



HARRAN ÜNİVERSİTESİ MÜHENDİSLİK DERGİSİ

HARRAN UNIVERSITY JOURNAL of ENGINEERING

e-ISSN: 2528-8733 (ONLINE)

URL: <http://dergipark.gov.tr/humder>

Hibrit Ateşböceği ve Parçacık Sürü Algoritmasının Kaotik Haritalar ile İyileştirilmesi

Improving Hybrid Firefly and Particle Swarm Algorithm with Chaotic Maps

Yazar(lar) (Author(s)): İbrahim Berkan AYDİLEK¹, Mehmet Emin TENEKECİ², İzzettin Hakan KARAÇİZMELİ³, Serkan KAYA⁴, Abdülkadir GÜMÜŞÇÜ⁵

¹ ORCID ID: 0000-0001-8037-8625

² ORCID ID: 0000-0001-5944-4704

³ ORCID ID: 0000-0002-8540-8050

⁴ ORCID ID: 0000-0002-1555-5493

⁵ ORCID ID: 0000-0002-5948-595X

Bu makaleye şu şekilde atıfta bulunabilirsiniz (To cite to this article): Aydilek İ.B., Tenekeci M.E., Karaçizmeli İ.H., Kaya S., Gümüşçü A., “Hibrit Ateşböceği ve Parçacık Sürü Algoritmasının Kaotik Haritalar ile İyileştirilmesi”, *Harran Üniversitesi Mühendislik Dergisi*, 4(2): 69-78, (2019).

Erişim linki (To link to this article): <http://dergipark.gov.tr/humder/archive>



Hibrit Ateşböceği ve Parçacık Sürü Algoritmasının Kaotik Haritalar ile İyileştirilmesi

İbrahim Berkan AYDİLEK¹, Mehmet Emin TENKEKİ^{2,*}, İzzettin Hakan KARAÇİZMELİ³, Serkan KAYA⁴, Abdülkadir GÜMÜŞÇÜ⁵

¹Harran Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, 63300, Haliliye/ŞANLIURFA

Öz

Optimizasyon, sınırlanmış durumlarda en uygun çözümü bulmak için uygulanan arama yöntemidir. Son yıllarda sürü tabanlı meta sezgisel algoritmaların yaygınlaşması ile başarılı sonuçlar elde edilmiştir. Bu algoritmaların güçlü yanları olduğu kadar zayıf yanları da bulunabilmektedir. Bu algoritmaların güçlü yanlarının birleştirilmesi sonucu daha iyi algoritmalar geliştirilmeye çalışılmaktadır. Buna örnek olarak ateşböceği algoritması ile parçacık sürü algoritması birlikte kullanılarak Hibrit Ateşböceği ve Parçacık Sürü Algoritması (HAPSO) geliştirilmiştir. Bu çalışmada, çözüm uzayında optimum çözümü arama için kullanılan klasik rasgele fonksiyonu yerine kaotik haritaların kullanımı denenmiştir. Arama işleminde parçacıkların hareketlerini belirleyen değişkenler kaotik harita ile elde edilen değerler ile değiştirilmektedir. Kaotik haritaların kullanımı rasgeleliğe göre daha dinamik bir arama işlemi gerçekleştirmektedir. Bu şekilde sömürü ve keşif dengelemesi sağlanmaktadır. Çalışmamızda Singer ve Iterative kaotik haritaları kullanılmıştır. Önerilen yöntemin başarısının ölçülebilmesi için CEC 2011 yarışmasında kullanılan ses sinyalinin sentezi için FM parametrelerin belirlenmesi problemi kullanılmıştır. Elde edilen sonuçlar önerilen yöntemin başarısını açık bir şekilde göstermektedir.

Makale Bilgisi

Başvuru: 06/05/2019

Düzeltilme: 19/06/2019

Kabul: 16/07/2019

Anahtar Kelimeler

Kaotik haritalar,
Hibrit yöntemler,
Meta sezgisel
algoritmalar,
Optimizasyon

Keywords

Chaotic maps,
Hybrid methods,
Meta heuristic algorithms,
Optimization

Improving Hybrid Firefly and Particle Swarm Algorithm with Chaotic Maps

Abstract

Optimization is the search method applied to find the optimal solution in restricted situations. In recent years, successful results have been achieved with the spread of swarm based meta heuristic algorithms. These algorithms may have weaknesses as well as strengths. By combining the strengths of these algorithms, better algorithms are being developed. As an example, Hybrid Firefly and Partical Swarm Optimization Algorithm (HFPSO) was developed by using firefly algorithm and particle swarm algorithm together. In this study, we have tried to use chaotic maps instead of the classical random function used to search the optimum solution in the solution space. In the scan process, the variables that determine the motion of the particles are replaced by the values obtained with the chaotic map. The use of chaotic maps performs a more dynamic search than randomness. In this way, exploitation and exploration balancing is provided. Singer and Iterative chaotic maps were used in our study. The problem which is the determination of the FM parameters for the synthesis of the audio signal, used in the CEC 2011 competition was used to measure the success of the proposed method. The obtained results clearly show the success of the proposed method.

1. GİRİŞ (INTRODUCTION)

Optimizasyon, en uygun çözümün verilen şartlar altında bulunmasıdır. Bir problem için oluşturulabilecek bir uygunluk fonksiyonunu en iyi yapacak parametrelerin çözüm uzayında bulunması, optimizasyon yöntemlerinin temel amacıdır. Özellikle son zamanlarda bilgi teknolojilerinde yaşanan büyük gelişim sonucu, optimizasyon yöntemlerinin kullanımı giderek artmıştır. Tüm optimizasyon yöntemlerinde bir rasgelelik söz konusudur. Bu rasgelelik optimizasyon işlemindeki çözüm uzayında aramanın lokal

*İletişim yazarı, e-mail: etenekeci@harran.edu.tr

minimumda kalmasına sebep olabilmektedir. Bu durum işlemin belirli bir kısır döngü içinde kalmasına sebep olmaktadır. Bu yüzden son zamanlarda yapılan çalışmalarda optimizasyonu belirli bir döngüden çıkarmak adına kaotik optimizasyon yöntemleri sıkça kullanılmaya başlanmıştır. Kaos, bir sistemin hiç beklenmedik şekilde çıkış vermesidir. Lokal minimum veya erken yakınsama problemlerinin yaşandığı fonksiyonlarda kaotik fonksiyonların ürettiği katsayılarla beraber metasezgisel algoritmaların daha iyi bir arama işlemi gerçekleştirdiği düşünülmektedir.

Meta sezgisel optimizasyon algoritmalarından olan Parçacık Sürü Optimizasyon yöntemi (PSO) kuş sürülerinin davranışları göz önünde bulundurularak önerilmiş olup kuş sürülerinin uçuş düzenlerinden esinlenerek ortaya çıkarılmıştır. PSO yöntemine ek olarak Xin-She Yang ve ark. ateşböceklerinin birbirlerini etkilemek için kurdukları haberleşme sisteminden esinlenerek ateşböceği optimizasyon yöntemini önermişlerdir. Doğadan esinlenerek önerilen optimizasyon yöntemlerinin yanında birden fazla optimizasyon yöntemi harmanlanarak hibrit modeller de önerilmiştir. Aydılek bu iki farklı optimizasyon yöntemini harmanlayarak Hibrit Ateşböceği Parçacık Sürü Optimizasyon (HAPSO) yöntemini önermiştir [1].

Literatürdeki araştırmalar, kaotik optimizasyon algoritmalarının, yerel minimum tuzağına yakalanmama ve daha iyi yakınsama süresi açısından klasik optimizasyon algoritmalarından daha başarılı olduğunu göstermektedir. Bu nedenle, son zamanlarda yapılan çalışmalar ele alındığında kaotik haritaların optimizasyon ve stokastik arama algoritmalarında kullanımının giderek arttığı görülmektedir. Kaotik optimizasyon algoritmaları, global optimizasyon problemleri için rastgele diziler üretmek için kaotik haritaları kullanır. Kaotik dizilerin olasılık yoğunluk fonksiyonu gibi istatistiksel özellikleri hesaba katıldığında, kaotik optimizasyon algoritmaları, klasik optimizasyon algoritmalarına göre yerel minimum noktalarından kaçınarak, global aramalarla optimum sonuçlara ulaşabilmektedir.

Yang ve diğ. (2014) çalışmalarında yüksek verimlilikli ve üstün performanslı kaos optimizasyon algoritması ortaya koymuştur. Yazarlar, kaotik Broyden–Fletcher–Goldfarb–Shanno (BFGS) algoritması önermiş ve performans analizini yapmışlardır [2].

Pluhacek ve diğ. (2015), meta-sezgisel performansın kaotik dizilerin uygulanmasıyla geliştiğini göstermek için PSO algoritması ile çalışmışlardır. Yazarlar, PSO algoritmasını kaotik rasgele sayılarla birleştirmişlerdir. Yazarlar kaotik PSO yönteminin daha başarılı olduğunu göstermişlerdir [3].

Hosseinpouarfard ve Javidi (2015), rastgele sayılar üretmek için Lorenz sistemi, Tent haritası ve Henon haritası kullanan yeni bir kaotik PSO algoritması önermektedir. Yazarlar önerilen algoritmanın performansının PSO, GA ve kaotik GA’ dan daha iyi olduğunu kanıtlamışlardır [4].

Yu ve diğ. (2017), yavaş yakınsama oranı ve kolayca yerel optimuma düşme gibi problemleri ortadan kaldıran dinamik bir kaotik ateşböceği algoritmasını ortaya koymuştur. Deneysel çalışmalar kaotik ateşböceği algoritmasının performansının klasik ateşböceği algoritmasından daha iyi olduğunu göstermiştir. [5]

Guvenc ve diğ. (2017), yavaş yakınsama problemini ortadan kaldırmak için on kaotik haritayı güve sürüsü algoritmasına (GSA) yerleştirmiştir. Deneysel sonuçlara göre, kaotik GSA’ nın yakınsama hızı orijinal GSA’ dan daha iyi olduğunu göstermiştir. Ayrıca, sinüzoidal haritanın diğer dokuz kaotik harita arasındaki en iyi harita olduğunu belirlemişlerdir [6].

Literatürde yapılan çalışmalar göz önünde bulundurulduğunda kaotik fonksiyonlar ile üretilen rasgele sayılar ile yapılan optimizasyon işlemlerinin, klasik optimizasyon yöntemlerine göre daha başarılı sonuçlara ulaştığı görülmektedir.

Bu çalışmamızda (HAPSO) algoritmasının rasgele şekilde değişen parametreleri, belirli kaos fonksiyonlarına bağlanarak arama uzayının daha başarılı şekilde taranması amaçlanmıştır. Böylelikle istenilen optimum değerler daha başarılı ve daha kısa sürede bulunabilecektir. Bu amaçla 2 farklı kaos

fonksiyonu kullanılarak HAPSO algoritması iyileştirilmiştir. Önerilen kaotik HAPSO algoritmasının performansı, CEC 2011 fonksiyonlarında test edilmiştir.

Makalenin geri kalan kısmında; bölüm 2' de de kaotik haritalardan, bölüm 3' Kaotik HAPSO yönteminden, bölüm 4' de test probleminde, bölüm 5'te denemeler ve elde edilen sonuçlardan ve bölüm 6' da ise yapılan çalışmanın sunduğu katkılardan bahsedilmiştir.

2. KAOTİK HARİTALAR (CHAOTIC MAPS)

Bu bölümde HAPSO algoritmasında rasgele seçilen değerlerin kaotik olarak belirlenmesi için kullanılacak kaotik haritalardan bahsedilecektir. Bu bahsedilen haritalar bir boyutlu olup tersi alınmamaktadır. Çalışmamızda 2 farklı harita denenmiştir. Bunlar Singer ve Iterative haritalardır. Hesaplamalar Singer haritası için Eşitlik 1 ve Iterative haritası için Eşitlik 2' de gösterildiği şekilde yapılmaktadır. Bu fonksiyonlarda dağılımı belirleyecek olan başlangıç noktası 0 ile 1 arasında rastgele seçilmiştir. [9] çalışmasında bu haritalar için en uygun başlangıç değeri olarak 0.7 belirlenmiştir.

$$x_{i+1} = \mu(7.86x_i - 23.31x_i^2 + 28.75x_i^3 - 13.302875x_i^4), \mu = 1.07 \quad (1)$$

$$x_{i+1} = \sin\left(\frac{a\pi}{x_i}\right) \quad (2)$$

3. KAOTİK HİBRİT ATEŞBÖCEĞİ PARÇACIK SÜRÜ OPTİMİZASYONU (CHAOTIC HYBRID FIREFLY AND PARTICAL SWARM OPTIMIZATION)

Son yıllarda, doğadan ilham alan optimizasyon algoritmalarının kullanımı yaygınlaşmıştır. Özellikle sürü tabanlı hareket ederek hedefe ilerleyen doğadaki davranışlar bilgisayarda modellenerek değişik problemlerin çözümünde kullanılmıştır. Çalışmamızda HAPSO algoritması kaotik haritalar kullanılması ile iyileştirilmiştir.

HAPSO algoritmasında kullanılan ateşböceği algoritması doğadan esinlenmiş bir, sürü tabanlı meta sezgisel algoritmadır. Bu algoritmada, ateşböceklerinin cinsiyetsiz olduğu varsayılmaktadır, bu şekilde her birey yinelemeli olarak diğerine yaklaşır. Bu davranışın modellenmesi ve optimizasyon algoritması olarak kullanılması Yang tarafından önerilmektedir [7]. Ateşböcekleri sürü içinde hareket ettikçe, her biri en parlak olana doğru yaklaşır. Ateş böceklerinin hareketlerini etkileyen, çevredeki diğer ateş böceklerinin ışıklarıdır. Işığın şiddeti ise parlaklığa ve mesafeye bağlı olarak değişir. Mesafe arttıkça ışığın yoğunluğu azalır. Bu nedenle, herhangi bir ateşböceği için uzaktaki ateşböceği yakındakilerden daha az etkili olacaktır. Bu faktörler göz önüne alınarak model geliştirilmiştir. Işık yoğunluğunun etkisi Eşitlik 3 ile ifade edilmiştir. $I(r)$, bir r mesafesine yayılacak ışığın miktarını, I_s ışık kaynağından yayılan ışığın miktarını ve r ise mesafeyi ifade eder.

$$I(r) = \frac{I_s}{r^2} \quad (3)$$

Parçacığın ışığı emilim miktarı ortamın özelliklerine göre değişir. Ateşböceklerinin çekiciliği Eşitlik 4' de ifade edilmiştir. Burada γ değeri emilim katsayısı olarak alınır. B_0 , ateşböceğinin 0 mesafesindeki çekiciliğini ifade eder.

$$B(r) = B_0 e^{-\gamma r^2} \quad (4)$$

Öncelikle ateş böceklerinin ilk konumları rasgele olarak belirlenir. Eşitlik 3 ile, X_i ' nin bir sonraki iterasyondaki yeni pozisyonu hesaplanır. Eşitlik 5' de gösterilen X_i ve X_j , iki ateşböceğini temsil eder. ϵ_i , rastgele değişkenlerin vektörüdür ve α , rastgeleliğin parametresidir. Algoritma, yinelemeli olarak hesaplanan her yeni pozisyondaki en uygun çözümü bulmaya çalışır.

$$X_i = X_i + B_0 e^{-\gamma r_{ij}^2} (X_j - X_i) + \alpha \epsilon_i \quad (5)$$

Bahsedilen işlemler ile her bir ateşböceği en uygun çözüme yaklaştırılır. Bu işlemler belirlenmiş sayıda gerçekleştirilerek optimum çözüme ulaşılmaya çalışılır.

HAPSO' nun diğer algoritması olan PSO, Kennedy ve Eberhart tarafından [8]' de önerilmiştir. Ateşböceği algoritması gibi, sürü davranışlarını taklit etmektedir. Yöntem, kuşların veya balıkların yiyecek çevresinde hareketlerini modelleyerek en uygun çözümü aramaktadır. Algoritma, her parçacığın yiyeceğe ulaşması için her adımdaki konumunu günceller. Her bir iterasyonda, parçacıklar için en iyi (pbest) ve parçacıkların tümü için en iyi çözüm olan global en iyi (gbest) değerleri göz önünde bulundurularak işlemler gerçekleştirilir.

$$v_i^{k+1} = w \cdot v_i^k + c_1 \cdot rand_1^k \cdot (pbest_i^k - x_i^k) + c_2 \cdot rand_2^k \cdot (gbest_i^k - x_i^k) \quad (6)$$

Eşitlik 6' de, x_i parçacıkların pozisyonu, w atalet ağırlığını ve v_i parçacıkların hızını belirtir. c_1 ve c_2 , hızlanma parametreleridir. $rand_1$ ve $rand_2$ [0, 1] aralığında değişen rastgele değerlerdir.

Eşitlik 7' de gösterildiği şekilde her bir parçacığın yeni pozisyonu hesaplanır. x_i^{k+1} parçacığın yeni pozisyonunu ifade ederken, x_i^k parçacığın mevcut pozisyonunu ve v_i^{k+1} ise hızını ifade etmektedir.

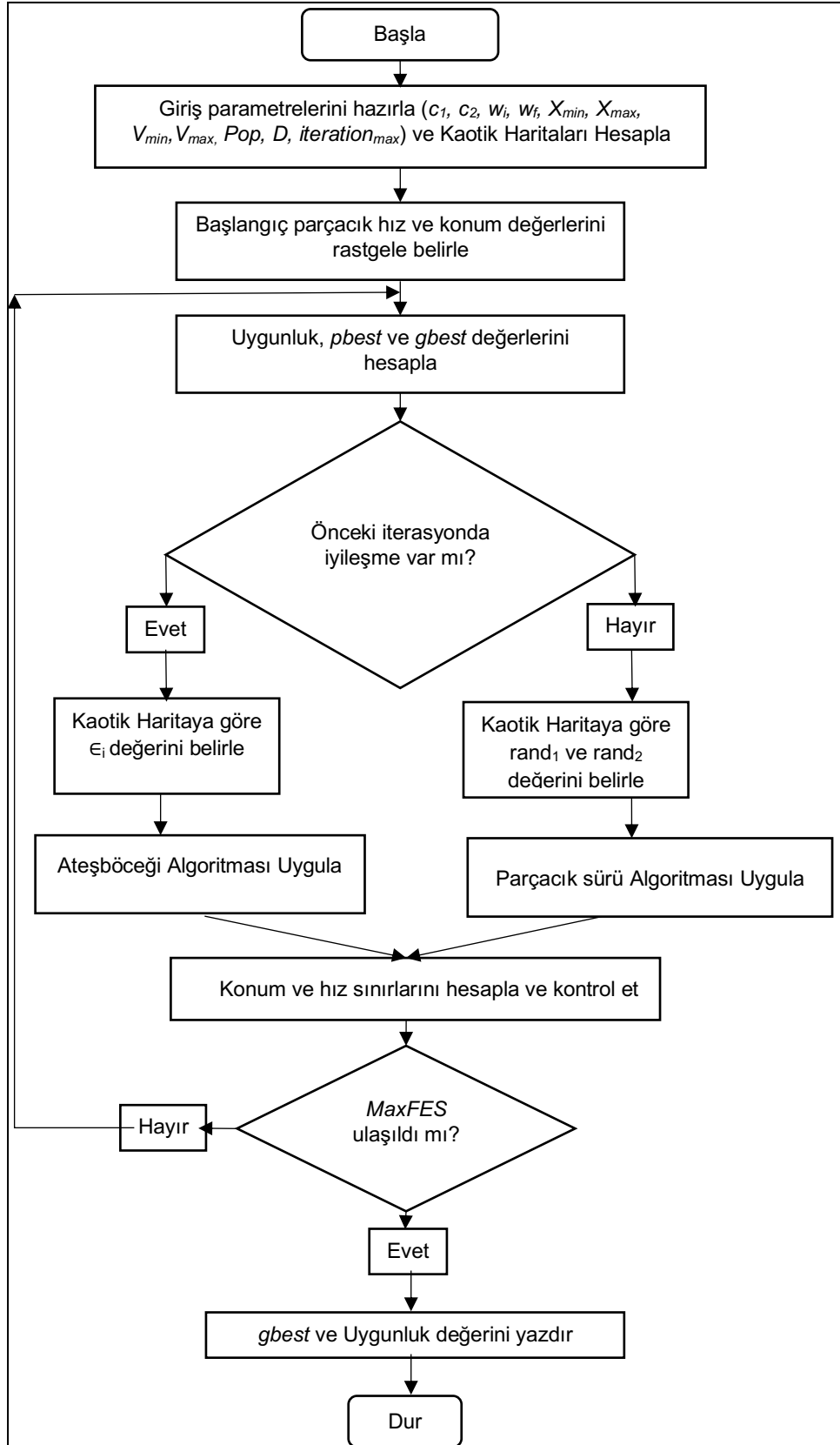
$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (7)$$

Metasezgisel algoritmalar ile optimizasyon uygulamalarında oldukça başarılı sonuçlar elde edilebilmektedir. Ancak yöntemlerin başarılı olduğu problemler veya arama yöntemleri farklılık gösterebilmektedir. Bu nedenle mevcut yöntemler birlikte hibrit kullanılarak daha etkin sonuçlar alınabilmektedir. Aydılek [1] tarafından önerilen HAPSO, ateşböceği algoritması ile parçacık sürü algoritmasının güçlü yanlarını birlikte kullanarak daha başarılı sonuçlar elde eden başarılı bir hibrit algoritmadır.

PSO algoritması V hız parametresi ve en iyi değerini koruması sebebiyle oldukça başarılı ve hızlı arama işlemi gerçekleştirmektedir. Bu özelliği ile global aramada etkin bir yöntem olarak literatürde yer etmiştir. Ancak bunun yanında yerel aramalarda salınım yaparak sonuca ulaşması gecikebilmektedir. Bundan dolayı sömürü özelliği daha zayıf kalmaktadır.

Ateşböceği algoritmasının ise hız parametresi ve geçmiş en iyi değeri tutabilecek yeteneği bulunmamaktadır. Bu sebepten yerel koşullara göre en uygun çözümü belirleyebilmektedir. Böylece başarılı sömürü gerçekleştirmektedir. Ancak PSO' ya göre daha yavaş global sonuca ulaşmaktadır.

Aydılek, HAPSO algoritmasında Ateşböceği ve PSO algoritmalarının arama kabiliyetlerini birleştirerek kullanmıştır. PSO' nun hızlı arama özelliği ile global çözümlerde ve ateşböceğinin başarılı sömürme özelliği ile yerel çözümlerde üstün olan özellikleri birleştirilmiştir. Önerdiği yöntemi ile yalnız PSO ve ateşböceği algoritmalarından daha başarılı sonuç elde ettiğini göstermiştir.



Şekil 1. Önerilen CHAPSO algoritmasının akış diyagramı

Çalışmamızda mevcut HAPSO algoritmasında parametre değerlerinin rasgele alınması yerine kaotik haritalardan üretilmiş sayıların kullanılması önerilmektedir. Parçacık sürü algoritmasında kullanılan Eşitlik 6' te bulunan $rand_1$ ve $rand_2$ parametreleri ve ateşböceği algoritmasındaki Eşitlik 8' de bulunan ϵ_i parametresi 0 ile 1 arasında rastgele üretilmektedir. Kullanılan rasgele fonksiyonlar düzenli (uniform)

formda olmaktadır. Çalışmamızda bu üç parametrenin kaotik haritalarla daha dinamik bir şekilde üretilen değerler ile belirlenmesinin başarı oranını arttıracacağı düşünülmektedir.

$$X_i(t+1) = X_i(t) + B_0 e^{-\gamma r_{i,j}^2} (X_i(t) - gbest^{t-1}) + \alpha \epsilon_i \quad (8)$$

Optimizasyon algoritmaları, sonuca ulaşmak için çözüm alanını taramaktadır. Tarama işlemi bazı parametrelerin değerlerini değiştirerek yapılmaktadır. Geleneksel algoritmalarda bu değişim için düzgün dağılıma sahip rastgele fonksiyonlar kullanılır. Çalışmamızda, daha karmaşık dağılımlara sahip olan kaotik haritalar rastgele değişkenlerin değerlerini belirlemede kullanılmıştır.

Giriş kısmında bahsedilen benzer çalışmalarda görüldüğü gibi son yıllarda optimizasyon algoritmalarının performansının rastgele fonksiyonlar yerine kaotik haritaların kullanılmasıyla artırıldığı görülmüştür.

4. SES DALGALARI İÇİN FREKANS MODULASYONUNDA PARAMETRE BELİRLEME PROBLEMİ (PARAMETER DETERMINATION PROBLEM FOR FREQUENCY MODULATION FOR SOUND WAVES)

Seslerin frekans modülasyonu ile sentezlenmesi modern müzik sistemleri için önemli bir rol oynamaktadır. Bu işlem için 6 boyutlu bir problemin optimize edilmesi gerekmektedir. Problemin karmaşık yapısından dolayı CEC 2011 [10] yarışmasında optimizasyon algoritmalarının karşılaştırılması için test problemi olarak kullanılmıştır.

Problemin modellenmesinde $X = \{a_1, \omega_1, a_2, \omega_2, a_3, \omega_3\}$ olacak şekilde 6 parametre kullanılmaktadır. Eşitlik 9 ile oluşturulan ses sinyali Eşitlik 10' da verilen sinyale benzediği oranda doğru sonuca ulaşılabilecektir.

$$y(t) = a_1 \cdot \sin(w_1 \cdot t \cdot \theta + a_2 \cdot \sin(w_2 \cdot t \cdot \theta + a_3 \cdot \sin(w_3 \cdot t \cdot \theta))) \quad (9)$$

$$y_0(t) = (1.0) \cdot \sin((5.0) \cdot t \cdot \theta - (1.5) \cdot \sin((4.8) \cdot t \cdot \theta + (2.0) \cdot \sin((4.9) \cdot t \cdot \theta))) \quad (10)$$

Problemin uygunluk fonksiyonu Eşitlik 9 ile $y(t)$ üretilen ses sinyalini temsil ederken, Eşitlik 10 ile elde edilen $y_0(t)$ gerçek ses arasındaki hatanın karesinin toplamıdır. Problemden $\theta = 2\pi/100$ ve parametreler $[-6.4, 6.35]$ arasında değeri alabilmektedir.

5. DENEYLER VE BULGULAR (EXPERIMENTS AND RESULTS)

Deneyel çalışma için parçacık büyüklüğü 50, iterasyon sayısı 1000 ve tekrar sayısı 30 olarak belirlenmiştir. Önceki bölümde açıklandığı gibi öncelikle HAPSO algoritması 2 farklı kaotik harita ile kaotik hale getirilmiştir. Ardından bu kaotik haritalar kullanılarak oluşturulan CHAPSO algoritmalarının performansları ile HAPSO algoritmasının performansı ve [11]' de incelenen 10 algoritmanın performansları karşılaştırılmıştır.

2 farklı kaotik harita ile kaotik hale getirilen HAPSO algoritmasının FM probleminde her harita için 30 tekrar olacak şekilde denemesinin ardından elde edilen sonuçların ortalamaları ve standart sapmaları Tablo 1' de verilmiştir. En düşük ortalama değere göre yapılan sıralama da yine Tablo 1' de görülmektedir. Bu sıralamaya göre en düşük ortalama sahip CHAPSO algoritması Singer haritası ile kaotik hale getirilen algoritmadır.

Tablo 1. Kaotik haritalara göre amaç fonksiyonu değerleri

Harita	Ortalama	Std. Sapma	Sıralama
Singer	14,10	5,9294	1
Iterative	14,45	4,8862	2

İki kaotik harita ile elde edilen sonuçların istatistiksel olarak da farklı olup olmadığını anlamak için Mann-Whitney testi ile medyan, Levene testi ile de varyans analizleri yapılmıştır. %95 güven düzeyi için sonuçlar (p-değerleri) Tablo 2’ de görülmektedir. Elde edilen sonuçlara göre, singer ve iterative haritalar için hem varyans değerleri hem de medyan değerleri farklı değildir. Tablo 3’ de ise bu iki harita ile elde edilen çözümlere ulaşma sürelerinin istatistiki bilgileri görülmektedir. Görüleceği üzere singer ve iterative haritaların çözüm süreleri aynıdır. Bu durum istatistiksel olarak da p-değeriyle desteklenmektedir.

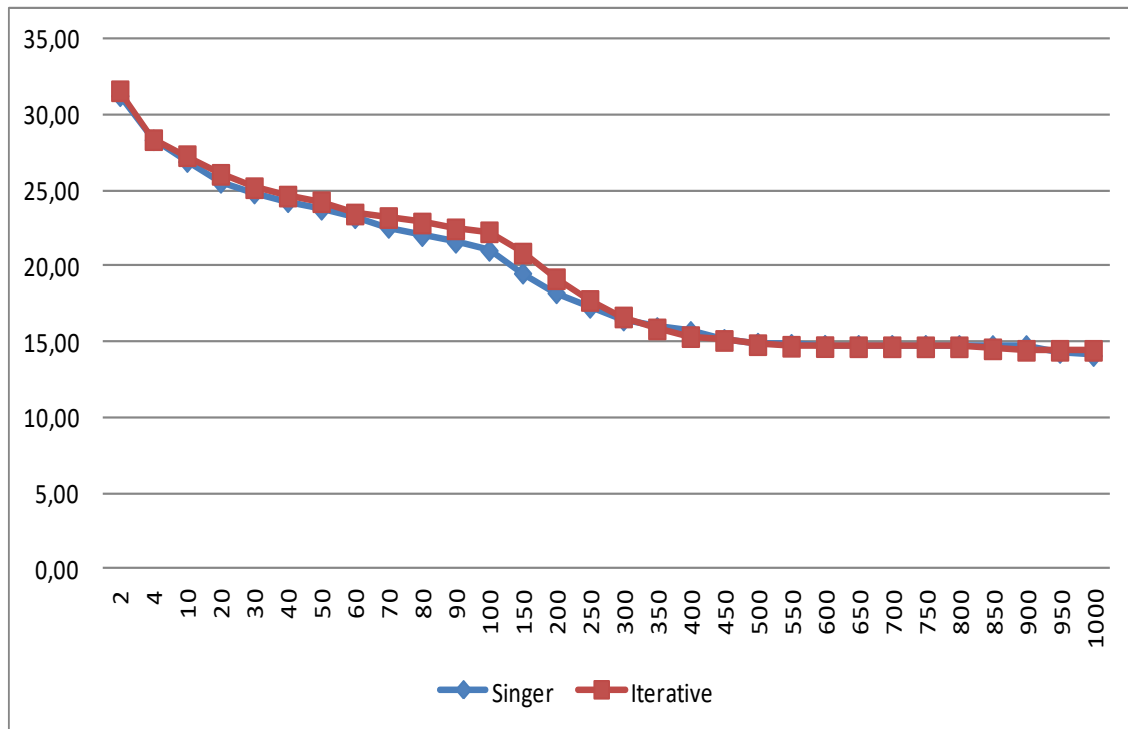
Tablo 2. Kaotik haritalara göre istatistiksel analiz sonuçları

Harita	Medyan Analizi	Varyans Analizi
Singer	-	-
Iterative	0,6467	0,2380

Tablo 3. Kaotik haritalara göre çözüm süreleri (sn)

Harita	Ortalama Süre	Std. Sapma	p-değeri
Singer	1,15	0,0559	-
Iterative	1,15	0,0586	0,7890

Şekil 2’de ise singer ve iterative haritalara göre CHAPSO algoritmasının yakınsama hızları görülmektedir. İki kaotik harita arasında yakınsama anlamında da belirgin bir fark yoktur. Her iki haritada da 1000 iterasyona aynı sürede ulaşılmıştır. Deney tekrarlarının ortalamalarına göre hazırlanan bu grafikte ise en iyi çözüme iniş hızlarının da birbirinden ayrılmadığı görülmektedir. Bu durumda performanslar arasında bir farklılık olmadığına göre her iki harita da HAPSO algoritmasının kaotik hale getirilmesi için tercih edilebilir.



Şekil 2. CHAPSO algoritması yakınsama grafiği

[11]' de incelenen 10 algoritmanın performansları ile HAPSO ve singer haritası kullanılan CHAPSO algoritmalarının performansları Tablo 4' de, HAPSO ve iterative harita kullanılan CHAPSO algoritmalarının performansları ise Tablo 5' de karşılaştırılmıştır. Gerek singer harita gerekse iterative harita kullanılan CHAPSO algoritması ile hem HAPSO algoritmasından hem de diğer 10 algoritmadan daha iyi ortalama değerler elde edilmiştir.

Tablo 4. KHAPSO (Singer Harita) ile Diğer Algoritmaların Performanslarının Karşılaştırması

Harita	Ortalama	Std. Sapma	P- değeri	Sıralama
Singer CHAPSO	14,10	5,9294	-	1
HAPSO [1]	17,45	6,3516	0,0080	2
Chaotic BFO [11]	17,79	2,0168	0,0020	3
FA [7]	17,92	5,5999	0,0010	4
PSO [8]	18,09	4,2074	0,0010	5
MFO [12]	18,11	5,7084	0,0010	6
GOA[13]	18,71	4,3338	0,0000	7
FPA [14]	18,92	2,3435	0,0000	8
SSA [15]	19,78	3,3975	0,0000	9
SCA [16]	21,19	6,2048	0,0000	10
BA [17]	22,21	3,2132	0,0000	11
BFO [18]	23,25	1,9390	0,0000	12

FA: Firefly algorithm, MFO: Moth-Flame Optimization, GOA: Grasshopper Optimization algorithm, FPA: Flower Pollination algorithm, SSA: Salp swarm algorithm, SCA: Sine Cosine algorithm, BA: Bat algorithm, BFO: Bacterial Foraging Optimization algorithm.

Tablo 5. KHAPSO (Iterative Harita) ile Diğer Algoritmaların Performanslarının Karşılaştırması

Harita	Ortalama	Std. Sapma	p- değeri	Sıralama
Iterative CHAPSO	14,45	4,8862	-	1
HAPSO [1]	17,45	6,3516	0,0040	2
Chaotic BFO [11]	17,79	2,0168	0,0010	3
FA [7]	17,92	5,5999	0,0010	4
PSO [8]	18,09	4,2074	0,0010	5
MFO [12]	18,11	5,7084	0,0010	6
GOA[13]	18,71	4,3338	0,0000	7
FPA [14]	18,92	2,3435	0,0000	8
SSA [15]	19,78	3,3975	0,0000	9
SCA [16]	21,19	6,2048	0,0000	10
BA [17]	22,21	3,2132	0,0000	11
BFO [18]	23,25	1,9390	0,0000	12

Tablo 4 ve Tablo 5' de Wilcoxon işaretli sıralar testi ile yapılan istatistiksel analizlerin p-değerleri de görülmektedir. Bu değerlere göre her iki tablonun birincisi CHAPSO algoritmasıdır. Yani singer veya

iterative harita ile elde edilen CHAPSO algoritması hem HAPSO algoritmasının performansını artırmıştır hem de tablolarda verilen diğer algoritmalarla istatistiksel olarak anlamlı olacak şekilde daha iyi sonuçlar vermiş, maliyet fonksiyonunun daha düşük değerlere ulaşmasını sağlamıştır.

5. SONUÇ (CONCLUSION)

Bu çalışmada hibrit bir algoritmadaki rasgele değişkenlerin değerlerini kaotik haritalar ile değiştirerek optimizasyon probleminde başarı oranı artırılmıştır. Literatürde bahsedilen güncel algoritmalar ile karşılaştırmalar yapılarak sonuçlar ifade edilmiştir. Çalışmada HAPSO algoritmasındaki rasgele değişkenler, haritalar tarafından oluşturulan kaotik değerlerle değiştirilmiştir. Kaotik haritaların doğasından dolayı arama uzayını taramada kaotik haritanın kullanılmasıyla yerel minimum tuzağına düşmeden daha başarılı sonuçlar elde edilmiştir. Bu çalışma ile benzer hibrit algoritmaların kaotik hale getirilerek daha başarılı hale getirilebileceği görülmüştür.

TEŞEKKÜR (ACKNOWLEDGMENTS)

Bu çalışma TÜBİTAK tarafından 118E355 numaralı, "Akış Tipi Çizelgeleme Probleminin Yeni Kaotik Metasezgisel Optimizasyon Algoritmaları ile Çözülmesi" başlıklı proje ile desteklenmiştir. Katkılarından dolayı teşekkür ederiz.

KAYNAKLAR (REFERENCES)

- [1] I. B. Aydilek, "A hybrid firefly and particle swarm optimization algorithm for computationally expensive numerical problems," *Applied Soft Computing*, vol. 66, pp. 232-249, May 2018.
- [2] D. X. Yang, Z. J. Liu, and J. L. Zhou, "Chaos optimization algorithms based on chaotic maps with different probability distribution and search speed for global optimization," *Communications in Nonlinear Science and Numerical Simulation*, vol. 19, no. 4, pp. 1229-1246, Apr 2014.
- [3] B. Liu, L. Wang, Y. H. Jin, F. Tang, and D. X. Huang, "Improved particle swarm optimization combined with chaos," *Chaos Solitons & Fractals*, vol. 25, no. 5, pp. 1261-1271, Sep 2005.
- [4] R. Hosseinpourfard, M. Javidi. Chaotic PSO using the Lorenz System: An Efficient Approach for Optimizing Nonlinear Problems. *Cankaya University Journal of Science and Engineering*, 12 (1), 2015, Retrieved from <http://dergipark.gov.tr/cankujse/issue/33130/368659>
- [5] W. X. Yu, J. N. Wang, Y. L. Li, and Z. H. Wang, "The chaos and stability of firefly algorithm adjacent individual," *TELKOMNIKA*, vol. 15, no. 4, pp. 1733-1740, 2017.
- [6] U. Guvenc, S. Duman, and Y. Hınıslıoğlu, "Chaotic moth swarm algorithm," presented at the IEEE International Conference on Innovations in Intelligent Systems and Applications, 2017.
- [7] X. S. Yang, "Firefly algorithms for multimodal optimisation," *Proc. 5th Symposium on Stochastic Algorithms Foundations and Applications*, pp. 169-178, 2009.
- [8] J. Kennedy and R. Eberhart, "Particle swarm optimization," presented at the Proceedings of ICNN'95 - International Conference on Neural Networks, Australia, 1995.
- [9] S. Saremi, S. Mirjalili, A. Lewis, Biogeography-based optimisation with chaos, *Neural Computing and Applications*, In press, 2014, Springer, DOI: <http://dx.doi.org/10.1007/s00521-014-1597-x>
- [10] S. Das, P.N. Suganthan, "Problem definitions and evaluation criteria for CEC 2011 competition on testing evolutionary algorithms on real world optimization problems." *Jadavpur University, Nanyang Technological University, Kolkata*, 2010, 341-359.
- [11] Q. Zhang, H. Chen, J. Luo, Y. Xu, C. Wu ve C. Li, "Chaos Enhanced Bacterial Foraging Optimization for Global Optimization", *IEEE Access*, 2018 (6), 64905-64919.
- [12] Mirjalili, S., Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowledge-Based Systems*, 2015. 89: p. 228-249.
- [13] Luo, J., et al., An Improved Grasshopper Optimization Algorithm with Application to Financial Stress Prediction. *Applied Mathematical Modelling*, 2018.

- [14] Yang, X.-S. Flower Pollination Algorithm for Global Optimization. 2012. Berlin, Heidelberg: Springer Berlin Heidelberg.
- [15] Mirjalili, S., et al., Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Advances in Engineering Software*, 2017. 114: p. 163-191.
- [16] Mirjalili, S., SCA: A Sine Cosine Algorithm for solving optimization problems. *Knowledge-Based Systems*, 2016. 96: p. 120-133.
- [17] Yang, X.S., A new metaheuristic Bat-inspired Algorithm, in *Studies in Computational Intelligence*. 2010. p. 65-74.
- [18] Passino, K.M., Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Systems*, 2002. 22(3): p. 52-67.