2.74E+01	2.73E+01	2.74E+01	2.74E+01
	Std	2.78E-01	3.80E-01	3.90E-01	3.38E-01	2.56E-01	2.33E-01	2.14E-01	2.38E-01
Schwefel	Min	-6.64E+03	-7.54E+03	-7.07E+03	-7.30E+03	-7.63E+03	-8.61E+03	-7.74E+03	-7.57E+03
	Maks	-5.05E+03	-6.01E+03	-6.09E+03	-6.16E+03	-6.10E+03	-6.15E+03	6.27E+03	-6.25E+03
	Ort	-5.80E+03	-6.53E+03	-6.46E+03	-6.64E+03	-6.60E+03	-6.78E+03	-6.12E+03	-6.71E+03
	Std	4.10E+02	3.34E+02	3.00E+02	3.00E+02	3.22E+02	4.86E+02	2.37E+03	3.29E+02
Penalized 2	Min	1.76E+00	1.44E+00	1.46E+00	1.34E+00	1.13E+00	1.27E+00	1.39E+00	1.25E+00
	Maks	2.70E+00	2.30E+00	2.20E+00	2.21E+00	2.19E+00	1.99E+00	2.20E+00	2.00E+00
	Ort	2.16E+00	2.01E+00	1.90E+00	1.84E+00	1.87E+00	1.79E+00	1.81E+00	1.79E+00
	Std	2.58E-01	2.19E-01	2.04E-01	1.78E-01	1.82E-01	1.52E-01	1.78E-01	1.81E-01
Kowalik	Min	3.23E-04	3.08E-04	3.08E-04	3.08E-04	3.08E-04	3.08E-04	3.08E-04	3.08E-04
	Maks	5.18E-04	3.48E-04	3.52E-04	3.57E-04	3.54E-04	3.23E-04	3.19E-04	3.31E-04
	Ort	4.02E-04	3.16E-04	3.15E-04	3.17E-04	3.14E-04	3.13E-04	3.13E-04	3.14E-04
	Std	5.18E-05	9.52E-06	9.6E-06	1.21E-05	9.99E-06	4.23E-06	3.55E-06	6.11E-06
Hartman6	Min	-3.09E+00	-3.32E+00	-3.32E+00	-3.32E+00	-3.32E+00	-3.32E+00	-3.32E+00	-3.32E+00
	Maks	-2.84E+00	-3.01E+00	-3.19E+00	-3.09E+00	-3.02E+00	-3.02E+00	-3.01E+00	-3.01E+00
	Ort	-3.01E+00	-3.26E+00	-3.31E+00	-3.29E+00	-3.22E+00	-3.24E+00	-3.28E+00	-3.21E+00
	Std	8.46E-02	1.02E-01	2.30E-02	7.47E-02	1.03E-01	1.14E-01	8.71E-02	1.13E-01
Shekel10	Min	-1.05E+01	-1.05E+01	-1.05E+01	-1.05E+01	-1.05E+01	-1.05E+01	-1.05E+01	-1.05E+01
	Maks	-5.02E+00	-5.11E+00	9.88E+00	7.68E+00	-8.39E+00	-5.12E+00	-8.36E+00	-8.56E+00
	Ort	-6.35E+00	-8.80E+00	-8.66E+00	-9.31E+00	-1.00E+01	-9.87E+00	-9.96E+00	-1.00E+01
	Std	2.23E+00	2.28E+00	4.00E+00	3.25E+00	4.72E-01	1.09E+00	5.41E-01	4.79E-01

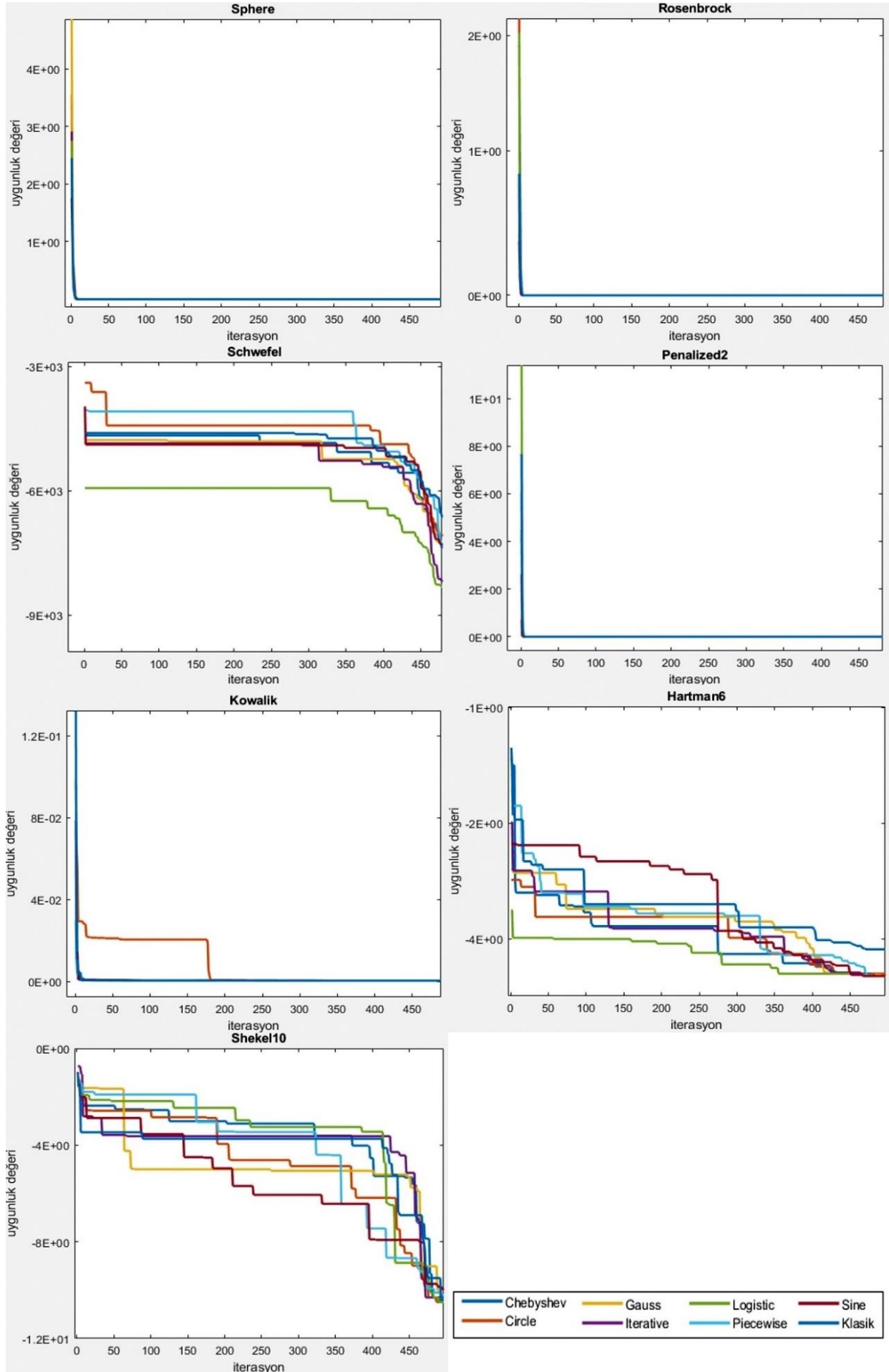
DOA ve KDOA'ların performansını değerlendirmek için minimum (Min), maksimum (Maks), ortalama (Ort) ve standart sapma (Std) değerleri kullanılmıştır. Elde edilen sonuçlar Tablo 3'te listelenmiştir. Tablo 3, incelendiğinde Sphere kıyaslama fonksiyonuna göre en iyi minimum değer Sine harita, Rosenbrock fonksiyonuna göre Circle ve Logistic harita, Schwefel fonksiyonuna göre Logistic harita, Penalized2 fonksiyonuna göre Iterative harita, Kowalik ve Hartman6 fonksiyonuna göre tüm haritalar ile elde edilmiştir. Shekel10 fonksiyonuna göre en iyi minimum değer bakımından birbirine çok yakın sonuçlar elde edilmiştir.

Tablo 3 değerlendirildiğinde, elde edilen ortalama değerlere göre tek modlu Sphere fonksiyonu için Sine harita, Rosenbrock fonksiyonu için Logistic harita ile en başarılı sonuçlar elde edilmiştir. , çok modlu Schwefel ve Penalized2, sabit boyutlu çok modlu Kowalik, Hartman6, Shekel10 fonksiyonlarında klasik DOA'dan daha iyi performans elde ettiği görülmektedir.

Klasik DOA ve KDOA'ların kıyaslama fonksiyonları için yakınsama grafikleri Şekil 4'te verilmiştir.

5. Sonuçlar

Gerçek dünya problemleri, doğrusal olmayan ve karmaşık sistemlerle temsil edilebilir. Bu sistemler kaotik haritalar ile daha net ifade edilebilir. Bu nedenle kaotik haritalar, metasezgisel algoritmalarla birlikte kullanıldığında daha iyi yakınsama yeteneği kazanmaktadır. Kaotik haritalar kullanılarak optimizasyon algoritmalarının çözüm kalitesi önemli ölçüde artırılabilir. Metasezgisel algoritmalarda, sömürü ve keşif aşaması arasındaki denge, algoritmanın etkin yakınsamasında önemli bir rol oynamaktadır. Bu çalışmada, kaotik haritalar DOA'ya entegre edilmiştir. Klasik DOA'nın daha hızlı yakınsama ve daha yüksek doğruluk elde etmek için yedi farklı kaotik harita KDOA'lar geliştirilmiştir. KDOA'nın performansı tek modlu, çok modlu ve sabit katsayılı çok modlu olmak üzere farklı kıyaslama fonksiyonları ile test edilmiştir. Bu çalışmada elde edilen sonuçlar, kaotik haritaların optimizasyon yöntemlerinin performansını artırdığını göstermektedir. DOA'da arama yapmak için rastgele değişkenler kullanmak yerine kaotik haritaları kullanmak daha uygun olabilir. Gelecekte DOA'nın dağıtılmış, çok amaçlı ve paralel versiyonları literatüre kazandırılabilir ve metasezgisel optimizasyon problemlerinde etkin bir şekilde kullanılabilir.



Şekil 4. Klasik DOA ve KDOA'ların yakınsama grafikleri

Kaynakça

- Arora, S., & Anand, P. (2019). Chaotic grasshopper optimization algorithm for global optimization. *Neural Computing and Applications*, 31(8), 4385-4405.
- Eberhart, R., & Kennedy, J. (1995, October). A new optimizer using particle swarm theory. In *MHS'95. Proceedings of the sixth international symposium on micro machine and human science* (pp. 39-43). Ieee.
- Einstein, A. (1956). *Investigations on the Theory of the Brownian Movement*. Courier Corporation.
- Hassan, B. A. (2021). CSCF: a chaotic sine cosine firefly algorithm for practical application problems. *Neural Computing and Applications*, 33(12), 7011-7030.
- Holland, J. H. (1992). Genetic algorithms. *Scientific american*, 267(1), 66-73.
- Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of global optimization*, 39(3), 459-471.
- Kaveh, A., & Mahdavi, V. R. (2014). Colliding bodies optimization: a novel meta-heuristic method. *Computers & Structures*, 139, 18-27.
- Mantegna, R. N. (1994). Fast, accurate algorithm for numerical simulation of Levy stable stochastic processes. *Physical Review E*, 49(5), 4677.
- Mirjalili, S. (2016). SCA: a sine cosine algorithm for solving optimization problems. *Knowledge-based systems*, 96, 120-133.
- Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in engineering software*, 69, 46-61.
- Onay, F. K., & Aydemir, S. B. (2022). Chaotic hunger games search optimization algorithm for global optimization and engineering problems. *Mathematics and Computers in Simulation*, 192, 514-536.
- Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. (2009). GSA: a gravitational search algorithm. *Information sciences*, 179(13), 2232-2248.
- Storn, R., & Price, K. (1997). Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4), 341-359.
- Yang, X. S. (2010). A new metaheuristic bat-inspired algorithm. In *Nature inspired cooperative strategies for optimization (NICSO 2010)* (pp. 65-74). Springer, Berlin, Heidelberg.
- Zhao, S., Zhang, T., Ma, S., & Wang, M. (2022). Sea-horse optimizer: a novel nature-inspired meta-heuristic for global optimization problems. *Applied Intelligence*, 1-28.
- Zhao, W., Wang, L., & Zhang, Z. (2019). A novel atom search optimization for dispersion coefficient estimation in groundwater. *Future Generation Computer Systems*, 91, 601-610.