

Salınlı Kaotik Ayçiçeği Optimizasyon Algoritması

Suna YILDIRIM^{1*}, Güngör YILDIRIM², Bilal ALATAŞ³

¹ Elazığ İl Özel İdaresi, Bilgi İşlem Birimi, Elazığ, Türkiye

² Bilgisayar Mühendisliği, Mühendislik Fakültesi, Fırat Üniversitesi, Elazığ, Türkiye

³ Yazılım Mühendisliği, Mühendislik Fakültesi, Fırat Üniversitesi, Elazığ, Türkiye

^{1*} sunayildirim23@gmail.com, ² gungor.yildirim@firat.edu.tr, ³ balatas@firat.edu.tr

(Geliş/Received: 23/03/2021;

Kabul/Accepted: 04/05/2021)

Öz: Uzun yıllardır metasezgisel algoritmalar güncelliğini korumakta, bu alanda farklı algoritmalar geliştirilmeye devam etmektedir. Son zamanlarda önerilen Ayçiçeği Optimizasyon Algoritması başarılı bir arama algoritması olarak öne çıkmaktadır. Bununla birlikte yeni bir algoritma olması nedeni ile geliştirilmeye açık ve kullandığı yöntem ile de esnek bir yapıdadır. Bu çalışma Ayçiçeği Optimizasyon Algoritmasını esas alan ve farklı bir keşif yöntemi kullanan Salınlı Kaotik Ayçiçeği Optimizasyon Algoritmasını önermektedir. Önerilen algoritmada birey üretimi ve algoritma yürütümü farklı kaotik haritalar kullanılarak gerçekleştirilmektedir. Bunun için Chebyshev, Circle, Logistic, Sine ve Tent kaotik haritaları kullanılmıştır. Performans karşılaştırmaları için kısıtlı ve kısıtsız test fonksiyonları tercih edilmiş ve mukayeseli sonuçlar sunulmuştur. Yapılan bu deneysel çalışma sonucunda, Salınlı Kaotik Ayçiçeği Algoritmasının, klasik ayçiçeği algoritması ile rekabet edebilir sonuçlar verdiği ispatlanmıştır.

Anahtar Kelimeler: Ayçiçeği optimizasyon algoritması, kaotik, optimizasyon

Oscillating Chaotic Sunflower Optimization Algorithm

Abstract: Metaheuristic algorithms have been a significant research area for years, and different algorithms in this area continue to be developed constantly. Recently proposed Sunflower Optimization Algorithm stands out as a successful search algorithm. The fact that it is a new algorithm and has a flexible structure makes it more open to development. This study proposes the Oscillating Chaotic Sunflower Optimization Algorithm, which is a different version of the Sunflower Optimization algorithm and uses an improved discovery method. Besides, in the proposed algorithm, plant production and other algorithm processes are able to be conducted using different chaotic maps. For this, chaotic maps of Chebyshev, Circle, Logistic, Sine and Tent have been preferred. For performance comparisons, constrained and unconstrained benchmark functions have been used and comparative results obtained have been presented. As a result of this experimental study, the Oscillating Chaotic Sunflower Algorithm has proven that it can achieve competitive results.

Key words: Sunflower Optimization Algorithm, chaotic, optimization

1. Giriş

Kombinasyonel ve nonlinear problemlerin çözümünde bitki zekası temelli sezgisel algoritmalar başarılarını ispatlamışlardır. Uzun yıllardır yapılan çalışmaların sonucunda bitkilerin cinsiyet kimliği ve bağışıklık sistemine sahip olduğu ve bitkilerin akıllı davranışlar sergilediği görülmüştür. Bu bilgilere dayanarak, bitkilerin sinir sistemine sahip olduğu düşünülmektedir. Örneğin köklerde, alınan ışık ve zehir verileri kök uçlarında büyüme merkezlerine iletilir ve kökler buna göre yönlendirme yapar. Diğer bir husus olarak, bitkilerin elektrik akımları yoluyla dış dünya ile temas kurduğu düşünülmektedir. Bitkiler tarafından yaprak biti veya tırtıla karşı gösterilen savunma mekanizması buna örnek olarak gösterilebilir. İlk saldırı gerçekleştikten sonra bitkiler tatlarını kötüleştirir veya düşmanlarını zehirleyebilecek salgılar üretir [1].

Bu makalede bir bitki zekası algoritması olan Ayçiçeği Optimizasyon(AO) Algoritmasına yönelik deneysel bir çalışma gerçekleştirilmiştir. Bu algoritma ayçiçeği bitkilerinin güneşe yönelim hareketlerini baz alır. Literatürde yeni olmasına karşın bazı problemlerin çözümünde kullanılmıştır. Qais ve diğ. [2] fotovoltajik modüllerin modellenmesi ve simülasyonu için bazı parametre seçimlerinde AO Algoritmasını kullanmışlardır. Parametre seçimlerinde tozlaşma oranı %5 ve eleme oranı %10 olarak alınmış ve başarılı sonuçlar elde edildiğini belirtmişlerdir. Gomes ve diğ. [3] yaptıkları çalışmada plaka benzeri yapıların hasar tespiti için bir optimizasyon yaklaşımı sunmuş ve önerilen tanımlamanın verimliliğini, Genetik Algoritma, AO ve geliştirilmiş bir AO

* Sorumlu yazar: sunayildirim23@gmail.com. Yazarların ORCID Numarası: ¹ 0000-0002-8246-0515, ² 0000-0002-4096-4838, ³ 0000-0002-3513-0329

Algoritmasını kullanarak karşılaştırmışlardır. Önerilen yaklaşımın sıkça kullanılan genetik algoritmalarından daha iyi sonuçlar verdiğini savunmuşlardır. Hussein ve diğ. [4] Proportional integral (PI) kontrolör parametrelerini seçmek için AO Algoritmasını uygulamışlardır. Elde edilen optimizasyon sonuçlarını Parçacık Sürü Optimizasyon algoritması sonuçları ile kıyaslamışlardır. Yuan ve diğ. [5], proton değişim membranlı yakıt hücresi (PEMFC) modelleri için parametrelerin optimum seçimi için yeni bir metodoloji önermektedir. PEMFC yığınının tahmini ve gerçek çıkış voltajı arasındaki hata karesi değerinin toplamını en aza indirmek için yeni geliştirilmiş bir AO Algoritması sunmuşlardır. Geliştirilmiş AO Algoritmasında, en iyi çözüme ulaşmak için kendi kendine uyarlanabilir ağırlıklandırma kullanılmıştır. Shaheen ve diğ. [6], güç sistemleri alanında optimum güç akışı problemini çözmek için AO algoritmasını kullanan yeni bir girişim sunmuşlardır. AO Algoritması amaç fonksiyonunu minimuma indirip, en iyi çözümü bulmak için kullanılmıştır. AO Algoritması, parçacık sürü optimizasyonu ve genetik algoritmaları kullanıp sonuçları karşılaştırmışlardır ve AO Algoritmasının daha iyi sonuçlar verdiğini belirtmişlerdir. Alshammari ve Guesmi [7] yaptıkları çalışmada, güç sistemi stabilizatörlerinin optimal ayarı için Logistic kaotik haritalama metodunu kullanan bir AO Algoritması geliştirmişlerdir. Sekiz test fonksiyonu kullanılmış ve kaotik AO Algoritmasının karşılaştırma için kullanılan tüm algoritmalarından daha iyi performans gösterdiğini belirtmişlerdir.

Öte yandan kaotik temelli optimizasyon algoritmaları bir çok problemin çözümünde farklı yaklaşımlar geliştirilmesine imkan sağlamıştır. Birçok metasezgisel algoritmanın kaotik versiyonu literatürde bulunmaktadır. Koupaei ve diğ. [8] yaptıkları çalışmada, çok/tek modlu amaç fonksiyonlarını optimize etmek için altın bölüm arama algoritmasının pratik bir versiyonunu tanıtmışlardır. Çalışma, doğrusal olmayan optimizasyon problemlerini çözmek için kaotik haritaların yeteneklerini ve altın bölüm arama yöntemini birleştiren yeni bir algoritma sunmuştur. Kaotik harita olarak Lorenz, Logistic ve Chebyshev haritaları kullanılmıştır. Önerilen algoritmanın performansını incelemek için yirmi test fonksiyonu ve bir gerçek dünya problemi kullanılmıştır. Deneysel sonuçlar, önerilen algoritmanın bazı son teknoloji yöntemlerle karşılaştırıldığında etkili ve verimli bir optimizasyon algoritması olduğunu ortaya koymuştur. Yang ve diğ. [9], kaotik haritaların yerel minimumdan daha kolay kaçabileceğini farklı kaotik haritalar kullanarak belirtmişlerdir. Makale, farklı kaotik haritaların ürettiği kaotik dizilerin hem olasılık dağılım özelliğinin hem de arama hızının yeni bir perspektifinden, kaotik optimizasyon algoritmalarının yüksek verimlilik ve üstün performansının doğal mekanizmasını ortaya koymuştur. Demir ve diğ. [10], optimizasyon yöntemlerinden daha yüksek sayısal sonuçlar elde etmek için 1B hibrit kaotik harita (Logistic haritası) tabanlı yeni sürü optimizasyon yöntemi önermişlerdir. Logistic-sinüs haritasının iyi bir istatistiksel sonuca sahip olma avantajını kullanmış ve bu avantaj, bu çalışmada global optimum değeri hesaplamak için doğrudan kullanılmıştır. Önerilen algoritma, sürü tabanlı bir optimizasyon algoritmasıdır ve küresel optimuma ulaşmak için yerel en iyi çözüme ulaşmak için Logistic-sinüs haritasının çekirdek değerini kullanmışlardır. Önerilen hibrit, kaotik harita tabanlı optimizasyon yöntemini test etmek için, yaygın olarak kullanılan sayısal kıyaslama fonksiyonları seçilmiştir. Önerilen kaotik optimizasyon yöntemi, sıkıştırma yayı tasarım problemi üzerinde de test edilmiştir. Xiang ve diğ. [11], yaptıkları çalışmada, kaotik aramayı PSO'ya bağlayan şemaların performansını ve eksikliklerini analiz etmişlerdir. Ardından, kaotik aramayı gerçekleştirmek için parçalı doğrusal kaotik harita (PWLCM- Piecewise Linear Chaotic Map) tanıtılmıştır ve PWLCM (PWLCPSO- Piecewise Linear Chaotic Particle Swarm Optimization) ile birleştirilmiş geliştirilmiş bir PSO (Particle Swarm Optimization) algoritması önerilmiştir. Tian [12], yerel optimuma yakalanmamak için, PSO algoritmasına kaotik haritalar (Tent haritası ve Logistic harita) ve Gauss mutasyonunu ekleyerek yeni bir parçacık sürüsü optimizasyonu önermiştir. Bir yandan, ilk popülasyonun kalitesini iyileştirmek için basit ama çok etkili bir yöntem olan kaotik harita kullanılmıştır. Öte yandan, maksimal odak mesafesine dayalı Gauss mutasyon mekanizması, algoritmanın yerel optimumdan kaçmasına yardımcı olmak ve partiküllerin global optimal veya daha yakın olana kadar çözüm uzayının diğer bölgelerinde aramaya devam etmesini sağlamak için benimsenmiştir. Garip ve diğ. [13], yaptıkları çalışmada, balina optimizasyon algoritmasının etkinliğini ve verimliliğini artırmak amacıyla 5 farklı kesikli kaotik harita kullanılmıştır. Global optimizasyon için kaotik haritalar ve balina optimizasyon algoritmasına dayalı CBWOA 1 (Chaotic-Based Whale Optimization Algorithm 1), CBWOA 2, CBWOA 3 ve CBWOA 4 algoritmaları önerilmiştir. WOA'da (Whale Optimization Algorithm) bulunan parametreleri ayarlamak için Tinkerbell, Sine, Logistic, Tent ve Henon kaotik haritalarından rastgele sayı dizileri kullanılmıştır. Geliştirilen algoritmaların doğruluğunu test etmek ve performans analizini yapmak için altı farklı tek modlu ve çok modlu kıyaslama fonksiyonu kullanılmıştır. Pluhacek ve diğ. [14], atalet ağırlıklı PSO algoritmasının performansını artırmak için üç farklı kaotik harita tabanlı sayı üreticinin kullanılmasını önermiştir. Bu çalışma, çeşitli test fonksiyonları üzerinde performans testinin sonuçlarını sunmaktadır. Farklı kaotik oluşturucular için elde edilen sonuçlar karşılaştırılmış ve kısaca analiz edilmiştir.

Bu çalışma, literatürdeki AO Algoritmalarından farklı olarak daha esnek bir keşif (exploration) sürecine olanak sağlayan Salınlı Kaotik Ayçiçeği Algoritmasını (SKAO) önermektedir. Bu algoritma, iterasyonlar

süresince yeni katılan bireylerin arama alanlarına osilasyon temelli konumlanmalarını esas almaktadır. Ayrıca önerilen algoritma, birey üretim ve algoritmik yürütümlerde kaotik haritaları kullanmaktadır. Önerilen yöntem, klasik AO Algoritmasına göre rekabetçi sonuçlar elde edebilmiş ve AO Algoritmasını kullanacak olan araştırmacılara güçlü bir alternatif olacağını ispatlamıştır.

Çalışmanın bundan sonraki organizasyonu şu şekildedir; klasik AO Algoritmasının detayları ve SKAO'nun kullandığı yaklaşım ikinci bölümde sunulacaktır. Performans testleri, deney parametreleri ve elde edilen sonuçlar üçüncü bölümde, sonuç değerlendirmeleri ise dördüncü bölümde paylaşılacaktır.

2. Ayçiçeği Optimizasyon Algoritması ve Sahnımlı Kaotik Ayçiçeği Optimizasyon Algoritması

Optimizasyon problemlerinin çözümünde kullanılan matematiksel yöntemler, işlem maliyetinin bazen çok büyük oranlara ulaşması veya her problem için matematiksel modelin çıkarılmasının zor olabileceği ihtimali nedeniyle dezavantajlı olabilir. Buna alternatif olarak metasezgisel yöntemler belirli bir modele bağlı kalmadan farklı türde problemlere uygulanabilir. Hesaplama maliyetinin daha düşük olması, kolay uyarlanabilir olması ve başarımları metasezgisel yöntemleri güncel tutmaktadır. Bu avantajlarıyla metasezgisel algoritmalar, mühendislikten sosyal bilimlere kadar pek çok alanda farklı problem tiplerine uygulanmıştır. Son yıllarda, önerilmiş başarılı metasezgisel optimizasyon algoritmalarından Ayçiçeği Optimizasyon Algoritması, ayçiçeklerinin güneş ışınlarını takip hareketlerini esas alır [15]. Ayçiçeklerinin amacı güneşe en iyi yönelimi arayıp bulmaktır. Her yeni günde, ayçiçekleri güne başlar ve güneşi takip etmeyi hedefler ve bu döngü her sabah devam eder. Gün sonunda ise ters yönde hareket ederek ertesi günün başlamasını beklerler. Ayçiçekleri, güneşe yakınlıklarına bağlı olarak, eğer güneşe yakınlarsa durumlarını korumak ya da güneşten uzaklarsa güneşe daha fazla yaklaşabilmek için daha büyük yönelimler göstermek isteyebilir. Komşu ayçiçekleri güneşin pozisyonuna göre kendi aralarında tozlaşarak yeni bireyler oluştururlar. Gerçek dünyada ayçiçekleri milyonlarca tozlaşma gerçekleştirebilecekken, algoritmada basitlik açısından her ayçiçeğinin bir tozlaşma gerçekleştirdiği varsayılır. Ayçiçeği Optimizasyon Algoritmasında, başlangıç popülasyonu oluşturulduktan sonra çözüme en yakın birey, güneş olarak adlandırılır. Diğer bireyler de güneşin pozisyonuna göre kendi aralarında tozlaşarak yeni bireyler oluştururlar. Yeni bireyler oluşturulduktan sonra tekrar çözüme en yakın birey bulunarak güneş güncellenir. Arama uzayının çok fazla artmaması için belli bir oranda ayçiçeği elenir ve buna eleme oranı denir. Ve sonlandırma kriteri sağlanana kadar bu işlemler tekrarlanır.

Ayçiçeği algoritmasında, bir etken güneşe olan uzaklıktır. Burada ters kare radyasyon yasası geçerlidir. Bu yasada, radyasyon yoğunluğu, uzaklığın karesiyle ters orantılıdır. Yani bir birey güneşe ne kadar uzaksa, o miktarın karesinin tersi kadar radyasyona maruz kalır. Bu etki Q_i ile gösterilip genel ifadesi Denklem 1'deki gibidir. Bu denklemde, P kaynağın gücü ve r_i de, i . birey ile anlık en iyi arasındaki mesafedir.

$$Q_i = \frac{P}{4\pi r_i^2} \quad (1)$$

Algoritmada temel prensip ayçiçeklerinin güneşe yöneliminin temsil edilmesidir. Bunun için popülasyondaki güneş (X^*), diğer ayçiçekleri için (X_i) referans olmaktadır. Referansa göre uzaklık kriterini dikkate alan yönelimi Denklem 2 ile ifade edilir.

$$\vec{s}_i = \frac{X^* - X_i}{\|X^* - X_i\|}, \quad i = 1, 2, \dots, n_p \quad (2)$$

Popülasyon bireylerinin güneşe doğru yönelim adımları ayçiçeği algoritmasının kritik parametrelerindedir. Popülasyon bireylerinin yönelim hızlarında i . birey ($i-1$). birey ile tozlaşarak rasgele pozisyonlu yeni bireyi oluşturur. Bu rasgele pozisyon, bireyler arasındaki mesafe ile orantılıdır. Güneşe yönelim adımlarının genel ifadesi Denklem 3'te verilmiştir. Bu denklemde P_i ilgili bireyler arasındaki tozlaşma olasılığını, λ ise eylemsizlik katsayısının temsil etmektedir.

$$d_i = \lambda P_i (\|X_i + X_{i-1}\|) \|X_i + X_{i-1}\| \quad (3)$$

Adım değerlerinin belirlenmesinde temel kriterlerden bir tanesi maksimum adım uzunluğudur (d_{max}). Denklem 4'te gösterilen bu parametre, problem tanımına göre öntanımlı üst sınır (X_{max}) ve alt sınır (X_{min})'in Öklid mesafesi ile doğru, popülasyondaki birey sayısı (N_{pop}) ile ters orantılıdır.

$$d_{max} = \frac{\|X_{max} - X_{min}\|}{2N_{pop}} \quad (4)$$

Yukarıda açıklanan temel parametreleri esas alan yeni birey Denklem 5 ile hesaplanır.

$$\vec{X}_{i+1} = \vec{X}_i + d_i \vec{s}_i \quad (5)$$

Klasik AO Algoritmasında her iterasyonda güneşe uzak olan bireyler, popülasyondan çıkarma oranına (mortality rate) göre arama sürecinden çıkarılmakta ve yerine yeni bireyler üretilmektedir. Rastgelelik dikkate alındığında yeni bireylerin eski bireyler ile benzer şekilde davranış göstermesi olasılık dahilindedir. Bu çalışmada keşif (exploration) sürecini daha esnek bir şekilde gerçekleştirme imkânı veren Salınlı Kaotik Ayçiçeği Algoritmasını (SKAO) önerilmiştir. AO'nun bu versiyonunda, iterasyonlar boyunca yeni eklenen bireylerin belirli bir kısmı için arama uzayı, popülasyon en iyisi ve arama uzayı maksimumu esas alınarak belirlenmektedir. Bunun için iki temel aşama geçilmektedir. Birinci aşama, eklenecek bireylerin belirli oranı (Y) için iterasyon en iyisi ve arama uzayı maksimum değeri arasında bir lokasyonun belirlenmesidir. Bu lokasyonunun belirlenmesinde, iki sınır ortasını ortalama değer olarak kullanan Gaussian Dağılım Fonksiyonundan elde edilen değer baz alınır. Birinci aşama adımlarını gösteren Denklem 6-7'de, D problem boyutunu göstermek üzere, $f \in D$, P_f ortalama değer, L_f arama lokasyon merkezini ve σ^2 varyansı temsil etmektedir.

$$P_f = \frac{x_f^* + x_f^{max}}{2} \quad (6)$$

$$L_f = N(P_f, \sigma^2) \quad (7)$$

Belirlenen lokasyon merkezi etrafında aday değerlerinin üretimi için trigonometrik bir yaklaşım kullanılır. Böylece keşif süreci daha esnek bir şekilde gerçekleştirilir. Denklem 8, Salınlı kaotik AO'da iterasyon i'de popülasyona katılan yeni aday değerlerinin belirlenmesini göstermektedir. Burada, η kullanılan kaotik harita fonksiyonundan dönen değeri, β ise ön tanımlı uygulama sabitini ifade etmektedir.

$$X_f^{yeni} = L_f(1 + \eta \cos(\frac{2\pi i}{\beta})) \quad (8)$$

Algoritma 1. Kaotik temelli Ayçiçeği Optimizasyon Algoritması sözde kodu

1. $X_{max}, X_{min} \leftarrow$ problem tanımlı üst sınır ve alt sınırlar
 2. $Itr_{max} \leftarrow$ maksimum iterasyon sayısı
 3. $N_{pop} \leftarrow$ popülasyon boyutu
 4. $k \leftarrow$ kaotik üreteç fonksiyonu tipi
 5. $O \leftarrow$ Popülasyondan çıkarma oranı (%)
 6. **for** $i=1$ 'den N_{pop}
 7. $X_i =$ BireyÜreteçFonksiyonu(k, X_{max}, X_{min})
 8. PopulasyonaEkle(X_i)
 9. **end for**
 10. AmaçHesapla()
 11. $X^* \leftarrow$ GüneşBul()
 12. AyçicekleriniGüneşeYonlendir()
 13. **while** ($i < Itr_{max}$)
 14. **for** $j=1$ 'den N_{pop}
 15. BireylerinVektörünüHesapla()
 16. GüneştenUzakBireyleriÇıkar(%O, %Y, i, σ β)
 17. YeniBireyleriDeğerlendir(X^*)
 18. **end for**
 19. AmaçHesapla()
 20. $X^* \leftarrow$ GüneşBul()
 21. **End while**
 22. EniyiÇözümGöster()
-

Bu fonksiyonu kullanmayan ve popülasyona yeni katılan diğer bireylerin belirlenmesi klasik AO'ya göre gerçekleştirilmiştir. Ancak tüm aday üretimlerinde (başlangıç popülasyonu dahil) kaotik harita fonksiyonları kullanılmıştır. Çalışma, SKAO'nun farklı kaotik varyasyonlarının performansını değerlendirmek için literatürde iyi bilinen kaotik harita fonksiyonlarını kullanmıştır. Algoritma 1, SKAO Algoritmasının sözde kodunu

göstermektedir. Kullanılan kaotik harita seçimi k parametresi ile gösterilmektedir. Algoritmada Denklem 6-8, güneşten uzak bireylerin çıkarılma fonksiyonunda yürütülmektedir. Bu fonksiyon i, Y, σ ve β parametrelerini kullanır.

Kaotik haritalar, matematiksel olarak basit deterministik dinamik bir sistemin rastgeleliğidir ve kaotik sistem rastgelelik kaynağı olarak düşünülebilir [16]. Bu çalışmada kaotik haritalar kullanılarak, SKAO uygulamasından daha verimli sonuçlar elde edilip edilemeyeceği incelenmiştir. Birey üretimi ve hareket davranışlarında bu kaotik haritalar kullanılmıştır. Tablo 1, çalışmada kullanılan kaotik haritaları göstermektedir. Kullanılan tüm kaotik haritalar, SKAO Algoritmasına eklenmiş ve bir fonksiyon seçim parametresi ile etkinleştirilmiştir. Böylece, [16, 17]'ye benzer şekilde, çalıştırılan SKAO Algoritması kullanılan kaotik harita ile adlandırılmıştır (örnek tentSKAO gibi).

3. Deneyleer

Bu bölümde, kaotik temelli Ayçiçeği Optimizasyon Algoritmalarının performans sonuçları paylaşılacaktır. Değerlendirmeler için bir önceki bölümde açıklanan kaotik üreteçleri kullanan Ayçiçeği algoritmalarının sonuçları ile klasik ayçiçeği algoritmasının sonuçları birlikte ele alınmıştır. Performans testleri için literatürde iyi bilinen ve Tablo 2'de verilen test fonksiyonları kullanılmıştır. Bu test fonksiyonlarından Rosenbrock Cubic/Line ve Rosenbrock Disk kısıtlı test fonksiyonlarıdır. Yapılan performans testlerinde problem boyutları Sphere fonksiyonunda beş, Rastrigin fonksiyonunda üç, diğer fonksiyonlarda ise iki olarak seçilmiştir. Bununla birlikte, birey boyutu ve iç iterasyon sayısı 20 olup çözüm iterasyon sayısı 5 olarak seçilmiştir. Böylece deneyler 95 iterasyon (referans eleman dışı) üzerinden gerçekleştirilmiştir. Deneylerde tozlaşma oranı (p) 0.05, eleme oranı (O) 0.1'dir. Performans testleri birbirinden bağımsız 20 deney ile gerçekleştirilmiştir. Performans sonuçlarında ilk olarak iterasyonlara göre elde edilen minimum değerler irdelenmiştir. Bunun için klasik ve kaotik Ayçiçeği Algoritmaları her bir test fonksiyon için birlikte değerlendirilmiştir. Son olarak ise tüm deneylerden elde edilen sonuçlara ait istatistik sonuçlar paylaşılmıştır.

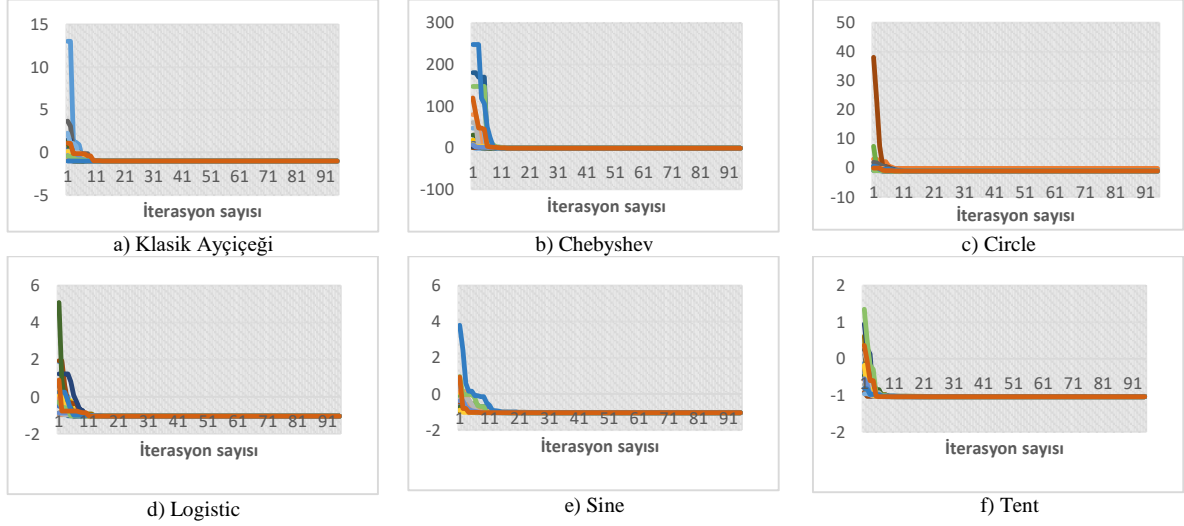
Tablo 1. Kaotik Haritalar

Adı	Fonksiyonu	Parametreleri
Sine [18]	$X_{n+1} = \frac{a}{4} \sin(\pi X_n)$	$0 < a \leq 4$
Logistic [19]	$X_{n+1} = \mu X_n (1 - X_n)$	$0 < \mu \leq 4$
Chebyshev [20]	$X_{n+1} = \cos(k \cos^{-1} X_n)$	$k \geq 2$
Circle [18]	$X_{n+1} = X_n + 0.2 - \left(\frac{1}{4\pi}\right) \sin(2\pi X_n) \text{mod}(1)$	
Tent [21]	$X_{n+1} = \begin{cases} \mu X_n, & X_n < \frac{1}{2} \\ \mu(1 - X_n), & \frac{1}{2} \leq X_n \end{cases}$	

Tablo 2. Test Fonksiyonları

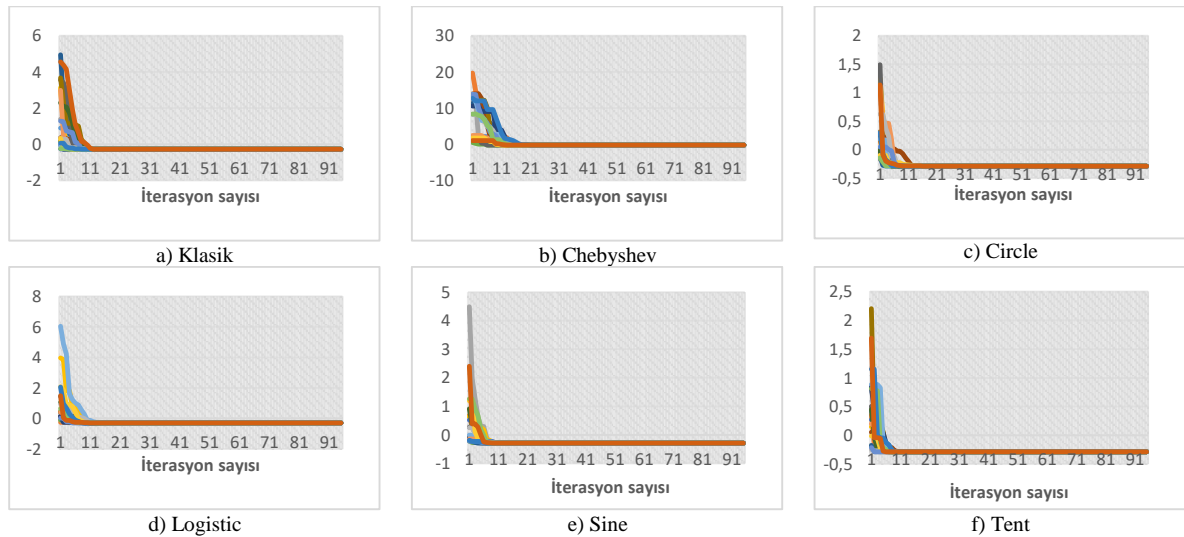
Adı	Fonksiyon Denklemi	Parametreler	Min
Sphere	$f(x) = \sum_{i=1}^N x_i^2$	$-100 < x_i < 100$	0
Rastrigin	$f(x) = 10N + \sum_{i=1}^N [x_i^2 - 10 \cos(2\pi x_i)]$	$-5.12 < x_i < 5.12$	0
Camel	$f(x) = \left(4 - 2.1x_1^2 + \frac{x_1^4}{3}\right)x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2$	$-200 < x_i < 200$	-1.031
Nonlinear	$f(x) = x_1^2 - 3x_1x_2 + 4x_2^2 + x_1 - x_2$	$-2 < x_i < 2$	-0.28
Rosenbrock Cubic/Line	$f(x, y) = (1 - x)^2 + 100(y - x^2)^2$	$(x - 1)^3 - y + 1 \leq 0$ $ve x + y - 2 \leq 0$	0
Rosenbrock Disk	$f(x, y) = (1 - x)^2 + 100(y - x^2)^2$	$x^2 + y^2 \leq 2$	0

İlk performans testleri, Camel test fonksiyonu için yapılmıştır. Şekil 1, bu deneylerin iterasyonlarından elde edilen iterasyon minimum değişimlerini göstermektedir. Klasik Ayçiçeği fonksiyonu deneylerinde elde edilen minimum değerlere %99 ve üzeri oranda yaklaşıma ortalama olarak 17 iterasyonda gerçekleşmiştir. Bu ortalama iterasyon değerleri, tentSKAO için 13, circleSKAO ve logisticSKAO için 16, chebyshevSKAO için 19 ve sineSKAO için 25'tir. Genel performanslar bir birine yakın olmakla beraber tentSKAO bu fonksiyon için daha başarılı olmuştur.



Şekil 1. Camel Test Fonksiyonu için iterasyonlara göre uygunluk değerinin minimum değişimleri

Bir sonraki performans testleri, Nonlineer test fonksiyonu için gerçekleştirilmiştir. Şekil 2 bu deneylerin iterasyonlarından elde edilen iterasyon minimumları değişim performanslarını göstermektedir. Klasik ve kaotik ayçiçeği fonksiyonu deneyleri karşılaştırıldığında; klasik yöntemde ortalama 8 iterasyonda, chebyshevSKAO'da 11 iterasyonda, circleSKAO'da 7 iterasyonda, logisticSKAO'da 10 iterasyonda, sineSKAO'da 7 iterasyonda ve tentSKAO'da 6 iterasyonda minimum değerlere ulaşılmıştır. Bu testlerde de yine en iyi sonucu tentSKAO vermiştir.

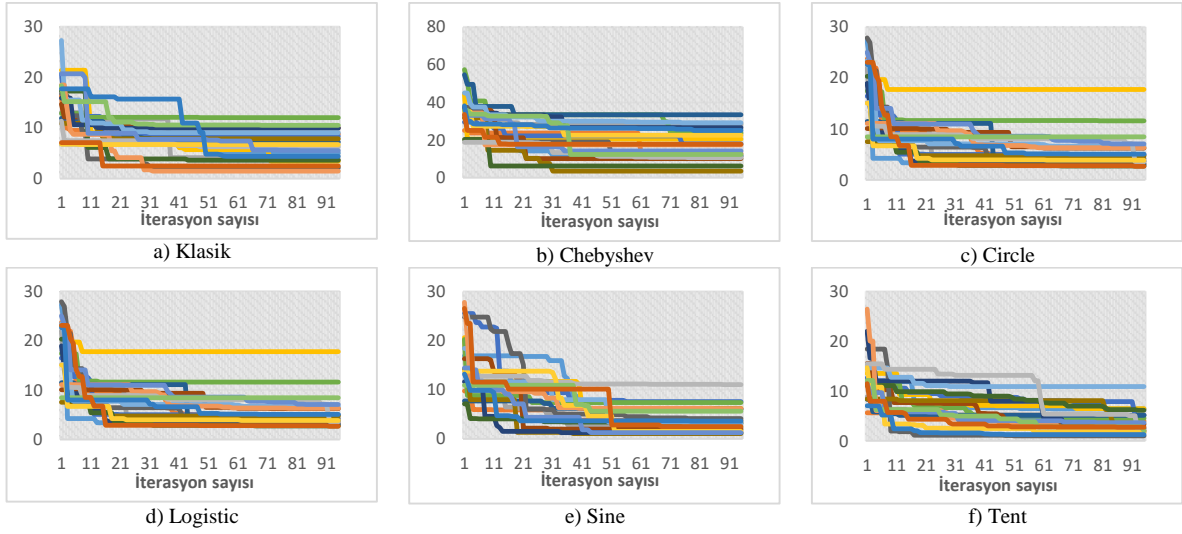


Şekil 2. Nonlineer Test Fonksiyonu için iterasyonlara göre uygunluk değerinin minimum değişimleri

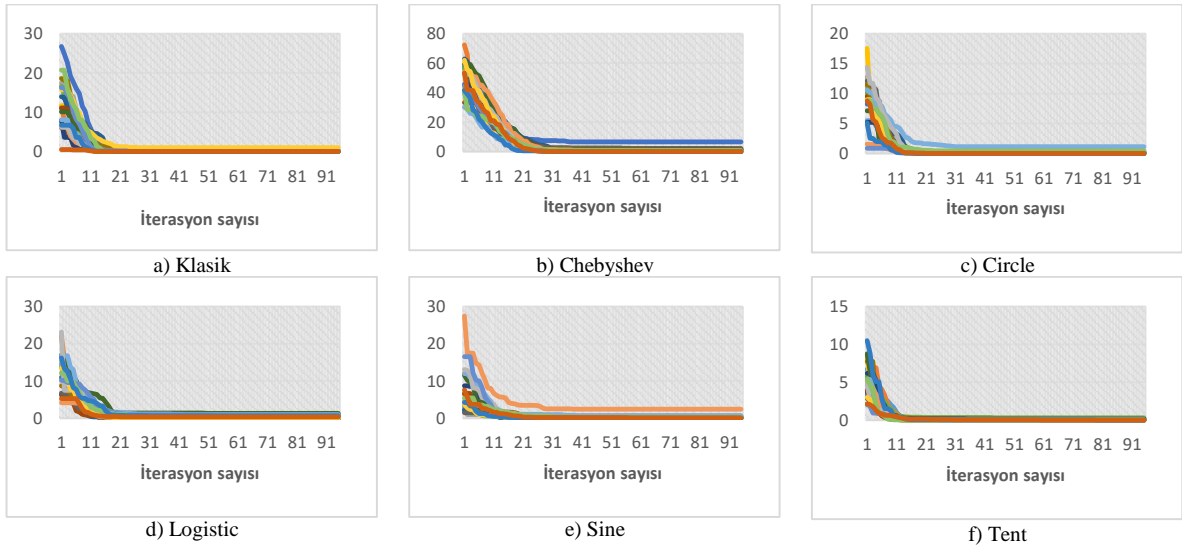
Sonraki performans testleri, Rastrigin test fonksiyonu için yapılmıştır. Şekil 3'te deneylerdeki iterasyon minimumları sonuçları gösterilmektedir. Bilindiği üzere Rastrigin test fonksiyonu pek çok yerel lokal minimum

çeren zor bir fonksiyondur. Minimumu sıfır olan Rastrigin fonksiyonu için deneylerde $d=3$ alınmıştır. Sonuçlar incelendiğinde genellikle metotların bu lokal minimumlara takıldığı gözlemlenmiştir. Minimuma ulaşma ortalama iterasyon sayılarına bakıldığında, klasik yöntemde 32. iterasyonda 1.471, chebyshevSKAO'da 31. iterasyonda 3.554, circleSKAO'da 16. iterasyonda 2.666, logisticSKAO'da 65. iterasyonda 0.885, sineSKAO'da 19. iterasyonda 1.026, tentSKAO'da ise 17. iterasyonda 1.012 minimum değerlerine ulaşılmıştır. Bu testlerde, en performanslı sonuç logistic ayçiçeği optimizasyon algoritmasında elde edilmiştir.

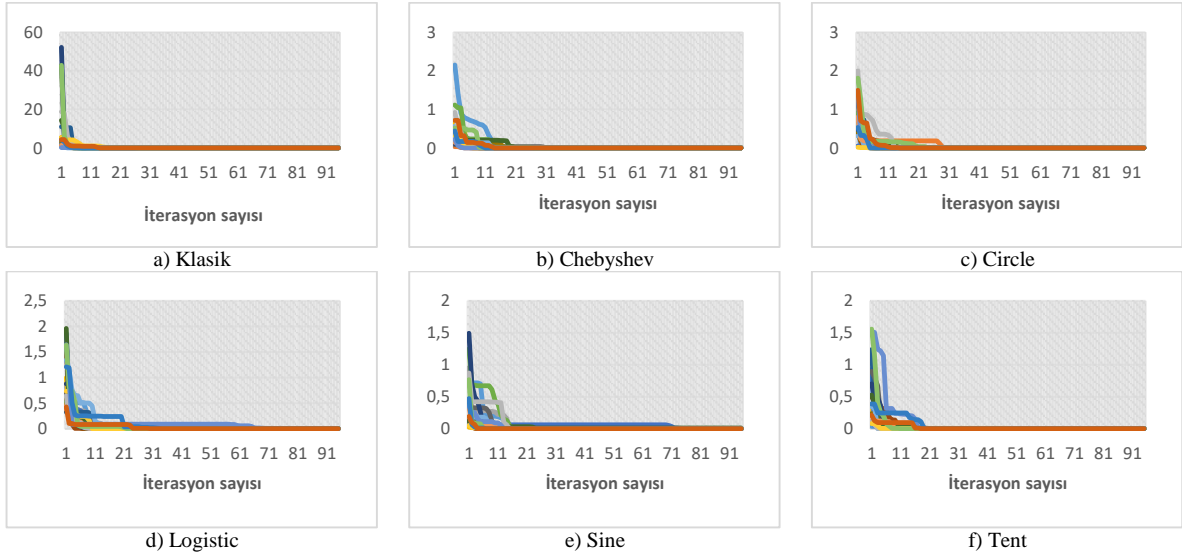
Sphere fonksiyonu için yapılan deneylerde elde edilen iterasyon-minimum değişimleri Şekil 4'te verilmiştir. Genel olarak bu test fonksiyonu için kaotik temelli AO metotlarından tentSKAO algoritmasının başarımı klasikAO algoritmasıyla hemen hemen aynıdır. Minimum değerlere ulaşılan ortalama iterasyon sayıları; klasikAO'da 22. iterasyon, chebyshevSKAO'da 47. iterasyon, circleSKAO'da 27. iterasyon, logisticSKAO'da 48. iterasyon, sineSKAO'da 43. iterasyon, tentSKAO'da 19. iterasyondur.



Şekil 3. Rastrigin Test Fonksiyonu için iterasyonlara göre uygunluk değerinin minimum değişimleri

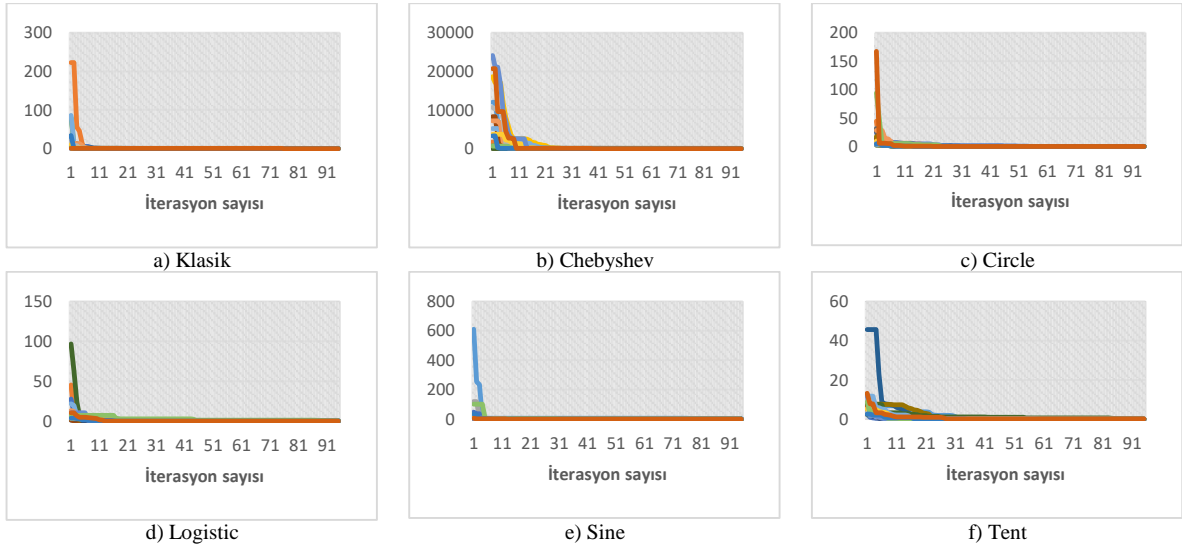


Şekil 4. Sphere Test Fonksiyonu için iterasyonlara göre uygunluk değerinin minimum değişimleri



Şekil 5. Rosenbrock Disk Test Fonksiyonu için iterasyonlara göre uygunluk değerinin minimum değişimleri

Rosenbrock Disk test fonksiyonu için yapılan deney sonuçları ve iterasyon değişimleri Şekil 5'te görülmektedir. Bu deneylerde hem klasik AO, hem de kaotik temelli AO'ların minimum değerlere ulaştığı ortalama iterasyon sayılarının birbirine yakın olduğu ve başarılı sonuçlar verdiği gözlemlenmiştir. KlasikAO ortalama 9. iterasyonda minimum değere ulaşmışken, chebyshevSKAO 10. iterasyonda, circleSKAO 15. iterasyonda, logisticSKAO 12. iterasyonda, sineSKAO 8. iterasyonda ve tentSKAO da 10. iterasyonda minimum değerlere ulaşmıştır. Son performans testleri, Şekil 6'da gösterildiği üzere Rosenbrock Cubic/Line test fonksiyonu için yapılmıştır. KlasikAO, kaotik temelli AO Algoritmalarına göre gözle görülür bir şekilde başarılı sonuçlar vermiştir. Minimum değere ulaşılan ortalama iterasyon sayısı klasikAO'da 10. iterasyondayken, kaotik temelli AO'larda; 25. iterasyonla sineSKAO, 28. iterasyonla logisticSKAO, 33. iterasyonla circleSKAO, 47. iterasyonla tentSKAO ve 53. iterasyonla chebyshevSKAO, klasikAO sonuçlarını izlemişlerdir.



Şekil 6. Rosenbrock Cubic/Line Test Fonksiyonu için iterasyonlara göre uygunluk değerinin minimum değişimleri

Tablo 3, deneylerde elde edilen tüm sonuçlara ait istatistiksel değerlendirmeleri karşılaştırmalı olarak göstermektedir. Tablo sonuçları, test fonksiyonlarına göre değerlendirildiğinde, minimum değeri -1.031 olan Camel fonksiyonu için her ne kadar tüm fonksiyonlar genel olarak performanslı sayılsa da, chebyshevSKAO, logisticSKAO, sineSKAO ve tentSKAO daha başarılıdır. Bu fonksiyon için KlasikAO ve circleSKAO, standart

sapma ve ortalama değerleri bakımından biraz daha geride kalmıştır. İkinci test fonksiyonu olan nonlineer fonksiyonuna ait gerçek minimum değeri -0.28'dir. Bu fonksiyon için tüm yöntemler son derece başarılı olmuştur. Tüm testler için her yöntem neredeyse aynı sonuçlara ulaşmayı başarmıştır. Rastrigin fonksiyonu, gerçek minimumun bulunması zor olan bir fonksiyondur. 0 olan gerçek minimum değerine ulaşamamış olup, en iyi sonucu logisticSKAO vermiştir. Onu, tentSKAO ve sineSKAO algoritmaları izlemiştir. Gerçek minimum değeri 0 olan Sphere fonksiyonunun performansları incelendiğinde, minimum değer parametresine göre en yakın değeri sineSKAO ve tentSKAO yakalamıştır. Bununla birlikte bu değere yaklaşan klasikAO ve circleSKAO da oldukça başarılıdır. Ortalama ve ortanca değerleri performansı açısından tentSKAO daha da ön plana çıkmaktadır. Kaotik temelli AOlarda içerisinde chebyshevSKAO ve logisticSKAO, klasikAO'ya göre bu fonksiyon için geride kalmışlardır. Kısıtlı fonksiyonlardan Rosenbrock Disk için gerçek minimum değeri sıfırdır. Bu fonksiyon için tüm yöntemler oldukça başarılıdır. Bununla birlikte minimum ve maksimum değerler dikkate alındığında klasikAO ve tentSKAO, küçük bir farkla diğerlerinden geride kalmıştır. Gerçek minimum değeri 0 olan bir diğer kısıtlı test fonksiyonu Rosenbrock Cubic/Line fonksiyonudur. Minimum değere ulaşma bakımından tüm yöntemler başarılı sayılabilir. Öte yandan, ortalama ve ortanca parametrelerine göre chebyshevSKAO biraz daha geride kalmıştır.

İterasyon ve istatistiksel sonuçlar beraber değerlendirildiğinde kaotikAO'ların klasikAO'ya göre rekabetçi ve başarılı performansı görülmüştür. Bununla birlikte tentSKAO diğer kaotik fonksiyonlara göre bir adım daha öne çıkmaktadır. Her ne kadar başarılı sayılsa da, chebyshevSKAO'nun diğer kaotikSKAO'lara göre küçük bir farkla geride kaldığı görülmüştür.

Tablo 3. Tüm deneyler için istatistiksel sonuçlar

		Camel	Nonlineer	Rastrigin	Sphere	R.Disk	R. Cubic/Line
Klasik	Min	-1.031	-0.285	1.471	2.812E-05	2.938E-07	3.420E-08
	Maks	-1.027	-0.285	12.046	1.029	0.028	5.369E-07
	Ortalama	-1.031	-0.285	6.416	0.094	0.001	9.042E-05
	Ortanca	-1.031	-0.285	6.479	0.019	5.437E-06	1.858E-05
	Std. Sapma	0.001	3.519E-06	3.014	0.226	0.006	8.745E-07
Chebyshev	Min	-1.031	-0.285	3.554	0.014	3.635E-08	5.997E-07
	Maks	-1.031	-0.285	33.335	6.490	6.193E-05	6.776
	Ortalama	-1.031	-0.285	17.778	0.890	1.001E-05	0.762
	Ortanca	-1.031	-0.285	17.572	0.399	5.116E-06	6.036E-07
	Std. Sapma	7.433E-05	8.59E-06	7.966	1.408	1.523E-05	1.857
Circle	Min	-1.031	-0.285	2.666	3.951E-05	3.611E-08	5.484E-07
	Maks	-0.215	-0.285	17.727	1.142	4.778E-05	0.001
	Ortalama	-0.988	-0.285	5.684	0.142	8.096E-06	9.958E-05
	Ortanca	-1.031	-0.285	4.793	0.067	3.477E-06	5.888E-06
	Std. Sapma	0.182	1.869E-06	3.502	0.249	1.299E-05	1.541E-07
Logistic	Min	-1.031	-0.285	0.885	0.001	6.099E-08	4.551E-07
	Maks	-1.031	-0.285	11.396	1.326	1.19E-07	0.952
	Ortalama	-1.031	-0.285	4.704	0.333	2.791E-05	0.070
	Ortanca	-1.031	-0.285	4.028	0.124	3.688E-06	1.608E-05
	Std. Sapma	1.889E-05	2.154E-06	3.303	0.390	6.675E-05	0.224
Sine	Min	-1.031	-0.285	1.026	1.14E-07	4.312E-08	5.800E-07
	Maks	-1.031	-0.285	11.0422	2.441	0.008	3.211
	Ortalama	-1.031	-0.285	4.258	0.236	2.325E-07	0.160
	Ortanca	-1.031	-0.285	3.662	0.032	4.254E-06	5.806E-06
	Std. Sapma	3.884E-05	6.959E-06	2.590	0.537	0.001	0.699
Tent	Min	-1.031	-0.285	1.012	1.02E-07	9.126E-08	5.518E-08
	Maks	-1.031	-0.285	10.909	0.291	0.002	0.110
	Ortalama	-1.031	-0.285	3.909	0.042	3.902E-07	0.008
	Ortanca	-1.031	-0.285	3.492	0.012	2.793E-06	2.361E-05
	Std. Sapma	2.574E-05	3.693E-06	2.356	0.068	5.632E-07	0.025

4. Sonuçlar

Bu makalede, ayçiçeği optimizasyon algoritmasına daha esnek bir keşif süreci gerçekleştiren Salımlı Kaotik Ayçiçeği Algoritması (SKAO) önerilmiştir. Önerilen algoritmanın esnekliği, hem kaotik hem de trigonometrik yaklaşımları kullanmasından kaynaklanmaktadır. Önerilen algoritmanın performansı hem klasik Ayçiçeği Algoritmasına hem de farklı kaotik haritalar kullanan SKAO'ya göre irdelenmiştir. Kaotik harita fonksiyonu olarak, Chebyshev, Circle, Logistic, Sine ve Tent haritaları kullanılmıştır. Bu yöntemlerin tamamı, 4 kısıtsız (Camel, Nonlineer, Rastrigin, Sphere) ve 2 kısıtlı (Rosenbrock Disk ve Rosenbrock Cubic/Line) test

fonksiyonlarından geçirilmiştir. Yapılan deneyler sonucunda, önerilen SKAO Algoritmasının oldukça rekabetçi sonuçlar ürettiği görülmüştür. Bu yönüyle, SKAO Algoritması hem yazarlar hem de bu alanda çalışan araştırmacılar için bir alternatif olabileceğini ispatlamıştır. Yazarlar, Kaotik Ayçiçeği Algoritmasıyla kural çıkarım bazlı optimizasyonu ve duygu analizi araştırmalarına odaklanacaklardır.

Kaynaklar

- [1] Akyol S, Alataş B. Plant intelligence based metaheuristic optimization algorithms. *Artificial Intelligence Review* 2017; 47:417–462. Doi: 10.1007/s10462-016-9486-6
- [2] Qais MH, Hasanien HM, Alghuwainem S. Identification of electrical parameters for three-diode photovoltaic model using analytical and sunflower optimization algorithm. *Applied Energy* 2019; 250: 109-117. Doi: 10.1016/j.apenergy.2019.05.013.
- [3] Gomes GF, Almeida FA. Tuning metaheuristic algorithms using mixture design: Application of sunflower optimization for structural damage identification. *Advances in Engineering Software* 2020; 149: 102877. Doi: 10.1016/j.advengsoft.2020.102877.
- [4] Hussien AM, Hasanien HM, Mekhamer SF. Sunflower optimization algorithm-based optimal PI control for enhancing the performance of an autonomous operation of a microgrid. *Ain Shams Engineering Journal* 2021; Doi: 10.1016/j.asej.2020.10.020.
- [5] Yuan Z, Wang W, Wang H, Razmjooy N. A new technique for optimal estimation of the circuit-based PEMFCs using developed Sunflower Optimization Algorithm. *Energy Reports* 2020; 6: 662-671. Doi: 10.1016/j.egy.2020.03.010.
- [6] Shaheen MAM, Hasanien HM, Mekhamer SF, Talaat HEA. Optimal Power of Power Systems Including Distributed Generation Units Using Sunflower Optimization Algorithm. *IEEE Access* 2019; 7: 109289-109300. Doi: 10.1109/Access.2019.2933489.
- [7] Alshammari BM, Guesmi T. New Chaotic Sunflower Optimization Algorithm for Optimal Tuning of Power System Stabilizers. *Journal of Electrical Engineering & Technology* 2020; 15: 1985-1997. Doi: 10.1007/s42835-020-00470-1.
- [8] Koupaei JA, Hosseini SMM, Ghaini FMM. A new optimization algorithm based on chaotic maps and golden section search method. *Engineering Applications of Artificial Intelligence* 2016; 50: 201-214. Doi: 10.1016/j.engappai.2016.01.034.
- [9] Yang D, Liu Z, Jilei Z. Chaos optimization algorithms based on chaotic maps with different probability distribution and search speed for global optimization. *Commun Nonlinear Sci Numer Simulat* 2014; 19: 1229-1246. Doi: 10.1016/j.cnsns.2013.08.017.
- [10] Demir FB, Tuncer T, Kocamaz AF. A chaotic optimization method based on logistic-sine map for numerical function optimization. *Neural Computing and Applications* 2020; 32: 14227-14239. Doi: 10.1007/s00521-020-04815-9.
- [11] Xiang T, Liao X, Wong K. An improved particle swarm optimization algorithm combined with piecewise linear chaotic map. *Applied Mathematics and Computation* 2007; 190: 1637-1645. Doi: 10.1016/j.amc.2007.02.103.
- [12] Tian D. Particle Swarm Optimization with Chaotic Maps and Gaussian Mutation for Function Optimization. *International Journal of Grid Distribution Computing* 2015; 8(4): 123-134. Doi: 10.14257/ijgdc.2015.8.4.12
- [13] Garip ZB, Çimen ME, Karayel D, Boz AF. The Chaos-Based Whale Optimization Algorithms Global Optimization. *Chaos Theory and Applications*; 2019. 1(1): 51-63.
- [14] Pluhacek M, Senkerik R, Zelinka I. Impact of Various Chaotic Maps on the Performance of Chaos Enhanced PSO Algorithm with Inertia Weight-An Initial Study. *Nostradamus: Modern Methods of Prediction, Modeling and Analysis of Nonlinear Systems, Advances in Intelligent Systems and Computing*; 2013; Berlin. 153-166. Doi: 10.1007/978-3-642-33227-2_18
- [15] Gomes GF, Cunha Jr SS, Anceleti Jr AC. A sunflower optimization (SFO) algorithm applied to damage identification on laminated composite plates. *Engineering with Computers* 2019; 35: 619-626. Doi: 10.1007/s00366-018-0620-8.
- [16] Alataş B, Akın E, Özer AB, Kaotik Haritalı Parçacık Sürü Optimizasyon Algoritmaları. *ELECO 2007 5th International Conference on Electrical and Electronics Engineering*; 2007; Bursa.
- [17] Yildirim G, Alatas B. New Adaptive Intelligent Grey Wolf Optimizer based Multi-Objective Quantitative Classification Rules Mining Approaches. *Journal of Ambient Intelligence and Humanized Computing* 2021; <https://doi.org/10.1007/s12652-020-02701-9>.
- [18] Yıldırım G, Aydın G, Alli H, Tatar Y. Hadoop ile Kaos Temelli FCW Optimizasyon Algoritmasının Analizi. *ELECO 2014 Elektrik-Elektronik-Bilgisayar ve Biyomedikal Mühendisliği Sempozyumu*; 2014; Bursa. 708-712.
- [19] May RM. Simple mathematical models with very complicated dynamics, *Nature* 1976; 261: 459-467.
- [20] Geisel T, Fairen V. Statistical Properties of Chaos in Chebyshev Maps. *Physics Letters* 1984; 105A: 263–266.
- [21] Li C, Luo G, Qin K, Li C. An image encryption scheme based on chaotic tent map. *Nonlinear Dynamics* 2017; 87: 127-133. Doi: 10.1017/s11071-016-3030-8.