

Lojistik-Gauss Harita Tabanlı Yeni Bir Kaotik Sürü Optimizasyon Yöntemi

Fahrettin Burak Demir^{1*}, Türker Tuncer², Adnan Fatih Kocamaz³

¹Malatya Turgut Özal Üniversitesi Bilgisayar Teknolojileri Bölümü, Malatya, Türkiye

(fahrettin.demir@ozal.edu.tr)

²Fırat Üniversitesi, Adli Bilişim Mühendisliği, Elazığ, Türkiye (turkertuncer@firat.edu.tr)

³İnönü Üniversitesi, Bilgisayar Mühendisliği, Malatya, Türkiye (fatih.kocamaz@inonu.edu.tr)

Received: Dec.26, 2018

Accepted: Jan.21, 2019

Published: Jun.1, 2019

Özet— Gerçek hayattaki bazı problemler, klasik matematiksel yöntemler kullanarak çözülememektedir. Bu sebeple, bu problemlerin çözümünde genellikle meta-sezgisel optimizasyon yöntemleri kullanılmaktadır. Meta-sezgisel optimizasyon yöntemlerinin başarı düzeyini artırmak için kullanılan yöntemlerden birisi de kaotik haritalardır. Bu makalede yeni bir hibrit kaotik harita önerilmiş ve önerilen hibrit kaotik harita kullanılarak yeni bir kaotik optimizasyon yöntemi geliştirilmiştir. Önerilen hibrit kaotik harita, lojistik ve gauss haritalarını kullanmaktadır. Optimizasyon aşamasında parçacıkların değerlerini güncellemek için lojistik-gauss haritası kullanılmıştır. Önerilen optimizasyon yöntemi de lojistik-gauss tabanlı kaotik sürü algoritması olarak adlandırılmıştır. Yeni optimizasyon yönteminin performansını test etmek için ise literatürde sıklıkla kullanılan 6 sayısal kıyaslama fonksiyonu tercih edilmiştir. Ve elde edilen değerler yine literatürde yer alan 3 farklı sürü tabanlı optimizasyon yöntemiyle karşılaştırılmıştır. Önerilen yöntem hemen bütün fonksiyonlar için daha optimum sonuçlar üretmiştir. Bu sayede sürü optimizasyon yönteminin, tuzaklı lokal optimum değerlerden uzak durması sağlanmaya çalışılmıştır.

Keywords : Kaotik optimizasyon yöntemleri, Lojistik harita, Gauss haritası, Kıyaslama fonksiyonları, Sürü optimizasyon algoritması.

Abstract— Some real life problems can not be solved using classical mathematical methods. For this reason, meta-heuristic optimization methods are usually used to solve these problems. One of the methods used to increase the success level of meta-heuristic optimization methods is chaotic maps. In this article, a new hybrid chaotic map was proposed and a new chaotic optimization method was developed using the proposed hybrid chaotic map. The proposed chaotic method uses logistics and gauss maps. The logistic-gauss map is used to update the values of the particles during the optimization phase. The proposed optimization method is called logistic-gauss chaotic particle swarm algorithm. In order to test the performance of the new optimization method, 6 numerical benchmark functions, which are frequently used in the literature, are preferred. And the obtained values are compared with the 3 different particle-based optimization methods in the literature. The proposed method produces more optimal results for almost all functions. In this way, it was tried to prevent the trapped local optimum solutions of the swarm optimization method.

Keywords : Chaotic optimization method, Logistic map, Gauss map, Benchmark Functions, Swarm optimizations

1.Giriş

Dünyada var olan hesapsal problemlerin büyük bir kısmının çözümünde bilgi teknolojileri yaygın olarak kullanılmaktadır. Bu problemlerin pek çoğu klasik matematiksel çözüme sahip değildir. Bu sebeple, bu problemlerin çözümünde uzunca bir süredir meta-sezgisel optimizasyon yöntemleri kullanılmaktadır. Meta-sezgisel optimizasyon yöntemlerinin temel amacı, sonlu bir arama uzayı içerisinde küresel en iyi noktayı(optimum) bulabilmektir [1]. Bu yöntemler en iyi çözüm değerlerini ve uygunluk fonksiyonları kullanarak arama alanını küçültmektedir. Bu sayede arama uzayı içerisindeki çok sayıda çözümden en iyisini bulmayı taahhüt etmektedir [2].

Meta-sezgisel yöntemler genellikle doğadaki olayları ve durumları modellemektedir. Ayrıca spor tabanlı, kültürel-sosyal tabanlı, müzik tabanlı, matematik tabanlı modellemeler de mevcuttur [3]. Bu modellemelere, balık ve kuş sürülerinin hareketlerini temel alarak geliştirilen Parçacık Sürü Optimizasyon Algoritması (PSO) [4], evrenin var oluşunu modelleyen Büyük Patlama–Büyük Çöküş Optimizasyon Algoritması (Big Bang-Big Crunch Optimization Algorithm-BB-BCOA) [5], parçacık güncellemeleri için matematiksel fonksiyonları kullanan Sinüs-Kosinüs Optimizasyon Algoritması (Sine-Cosine Optimization Algorithm-SCOA) [6,7], evrimsel optimizasyon algoritmalarını modelleyen, aynı zamanda kültürel-sosyal bir algoritma da olan ve en iyi ülkenin emperyalist olarak seçildiği Emperyalist Yarışmacı Algoritması [8] ve müzisyenlerin çaldıkları notaları modelleyen Armoni Arama Algoritması [9] (Harmony Search-HS) örnek verilebilir.

Kaotik haritalar, karmaşık problemleri modellemede, sayısal analiz yöntemlerinde, görüntü şifrelemede, kriptolojide ve sezgisel optimizasyon yöntemlerinde sıkça kullanılmaktadır. Meta-sezgisel optimizasyon yöntemlerindeki gibi bu çalışmada da uzun periyotlu rastsal sayı dizileri bulunmaktadır. Oluşturulacak sayıların aynı olmaması, yayılmış değerler alması, saklama maliyetlerinin düşük olması ve fazla zamana ihtiyaç duymamaları istenilen durumlardır. Kaotik haritaları kullanarak yerel optimumlardan kaçmak mümkündür [10]. Kaotik sayı dizileri; yapay sinir ağlarında [11], kaotik beyin fırtınası optimizasyon algoritmasında [12], kaotik yaklaşımla çok amaçlı transformator tasarımında [13], veri iletişim sürecinde şifreleme ve sıkıştırma için kaotik yöntemlerin analizinde [14] kullanılmıştır.

Kaotik sayı dizilerinin yapılandırılması ve oluşturulması hızlı ve kolaydır. Basit başlangıç parametreleri ve fonksiyonları ile istenilen uzunlukta dizileri elde etmek oldukça basittir [10,15]. Kaotik haritalarla üretilen sayıların geniş yayımlı ve tahmin edilemez ve periyodik olmayan bir özellikte olduğu teorik olarak ispatlanmıştır [10,16].

Çalışmanın öne çıkan yönleri aşağıdaki gibidir.

- Bu çalışmada yeni bir tek boyutlu hibrit kaotik harita olan lojistik-gauss harita önerilmiştir.
- Kaotik haritalar istatistiksel olarak iyi sonuçlar üretmektedir. Bu özelliği doğrudan optimizasyon yönteminde kullanmak için diferansiyel tabanlı yeni bir kaotik optimizasyon yöntemi önerilmiştir. Önerilen yöntemde hız parametresi lojistik-gauss harita kullanılarak hesaplanmıştır.

Çalışmanın 2. bölümünde kaotik haritanın tanımı ve çalışmada kullanılan kaotik haritalara ait bilgiler verilmiştir. 3. bölümde, önerilen yeni Kaotik Lojistik-Gauss tabanlı Sürü Optimizasyon Algoritması(KLG-SOA) tanıtılmıştır. Çalışmanın 4. bölümünde deneysel sonuçlara, 5. bölümde ise sonuçlar ve önerilere yer verilmiştir.

2. Materyal

Bu makalede yeni bir kaotik harita elde etmek için lojistik ve gauss haritaları hibrit bir şekilde kullanılmıştır. Ayrıca bu hibrit yöntemle lojistik ve gauss haritaların kaotik etki alanı iyileştirilmiştir. Bu bölümde, kaotik haritalar, lojistik harita, gauss haritası ve hibrit kaotik model anlatılmıştır.

2.1. Kaotik Haritalar

Karmaşık yapıları modellemede ve bunlara ait sayısal analizlerde, uzun periyotlu sayı dizileri sıklıkla kullanılmaktadır. Bu uzun periyotlu sayı dizilerinin hafızada çok fazla yer kaplamaması adına olabilecek en iyi başlangıç değerleri için en iyi veri kümesinin seçilmesi gerekmektedir. Kaos, başlangıç şartlarına ve belirlenen parametrelere bağlı olarak düzensizlik içerisinde düzeni ifade eden, içerisinde periyodik ve doğrusal olmayan dinamik sistemleri barındıran deterministik ve rastsal bir süreçtir [15]. Kaotik haritalar ise ayrık zamanlı dinamik sistemlerdir.

$$x_{i+1} = f(x_i), 0 < x_i < 1, i = 0, 1, 2, \dots \quad (1)$$

Eşitlik 1 kaotik haritaların temel matematiksel tanımını temsil etmektedir [15]. Optimizasyon tekniklerinde de sıklıkla kullanılan kaotik haritalar şunlardır: Chebyshev, Daire, Gauss, İteratif, Lojistik, Parçalı, Sinüs, Singer, Sinüsoidal ve Çadır [20].

2.2. Lojistik Harita

Lojistik haritalar literatürde sıklıkla kullanılan haritalardan birisidir. Lojistik haritalar rastgele sayı üretiminde, imge şifreleme ve optimizasyon yöntemlerinde kullanılmaktadır [15]. Lojistik haritaya ait matematiksel tanım aşağıdaki gibidir.

$$x_{i+1} = rx_i(1 - x_i), r \in (3.57, 4), x \in (0, 1) \text{ and } x \neq \{0.25, 0.5, 0.75\} \quad (2)$$

Bu harita kullanılarak 0,1 aralığında rastgele sayılar üretilmektedir [17].

2.3. Gauss Harita

Literatürde sıklıkla kullanılan diğer bir harita da Gauss haritasıdır. Gauss haritasına ait matematiksel tanım aşağıda verilmiştir [18].

$$X_{r+1} = \begin{cases} 0 & , X_r = 0 \\ 1/X_r \bmod(1) & , X_r \in (0, 1) \end{cases} \quad (3)$$

$$1/X_r \bmod(1) = \frac{1}{X_r} - \left\lfloor \frac{1}{X_r} \right\rfloor$$

Bu harita da (0,1) aralığında sayı üretmektedir.

2.4. Hibrit Lojistik-Gauss Haritası

Bu çalışmada önerilen yeni kaotik harita, lojistik ve gauss haritalarını kullanmaktadır. Ve önerilen haritanın temel amacı, lojistik ve gauss haritalarından daha iyi kaotik özellikler gösteren yeni bir harita oluşturmaktır. Önerilen lojistik-gauss haritasına ait matematiksel tanım aşağıda verilmiştir.

$$r(i+1) = \bmod\left(\frac{1}{4 * r(i) * (1 - r(i))}, 1\right) \quad (4)$$

3. Önerilen Kaotik Lojistik-Gauss Tabanlı Sürü Optimizasyon Yöntemi

Bu çalışmada, lojistik-gauss tabanlı sürü optimizasyon yöntemi önerilmiştir. Bu yöntemin en temel amacı global optimum noktayı bulmaktır. Literatürde kaotik arama algoritmaları ilk defa Hamaizia ve ark. [19] tarafından önerilmiştir. Bu yöntemde Hamaizia ve ark.'nın önerdiği yöntemin aksine adım

büyüklüğü değil lojistik-gauss haritasının ürettiği değerler kullanılmaktadır. Önerilen yöntem; en iyi değeri seçme, tohum değerini oluşturma, hibrit harita yöntemi ile rastgele sayı üretme, parçacık güncelleme ve en iyi değeri güncelleme aşamalarından meydana gelmektedir.

Önerilen yöntemin sözde kodu Şekil 1.'de verilmiştir. Burada 5.satırda hibrit denklemin kodu verilmiştir. 8. satırda ise parçacık adım hesabına ait denklem verilmiş olup, sonraki adımlarda pbest 'in hesaplanması ve güncellenmesi işlemleri gösterilmiştir.

```

1: Parametreleri Belirle (LB: Alt Sınır, UB: Üst Sınır, MI: Maksimum İterasyon, pn: Parçacık Sayısı)
2: Parçacıkları Başlat: Parçacık Değerlerini Rastsal Üret
3: Her Parçacığı Hesapla ve Kişisel En İyi Değeri Hesapla (pbest)
4: for k=1 : MI do
5: r(i+1)=mod(1/4*r(i)*(1-r(i)),1);
6: for i=1 : pn do
7: r Değerini Oluştur
8: pi=pi + ri(pbest - pi)
9: if pi<LB veya pi>UB then
10: pi=pbest
11: end if
12: Uygunluk Fonksiyonlarını Kullanarak Parçacık Değerlerini Güncelle
13: if pi<pbest then
14: pi=pbest
15: end if
16: if pbest=istenilen_deger then
17: break;
18: end if
19: end for i

```

Şekil 1. Lojistik-Gauss Tabanlı Kaotik Sürü Optimizasyonuna Ait Sözde Kod

4. Deneysel Sonuçlar

Bu bölümde önerilen yöntemin performansı 6 adet kıyaslama fonksiyonu kullanılarak test edilmiştir. Ayrıca elde edilen bu değerler, literatürde sıklıkla kullanılan 3 farklı sürü tabanlı optimizasyon yöntemiyle de karşılaştırılmıştır. Test seti olarak kullanılan fonksiyonlar Tablo.1'de verilmiştir [19].

Tablo 1. Kullanılan Test Seti

Fonksiyon	Adı	Test Fonksiyonu	Aralık	Global Optimum
F1	Sphere	$f_1(x) = \sum_{i=1}^n x_i^2$	$[-100, 100]^n$	$[0]^n$
F2	Schwefel's 2.22	$f_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	$[-10, 10]^n$	$[0]^n$
F3	Schwefel's 1.2	$f_3(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	$[-100, 100]^n$	$[0]^n$
F4	Schwefel's 2.21	$f_4(x) = \max(x_i)$	$[-100, 100]^n$	$[0]^n$
F5	Rosenbrock	$f_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-30, 30]^n$	$[0]^n$
F6	Step	$f_6(x) = \sum_{i=1}^n (x_i + 0.5)^2$	$[-100, 100]^n$	$[0.5]^n$

Elde edilmiş olan sonuçların diğer optimizasyon yöntemleriyle karşılaştırmalarının yapılabilmesi için Tablo.1'de verilen veri seti kullanılmıştır. Bu sayede elde edilen sonuçların anlamlandırılabilmesi

de sağlanmıştır. Performans değerlendirilmesi yapabilmek için önerilen kaotik lojistik-gauss tabanlı sürü algoritması yöntemi, Kaotik Bio-Coğrafya Tabanlı Optimizasyon Algoritması (KBTOA), Kaotik Kril Sürüsü Optimizasyonu (KKSOA) ve Kaotik Yerçekimi Arama Optimizasyonu Algoritması (KYAOA)'na ait kaotik algoritmalar ile karşılaştırılmıştır. Tablo.2'de optimizasyon yöntemlerinin kullanmış olduğu parametreler gösterilmiştir.

Tablo 2. Optimizasyon Yöntemlerinde Kullanılan Parametreler

Metot	Popülasyon	Maksimum İterasyon	Boyut	Diğer Veriler
KBTOA	40	500	30	$\mu=0.005, \mu=0.8$
KKSOA	40	500	30	$N^{\max}=0.01, V_f=0.02, D^{\max}=0.005$
KYAOA	40	500	30	$G_0=100, \alpha=20$

Tablo 2'deki gibi parametreler belirlenen kaotik optimizasyon yöntemleri 30 kez çalıştırılarak en iyi değerler kaydedilmiş ve Tablo 3' te elde edilen sonuçlar gösterilmiştir.

Tablo 3. Kaotik Haritalı Optimizasyon Algoritmalarıyla Karşılaştırma Sonuçları

Fonksiyon	KBTOA	KKSOA	KYAOA	KLG-SOA
F1	1.77×10^0	7.89×10^{-1}	1.18×10^{-16}	1.7842e-128
F2	1.20×10^{-1}	2.31×10^2	2.70×10^{-1}	4.5880e-139
F3	7.55×10^3	4.31×10^2	6.55×10^2	1.6587e-279
F4	1.71×10^1	9.71×10^0	4.86×10^0	7.4665e-134
F5	4.18×10^2	1.91×10^2	5.41×10^1	0
F6	1.45×10^0	5.15×10^{-1}	7.77×10^{-17}	0

Tablo.3'teki sonuçlarda görüldüğü üzere, önerilen yöntem literatürde daha önce önerilmiş 3 adet optimizasyon yöntemiyle ve 6 adet kıyaslama fonksiyonu kullanılarak karşılaştırılmıştır. Karşılaştırma sonucu ile önerilen yöntemin tüm fonksiyonlarında en iyi/en iyiye yakın değerler elde ettiği gösterilmiştir.

5. Sonuçlar ve Öneriler

Sürü tabanlı optimizasyon yöntemi oluştururken dikkat edilmesi gereken en önemli nokta parçacıkların nasıl güncelleneceğine karar vermektir. Bu çalışmada lojistik-gauss haritası tabanlı yeni bir kaotik sürü optimizasyonu yöntemi önerilmiştir. Lojistik-gauss haritasının bu çalışmadaki en önemli katkısı rastgele sayı üretiminde başarılı sonuçlar elde etmiş olmasıdır. Önerilen yöntemi test etmek için literatürde yaygın olarak kullanılan amaç fonksiyonları ve kaotik optimizasyon yöntemleri kullanılmıştır. Deneysel sonuçlar, önerilen yöntemin test setinde başarılı sonuçlar ürettiğini açık bir şekilde ifade etmektedir. Bu başarının temel sebebi, kaotik haritaların sadece rastgele sayı üretici olarak kullanılmaması aynı zamanda parçacık güncelleme işleminde de kullanılmasıdır.

Bu çalışma, gelecekteki çalışmalarda gerçek dünya problemlerinin de bu yöntemle çözülebileceğini ve tüm kaotik haritaların sunulan yapı içerisinde kullanılabilceğini göstermektedir.

References:

1. Canayaz M., Karcı, A. 2015. Investigation of cricket behaviours as evolutionary computation for system design optimization problems. *Measurement*, 68, 225-235.
2. Akyol S., Alataş, B. 2012. Güncel sürü zekâsı optimizasyon algoritmaları. *Nevşehir Üni. Fen Bil. Ens. Der.* 1, 36-50.
3. Mirjalili S., Lewis A. 2016, The Whale Optimization Algorithm, *Advances in Engineering Software*, 95:51,67.
4. Kennedy J. And Eberhart R. C. 1995. Particle Swarm Optimization. *Proc. IEEE Conf. Neural Networks*, Perth, Australia, 1942-198,1995.
5. Prayogo D., Cheng M.Y., Wu Y.W., Herdany A.A., Prayogo H. 2018. Differential Big Bang - Big Crunch algorithm for construction-engineering design optimization, *Automation in Construction* 85 (2018) 290–304.
6. Mirjalili S. 2016. SCA: A Sine Cosine Algorithm for solving optimization problems, *Knowledge-Based Systems* 96 (2016) 120–133.
7. Demir G., Tanyıldızı, E., The Use of Sine Cosine Algorithm (SCA) in Solution of Optimization Problems, *Science and Eng. J of Fırat Univ.* 29(1), 227-238.
8. Atashpaz-Gargari, E. ve Lucas C. 2007. “Imperialist Competitive Algorithm: An Algorithm for Optimization Inspired by Imperialistic Competitions”, *IEEE Congress on Evolutionary Computation*, pp. 4661-4667.
9. Geem, Z.W. ve Kim, J-H, Loganathan, G.V. 2001. “A New Heuristic Optimization Algorithm: Harmony Search”, *Simulation*, Vol. 76, No. 2, pp. 60–68, 2001.
10. Tanyıldızı, E. Ve Cıgal, T. 2017. “Kaotik Haritalı Balina Optimizasyon Algoritmaları”, *Fırat Üniv. Müh. Bil. Dergisi*, 29(1), 309-319,2017.
11. Nozawa, H. 1992. A neural network model as globally coupled map and application based on chaos, *Chaos*, 377–386.
12. Yang, Z., Shi Y. 2015. Brain storm optimization with chaotic operation, *Advanced Computational Intelligence (ICACI), 2015 Seventh International Conference*, 978-1-4799- 7259-3.
13. Coelho L.D., Mariani V.C, Guerra F.A., Luz M.V.F, Leite J.V., (2014), Multiobjective Optimization of Transformer Design Using a Chaotic Evolutionary Approach, *IEEE Transactions on Magnetics*, Volume 50, Issue 2, 669-672.
14. Özkaynak, F., Özer, A. B., Sırma, Y., 2012, *Signal Processing and Communications Applications Conference (SIU)*, 10.1109/SIU.2012.6204450.
15. Alataş B., Akın E., Özer B. 2007. Kaotik Haritalı Parçacık Sürü Optimizasyon Algoritmaları, *Eskişehir Osmangazi Üniversitesi Elektrik Elektronik Bilgisayar Biyomedikal Mühendisliği Ulusal Kongresi*.
16. Özkaynak F. 2015. A novel method to improve the performance of chaos based evolutionary algorithms. *Optik-International Journal for Light and Electron Optics* 126 (24), 5434-5438.
17. Zhenyu G., Bo C., Min Y., Binggang C. 2006. Self-adaptive chaos differential evolution. In: Jiao L, Wang L, Gao X, Liu J, Wu F (eds) *Advances in natural computation. Lecture notes in computer science*, vol 4221. Springer, Berlin, Heidelberg, pp 972–975.

18. Peitgen H., Jurgens H. and Saupe D., 1992. Chaos and Fractals. Berlin, Germany: Springer-Verlag, 1992.
19. Saremi S., Mirjalili S., Lewis A., 2014. Biogeography-based optimization with chaos, Neural Computing & Applications (2014) 25:1077–1097.
20. Mirjalili S., Gandomi A.H., (2017) Chaotic gravitational constants for the gravitational search algorithm, Applied Soft Computing, vol 53. Pages 407-419.