



Yüksek Boyutlu Nümerik Optimizasyon Problemlerinin Çözümünde Kelebek Optimizasyon Algoritmasının Performansının Değerlendirilmesi

Evaluation of The Performance of Butterfly Optimization Algorithm in Solving High-Dimensional Numerical Optimization Problems

¹Ceren BAŞTEMUR KAYA , ²Ebubekir KAYA 

¹Nevşehir Hacı Bektaş Veli Üniversitesi, Bilgisayar Teknolojileri Bölümü, Merkez/Nevşehir, Türkiye

²Nevşehir Hacı Bektaş Veli Üniversitesi, Bilgisayar Mühendisliği Bölümü, Merkez/Nevşehir, Türkiye

¹ceren@nevsehir.edu.tr, ²ebubekir@nevsehir.edu.tr

Araştırma Makalesi/Research Article

ARTICLE INFO

Article history

Received : 5 September 2022

Accepted : 16 October 2022

Keywords:

Butterfly Optimization Algorithm, High-Dimensional Problems, Global Optimization, Metaheuristic Algorithms

ABSTRACT

The butterfly optimization algorithm is one of the metaheuristic algorithms that models the foraging behavior of butterflies. It appears in the literature that it has been used successfully in the solution of many problems. Control parameters and dimension of problem directly affect the performance of a metaheuristic algorithm. When possible, problems are evaluated, some may be low-dimensional, while others may be high-dimensional. At the same time, it is expected that metaheuristic algorithms are successful on high-dimensional problems as well. In this study, the performance of the butterfly optimization algorithm is evaluated in solving high-dimensional numerical optimization problems. The performance of butterfly optimization algorithm is compared with artificial bee colony algorithm, harmony search and flower pollination algorithm. Wilcoxon signed-rank test is used to analyze the significance of the results. The results show that the butterfly optimization algorithm is generally more successful than other algorithms in solving high-dimensional numerical optimization problems.

© 2022 Bandırma Onyedi Eylül University, Faculty of Engineering and Natural Science. Published by Dergi Park. All rights reserved.

MAKALE BİLGİSİ

Makale Tarihleri

Gönderim : 5 Eylül 2022

Kabul : 16 Ekim 2022

Anahtar Kelimeler:

Kelebek Optimizasyon Algoritması, Yüksek Boyutlu Problemler, Global Optimizasyon, Meta-Sezgisel Algoritmalar

ÖZET

Kelebek optimizasyon algoritması, kelebeklerin yiyecek arama davranışını modelleyen güncel meta-sezgisel algoritmalarından biridir. Pek çok problemin çözümünde başarılı bir şekilde kullanıldığı literatürde görülmektedir. Bir meta sezgisel algoritmanın başarısını, kontrol parametreleri ve problem boyutu doğrudan etkilemektedir. Muhtemel problemler değerlendirildiğinde, bazıları düşük boyutlu iken, bazıları yüksek boyutlu olabilmektedir. Bununla birlikte, meta-sezgisel algoritmaların yüksek boyutlu problemler üzerinde de başarılı olması beklenmektedir. Bu çalışmada, yüksek boyutlu nümerik optimizasyon problemlerinin çözümünde kelebek optimizasyon algoritmasının performansı değerlendirilmiştir. Kelebek optimizasyon algoritması; yapay arı koloni algoritması, armoni arama algoritması ve çiçek tozlaşma algoritması ile karşılaştırılmıştır. Sonuçların anlamlılığının analizi için Wilcoxon işaretli sıralar testi uygulanmıştır. Sonuçlar, çok boyutlu nümerik optimizasyon problemlerinin çözümünde, kelebek optimizasyon algoritmasının diğer algoritmalarından genel olarak daha başarılı olduğunu göstermiştir.

© 2022 Bandırma Onyedi Eylül Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi. Dergi Park tarafından yayımlanmaktadır. Tüm Hakları Saklıdır.

1. GİRİŞ

Gerçek dünya karmaşık, çok yönlü, kısa zamanda en iyi sonuçların elde edilebileceği çözüm yöntemlerine ihtiyaç duyan problemler içermektedir. Bu ihtiyaç doğrultusunda karmaşık problemleri gerçek zamanlı çözebilen algoritmalar geliştirilmiştir [1]. Özellikle doğadan ilham alan meta-sezgisel yöntemler performans açısından iyi sonuçlar vermektedir [2]. Her meta-sezgisel algoritma, iyi bilinen bir doğal olayı modellemeye çalışır ve modelini optimizasyon problemlerini çözmek için uygular. Meta-sezgisel algoritmaları; evrimsel, sürü zekası tabanlı, fizik tabanlı ve insan tabanlı olarak dört başlık altında gruplandırmak mümkündür [3]. Doğadaki canlıların yiyecek arama, eş seçme ve hayatta kalmak için sergiledikleri davranışlar incelendiğinde bazı zeki yaklaşımların olduğu görülmektedir. Bu yaklaşımlar meta-sezgisel algoritmalara dönüştürüldüğünde, gerçek dünya problemlerini çözmeye çok iyi sonuçlar verdikleri görülmüştür [4]. Özellikle endüstriyel ve bilimsel çalışmalar için karmaşık optimizasyon problemlerinin çözümü büyük önem arz etmektedir. Bu algoritmalar optimum çözüme yakınsayabilir ve kısa sürede etkili çözümler üretebilir [5]. Bu durum meta-sezgisel algoritmaların yoğun bir şekilde kullanılmasının önemli sebeplerinden biridir ve pek çok alandaki optimizasyon problemlerinin çözümünde kullanılmaktadırlar [6,7].

Literatür incelendiğinde farklı davranışların modellendiği birçok meta-sezgisel algoritmaya rastlamak mümkündür. Yapay Arı Kolonisi (Artificial Bee Colony-ABC) [8], Genetik Algoritma (Genetic Algorithm-GA) [9], Karınca Koloni Optimizasyonu (Ant Colony Optimization-ACO) [10], Parçacık Sürü Optimizasyonu (Particle Swarm Optimization-PSO) [11], Yapay Bağışıklık Sistemi (Artificial Immune System-AIS) [12] popüler olan meta-sezgisel algoritmalarından bazılarıdır.

Doğadan ilham alan meta-sezgisel yöntemlerden biri de kelebek optimizasyon algoritmasıdır (Butterfly Optimization Algorithm-BOA). BOA kelebelerin yiyecek arama ve eş seçme davranışlarını temel almaktadır. Kelebekler nektar vb. yiyecekleri ararken veya eş seçme için partnerin yerini belirlerken koku duyarını kullanmaktadır. Bu strateji esas alınarak algoritma geliştirilmiştir [13]. Bu algoritma yeni geliştirilmiş olmasına rağmen kısa sürede pek çok gerçek dünya probleminin çözümünde kullanıldığı görülmektedir ve bu çalışmalardan bazıları literatür bölümünde sunulmuştur. Meta-sezgisel algoritmalara ait kontrol parametreleri ve problem boyutu performansı etkileyen önemli hususlardandır. Özellikle gerçek dünyadaki problemler düşük veya yüksek boyutlu olabilmektedirler. BOA açısından bakıldığında; düşük boyutlu problemlerin çözümü üzerine pek çok çalışma bulunmasına rağmen yüksek boyutlu problemler açısından sınırlı sayıda çalışma olduğu görülmektedir. Literatürde BOA'nın yüksek boyutlu nümerik optimizasyon problemlerinin çözümüne yönelik çalışmanın olmayışı bu çalışmayı yenilikçi kılmaktadır. Bu çalışma kapsamında, gelecekte yapılacak çalışmalara yol göstermek amacıyla, yüksek boyutlu nümerik optimizasyon problemlerinin çözümünde BOA'nın performansı değerlendirilmiştir. Bununla birlikte, ilgili problemlerin çözümü için daha etkili olabilecek kontrol parametre değerlerinin tespitine yönelik uygulamalar gerçekleştirilmiştir. Ayrıca, BOA'nın performansı, literatürdeki popüler bazı meta-sezgisel algoritmalarla karşılaştırılmıştır.

Bu çalışmanın ilerleyen bölümleri şu şekilde organize edilmiştir: 2. bölümde literatür çalışması verilmiştir. 3. bölümde BOA'nın detayları anlatılmıştır. 4. bölümde simülasyon sonuçları sunulmuştur. Son bölümde ise tartışma ve sonuca yer verilmiştir.

2. LİTERATÜR ÇALIŞMASI

Yıldız ve ark. [14] BOA'yı otomotiv endüstrisinde kullanmıştır. Çalışmada bileşenlerin bağlantıları, şekilleri, ağırlıkları vb. özelliklerinde çalışılmış ve BOA'nın optimum bileşen tasarlamasında yararlı olduğu vurgulanmıştır. Tubishat ve ark. [15] makine öğreniminde BOA'nın geliştirilen bir versiyonu olan dinamik BOA'yı önermişlerdir. UCI deposundan alınan 20 veri seti kullanılarak GA, çekirge optimizasyon algoritması (Grasshopper Optimization Algorithm-GOA), PSO, karınca aslanı optimizasyonu (Antlion Optimization Algorithm-AOA), sinüs kosinüs algoritması (Sine Cosine Algorithm-SCA), BOA ve dinamik BOA'nın sonuçları karşılaştırılmıştır. Sınıflandırma doğruluğu, uygunluk değerleri, seçilen özneliklerin sayısı, istatistiksel sonuçlar ve yakınsama eğrileri gibi ölçütlerinin çoğunda dinamik BOA'nın daha iyi performans gösterdiği belirtilmiştir.

Arora ve ark. [16] mekanik tasarım optimizasyon problemlerini çözmek amacıyla modifiye edilmiş BOA'yı önermişlerdir. 15 test fonksiyonu ve 3 mühendislik tasarım problemi ile algoritmayı doğrulamışlardır. BOA, PSO, kovaryans matris adaptasyonu evrim stratejisi, ateşböceği algoritması (Firefly Algorithm-FA), uyarlanabilir diferansiyel evrim algoritması (Differential Evolution Algorithm-DEA), N-M SDSA ve Q-N LSA sonuçları karşılaştırılmış ve modifiye edilmiş BOA'nın rekabetçi sonuçlar ortaya koyduğu ifade edilmiştir.

Arora ve Singh [17] BOA uygulaması kullanan düğüm lokalizasyon şeması önermiştir. Önerdikleri şemanın performansı ile PSO ve FA şemalarının performansını karşılaştırmışlardır. BOA uygulaması kullanan şemanın daha doğru ve tutarlı konum gösterdiğini belirtmişlerdir.

Arora ve Singh [18] BOA'nın performansını arttırmak amacıyla 10 kaotik harita kullanmıştır. Önerdikleri kaotik kelebek optimizasyon algoritmalarını tek modlu ve çok modlu test fonksiyonlarında ve mühendislik tasarım problemlerinde doğrulamışlardır. Kaotik haritaların performansı önemli ölçüde arttırabildiğini ifade etmişlerdir.

Sharma [19] BOA'nın yeni bir varyantı olan çift yönlü BOA'yı önermiştir. Farklı kıyaslama problemlerinde algoritmanın performansını değerlendirmişlerdir. GA, PSO, DEA, BOA ve önerilen çift yönlü BOA ait sonuçlar kıyaslanmış ve önerilen varyantın rekabetçi sonuçlar gösterdiği belirtilmiştir.

Mortazavi ve Moloodpoor [20] BOA'nın kapasitesini geliştirmek amacıyla sanal kelebek adı verilen yardımcı bir kavramı ele alarak, modifiye bir versiyon olan bulanık BOA'yı önermişlerdir. Önerilen BOA yaklaşımı, standart BOA da dahil olmak üzere bazı algoritmalarla karşılaştırılmıştır. Sonuç olarak geliştirilen versiyon ile BOA'nın arama kabiliyetinin geliştirildiği vurgulanmıştır.

Sharma ve ark. [21] BOA'nın keşif yeteneği ile simbiyotik organizmalar arama algoritmasının problem çözme avantajlarını birleştirerek kombinatoriyal optimizasyon yaklaşımını tanıtmışlardır. İlgili yaklaşımın performansını, bazı optimizasyon algoritmaları ile karşılaştırmışlardır. Sonuçlarda önerilen yöntemin daha üstün ve güvenilir olduğunu belirtmişlerdir.

Arora ve Singh [22] BOA'yı yerel optimumda çalışmasının daha etkili olmasını ve doğruluk seviyesinin artırılmasını sağlayacak şekilde geliştirmişlerdir. Geliştirilen versiyon bazı problemlere uygulanmıştır. ABC algoritması, FA ve PSO ile performansını karşılaştırmışlardır. Sonuçlar incelendiğinde önerilen versiyonun çözüme ulaşması ve yakınsama oranının daha iyi olduğunu ifade etmişlerdir.

Jalali ve ark. [23] yapay sinir ağı ve BOA'nın birleşimine dayalı olan sınıflandırma yöntemi kullandı. İyi bilinen iki veri seti kullanılarak farklı yöntemlerle önerilen sınıflandırma yöntemini karşılaştırmışlardır. Deneysel sonuçlar incelendiğinde diğer yöntemlere üstünlük sağlandığı vurgulanmıştır.

Sharma ve ark. [24] yakınsamayı geliştirmek amacıyla çift yönlü arama yapılan BOA'yı önermiştir. Üç gerçek dünya problemi ve yedi tek modlu problem ele alınarak önerilen algoritmanın performansı değerlendirilmiştir.

3. KELEBEK OPTİMİZASYON ALGORİTMASI

Kelebekler koku, görme, tat alma, dokunma ve işitme duyularını kullanarak hayati fonksiyonlarını sürdürürler. Başta yiyecek ve eş seçme için partner bulma olmak üzere, göç etmede, tehlikelerden kaçmada ve yumurtlama için uygun yer bulmada bu duyularından yararlanırlar. Ancak duyularının içinde koku en önemli olanıdır [25]. Koku almak için kullandıkları duyu alıcıları anten, bacak, el parmakları vb. uzuvlarında bulunmaktadır. Duyu alıcıları kemoreseptör olarak adlandırılan vücut yüzeyindeki sinir hücreleridir [26]. Kemoreseptör sayesinde nektarın kaynağını bulabilirler. Ayrıca kemoreseptörler güçlü genetik yapılarını sürdürmek amacıyla en iyi çiftleme partneri bulma konusunda rehberlik ederler. Dişi kelebeğin yaydığı koku salgısı feromondur. Erkek kelebek feromon sayesinde dişiyi tanımlayabilir.

BOA'da kelebekler arama araçlarıdır. Bir kelebek uygunluk durumuna bağlı olarak koku üretecektir. Uygunluk bir konumdan başka bir konuma geçme durumuna göre değişmektedir. Mesafe boyunca kelebek koku yayar ve diğer kelebekler bu kokuyu hissedebilir, başka kelebeklerle paylaşabilir. Böylece kolektif bir sosyal bilgi ağı oluşturulabilir. BOA'da bir kelebeğin başka bir kelebeğin kokusunu aldığı anda ona doğru hareket etmesi global arama olarak adlandırılmaktadır. Bir kelebeğin koku algılamadığında rastgele hareket etmesi ise yerel arama olarak adlandırılır [13].

Kelebeğin yaydığı her kokunun kendine özgü yapısı vardır. Ses, ışık, sıcaklık vb. bir uyarıcı/yöntem tarafından koku etkilenebilir. BOA'da algılama yöntemi; duyu modalite (c), uyarıcı yoğunluğu (I) ve güç katsayısı (a) parametrelerine dayanmaktadır. Duyusal modalitede (c), algılayıcılar tarafından kullanılan ham girdi kullanılır. Duyusal enerji ölçülür ve aynı şekilde işlenir. Buradaki farklı yöntemler ses, ışık, sıcaklık, koku olabilir. BOA'da yöntem kokunun algılanan büyüklüğüdür (f). Uyarıcı yoğunluğu (I) uyarıcının büyüklüğünü ifade eder. Güç katsayısı (a) ise yoğunluğun yükseltildiği katsayıdır. a parametresi düzenli ifadeye, doğrusal tepkiye ve tepki sıkışmasına izin verir. Tepki genişlemesinde I artarsa; f, I'dan daha hızlı artar. Tepki sıkışmasında I arttığında; f, I'dan daha yavaş artar. Doğrusal tepkide I arttığında; f orantılı olarak artar. BOA'da I'nın büyüklüğünü takip etmek için tepki sıkışması kullanılır [13, 17, 27, 28].

Kelebeklerin doğal varlığı I'nın varyasyonuna ve f'nin formülasyonuna dayanmaktadır. Kokuyu diğer yöntemlerden ayırt etmek için c kullanılır. Daha az I'lı kelebek, daha çok I'lı kelebeğe hareket ettikçe; f, I'dan daha hızlı artar. Bu yüzden f'nin a ile elde edilen bir absorpsiyon derecesi ile değişmesine izin verilir [13].

$$f = cI^a \quad (1)$$

Uygulamadaki durumların çoğunda a ve c [0, 1] aralığında alınabilir. a=1 olması durumu kokunun idealize edilmiş bir ortamda yayıldığını gösterir. a=0 ise, herhangi bir kelebeğin yaydığı kokunun diğer kelebekler tarafından hiç hissedilemeyeceği anlamına gelir. Teorik olarak c ∈ [0,∞]'dir. c optimizes edilecek sistemin karakteristiği ile belirlenir. a ve c değerleri, algoritmanın yakınsama hızını önemli ölçüde etkiler [13].

BOA'da üç aşama bulunmaktadır: (1) başlatma aşaması, (2) iterasyon aşaması ve (3) son aşamadır. BOA'nın her çalışmasında, önce başlatma aşaması yürütülür, ardından tekrarlamalı bir şekilde arama yapılır ve son aşamada en iyi çözüm bulunduğu anda algoritma sonlandırılır. Başlatma aşamasında algoritmanın tarafsız fonksiyonu ve çözüm aralığı tanımlanır. Parametre değerleri atanır. Sonrasında optimizasyon için ilk kelebek popülasyonu oluşturulur. Kelebeklerin konumları, koku ve uygunluk değerleri ile birlikte arama aralığında rastgele oluşturulur. İterasyon aşamasında oluşturulan yapay kelebeklerle arama başlatılır. Her iterasyonda çözüm aralığındaki kelebekler yeni pozisyonlara hareket eder ve uygunluk değerleri değerlendirilir. Algoritmada önce çözüm aralığındaki tüm kelebeklerin uygunluk değerleri hesaplanır. Sonrasında kelebekler Denklem 1 kullanılarak koku üretir. Algoritma

global arama ve yerel arama adımlarını içerisinde barındırır. Global arama aşamasında Denklem 2 kullanılarak en uygun çözüm bulunur [13].

$$x_i^{t+1} = x_i^t + (r^2 \times g^* - x_i^t) \times f_i \quad (2)$$

Burada x_i^t 'de, t iterasyon sayısındaki i'inci kelebek için çözüm vektörü x_i^t 'dir. g^* , mevcut iterasyondaki tüm çözümler arasında bulunan mevcut en iyi çözümü temsil eder. i'inci kelebeğin kokusu f_i ile temsil edilir ve r, [0, 1] aralığında rastgele bir sayıdır. Yerel arama aşaması şu şekilde temsil edilebilir:

$$x_i^{t+1} = x_i^t + (r^2 \times x_j^t - x_k^t) \times f_i \quad (3)$$

Burada x_j^t ve x_k^t , çözüm uzayından j'inci ve k'inci kelebekleridir. x_j^t ve x_k^t aynı kümeye aitse ve r [0, 1] aralığında rastgele bir sayıysa, Denklem 3 ile yerel rastgele gezinti yapılır. BOA'da ortak global arama ile yoğun yerel arama arasında geçiş yapmak için bir değiştirme (anahtarlama) olasılığı p kullanılır. Durdurma kriterlerine ulaşıncaya kadar iterasyon aşamasına devam edilir. Durdurma kriterleri; kullanılan maksimum CPU süresi, ulaşılan maksimum iterasyon sayısı, iyileştirme yapılmayan maksimum iterasyon sayısı, belirli bir hata oranı değerine ulaşılması veya diğer uygun kriterler gibi farklı şekillerde tanımlanabilir. İterasyon aşaması sona erdiğinde, algoritma en iyi uygunluk değeri ile bulunan en iyi çözümü verir [13].

BOA'nın algoritmik yapısı Algoritma 1'de sunulmuştur.

Algoritma 1. Kelebek optimizasyon algoritması [13]

1. Uygunluk fonksiyonu $f(\mathbf{x})$, $\mathbf{x}=(x_1, x_2, \dots, x_{dim})$, $dim=$ boyut parametrelerini belirle
 2. n kelebeğin başlangıç popülasyonu oluştur $x_i=(i=1, 2, \dots, n)$
 3. x_i 'deki uyaran yoğunluğu I_i , $f(x_i)$ kullanarak belirle
 4. Sensör modalitesi c'yi, güç katsayısı a'yı ve değiştirme (anahtarlama) olasılığı p'yi tanımla
 5. **while** durdurma kriteri karşılanmadı **do**
 6. **for each** popülasyondaki her bf kelebeği için **do**
 7. Denklem 1 kullanarak bf için koku hesapla
 8. **end for**
 9. En iyi bf'yi bul
 10. **for** popülasyondaki her bf kelebeği için **do**
 11. [0,1] arasında rastgele bir r sayısı üret
 12. **if** $r < p$ **then**
 13. Denklem 2'yi kullanarak en iyi kelebeğe/çözüme doğru ilerle
 14. **else**
 15. Denklem 3'ü kullanarak rastgele hareket et
 16. **end if**
 17. **end for**
 18. a'nın değerini güncelle
 19. **end while**
-

4. SİMÜLASYON SONUÇLARI

Çok boyutlu nümerik optimizasyon problemi çözümünde BOA'nın performansı değerlendirilmiştir. Uygulamalarda Tablo 1'de verilen ve literatürde yoğun bir şekilde kullanılan klasik test fonksiyonları üzerinde çalışılmıştır. Verilen yedi test fonksiyonundan beş tanesi tek modlu (U- unimodal), iki tanesi ise çok modlu (M-multi-modal) 'dur. Tüm test fonksiyonlarının global minimum değeri sıfırdır. Problem boyutu 100, 200 ve 300 için analizler gerçekleştirilmiştir. Bilindiği üzere popülasyon büyüklüğü ve jenerasyon sayısı performansı etkileyen kriterlerdendir. Bu yüzden uygulamalarda popülasyon büyüklüğü olarak 10, 20 ve 50 için sonuçlar elde edilmiştir. Bu popülasyon büyüklükleri için uygulanan maksimum jenerasyon sayıları sırasıyla 2000, 1000 ve 400'dür. İstatistiksel sonuçların elde edilebilmesi ve sonuçların güvenilirliği açısından her uygulama 30 kere çalıştırılmıştır ve ortalama uygunluk değerleri (f) elde edilmiştir. Bununla birlikte, 30 kere çalıştırma sonuçları kullanılarak standart sapma (σ) değerleri de hesaplanmıştır.

Tablo 2'de problem boyutu 100 için BOA ile elde edilen sonuçlar verilmiştir. F1, F2, F3, F5 ve F6 problemlerinde en iyi uygunluk değeri n değeri 10 olduğu zaman elde edilmiştir. Bu test fonksiyonlarında popülasyon sayısı azaldıkça, daha etkili uygunluk değerleri elde edilmiştir. F4 probleminde tüm popülasyon büyüklükleri için 0 (sıfır) değerine ulaşılmıştır. F7 probleminde en başarılı sonuç n değeri 20 olduğu zaman bulunmuştur. Bununla birlikte n değeri 50 olduğu zaman elde edilen sonuçta oldukça başarılıdır.

Tablo 3'de problem boyutu 200 için BOA ile elde edilen sonuçlar sunulmuştur. d değeri 200 olduğunda F1, F2, F3, F5 ve F6 problemleri için en iyi sonuçlar n değerinin 10 olması durumunda bulunmuştur. F4 probleminde tüm popülasyon büyüklükleri için 0 (sıfır) değeri elde edilmiştir. F7 probleminde en başarılı sonuca n değeri 50 olduğu zaman ulaşılmıştır.

Tablo 4'de problem boyutunun 300 olması durumunda BOA ile elde edilen sonuçlar verilmiştir. d değerinin 100 ve 200 olması durumunda F1, F2, F3, F5 ve F6 problemlerinde olduğu gibi en başarılı sonuca n değeri 10 olduğu

zaman ulaşılmıştır. F4 probleminde tüm popülasyon büyüklükleri için 0 (sıfır) değeri bulunmuştur. F7 probleminde n değeri 50 olduğu zaman en iyi uygunluk değerine ulaşılmıştır. Tablo 2, 3 ve 4 genel olarak değerlendirildiğinde f değerlerinin kabul edilebilir olduğu görülmektedir. Buradaki f değeri, 30 çalıştırmaya ait ortalama uygunluk değeridir. f, tüm çalıştırma sonuçları hakkında genel bilgi vermektedir. Sonuçların f değerine yakın ve uzaklığını analiz etmek için σ değerine bakılmalıdır. Çoğunlukla tüm problemlerde, farklı popülasyon büyüklükleri ve farklı problem boyutları için bulunan σ değerlerinin f değerinden daha küçük olduğu görülmektedir. Bu durum 30 çalıştırmadaki her bir uygunluk değerinin ortalama uygunluk değerine yakın olduğunu göstermektedir. Yani, çoğunlukla benzer sonuçlar çıkmaktadır. Bu durum algoritmanın gürbüz (robust) olduğunun bir işaretidir.

Tablo 1. Uygulamalarda kullanılan test problemleri.

Fonksiyon No	Arama Aralığı	Fonksiyon Adı	Formül	Global Minimum	Tür
F1	[-10,10]	Sphere	$f(x) = \sum_{i=1}^n x_i^2$	0	U
F2	[-10,10]	High conditioned elliptic	$f(x) = \sum_{i=1}^n (10^6)^{i-1} x_i^2$	0	U
F3	[-10,10]	Ackley	$f(x) = -20 \exp \left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right) - \exp \left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i) \right) + 20 + e$	0	M
F4	[-100,100]	Step	$f(x) = \sum_{i=1}^n (x_i + 0.5)^2$	0	U
F5	[-1,1]	Sum of Different Powers	$f(x) = \sum_{i=1}^n x_i ^{i+1}$	0	U
F6	[-10,10]	Sum Squares	$f(x) = \sum_{i=1}^n i x_i^2$	0	U
F7	[-5,5]	Levy	$f(x) = \sin^2(\pi w_1) + \sum_{i=1}^{n-1} (w_i - 1)^2 [1 + 10 \sin^2(\pi w_i + 1)] + (w_d - 1)^2 + [1 + \sin^2(2\pi w_d)]$ $w_i = 1 + \frac{x_i - 1}{4}$	0	M

Problem boyutu 100, 200 ve 300 ile elde edilen sonuçlar ABC algoritması, FPA ve HS ile karşılaştırılmıştır. ABC algoritmasının “Limit” kontrol parametresinin değeri (popülasyon büyüklüğü x problem boyutu)/2 alınmıştır. HS’nin PAR ve HMCR kontrol parametre değerleri sırasıyla 0.3 ve 0.95’dir. FPA’nın değişim olasılığı (switch probability) ise 0.8 alınmıştır. İlgili meta-sezgisel algoritmalar ile elde edilen sonuçlar sırayla Tablo 5, Tablo 6 ve Tablo 7’de verilmiştir. Tüm algoritmalar için koloni büyüklüğü ve maksimum jenerasyon sayısı sırasıyla 10 ve 2000 alınmıştır. İstatiksel analizlerin gerçekleştirilebilmesi için her uygulama 30 kere çalıştırılmış ve ortalama uygunluk değeri hesaplanmıştır. Tablolarda yer alan uygunluk değerleri 30 çalıştırmaya ait ortalama değerlerdir. F1, F2, F3, F4, F5 ve F6 problemlerinde en iyi sonuç BOA ile elde edilmiştir. BOA’dan sonra ise en iyi sonuç ABC algoritması ile bulunmuştur. F7 probleminde ise en iyi sonuca ABC algoritması ile ulaşılmıştır. ABC algoritmasından sonra en iyi sonuç BOA’ya aittir.

Elde edilen sonuçların istatistiksel olarak anlamlılığını belirlemek için Wilcoxon işaretli sıra testi uygulanmıştır. F7 dışındaki tüm örneklerde p=0.000 sonucu elde edilmiştir. Bu durum F7 dışındaki tüm problemlerde BOA’nın diğer algoritmalarından anlamlı olarak başarılı olduğunu göstermektedir. F7 probleminde ise ABC algoritması lehine anlamlı sonuçlar bulunmuştur. F7 probleminde ABC algoritması dışındaki sonuçlarda; BOA’nın FPA ve HS’ye göre istatistiksel olarak anlamlı sonuçlara sahip olduğu tespit edilmiştir.

Tablo 2. BOA ile elde edilen sonuçlar (d=100).

Test Fonksiyonu	Popülasyon Büyüklüğü					
	n=50		n=20		n=10	
	f	σ	f	σ	f	σ
F1	9.00721e-11	6.09649e-12	1.77197e-14	1.01097e-15	2.44174e-17	9.18958e-19
F2	1.14347e-10	8.37831e-12	2.07107e-14	1.15262e-15	2.63364e-17	1.23917e-18
F3	3.08325e-08	1.98785e-09	1.1004e-11	4.78211e-13	1.93623e-14	1.42112e-15
F4	0	0	0	0	0	0
F5	4.74446e-13	5.1679e-13	3.46646e-16	3.113e-16	6.94086e-19	3.33583e-19
F6	1.05061e-10	5.64313e-12	1.9456e-14	9.78891e-16	2.54411e-17	9.99414e-19
F7	9.2193	0.196196	9.20221	0.244644	9.29592	0.214403

Tablo 3. BOA ile elde edilen sonuçlar (d=200).

Test Fonksiyonu	Popülasyon Büyüklüğü					
	n=50		n=20		n=10	
	f	σ	f	σ	f	σ
F1	9.41047e-11	5.79545e-12	1.82942e-14	1.00866e-15	2.44096e-17	1.00092e-18
F2	1.16376e-10	5.89108e-12	2.09957e-14	1.2112e-15	2.70309e-17	1.17804e-18
F3	3.13705e-08	1.37342e-09	1.1155e-11	4.95147e-13	1.99544e-14	1.71207e-15
F4	0	0	0	0	0	0
F5	3.02649e-13	2.24052e-13	2.23141e-16	2.56049e-16	6.97082e-19	2.24924e-19
F6	1.08794e-10	7.93772e-12	1.99807e-14	1.01787e-15	2.57271e-17	1.15855e-18
F7	18.516	0.0494321	18.5419	0.0392034	18.5961	0.0421195

Tablo 4. BOA ile elde edilen sonuçlar (d=300).

Test Fonksiyonu	Popülasyon Büyüklüğü					
	n=50		n=20		n=10	
	f	σ	f	σ	f	σ
F1	9.43137e-11	4.56474e-12	1.85061e-14	1.02513e-15	2.44954e-17	7.43834e-19
F2	1.20001e-10	5.77089e-12	2.07803e-14	1.21436e-15	2.64329e-17	9.30073e-19
F3	3.1065e-08	1.24352e-09	1.11366e-11	6.02741e-13	1.97175e-14	1.6281e-15
F4	0	0	0	0	0	0
F5	2.66934e-13	1.24623e-13	3.1111e-16	2.57464e-16	6.70836e-19	3.70877e-19
F6	1.11008e-10	6.08555e-12	2.0107e-14	8.2465e-16	2.58377e-17	1.02885e-18
F7	27.6446	0.0251591	27.6605	0.0346833	27.7045	0.0401644

Tablo 5. BOA ve diğer algoritmalar ile elde edilen sonuçlar (d=100).

Test Fonksiyonları	BOA	ABC	FPA	HS
F1	2.44174e-17	5.43198e-04	2.04583e+02	3.57534e+02
F2	2.63364e-17	56.6816	5.14954e+06	2.56713e+06
F3	1.93623e-14	0.701012	6.18034	7.96431
F4	0	3.1	22400.7	37380.1
F5	6.94086e-19	3.90688e-11	0.00084601	3.80221e-05
F6	2.54411e-17	0.00638635	9388.18	14963.1
F7	9.29592	0.164277	19.6359	22.6103

Tablo 6. BOA ve diğer algoritmalar ile elde edilen sonuçlar (d=200).

Test Fonksiyonları	BOA	ABC	FPA	HS
F1	2.44096e-17	7.07583	550.125	1339.66
F2	2.70309e-17	104724	1.84675e+07	1.79368e+07
F3	1.99544e-14	2.76478	6.64731	9.77193
F4	0	419.967	57651.7	133295
F5	6.97082e-19	1.22028e-05	0.000931957	0.000298581
F6	2.57271e-17	136.156	50753.1	122176
F7	18.5961	1.56232	59.8887	89.8553

Tablo 7. BOA ve diğer algoritmalar ile elde edilen sonuçlar (d=300).

Test Fonksiyonları	BOA	ABC	FPA	HS
F1	2.44954e-17	91.1405	968.722	2696.33
F2	2.64329e-17	4.85842e+06	3.60536e+07	5.0605e+07
F3	1.97175e-14	4.67604	6.98881	10.7026
F4	0	8701.97	93790.2	272627
F5	6.70836e-19	0.000151676	0.00447091	0.000469817
F6	2.58377e-17	10989.4	134007	364937
F7	27.7045	16.1303	114.89	181.578

5. TARTIŞMA VE SONUÇ

Problem boyutunun düşük ve yüksek olması meta-sezgisel optimizasyon algoritmalarının performansını etkilemektedir. BOA'nın düşük boyutlu optimizasyon problemlerinde başarılı olduğu literatürde görülmektedir. Bu çalışma kapsamında yüksek boyutlu nümerik optimizasyon problemlerinde BOA'nın performansı değerlendirilmiştir. Bununla birlikte, farklı popülasyon büyüklükleri ve problem boyutları üzerinde sonuçlar incelenmiştir. BOA için değerlendirildiğinde yüksek boyutlu problemlerde popülasyon sayısının azalması performansı arttırmıştır. Özellikle n değerinin 10, 20 ve 50 olması durumunda elde edilen sonuçlar düşünüldüğünde en iyi sonuçlar çoğunlukla 10'da bulunmuştur. Problem boyutu 100, 200 ve 300 için elde edilen sonuçlardan problem boyutunun artması çözüm kalitesinin düşmesine sebep olmaktadır.

Algoritmanın performansını etkileyen önemli hususlardan biride problemin yapısıdır. Kullanılan test fonksiyonları tek ve çok modlu olmak üzere iki aşamada değerlendirilmiştir. Tek modlu problemlerde global minimuma ulaşmak daha kolayken, çok modlu problemlerin yapısı gereği global minimuma ulaşmak daha zordur. BOA ile elde edilen sonuçlar değerlendirildiğinde, tek modlu problemlerde bulunan sonuçların çok modlu problemlere göre daha etkili olduğu görülmektedir. Buna karşın, bu çalışmada yüksek boyutlu problemler incelenmiştir. Çok modlu problemlerde yüksek boyutların kullanılmasına rağmen, BOA ile başarılı sonuçların elde edildiği tespit edilmiştir. Özellikle BOA'nın çözüm üretme mekanizmasında global en iyi çözümün kullanılması hızlı yakınsamaya sahip olmasının temel sebeplerinden biridir. Bununla birlikte, sensör modalitesi, güç katsayısı ve değiştirme olasılığı gibi kontrol parametreler problem yapısına göre ayaralanarak etkili çözümlere ulaşılabilmektedir. Bu özellikler BOA'nın tek modlu problemlerin çözümünde başarılı olmasının önemli sebeplerindedir. Bununla birlikte, çok modlu problemlerin çözümünde de etkili olabileceği görülmüştür.

BOA'nın performansını değerlendirmek için popüler sezgisel algoritmalar ABC, FPA ve HS ile karşılaştırılmıştır. Her üç problem boyutu için BOA algoritması ile elde edilen sonuçların diğer algoritmalarından daha başarılı olduğu gözlemlenmiştir. Bir algoritma ile elde edilen ortalama uygunluk değeri, algoritmanın başarısı hakkında genel bir fikir verir. Bununla birlikte, elde edilen sonuçların anlamlılığını incelemek gerekmektedir. Çözüm kalitesinin daha iyi olması, sonuçların anlamlı olduğunu göstermez. Elde edilen sonuçların anlamlılığını tespit etmek için Wilcoxon işaretli sıra testi uygulanmıştır. $p=0.05$ anlamlılık derecesine göre analizler gerçekleştirilmiştir. $p \leq 0.05$ olması durumunda anlamlı bir fark olduğundan söz edilebilir. Aksi durumda sonuçlar arasında anlamlı bir fark olmayacaktır. Wilcoxon işaretli sıra testi sonuçları analiz edildiğinde, genel olarak BOA'ya ait sonuçlarda anlamlı bir farkın olduğu görülmüştür. Düşük boyutlu problemlerde algoritmaların performanslarının etkili olması muhtemeldir. Fakat problem boyutunun artması, ilgili problemi daha karmaşık hale getirmektedir. Problemin zor ve karmaşık hale gelmesi, çözüm kalitesini olumsuz etkileyen bir durumdur. Özellikle bir algoritmanın yüksek boyutlu problemlerde de iyi sonuçlara ulaşması güçlü bir yapıya sahip olduğunun bir göstergesidir. BOA ile elde edilen sonuçlarda yüksek boyutlu problemlerde dahi çok başarılı sonuçlar elde edildiği görülmüştür.

Bu çalışma kapsamında yüksek boyutlu test fonksiyonları üzerinde analizler gerçekleştirilmiş ve BOA'nın başarılı olduğu tespit edilmiştir. Bu sebeple gelecekte yapılacak çalışmalarda test problemleri dışında yüksek boyutlu gerçek dünya problemlerinin çözümünde BOA'nın performansının değerlendirilmesi literatüre önemli katkı sağlayacaktır. Bununla birlikte probleme özgü olarak BOA'nın yeni varyantlarının geliştirilmesi mümkündür. Geliştirilecek yeni varyantların yüksek boyutlu optimizasyon problemlerinin çözümündeki performansı değerlendirilebilir. BOA'nın sensör modalitesi, güç katsayısı ve değiştirme olasılığı kontrol parametrelerinin belirlenmesine yönelik adaptif yaklaşımlar geliştirilebilir.

Yazar Katkıları

Yazarlar çalışmaya eşit oranlı katkı sunmuşlardır.

Çıkar Çatışması

Makale yazarları aralarında herhangi bir çıkar çatışması olmadığını beyan ederler.

KAYNAKÇA

- [1] A. Gogna, and A. Tayal, "Metaheuristics: Review and application", *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 25, no. 4, pp. 503-526, 2013.
- [2] X.-S. Yang, "Nature-inspired metaheuristic algorithms", *Luniver press*, 2010.
- [3] S. Aslan, and S. Demirci, "Immune plasma algorithm: a novel meta-heuristic for optimization problems", *IEEE Access*, vol. 8, pp. 220227-220245, 2020.
- [4] I. Fister Jr, X.-S. Yang, I. Fister, J. Brest, and D. Fister, "A brief review of nature-inspired algorithms for optimization", 2013.
- [5] I. H. Osman, and G. Laporte, "Metaheuristics: A bibliography", *Springer*, 1996.
- [6] S. Aslan, and S. Aslan, "Gürültü minimizasyonu için kafes tabanlı yeni bir yapay arı kolonisi algoritması", *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, vol. 38, no. 1, pp. 15-27, 2023.
- [7] S. Aslan, and T. Erkin, "An immune plasma algorithm based approach for UCAV path planning", *Journal of King Saud University-Computer and Information Sciences*, 2022.
- [8] D. Karaboga, "An idea based on honey bee swarm for numerical optimization", *Technical report-tr06*, Erciyes University, Engineering Faculty, 2005.
- [9] K.-S. Tang, K.-F. Man, S. Kwong, and Q. He, "Genetic algorithms and their applications", *IEEE Signal Processing Magazine*, vol. 13, no. 6, pp. 22-37, 1996.
- [10] M. Dorigo, V. Maniezzo, and A. Colomi, "Ant system: Optimization by a colony of cooperating agents", *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol.26, no.1, pp. 29-41, 1996.
- [11] J. Kennedy, and R. Eberhart, "Particle swarm optimization", In *Proceedings of ICNN'95-International Conference on Neural Networks*, IEEE, 1995.
- [12] J. D. Farmer, N. H. Packard, and A. S. Perelson, "The immune system, adaptation, and machine learning", *Physica D: Nonlinear Phenomena*, vol. 22, no. 1-3, pp. 187-204, 1986.
- [13] S. Arora, and S. Singh, "Butterfly optimization algorithm: A novel approach for global optimization", *Soft Computing*, vol. 23, no. 3, pp. 715-734, 2019.
- [14] B. S. Yıldız, A. R. Yıldız, E. İ. Albak, H. Abderazek, S. M. Sait, and S. Bureerat, "Butterfly optimization algorithm for optimum shape design of automobile suspension components", *Materials Testing*, vol. 62, no. 4, pp. 365-370, 2020.
- [15] M. Tubishat, M. Alswaiti, S. Mirjalili, M. A. Al-Garadi, and T. A. Rana, "Dynamic butterfly optimization algorithm for feature selection", *IEEE Access*, pp. 194303-194314, 2020.
- [16] S. Arora, S. Singh, and K. A. Yetilmezsoy, "Modified butterfly optimization algorithm for mechanical design optimization problems", *Journal of The Brazilian Society of Mechanical Sciences and Engineering*, vol. 40, no. 1, pp. 1-17, 2018.
- [17] S. Arora, and S. Singh, "Node localization in wireless sensor networks using butterfly optimization algorithm", *Arabian Journal for Science and Engineering (Springer Science and Business Media BV)*, vol. 42, no. 8, 2017.
- [18] S. Arora, and S. Singh, "An improved butterfly optimization algorithm with chaos", *Journal of Intelligent and Fuzzy Systems*, vol. 32, no. 1, pp. 1079-1088, 2017.
- [19] T. K. Sharma, "Enhanced butterfly optimization algorithm for reliability optimization problems", *Journal of Ambient Intelligence and Humanized Computing*, pp. 1-25, 2020.
- [20] A. Mortazavi, and M. Moloodpoor, "Enhanced butterfly optimization algorithm with a new fuzzy regulator strategy and virtual butterfly concept", *Knowledge-Based Systems*, vol. 228, pp. 107291, 2021.
- [21] S. Sharma, A. K. Saha, and S. Nama, "An enhanced butterfly optimization algorithm for function optimization", in *Soft Computing: Theories and Applications*, Springer, pp. 593-603, 2020.
- [22] S. Arora, and S. Singh, "An improved butterfly optimization algorithm for global optimization", *Advanced Science, Engineering and Medicine*, vol. 8, no. 9, pp. 711-717, 2016.
- [23] S. M. J. Jalali, S. Ahmadian, P. M. Kebria, A. Khosravi, C. P. Lim, and S. Nahavandi, "Evolving artificial neural networks using butterfly optimization algorithm for data classification", in *International Conference on Neural Information Processing*, Springer, 2019.
- [24] T. K. Sharma, A. K. Sahoo, and P. Goyal, "Bidirectional butterfly optimization algorithm and engineering applications", *Materials Today: Proceedings*, pp. 736-741, 2021.
- [25] R. B. Blair, A. E. Launer, "Butterfly diversity and human land use: Species assemblages along an urban gradient". *Biological Conservation*, vol. 80, no. 1, pp. 113-125, 1997.
- [26] E. Pollard, and T. J. Yates, "Monitoring butterflies for ecology and conservation: The british butterfly monitoring scheme", *Springer Science and Business Media*, 1994.
- [27] J. C. Baird, and E. J. Noma, "Fundamentals of scaling and psychophysics", *John Wiley and Sons*, 1978.
- [28] D. M. MacKay, "Psychophysics of perceived intensity: A theoretical basis for fechner's and stevens' laws", *Science*, vol. 139, no. 3560, pp. 1213-1216, 1963.