



Global Optimizasyon Problemleri için Kaotik Bonobo Algoritması

Sümeyye Bazna^{1*}, Sinem Akyol²

^{1*} Adalet Bakanlığı, Adli Sicil ve İstatistik Genel Müdürlüğü, Ankara, Türkiye, (ORCID: 0000-0002-5286-1668), sumeyyebazna@gmail.com

² Fırat Üniversitesi, Mühendislik Fakültesi, Yazılım Mühendisliği Bölümü, Elazığ, Türkiye (ORCID: 0000-0001-9308-3500), sakyol@firat.edu.tr

(1st International Conference on Applied Engineering and Natural Sciences ICAENS 2021, November 1-3, 2021)

(DOI: 10.31590/ejosat.1012463)

ATIF/REFERENCE: Bazna, S. & Akyol, S. (2021). Global Optimizasyon Problemleri için Kaotik Bonobo Algoritması. *Avrupa Bilim ve Teknoloji Dergisi*, (28), 1028-1038.

Öz

Optimizasyon algoritmaları, global optimumdan feragat edilerek yaklaşık en iyi çözümü bulmayı amaçlayan algoritmalar. Bu çalışmada incelenen Bonobo Optimizasyon (BO) Algoritması ise sürü zekasına dayanan bir algoritma olup, bonoboların sosyal davranışlarının ve üreme stratejilerinin matematiksel modellenmesine dayanmaktadır. Bonobolar, yaşadıkları topluluk içinde çeşitli büyüklük ve özelliklerde gruplar oluşturarak, farklı amaçlar için birbirlerinden ayrılıp bir süre sonra yeniden bir araya gelmektedirler. Bonoboların üreme stratejisi incelendiğinde rasgele çiftleşme, kısıtlayıcı çiftleşme, konsorsiyum ve grup dışı çiftleşme gibi dört farklı stratejiyi benimsedikleri görülmektedir. Bonoboların bu doğal davranışları çeşitli optimizasyon problemlerini çözmek için kullanılmıştır. BO'yu diğer sürü zekasına dayalı algoritmalarından ayıran en önemli özelliği ise arama ajanlarının güncelleme mekanizmaları ve bunlarla ilişkili parametreler ve çiftleşme ortaklarının seçim yöntemidir. Bu çalışmada BO incelenip, algoritmada kullanılan parametreler Chebyshev, Circle, Gauss, Iterative, Logistic ve Tent kaotik haritaları kullanılarak yeniden üretilmiştir. Performansları karşılaştırmak için sekiz adet kalite testi fonksiyonu kullanılmıştır. Buna göre kaotik haritalar kullanılarak oluşturulan yeni algoritmalarından elde edilen sonuçların, klasik BO'ya göre daha verimli olduğu görülmüştür.

Anahtar Kelimeler: Bonobo Optimizasyonu, Kaotik Haritalar, Metasezgisel Algoritmalar.

Chaotic Bonobo Algorithm for Global Optimization Problems

Abstract

Optimization algorithms are algorithms that aim to find the approximate best solution by sacrificing the global optimum. The Bonobo Optimization Algorithm examined in this study is an algorithm based on herd intelligence and is based on mathematical modeling of bonobos' social behavior and reproductive strategies. Bonobos form groups of various sizes and characteristics within the community they live in, leaving each other for different purposes and reuniting after a while. When the reproductive strategy of bonobos is examined, it is seen that they adopt four different strategies such as random mating, restrictive mating, consortium and out-group mating. These natural behaviors of bonobos have been used to solve various optimization problems. The most important feature that distinguishes the Bonobo Optimization Algorithm from other algorithms based on swarm intelligence is the update mechanisms of the search agents and the parameters associated with them, and the selection method of mating partners. In this study, the Bonobo Algorithm was examined and the results obtained by reproducing the random parameters used in the algorithm with chaotic maps were evaluated. It was analyzed in this study and reconstructed with the Chebyshev, Circle, Gauss, Iterative, Logistics and Tent chaotic maps used here. Eight quality tests were used for their performance. Accordingly, chaotic maps seem to be more efficient than classical BO from the results obtained from the new models.

Keywords: Bonobo Optimization, Chaotic Maps, Metaheuristic Algorithms.

* Sorumlu Yazar: sumeyyebazna@gmail.com

1. Giriş

Optimizasyon, kelime anlamı olarak bir kaynağın ya da durumun en iyi ve en etkili şekilde kullanılması olarak tanımlanmıştır. Bilgisayar biliminin gelişmesiyle birlikte problemlerin karmaşıklığı da artmıştır. Çözümü çok zor olan yada çok fazla zaman gerektiren problemler için global optimumdan feragat edilerek yaklaşık çözümü bulmaya yarayan meta-sezgisel algoritmalar geliştirilmiştir. Doğada çeşitli ilke ve mekanizmalara bağlı olarak gerçekleşen bir çok olay vardır ve bu olayların matematiksel modellenmesiyle optimizasyon problemleri oldukça verimli şekillerde çözülebilmektedir [1].

Literatüre göre stokastik yöntemlerle ilgili teorik çalışmalar 3 ana başlık altında sınıflandırılmaktadır. Bunlar; var olan algoritmaların iyileştirilmesi, farklı algoritmaların hibritleştirilmesi ve yeni algoritmaların tanıtılmasıdır. Var olan algoritmaların iyileştirilmesinde araştırmacılar, değiştirilmiş ya da tamamen yeni operatörler ile mevcut algoritmanın performansını iyileştirmeye çalışmaktadırlar. Evrimsel operatörler [1], arama alanı azaltma tekniği [2] ve yeniden başlatma stratejileri [3] bu kategorinin örnekleri olarak sayılabilir. Farklı algoritmaların hibritleştirilmesi yönteminde, genel problemi çözmek adına farklı algoritmalar hibritleştirilmektedir [4]. Yeni algoritmaların tanıtımı ise araştırmacılar tarafından denenmiş ve başarılı sonuçlar almış algoritmaların tanıtılması ile ilgilidir. Önerilen tüm bu algoritmaların ortak özelliği ise evrimsel bir süreçten, topluluk halinde yaşayan canlıların sürü davranışlarından, insanların yaşam tarzlarından ya da çeşitli doğal olayların matematiksel modellenmesi ile ortaya çıkmış olmalarıdır [1].

Biyolojik evrimden ilham alınarak Genetik Algoritma (GA) [5] ve Diferansiyel Gelişim Algoritması (DE) [6] gibi çeşitli Evrimsel Algoritmalar (EA) geliştirilmiştir. Yine sürü zekasına dayalı Parçacık Sürü Optimizasyonu (PSO) [7] ve Yarasa Algoritması (BA) [8] geliştirilmiştir. Temelde geliştirilen bu algoritmaların tamamı kuş sürüsü, karınca sürüsü, balık sürüsü ya da virüs-bakteri büyümesi gibi canlıların sürü davranışlarının taklit edilmesi sonucunda geliştirilmiştir. İnsan davranışlarının konu alan algoritmaların ise saydığımız diğer algoritmalara kıyasla daha karmaşık ve üstün olması beklenmektedir [9]. Bu nedenle, Beyin Fırtınası Optimizasyonu (BSO) [9], Öğretme-Öğretme Temelli Algoritma (TLBO) [10] gibi insan davranışları temel alınarak geliştirilen algoritmaların daha büyük ölçekli problemleri çözebilmesi beklenmektedir. Literatürde önemli sayıda optimizasyon problemi meta-sezgisel algoritmalar yardımıyla çözülmektedir. Bu durumda şu soru gündeme gelmektedir: Daha fazla optimizasyon algoritmasına ihtiyaç var mıdır? Varsa neden? Bu sorunun cevabı "No Free Lunch" diye bilinen teoreme verilmektedir [11]. Bu teoreme göre hiçbir optimizasyon probleminin eşit kolaylık ve verimlilikte çözülemeyeceği ispatlanmıştır. Bir algoritmanın belirli bir problemdeki üstün performansını başka bir problemde garanti etmemesi ile bu ispatın sağlanması yapılabilmektedir. Dolayısıyla bu teorem farklı problemlerin farklı optimizasyon algoritmaları ile çözüleceğini savunmaktadır. Meta-sezgisel algoritmaların çoğunun doğası gereği uyarlanabilir kontrol parametrelerine sahip olması gerektiği görülmektedir yani algoritmanın daha verimli çalışması için o algoritmaya özgü parametrelerin evrim sırasında kendisini güncellemesi gerekmektedir [1].

Bu gerçeklerden hareketle BO geliştirilmiştir. BO, bonoboların sosyal yaşamlarından ve üreme stratejilerinden ilham

alınarak geliştirilen bir algoritmadır. Bonoboların bu doğal davranışları matematiksel olarak modellenmiş ve kullanılan kontrol parametrelerin doğası gereği uyarlanabilir olduğu bulunmuştur.

Literatür incelendiğinde, Abdelghany vd. çalışmalarında, BO'nun geliştirilmiş bir versiyonunu güneş pillerinin farklı matematiksel modellerinin bilinmeyen parametrelerinin doğru değerlerini elde etmek için kullanmışlardır ve başarılı sonuçlar elde etmişlerdir [12]. Sultan vd. çalışmalarında farklı ticari PEMFC yığınlarının bilinmeyen parametrelerini belirlemek için BO kullanmışlardır ve elde edilen sonuçlar iyi bilinen optimizasyon algoritmalarıyla karşılaştırılmıştır [13]. Das ve Pratihari, BO'nun performansını, CEC'13 ve CEC'14 test fonksiyonlarında test ederek son zamanların diğer verimli ve popüler optimizasyon algoritmaları ile karşılaştırmışlardır ve önerilen BO'nun sonuçlarından istatistiksel olarak üstün performans elde etmişlerdir [14]. Das ve Pratihari bir diğer çalışmada ise makinelerin daha iyi üretim hızı ve genel bakım maliyetinin en aza indirilmesi için önleyici bakım aralığını genetik algoritma (GA), parçacık sürü optimizasyonu (PSO) ve BO ile optimize etmişlerdir ve BO'nun GA ve PSO'ya kıyasla daha iyi sonuçlar verdiğini gözlemlemişlerdir [15]. Das vd. çalışmada farklı optimizasyon problemlerini çözmek için BO'nun üç farklı çok amaçlı versiyonunu önermişlerdir ve bu üç farklı versiyondan elde edilen sonuçları karşılaştırmışlardır [16].

Meta-sezgisel algoritmaların performanslarını arttırmak için kaotik haritalar kullanılabilir. Bu çalışmada BO'nun performansını arttırmak için Chebyshev, Circle, Gauss, Iterative, Logistic ve Tent kaotik haritaları kullanılarak KBO1 ve KBO2 algoritmaları önerilmiştir. Bu algoritmaların klasik BO'ya göre performanslarını test etmek için iki tane unimodal (Sphere ve Rosenbrock), iki tane multimodal (Schwefel ve Ackley), iki tane fixed-dimensional multimodal (Foxholes ve Kowalik) ve iki tane de Cec'2019 (Storn's Chebyshev Polynomial Fitting Problem ve Lennard-Jones Minimum Energy Cluster) kalite testi fonksiyonları kullanılmıştır. Elde edilen sonuçlara göre Kaotik BO algoritmalarının, klasik BO'ya göre daha iyi sonuçlar verdiği gözlemlenmiştir.

Makalenin ikinci bölümünde BO'nun matematiksel modeli açıklanmıştır ve algoritmanın adımları anlatılarak akış diyagramı verilmiştir. Üçüncü bölümde, kaotik BO algoritması anlatılmıştır. Kullanılan kaotik haritalar hakkında bilgi verilmiştir. Dördüncü bölümde deneysel sonuçlar tablo ve grafik halinde sunulmuş, açıklanmıştır. Beşinci bölümde ise sonuçlar üzerinde tartışılmıştır.

2. Materyal ve Metot

2.1. BO Algoritmasının Matematiksel Modellemesi

BO'da, problemin çözümü için bonoboların sosyal davranışları ve üreme stratejileri matematiksel olarak modellenmektedir. Diğer popülasyon tabanlı algoritmalarla benzer şekilde BO da sabit bir popülasyon büyüklüğüne sahiptir ve popülasyonun başlatılması rasgele gerçekleşmektedir. Algoritmada popülasyondaki her bir çözüm bonobo olarak adlandırılmaktadır ve tüm bonoboların uygunluk değerleri karşılaştırılmaktadır. Bir bonobo topluluğunda baskınlık hiyerarşisinde en iyi sıraya sahip olan bonobo, alfa bonobo olarak adlandırılmaktadır ve alfa bonobo en iyi uygunluk değerine sahip mevcut en iyi çözüm olarak kabul edilmektedir. Bununla beraber, BO'nun kullanıcı tanımlı olmayan parametreleri belirlenen başlangıç değerleri ile başlatılmaktadır. BO'nun bir diğer önemli

özelliği ise algoritmada kullanılan bütün rasgele sayıların 0.0 ile 1.0 arasında oluşturulmasıdır [1].

2.1.1. BO'nun Kullanıcı Tanımlı Olmayan Parametrelerinin Başlatılması

Algoritmada kullanılan pozitif faz sayısı (ppc), negatif faz sayısı (npc), faz değişikliği (cp), ekstra grup eşleşme olasılığı (P_{xgm}), geçici alt grup boyut faktörü ($tsgs_{factor}$), faz olasılığı (P_p) ve yönlü olasılık (P_d) aşağıdaki gibi başlatılmaktadır:

$$\begin{aligned} ppc &= 0, \\ npc &= 0, \\ cp &= 0, \\ P_{xgm} &= P_{xgm_initial}, \\ tsgs_{factor} &= tsgs_{factor_initial}, \\ P_p &= 0.5, \\ P_d &= 0.5 \end{aligned}$$

$P_{xgm_initial}$ ve $tsgs_{factor_initial}$ sırasıyla P_{xgm} ve $tsgs_{factor}$ 'un başlangıç değerleridir.

2.1.2. Pozitif Faz-Negatif Faz

Yeterli yiyecek, yüksek çiftleşme oranı, korunma ve barışçıl yaşam koşullarının olduğu durumlar pozitif faz durumlarıdır. Geliştirilen algoritmada en iyi çözümde (α_{bonobo}) olumlu bir gelişme gözlemlendiğinde bu aşamaya pozitif faz (PP) denmektedir. Elde edilen en iyi çözümde (α_{bonobo}) herhangi bir gelişme gözlenmemesi durumunda ise negatif fazdan (NP) bahsedilmektedir. Algoritma pozitif fazdan geçiyorsa, pozitif faz sayısı (ppc) bir artırılmaktadır. Benzer şekilde algoritma negatif fazdan geçiyorsa, negatif faz sayısı (npc) bir artırılmaktadır. Hem npc hem de ppc 'nin başlangıç değerleri 0 olarak başlatılmakta ve parametrelerin herhangi biri artırıldığında, diğer parametrenin değeri 0'a eşitlenmektedir. Burada önemli olan nokta bir bonobo topluluğunda alfa bonobonun statüsü zaman zaman değişikliğe tabi tutulabilmektedir. Yani bir alfa bonobonun sonsuza kadar alfa bonobo olarak kalacağına garantisizdir. Bir bonobonun mevcut alfa bonoboya kıyasla daha güçlü olduğu tespit edilirse, hak eden bonoboya alfa bonobo statüsü verilmektedir [1].

2.1.3. Fisyon-Füzyon Sosyal Stratejisini Kullanan Bonobonun Seçimi

Bonoboların benimsedikleri 4 farklı üreme stratejisi vardır. Bu stratejiler; rasgele çiftleşme, kısıtlayıcı çiftleşme, konsorsiyum ve grup dışı çiftleşmedir [17]. Algoritmanın geçtiği fazın durumuna bağlı olarak, bu 4 stratejiden biri kullanılmaktadır. Örnek olarak pozitif faz (PP) senaryosunu ele alacak olursak:

PP fiziksel olarak yeterli gıdanın olduğu, diğer topluluklardan korunmanın, çiftleşme başarısının ve bonobolar arasında genetik olarak farklılıkların olduğu durumları temsil etmektedir. Böyle bir durumda rasgele çiftleşme ve kısıtlayıcı çiftleşme olasılığı daha yüksek olmaktadır. Rasgele çiftleşmede hem alfa bonobo hem de alt sıradaki erkek bonobolar çiftleşmeye katılabilmektedir. Ancak kısıtlayıcı çiftleşmeye alfa bonobo ile beraber sadece en iyi uygunluk değerlerine sahip olan bonobolar katılabilmektedir. Olumsuz ortam şartlarını temsil eden negatif faz (NP) durumunda ise konsorsiyum ve grup dışı çiftleşme olasılıkları daha yüksektir [1].

2.1.4.1. Rasgele ve Kısıtlayıcı Çiftleşme Stratejileri

Yeni bir bonobonun yaratılmasına çiftleşme stratejisine karar verilerek başlanmaktadır. Bu durum için faz olasılığı denen (P_p) bir parametre tanımlanmaktadır. P_p 'nin başlangıç değeri 0.5 olarak ayarlanmış olup her iterasyonda güncellenmektedir. 0 ile 1 arasında üretilen rasgele bir sayının (say, r) P_p 'den küçük veya eşit olması durumunda Denklem 2'den yararlanılarak yeni bonobo yaratılmaktadır.

$$new_bonobo_j = bonobo_j^i + r1 * scab * (\alpha_j^{bonobo} - bonobo_j^i) + (1 - r1) * scsb * flag * (bonobo_j^i - bonobo_j^p) \quad (2)$$

Burada, new_bonobo_j yeni neslin j 'nci değişkeni, α_j^{bonobo} alfa bonobonun j 'nci değişkeni, $bonobo_j^i$ i 'nci bonobonun j 'nci değişkeni, $bonobo_j^p$ p 'nci bonobonun j 'nci değişkeni, $r1$ 0 ile 1 arasında rasgele sayı, $scab$ α_{bonobo} için paylaşım katsayısı, $scsb$ seçilen p 'nci bonobo için paylaşım katsayısı olmaktadır ve $flag$ ise sadece 1 veya -1 değeri alabilir.

i 'nci bonobonun uygunluk değerinin p 'nci bonobonunkinden daha iyi olduğu görülüyorsa, (yani α_{bonobo} ile birlikte diğer bonoboların da çiftleşmeye katılabileceği bir rasgele çiftleşme durumudur.) $flag$ 1'e eşitlenmektedir. Çiftleşmeye sadece α_{bonobo} ile birlikte daha yüksek rütbeli bonoboların katıldığı durumda $flag$ -1'e eşitlenmektedir (kısıtlayıcı çiftleşme durumu).

2.1.4.2. Konsorsiyum ve Grup Dışı Çiftleşme Stratejileri

Bu tür eşleşmeler rasgele sayının faz olasılığından büyük olduğu durumlarda görülmektedir. ($r > P_p$) Ayrıca (0,1) aralığında oluşturulan (say, r_2), grup dışı eşleşme olasılığından (P_{xgm}) küçük veya eşitse, çözüm güncellemesi ekstra grup eşleşmesi ile gerçekleştirilmektedir.

- α_j^{bonobo} 'nun j 'nci değişkeni $\geq bonobo_j^i$ 'nin i 'nci değişkeni ise Denklem 3 ve Denklem 4 kullanılmaktadır.

$$\beta_1 = e^{(r_2^2 + r_4 - \frac{2}{r_4})} \quad (3)$$

$$new_bonobo_j = bonobo_j^i + \beta_1 \times (Var_max_j - bonobo_j^i) \quad (4)$$

- $(say, r_3) \leq P_d$ ise Denklem 5 ve Denklem 6 kullanılmaktadır.

2.1.5. Değişken Sınırlama Koşulları

$$\beta_2 = e^{(-r_4^2 + 2r_4 - \frac{2}{r_4})} \quad (5)$$

$$new_bonobo_j = bonobo_j^i - \beta_2 \times (bonobo_j^i - Var_min_j) \quad (6)$$

- $\alpha_j^{bonobo} < bonobo_j^i$ ise Denklem 3 ve Denklem 7 kullanılmaktadır.

$$new_bonobo_j = bonobo_j^i - \beta_1 \times (bonobo_j^i - Var_min_j) \quad (7)$$

- $r3 \leq P_d$ veya $r3 > P_d$ ise Denklem 5 ve Denklem 8 kullanılmaktadır.

$$new_bonobo_j = bonobo_j^i + \beta_2 \times (Var_max_j - bonobo_j^i) \quad (8)$$

P_d 'nin başlangıç değeri 0.5 olarak ayarlanmaktadır ve daha sonra evrimin durumuna göre güncellenmektedir. β_1 ve β_2 new_bonobo_j değerini hesaplamak için iki ara parametredir. j , 1 ile d arasında değişmektedir (toplam karar değişkenlerinin sayısı). $r4$, 0 ile 1 arasında rasgele sayıdır (ancak 0 olamaz). Var_max_j ve Var_min_j sırasıyla j 'ninci değişkene karşılık gelen üst ve alt sınır değerleridir.

$r2 > P_{xgm}$ durumunda konsorsiyum çiftleşme stratejisi Denklem 9'daki gibi kullanılarak yeni bonobolar yaratılmaktadır.

$$new_bonobo_j = \begin{cases} bonobo_j^i + flag \times e^{-r5} \times (bonobo_j^i - bonobo_j^p) \\ \text{if } (flag = 1 || r6 \leq P_d), \\ bonobo_j^p, \text{ otherwise} \end{cases} \quad (9)$$

Burada $r5$ ve $r6$ rasgele iki sayı ve P_d ise yönlü olasılıktır. Özetle; pozitif evrede (PP) bonoboların, alfa bonoboya doğru hareket etme olasılıklarının daha fazla olduğu, evrimin negatif evresinde (NP) rasgele hareket eğiliminin daha fazla olduğu gözlemlenmektedir.

2.1.5. Değişken Sınırlama Koşulları

Eğer yavru, new_bonobo_j 'nin Var_max_j değerinden daha büyük olduğu bulunursa, new_bonobo_j 'ye Var_max_j 'ninkine eşit bir değer atanmaktadır.

Benzer şekilde, new_bonobo_j 'nin Var_min_j değerinden daha az olduğu görüldüğünde, new_bonobo_j 'nin değeri Var_min_j 'ye eşit alınmaktadır.

2.1.6. Yavruların Kabul Kriterleri

new_bonobo_j 'nin uygunluk değeri $bonobo^i$ 'ninkinden daha iyi bulunursa veya rasgele sayı P_{xgm} 'den küçük veya eşitse, yavru kabul edilmektedir ve $bonobo^i$ olmaktadır. Bu durumda popülasyonda new_bonobo_j 'nin yerini almaktadır. Yeni bonobonun uygunluğunun alfa bonobonunkinden daha iyi olduğu görülüyorsa, yeni bonobo alfa bonobo olarak bildirilmektedir.

2.1.7. Parametrelerin Güncellenmesi

Mevcut yinelemeden yeni elde edilen alfa bonobonun, önceki yinelemeye kıyasla uygunluk değeri açısından daha iyi bir çözüm olduğu görülürse, BO parametreleri aşağıdaki gibi güncellenmektedir:

$$\begin{aligned} npc &= 0, \\ ppc &= ppc + 1, \\ cp &= \text{minimum}(0.5, ppc \times rcpp), \\ P_{xgm} &= P_{xgm_initial}, \\ P_p &= 0.5 + cp, \\ P_d &= P_p, \\ tsgs_{factor} &= \text{minimum}(tsgs_{factor_max}, \\ &\quad (tsgs_{factor_initial} + ppc \times rcpp^2)) \end{aligned}$$

Burada $rcpp$ ve cp sırasıyla faz olasılığındaki değişim oranı ve fazdaki değişim miktarıdır. Faz olasılığının başlangıç değerinin, başlangıçta her iki faza da eşit önem vermek için 0.5 olarak seçildiğine dikkat edilmelidir. Ancak, faz olasılığının bu değeri, amaç fonksiyonunun ve algoritmanın davranışına göre

iterasyonların ortaya çıkmasıyla değişecektir. ($tsgs_{factor_max}$), geçici sosyal grup boyut faktörünün ($tsgs_{factor}$) maksimum değeridir ve kullanıcı tarafından algoritmanın başında ayarlanmaktadır. ($P_{xgm_initial}$) ve ($tsgs_{factor_initial}$) sırasıyla grup dışı çiftleşme olasılığının (P_{xgm}) ve geçici alt grup boyut faktörünün ($tsgs_{factor}$) başlangıç değerleridir.

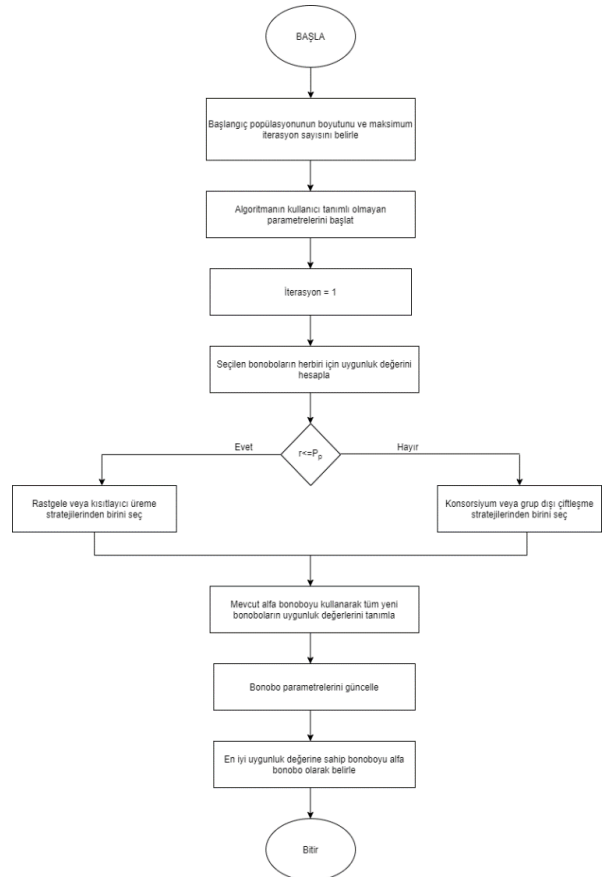
Burada ($P_{xgm_initial}$), algoritmanın kullanıcı tanımlı bir parametresidir. Ancak, ($tsgs_{factor_initial}$) Denklem 10 kullanılarak hesaplanır.

$$tsgs_{factor_initial} = 0.5 \times tsgs_{factor_max} \quad (10)$$

Başka bir durumda, mevcut yinelemede, alfa bonobonun uygunluk değerinde öncekine kıyasla herhangi bir gelişme olmazsa, parametreler aşağıdaki gibi güncellenmektedir:

$$\begin{aligned} ppc &= 0, npc = npc + 1, \\ cp &= -\text{minimum}(0.5, npc \times rcpp), \\ P_{xgm} &= \text{minimum}(0.5, (P_{xgm_initial} + npc \\ &\quad \times rcpp^2)), \\ P_p &= 0.5 + cp, \\ P_d &= P_p, \\ tsgs_{factor} &= \text{maximum}(0, (tsgs_{factor_initial} - npc \\ &\quad \times rcpp^2)). \end{aligned}$$

Bonobo optimizasyon algoritmasının akış diyagramı Şekil 1'de verilmiştir.



Şekil 1. BO'nun akış diyagramı

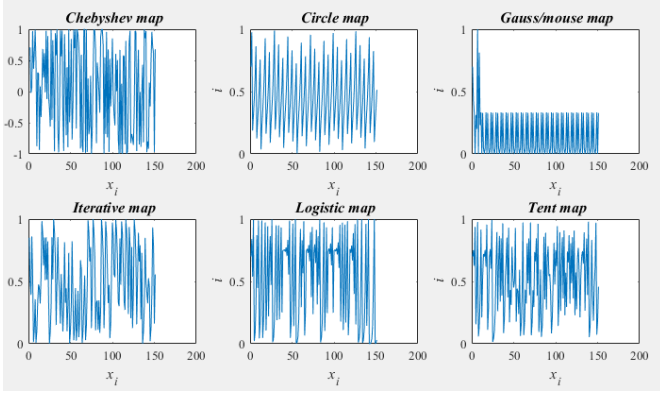
2.2. Kaotik Bonobo Optimizasyonu

Bu çalışmada BO'nun iki farklı kaotik versiyonu KBO1 ve KBO2 önerilmektedir. KBO1'de alfa bonobonun paylaşım katsayısı sabiti scab sabiti için, KBO2'de ise seçilen bonobonun paylaşım katsayısı olan scsb sabiti için kaotik sayılar üretilmiştir.

Kullanılan haritalar Tablo 1'de sunulmaktadır, Şekil 2'de ise kullanılan kaotik haritalar görselleştirilmiştir.

Tablo 1 Kullanılan kaotik haritalar

Kaotik harita	Formül
Circle map	$X_{n+1} = X_n + b - \left(\frac{a}{2\pi}\right) \sin(2\pi X_n) \text{ mod } 1$
Gauss map	$x_{n+1} = \exp(-\alpha x_n^2) + \beta_1$
Chebyshev map	$T_n(x) = \cos(n \times \arccos(x))$
Logistic map	$X_{n+1} = rX_n(1 - X_n)$
Tent map	$X_{n+1} = \mu \left(1 - 2 \left X_n - \frac{1}{2}\right \right)$
Iterative map	$x_{i+1} = \sin\left(\frac{a\pi}{x_i}\right), a = 0.7$



Şekil 2. Kullanılan Kaotik haritaların görselleştirilmesi

2.2.1. Alfa Bonobonun Paylaşım Katsayısı için Kaotik Haritalar

Rasgele ve kısıtlayıcı çiftleşme stratejisinde, yeni bir bonobonun yaratılmasında kullanılan alfa bonobo için paylaşım katsayısı olan scab değeri Şekil 3'teki gibi elde edilmiştir. Kullanılan kaotik haritalar 0 ile 1 arasında değerler üretmektedir. scab değeri 1 ile 2 arasında olması gerektiği için x(i+1) değerine 1 eklenmiştir.

```
x1=0.7
for i=1:fonskyion deęerlendirme sayısı
...
x(i+1)=seçilen kaotik haritası
scab=(x(i+1))+1
...
end
```

Şekil 3 scab değerinin kaotik haritalar kullanılarak elde edilmesi

2.2.2. Seçilen Bonobonun Paylaşım Katsayısı için Kaotik Haritalar

Rasgele ve kısıtlayıcı çiftleşme stratejisinde, yeni bir bonobonun yaratılmasında kullanılan rasgele seçilen bonobo için

paylaşım katsayısı olan scsb değeri Şekil 5'teki gibi elde edilmiştir.

```
x1=0.7
for i=1:fonskyion deęerlendirme sayısı
...
x(i+1)=seçilen kaotik haritası
scsb=(x(i+1))+1
...
end
```

Şekil 4 scsb değerinin kaotik haritalar kullanılarak elde edilmesi

3. Deneysel Sonuçlar

Bu algoritmaların klasik BO'ya göre performanslarını test etmek için unimodal test fonksiyonlarından Sphere ve Rosenbrock, multimodal test fonksiyonlarından Schwefel ve Ackley, fixed-dimension multimodal test fonksiyonlarından Foxholes ve Kowalik ve Cec'2019 test fonksiyonlarından Storn's Chebyshev Polynomial Fitting Problem ve Lennard-Jones Minimum Energy Cluster kullanılmıştır. Bu fonksiyonların açıklaması Tablo 2'de verilmiştir.

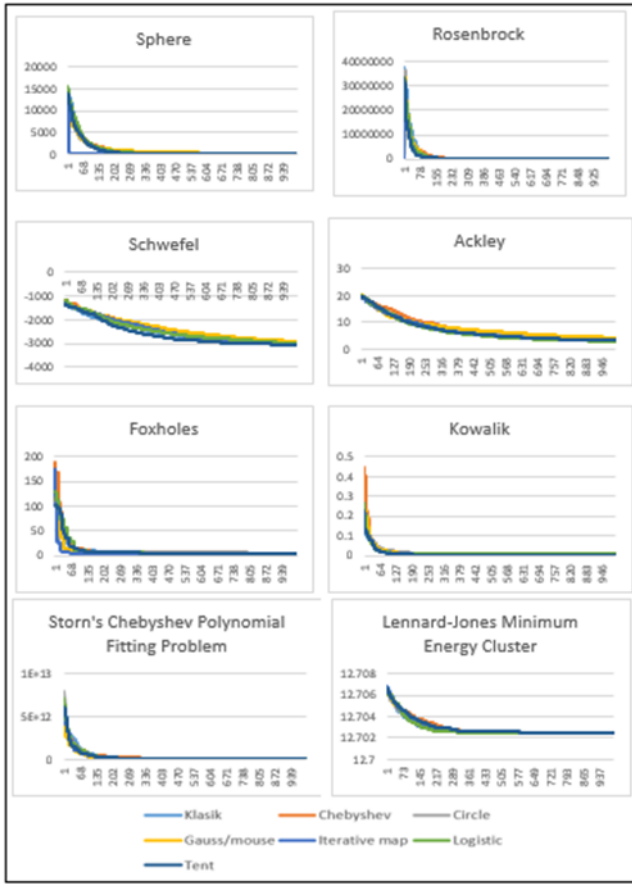
Tablo 2. Kullanılan kalite testi fonksiyonları

Name	Function Equation	Parameters	MIN	Problem Size
Sphere	$f(x) = \sum_{i=1}^N x_i^2$	$-100 < x_i < 100$	0	10
Rosenbrock	$f(x) = \sum_{i=1}^{N-1} [100(x_i^2 - x_{i+1})^2 + (1 - x_i)^2]$	$-30 < x_i < 30$	0	10
Schwefel	$f(x) = \sum_{i=1}^N (-x_i \sin(\sqrt{ x_i }))$	$-500 < x_i < 500$	-418.9829 * n	10
Ackley	$f(x) = -20 \exp \left[-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right] - \exp \left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i) \right) + e + 20$	$-32 < x_i < 32$	0	10
Foxholes	$f(x) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{25} (x_i - a_{i,j})^6} \right)^{-1}$	$-65.536 < x_i < 65.536$	1	2
Kowalik	$f(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]^2$	$-5 < x_i < 5$	0.00030	4
Storn's Chebyshev Polynomial Fitting Problem	$f(x) = p_1 + p_2 + p_3$ $p_1 = \begin{cases} (u - d)^2 & \text{if } u < d, \\ 0 & \text{otherwise;} \end{cases} \quad u = \sum_{j=1}^D x_j (1.2)^{D-j}$ $p_2 = \begin{cases} (v - d)^2 & \text{if } v < d, \\ 0 & \text{otherwise;} \end{cases} \quad v = \sum_{j=1}^D x_j (-1.2)^{D-j}$ $p_k = \begin{cases} (w_k - 1)^2 & \text{if } w_k > 1 \\ (w_k + 1)^2 & \text{if } w_k < 1 \\ 0 & \text{otherwise;} \end{cases} \quad w_k = \sum_{j=1}^D x_j \left(\frac{2k}{m} - 1 \right)^{D-j}$ $p_3 = \sum_{k=0}^m p_k, \quad k = 0, 1, \dots, m, \quad m = 32D.$	$-8192 < x_i < 8192$	1	9
Lennard-Jones Minimum Energy Cluster	$d = 72.661$ $f(x) = 12.7120622568 + \sum_{i=1}^{n-1} \sum_{j=i+1}^n \left(\frac{1}{d_{i,j}^2} - \frac{2}{d_{i,j}} \right),$ $d_{i,j} = \left(\sum_{k=0}^2 (x_{3i+k-2} - x_{3j+k-2})^2 \right)^3, \quad n = D/3$	$-4 < x_i < 4$	1	18

Alfa bonobo için paylaşım katsayısı olan *scab* değeri üretilirken kullanılan kaotik haritaların uygunluk değerlerinin değişimi Şekil 4'te verilmiştir. KBO1 için Chebyshev, Gauss, Logistic, Circle,

Iterative ve Tent kaotik haritaları kullanılarak üretilen *scab* değeri Sphere, Schwefel, Foxholes, Rosenbrock, Ackley, Kowalik, Storn's Chebyshev Polynomial Fitting Problem ve Lennard-

Jones Minimum Energy Cluster olmak üzere toplam 8 tane kalite testi fonksiyonu ile test edilmiştir. Elde edilen test sonuçları Tablo 3'te verilmiştir.



Şekil 4 Fonksiyon değerlendirme sayısına göre KBO1 algoritmasının uygunluk değerinin değişimi

Tablo 3. KBO1 için istatistiksel sonuçlar

		Klasik	Chebyshev	Circle	Gauss	Iterative	Logistic	Tent
Sphere	Min	1,63E-02	1,19E-01	9,25E-02	7,56E+00	3,78E-02	1,50E-02	1,37E-02
	Maks	1,31E+01	1,65E+01	2,58E+01	3,56E+02	5,36E+01	2,86E+00	1,68E+01
	Ortalama	2,10E+00	4,65E+00	6,43E+00	1,07E+02	3,65E+00	6,27E-01	1,47E+00
	Std. Sapma	3,231904101	4,893628745	6,684469893	101,6747831	11,85851553	0,794586764	3,664736117
Rosenbrock	Min	5,22E+00	2,91E+01	1,59E+01	1,62E+02	6,74E+00	8,78E+00	1,20E+01
	Maks	3,81E+03	1,85E+03	2,79E+03	1,11E+05	2,20E+03	4,22E+02	2,67E+03
	Ortalama	6,66E+02	3,18E+02	4,44E+02	1,89E+04	3,22E+02	1,59E+02	5,41E+02
	Std. Sapma	1124,037408	428,0650395	679,0225373	32545,37764	480,2474793	118,3905132	702,5170428
Schwefel	Min	-3596,66	-3493,51	-3606,8	-3408,66	-3695,49	-3647,77	-3599,91
	Maks	2,45E+03	-2,41E+03	2,42E+03	-2,60E+03	2,32E+03	2,47E+03	-2458,04
	Ortalama	-3021,46	-3055,24	-3005,66	-2924,01	-3089,56	-3018,92	-3088,48
	Std. Sapma	3,12E+02	3,46E+02	3,14E+02	2,24E+02	3,51E+02	2,95E+02	321,0434
Ackley	Min	3,84E-01	1,23E+00	2,05E+00	2,17E+00	1,914171	0,166056	1,698925
	Maks	6,09E+00	5,72E+00	6,54E+00	6,95E+00	7,288041	6,635179	5,995996
	Ortalama	3,46E+00	3,15E+00	3,73E+00	4,34E+00	3,082541	2,986883	3,209781
	Std. Sapma	1,591398	1,251946	1,163788	1,430327	1,321904	1,407412	0,986543

Foxholes	Min	0,998004	0,998004	0,998004	0,998004	0,998004	0,998004	0,998004
	Maks	15,50382	10,76318	18,30431	15,50382	12,67051	15,50382	12,67051
	Ortalama	4,179932	2,836754	6,065021	4,284551	1,976101	4,774352	5,217781
	Std. Sapma	4,200027	2,56514	5,509092	3,792022	2,864698	4,103282	3,713161
Kowalik	Min	0,000327	0,000566	0,000382	0,000563	0,000311	0,000409	0,00044
	Maks	0,020363	0,020363	0,020364	0,022573	0,020363	0,020363	0,022073
	Ortalama	0,009659	0,007789	0,007729	0,005491	0,003869	0,008611	0,003295
	Std. Sapma	0,009745	0,009473	0,009519	0,007962	0,007118	0,009848	0,006354
Storn's Chebyshev Polynomial Fitting Problem	Min	2,39E+08	1,17E+09	2,17E+08	2,91E+09	6,35E+08	5,71E+08	3,16E+08
	Maks	1,55E+11	1,05E+11	1,1E+11	5,52E+10	3,93E+10	9,68E+10	1E+11
	Ortalama	1,79E+10	2,79E+10	2E+10	1,7E+10	1,13E+10	2,13E+10	1,23E+10
	Std. Sapma	3,58E+10	2,95E+10	2,67E+10	1,53E+10	1,05E+10	2,87E+10	2,26E+10
Lennard-Jones Minimum Energy Cluster	Min	1,27E+01	1,27E+01	1,27E+01	1,27E+01	1,27E+01	1,27E+01	1,27E+01
	Maks	1,27E+01	1,27E+01	1,27E+01	1,27E+01	1,27E+01	1,27E+01	1,27E+01
	Ortalama	1,27E+01	1,27E+01	1,27E+01	1,27E+01	1,27E+01	1,27E+01	1,27E+01
	Std. Sapma	4,25E-06	1,95E-05	3,69E-06	3,26E-05	3,76E-07	0,000204	2,26E-05

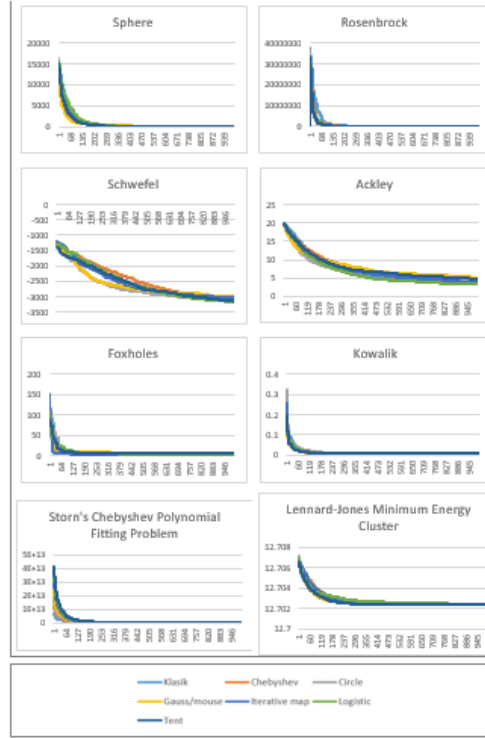
Tablo 3'teki bulgular incelendiğinde en iyi minimum değerleri; Sphere test fonksiyonuna göre Tent map, Rosenbrock test fonksiyonuna göre klasik bonobo algoritması; Schwefel ve Kowalik test fonksiyonlarına göre Iterative map, Ackley test fonksiyonuna göre Logistik map, Storn's Chebyshev Polynomial Fitting Problem test fonksiyonuna göre Circle map vermiştir. Foxholes ve Lennard- Jones Minimum Energy Cluster test fonksiyonuna göre birbirine çok yakın minimum sonuçlar bulunmuştur.

Aynı şekilde Tablo 3'e göre en iyi ortalama değerleri; Sphere, Rosenbrock ve Ackley test fonksiyonlarına göre Logistic, Schwefel, Foxholes ve Storn's Chebyshev Polynomial Fitting Problem test fonksiyonlarına göre Iterative map, Kowalik test

fonksiyonuna göre Tent map vermiştir. Lennard- Jones Minimum Energy Cluster test fonksiyonuna göre birbirine çok yakın minimum sonuçlar bulunmuştur.

Seçilen bonobo için paylaşım katsayısı olan *scsb* değeri üretilirken kullanılan kaotik haritaların uygunluk değerlerinin değişimi Şekil 6'da verilmiştir. KBO2 için Chebyshev map, Gauss map, Logistic map, Circle map, Iterative map ve Tent kaotik haritaları kullanılarak üretilen *scsb* değeri Sphere, Schwefel, Foxholes, Rosenbrock, Ackley, Kowalik, Storn's Chebyshev Polynomial Fitting Problem ve Lennard- Jones Minimum Energy Cluster olmak üzere toplam 8 tane kalite testi fonksiyonu ile test edilmiştir. Elde edilen test sonuçları Tablo 4'te verilmiştir.

Şekil 6 *scsb* değerinin kaotik haritalar kullanılarak elde edilmesi



Tablo 4. KBO2 için istatistiksel sonuçlar

		Klasik	Chebyshev	Circle	Gauss	Iterative	Logistic	Tent
Sphere	Min	0,016329708	0,154117853	0,251088192	2,859465646	0,025357535	0,012869884	0,011770677
	Maks	13,13547834	38,59750631	127,7587614	334,0771476	5,325510953	33,73332785	33,59947458
	Ortalama	2,101447548	4,230725927	21,96240175	69,17022755	1,32075202	3,33770931	3,839655753
	Std. Sapma	3,231904101	8,591700517	32,57699467	83,3405498	1,58289825	8,27272374	7,744874573
Rosenbrock	Min	5,216308756	13,51794909	7,896805802	237,126786	5,423793267	11,98071059	20,16582213
	Maks	3808,664931	9151,345743	4599,29451	63584,87095	19454,93423	3614,203656	1874,322702
	Ortalama	666,0955666	939,9375578	814,1178184	11970,57908	1293,475353	343,1381159	404,4154969
	Std. Sapma	1124,037408	2047,69443	1302,642058	13948,39756	4300,654478	812,5541485	561,9078643
Schwefel	Min	-3596,66	-3521,49	-3471,65	-3716,5	-3752,1	-3568,78	-3768,77
	Maks	-2449,66	-2568,83	-2616,74	-2233,49	-2386,61	-2740,82	-2498,76
	Ortalama	-3021,46	-3066,09	-3145,89	-2979,69	-3046,51	-3165,84	-3147,03
	Std. Sapma	311,6576	217,7573	212,0611	371,7339	369,6563	259,3259	342,0656
Ackley	Min	3,84E-01	1,32E+00	1,909969	3,341026	0,048592	0,139737	1,853678
	Maks	6,09E+00	5,72E+00	11,34514	7,613668	8,289261	6,767626	7,900858
	Ortalama	3,46E+00	3,25E+00	4,221874	5,1559	3,849412	3,265209	4,534949
	Std. Sapma	1,591398	1,015976	2,106425	1,288601	1,660481	1,450572	1,857888
Foxholes	Min	0,998004	0,998004	0,998004	1,992031	0,998004	0,998004	0,998004
	Maks	15,50382	15,50382	21,07269	17,37441	7,873993	10,76318	12,67051
	Ortalama	4,179932	4,627346	5,868512	6,9052	1,932364	3,746592	4,973943
	Std. Sapma	4,200027	3,8596	5,564256	4,909257	2,289157	3,62708	3,930296
Kowalik	Min	0,000327	0,000363	0,000377	0,000651	0,000493	0,000583	0,000309
	Maks	0,020363	0,020363	0,021016	0,028218	0,020363	0,020363	0,022276
	Ortalama	0,009659	0,003643	0,006164	0,007935	0,007544	0,008598	0,008734

	Std. Sapma	0,009745	0,00721	0,008542	0,010245	0,009309	0,009723	0,009954
Storn's Chebyshev Polynomial Fitting Problem	Min	2,39E+08	1,12E+09	1,43E+09	1,76E+09	2,44E+08	9,16E+08	1,83E+09
	Maks	1,55E+11	7,07E+10	7,58E+10	3,02E+11	5,14E+10	1,31E+11	1,43E+11
	Ortalama	1,79E+10	1,71E+10	1,89E+10	8,79E+10	1,29E+10	2,19E+10	1,78E+10
	Std. Sapma	3,58E+10	1,93E+10	2,05E+10	9,34E+10	1,32E+10	3,1E+10	3,11E+10
Lennard-Jones Minimum Energy Cluster	Min	12,7024	12,7024	12,7024	12,7024	12,7024	12,7024	12,7024
	Maks	12,70242	12,70241	12,7025	12,70245	12,70252	12,70242	12,7024
	Ortalama	12,70241	12,7024	12,70241	12,70241	12,70241	12,7024	12,7024
	Std. Sapma	4,25E-06	2,53E-07	2,72E-05	1,2E-05	2,5E-05	3,37E-06	4,52E-08

Tablo 4'teki bulgular incelendiğinde en iyi minimum değerleri; Sphere, Schwefel ve Kowalik test fonksiyonlarına göre Tent map, Rosenbrock ve Storn's Chebyshev Polynomial Fitting Problem test fonksiyonuna göre klasik bonobo algoritması, Ackley test fonksiyonuna göre Iterative map, Foxholes test fonksiyonuna göre Gauss map vermiştir. Lennard-Jones Minimum Energy Cluster test fonksiyonuna göre birbirine çok yakın minimum sonuçlar bulunmuştur.

4. Sonuç

Bu çalışmada klasik BO algoritmasındaki *scab* ve *scsb* sabit parametrelerinin üretilmesinde altı farklı kaotik harita kullanılarak KBO1 ve KBO2 algoritmaları elde edilmiştir. Elde edilen KBO1 ve KBO2 algoritmaları ile klasik BO algoritması, sekiz farklı kalite testi fonksiyonu kullanılarak performansları karşılaştırılmıştır. Yapılan çalışmaya göre KBO1 ve KBO2 algoritmalarından elde edilen sonuçlar, klasik BO algoritmasından elde edilen sonuçlara göre, genel olarak daha iyi olduğu görülmektedir. Benzer şekilde yakınsama grafikleri göz önüne alındığında kaotik BO algoritmalarının, klasik BO'ya göre iyi sonuçlara daha erken ulaştığı görülmektedir.

KBO1 ve KBO2'den elde edilen bu rekabetçi sonuçlar göz önüne alınırsa, diğer kaotik haritalarında BO algoritmasına uyarlanabileceği düşünülmektedir. Gelecek çalışmalar için, bu kaotik BO algoritmalarının çok amaçlı versiyonu geliştirilerek gerçek dünya problemleri için kullanılması düşünülmektedir.

Kaynakça

- [1] A. K. Das and D. K. Pratihari, (2019, June). A new bonobo optimizer (BO) for real-parameter optimization, *2019 IEEE Region 10 Symposium (TENSYP)*, pp. 108-113. IEEE.
- [2] A. K. Das and D. K. Pratihari, (2018). A directional crossover (DX) operator for real parameter optimization using genetic algorithm, *Applied Intelligence*.
- [3] Das, A. K., & Pratihari, D. K. (2019). A New Search Space Reduction Technique for Genetic Algorithms. *In Contemporary Advances in Innovative and Applicable Information Technology* (pp. 111-119). Springer, Singapore.
- [4] Das, A. K., & Pratihari, D. K. (2017, December). A novel restart strategy for solving complex multi-modal optimization problems using real-coded genetic algorithm. *In International Conference on Intelligent Systems Design and Applications* (pp. 32-41). Springer, Cham.
- [5] Yun, Y., Chung, H., & Moon, C. (2013). Hybrid genetic algorithm approach for precedence-constrained sequencing

problem. *Computers & Industrial Engineering*, 65(1), 137-147.

Benzer şekilde Tablo 4'e göre en iyi ortalama değerleri; Sphere, Foxholes ve Storn's Chebyshev Polynomial Fitting Problem test fonksiyonlarına göre Iterative, Rosenbrock ve Schwefel test fonksiyonlarına göre Logistic, Ackley ve Kowalik test fonksiyonlarına göre chebyshev map vermiştir. Lennard-Jones Minimum Energy Cluster test fonksiyonuna göre birbirine çok yakın minimum sonuçlar bulunmuştur.

- [6] Holland, J. H. (1975). An introductory analysis with applications to biology, control, and artificial intelligence. *Adaptation in Natural and Artificial Systems. First Edition, The University of Michigan, USA*.
- [7] 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.
- [8] Kennedy, J., & Eberhart, R. (1995, November). Particle swarm optimization. *In Proceedings of ICNN'95-international conference on neural networks* (Vol. 4, pp. 1942-1948). IEEE.
- [9] 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.
- [10] Shi, Y. (2015). An optimization algorithm based on brainstorming process. *In Emerging Research on Swarm Intelligence and Algorithm Optimization* (pp. 1-35). IGI Global.
- [11] Rao, R. V., Savsani, V. J., & Vakharia, D. P. (2011). Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems. *Computer-Aided Design*, 43(3), 303-315.
- [12] Abdelghany, R. Y., Kamel, S., Sultan, H. M., Khorasy, A., Elsayed, S. K., & Ahmed, M. (2021). Development of an Improved Bonobo Optimizer and Its Application for Solar Cell Parameter Estimation. *Sustainability*, 13(7), 3863.
- [13] Sultan, H. M., Menesy, A. S., Kamel, S., Tostado-Véliz, M., & Jurado, F. (2020, June). Parameter identification of proton exchange membrane fuel cell stacks using bonobo optimizer. *In 2020 IEEE International Conference on Environment and Electrical Engineering and 2020 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe)* (pp. 1-7). IEEE.
- [14] Das, A. K., & Pratihari, D. K. (2021). Bonobo optimizer (BO): an intelligent heuristic with self-adjusting parameters

- over continuous spaces and its applications to engineering problems. *Applied Intelligence*, 1-33.
- [15] Das, A. K., & Pratihar, D. K. (2019, July). Optimal preventive maintenance interval for a Crankshaft balancing machine under reliability constraint using Bonobo Optimizer. In *IFTToMM World Congress on Mechanism and Machine Science* (pp. 1659-1668). Springer, Cham.
- [16] Das, A. K., Nikum, A. K., Krishnan, S. V., & Pratihar, D. K. (2020). Multi-objective Bonobo Optimizer (MOBO): an intelligent heuristic for multi-criteria optimization. *Knowledge and Information Systems*, 62(11), 4407-4444.
- [17] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE transactions on evolutionary computation*, vol. 1, pp. 67-82, 1997.