

# Rössler Tabanlı Kaotik Farksal Gelişim Algoritması

## Rössler Based Chaotic Differential Evolution Algorithm

Uğur YÜZGEÇ<sup>1\*</sup>, Mehmet ESER<sup>2</sup>

**Özet-** Bu çalışmada optimizasyon problemlerinin çözümünde sıklıkla kullanılan evrimsel algoritmalarından farksal gelişim algoritmasının (FGA) temelini oluşturan rastgele sayı üretim süreci yerine, Rössler tabanlı kaotik sayı üretici önerilmiştir. Rössler sistemi Runge-Kutta yöntemi ile çözülerek elde edilen çıktılar Kaotik FGA (KFGA) yapısında kullanılmıştır. Önerilen kaotik tabanlı FGA yapısının performansını değerlendirmek için literatürden farklı özelliklere sahip on tane optimizasyon test problemi seçilmiştir. Önerilen yapının optimizasyon problemlerinin çözümünde çeşitlilik sağladığı ve test problemlerinin global minimum noktalarını başarı ile bulduğu tespit edilmiştir.

**Anahtar Kelimeler-** Kaotik, FGA, Rössler, Optimizasyon

**Abstract-** In this study, Rössler based the chaotic number generator was proposed instead of the random number generators which are the basis of Differential Evolution Algorithm (DE) that is the most used evolutionary algorithms in solving optimization problems. The outputs obtained by solving Rössler chaotic system with Runge-Kutta method were used in the structure Chaotic DE (CDE). For evaluating the performance of the proposed chaotic based DE structure, ten optimization test problem that have different characteristics were selected from literature. It was observed that the proposed structure provides diversity in the solution of the optimization problems and finds effectively the global minimum points of test problems.

**Keywords-** Chaotic, DE, Rössler, Optimization

### I. GİRİŞ

Optimizasyon, bir gerçel fonksiyonu en küçükleme ya da en büyükleme amacı ile gerçel ya da tamsayı değerlerini, tanımlı bir aralıkta seçerek, bir fonksiyona yerleştirip, sistematik olarak bir problemi incelemek ya da çözmek işlemlerini ifade etmektedir. Optimizasyon problemlerini çözmek için günümüzde, matematiksel ve sezgisel teknikleri kullanan, birçok optimizasyon algoritması bulunmaktadır. Matematiksel (klasik) algoritmalar genellikle probleme özgü tasarlanan ya da problemin tüm çözüm uzayını tarayarak problemi çözmeye çalışan algoritmalarlardır. Büyük boyutlu doğrusal olmayan problemlerde, çözümü bulmaları çok maliyetli ve uzun süreler almaktadır. Sezgisel algoritmalar, çözüm uzayı geniş olan problemlerde, tüm çözüm uzayını taramadan, sezgisel olarak çok kısa sürelerde çözümü ya da çözüme en yakın sonucu elde edebilen algoritmalarlardır. Bunların başlıcaları Genetik Algoritma, Benzetilmiş Tavlama, Farksal Gelişim Algoritması, Yapay Arı Kolonisi Algoritması, Parçacık Sürü Optimizasyonu olarak sayılabilir [1-4].

Bu çalışmada Farksal Gelişim Algoritması, başlangıç popülasyonunun oluşturulması, mutasyon ve çaprazlama işlemlerinde bireylerin seçilmesi işlemlerinde kullanılan rastgele sayı üreticileri yerine kaotik sistemlerden Rössler çekicisinin Runge-Kutta yöntemi ile çözülmesiyle elde edilen kaotik sayı üretici kullanılmasıyla geliştirilmiştir.

### II. RÖSSLER KAOTİK SİSTEMİ

Rössler sistemi üç doğrusal olmayan fark denklemden oluşmaktadır. Bu fark denklemler, Rössler çekicisinin fraktal özellikleri ile ilişkilendirilmiş kaotik dinamikleri sergileyen sürekli zaman dinamik sistemleri tanımlar. Rössler çekicisi, Lorenz çekicisine benzer şekilde davranır, ancak niteliksel olarak analiz etmek daha kolaydır. 1976 yılında, Rössler'in önerdiği [5,6] kaotik denklem sistemi aşağıda verilmiştir.

$$\begin{aligned}\dot{x} &= -y - z \\ \dot{y} &= x + a.y \\ \dot{z} &= b + z.(x - c)\end{aligned}\tag{1}$$

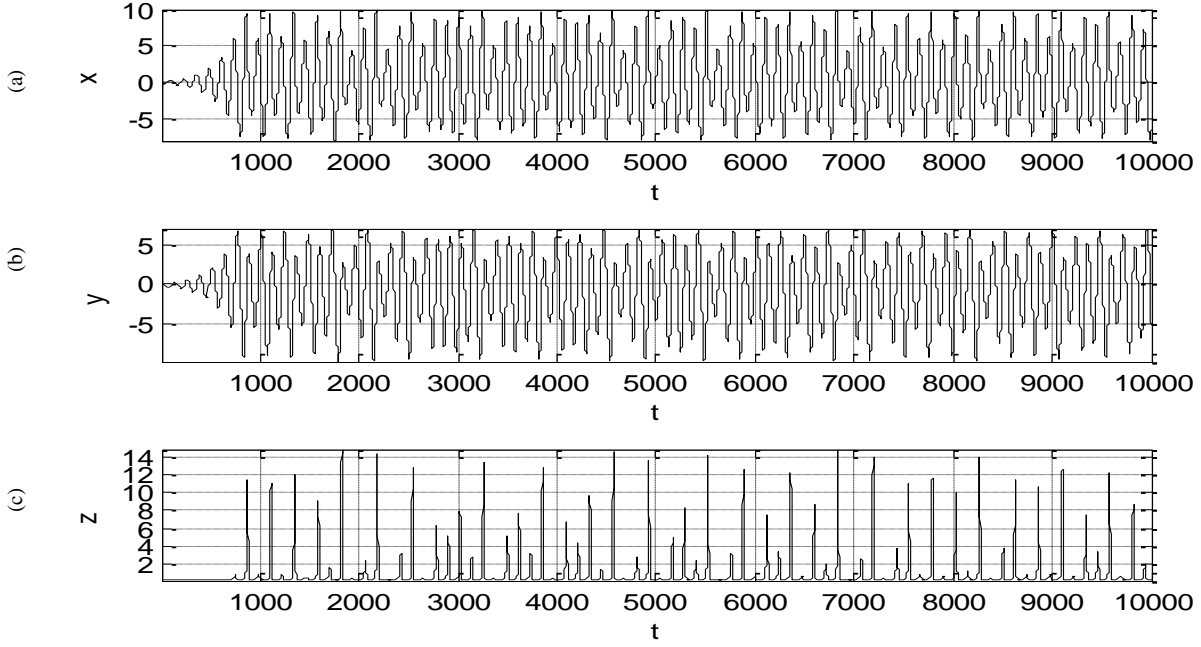
<sup>1\*</sup> Sorumlu yazar iletişim: [ugur.yuzgec@bilecik.edu.tr](mailto:ugur.yuzgec@bilecik.edu.tr)

Bilgisayar Mühendisliği, Bilecik Şeyh Edebali Üniversitesi, Mühendislik Fakültesi, Gülümbe, Bilecik

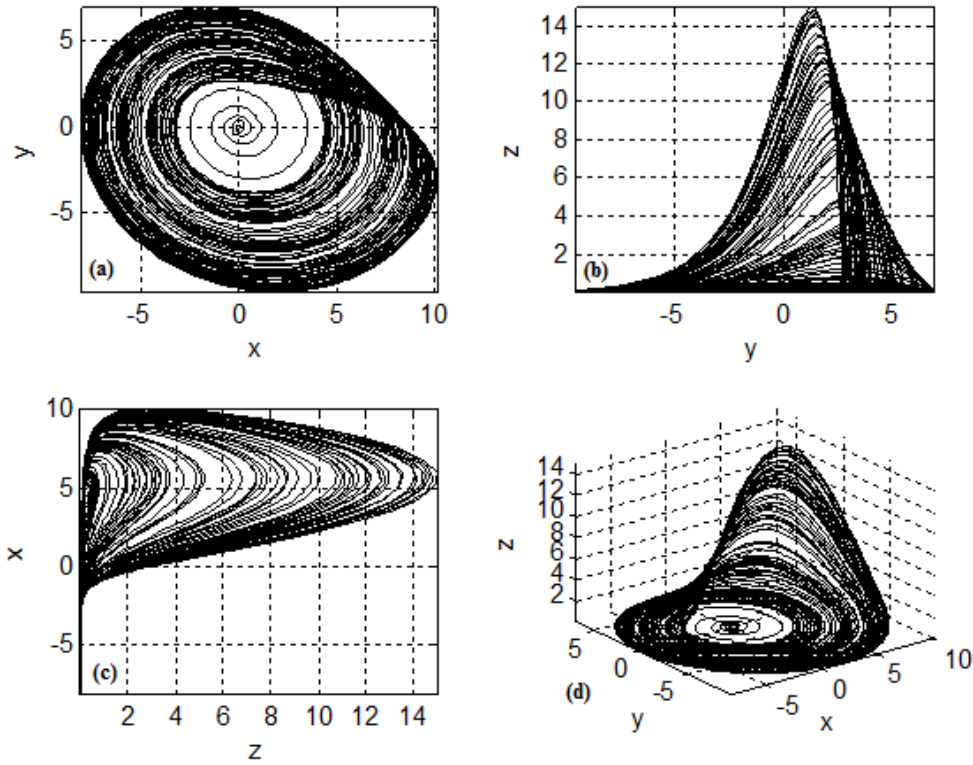
<sup>2</sup> İletişim: [mehmet.eser@bilecik.edu.tr](mailto:mehmet.eser@bilecik.edu.tr)

Bilgisayar Mühendisliği, Bilecik Şeyh Edebali Üniversitesi, Mühendislik Fakültesi, Gülümbe, Bilecik

Sistem, bir adet ikinci dereceden doğrusal olmayan terim ( $xz$ ) olmak üzere toplam yedi terim içermektedir.  $a = 0.2$ ,  $b = 0.2$  ve  $c = 5$  parametreleri ve başlangıç şartları için kaotik çözümler elde edilmiştir. Rössler sisteminin  $x_0 = 0.1$ ,  $y_0 = 0.1$ ,  $z_0 = 0.1$  başlangıç koşulları için Runge-Kutta yöntemi ile bulunan durum değişkenlerinin zamana bağlı değişimleri Şekil 1'de gösterilmiştir. Şekil 2'de ise Rössler sisteminin faz diyagramları ve üç eksen görünümü verilmiştir.



Şekil 1. Runge-Kutta metodu ile çözümlenmiş Rössler sisteminin x,y ve z durum değişkenlerinin zamana göre değişimi ( $x_0 = 0.1$ ,  $y_0 = 0.1$ ,  $z_0 = 0.1$ )



Şekil 2. (a) (b) (c) Runge-Kutta metodu ile çözümlenmiş Rössler Çekicisine ait faz diyagramları ve (d) üç eksen görünümü ( $x_0 = 0.1$ ,  $y_0 = 0.1$ ,  $z_0 = 0.1$ )

### III. RÖSSLER TABANLI KAOTİK FARKSAL GELİŞİM ALGORİTMASI

Farksal Gelişim Algoritması (FGA) Price ve Storn tarafından 1995 yılında geliştirilmiş, temel fikrinde popülasyondaki iki bireyin arasındaki farkın bir üçüncü bireye ilave edilmesine dayanan bir sezgisel optimizasyon tekniğidir [7-9]. FGA yapısında çaprazlama ve mutasyon olmak üzere iki operatör kullanılmaktadır. Bu çalışmada önerilen algorithmada klasik FGA yapısının temelini oluşturan rastgele sayı üretici yerine, kaotik Rössler denklemlerinin çözümlerinin ortalaması alınarak birbirinden farklı bireylerin seçimi için sayılar üretilmektedir.

#### A. Mutasyon Operatörü

FGA yapısı, bir bireye mutasyon işlemini, rastgele seçilmiş birey çiftinin ağırlıklaştırılmış farkının bu bireye ilave edilmesiyle gerçekleştirir [10]. Literatürde en sık kullanılan beş FGA stratejisi için mutasyona uğramış bireyler ( $v_{i,g}$ ) aşağıdaki şekilde bulunmaktadır:

$$v_{i,g+1} = x_{i,g} + F(x_{b,g} - x_{i,g}) + F(x_{r1,g} - x_{r2,g}) \quad (2)$$

$$v_{i,g+1} = x_{r3,g} + F(x_{r1,g} - x_{r2,g}) \quad (3)$$

$$v_{i,g+1} = x_{b,g} + F(x_{r1,g} - x_{r2,g}) \quad (4)$$

$$v_{i,g+1} = x_{b,g} + F(x_{r1,g} - x_{r2,g}) + F(x_{r3,g} - x_{r4,g}) \quad (5)$$

$$v_{i,g+1} = x_{r1,g} + F(x_{r2,g} - x_{r3,g}) + F(x_{r4,g} - x_{r5,g}) \quad (6)$$

Bu denklemlerde  $r_1, r_2, r_3, r_4, r_5 \in [1, NP]$  rastgele birey indisleridir.  $x_{b,g}$  popülasyon içerisindeki en iyi maliyet fonksiyonuna sahip bireyi göstermektedir. Rastgele birey indisleri birbirinden farklı olmak zorundadır. Rastgele seçilen bireylerin fark vektörlerinin yükseltilmesinde kullanılan ölçekleme faktörü ( $F$ ) sabit bir değerdir ve  $0 \leq F \leq 2$  aralığında değişmektedir [11].

#### B. Çaprazlama Operatörü

FGA yapısı içerisinde yeni bireyler çaprazlama operatörü ile aşağıdaki şekilde bulunabilir:

$$u_{i,g+1} = \begin{cases} v_{i,g+1}, & \text{if } r \leq CR \\ x_{i,g}, & \text{if } r > CR \end{cases} \quad (7)$$

Bu denklemde  $r \in [0,1]$  rastgele sayı üreticini ve  $u_{i,g}$  deneme bireylerini göstermektedir. Çaprazlama olasılık sabiti ( $CR$ )  $\in [0,1]$  genellikle kullanıcı tarafından belirlenir. Eğer  $CR = 1$  ise, oluşacak her yeni birey mutant olarak belirlenir, yani rastgele seçilen fark vektörüne göre seçilir. Diğer taraftan  $CR = 0$  durumunda, tüm bireyler önceki jenerasyondan gelir [11,12].

#### C. Seçim İşlemi

Bir sonraki iterasyondaki bireylerin  $x_{i,g}$  tespitinde seçim işleminin işletilmesi sırasında, deneme bireyinin ( $u_{i,g+1}$ ) maliyet fonksiyonu değeri hedef bireyin ( $x_{i,g}$ ) maliyet fonksiyonu ile karşılaştırılır ve sonraki iterasyona, optimizasyon problemine göre yeni birey seçilir. Aşağıdaki eşitlikte maksimum problemi için seçim operatörünün denklemi verilmiştir:

$$x_{i,g+1} = \begin{cases} u_{i,g+1}, & \text{if } f(u_{i,g+1}) > f(x_{i,g}) \\ x_{i,g}, & \text{yoksa} \end{cases} \quad (8)$$

Burada  $f$  maliyet fonksiyonunu göstermektedir. Eğer deneme bireyi ( $u_{i,g+1}$ ) hedef bireyden ( $x_{i,g}$ ) daha iyi bir maliyet fonksiyonuna sahipse, bir sonraki iterasyonda kullanılacak birey ( $x_{i,g+1}$ ) deneme bireyi ( $u_{i,g+1}$ ) olarak seçilir. Aksi durumda popülasyondaki hedef bireyde değişiklik olmaz [11].

### IV. RÖSSLER TABANLI FGA'NIN PERFORMANSI

Bu çalışmada Rössler tabanlı FGA yapısının performansının test edilebilmesi için literatürden alınan matematiksel ifadeleri Tablo 1'de verilen optimizasyon test fonksiyonları kullanılmıştır. Geliştirilen Rössler tabanlı FGA yapısı, 10 farklı optimizasyon test fonksiyonu kullanılarak, Intel(R) Core(TM)2 Duo , P7350@ 2.00 GHz CPU, 4 GB RAM donanımlı, 64 bit işletim sistemine sahip bir bilgisayarda oluşturulmuştur. Yapılan optimizasyon uygulamalarında algoritma parametreleri, NP (Popülasyon değeri) için 20, maksimum iterasyon sayısı için 200, F (ölçekleme faktörü) için 1.2 ve CR (çaprazlama sabiti) için 0.7 olarak alınmıştır. Literatürden

alınmış olan on adet optimizasyon test fonksiyonu için DE/rand/1/bin FGA stratejisi (bkz. Eşitlik 3) kullanılarak, Rössler tabanlı KFGA yapısı ile elde edilen sonuçlar Tablo 2’de sunulmuştur. Test fonksiyonlarında problem boyutu  $d=2$  olarak alınmıştır.

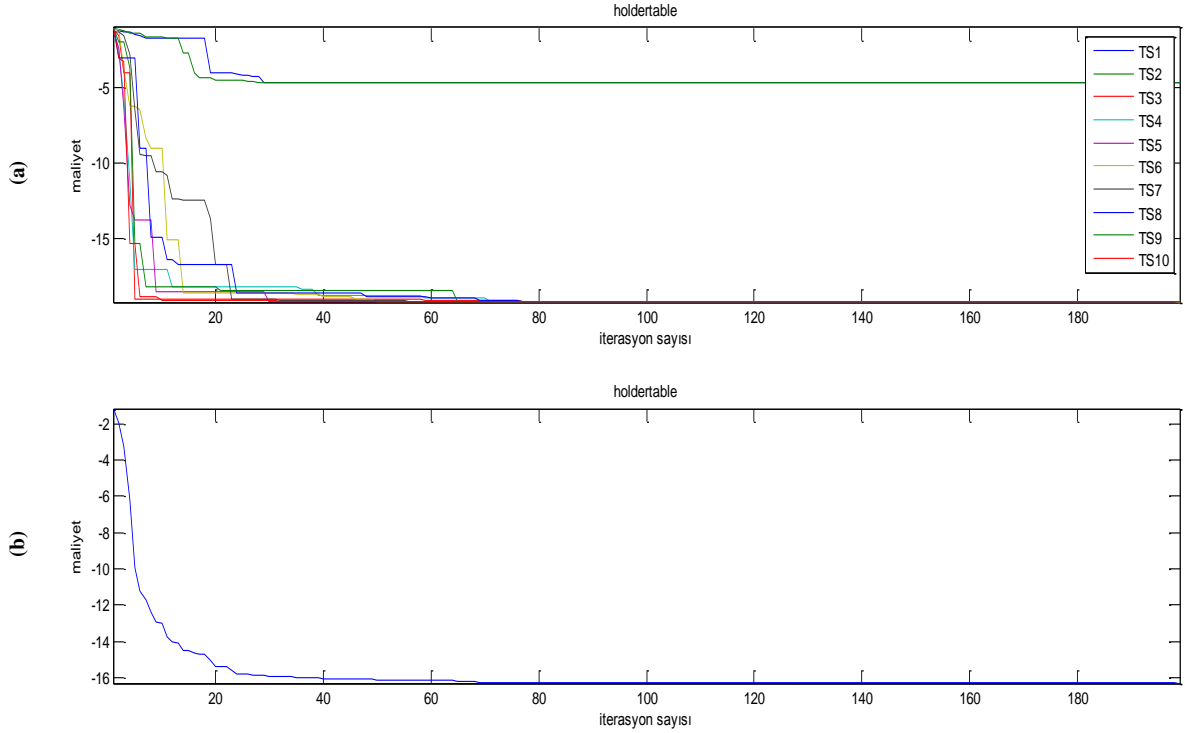
**Tablo 1.** Kullanılan optimizasyon test fonksiyonları

İsim	Problem
Ackley ( $f1$ )	$f(x) = -a \exp\left(-b \sqrt{\frac{1}{d} \sum_{i=1}^d x_i^2}\right) - \exp\left(\frac{1}{d} \sum_{i=1}^d \cos(cx_i)\right) + a + \exp(1)$
Holdertable ( $f2$ )	$f(x) = -\left \sin(x_1)\cos(x_2)\exp\left(\left 1 - \frac{\sqrt{x_1^2 + x_2^2}}{\pi}\right \right)\right $
Rastrigin ( $f3$ )	$f(x) = 10d + \sum_{i=1}^d [x_i^2 - 10\cos(2\pi x_i)]$
Rosenbrock ( $f4$ )	$f(x) = \sum_{i=1}^{d-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$
Guinta ( $f5$ )	$f(x) = 0.6 + \sum_{i=1}^n \left[\sin^2\left(1 - \frac{16}{15}x_i\right) - \frac{1}{50}\sin\left(4 - \frac{64}{15}x_i\right) - \sin\left(1 - \frac{16}{15}x_i\right)\right]$
Penholder ( $f6$ )	$f(x) = -\exp\left[\exp\left(\left -\frac{\sqrt{x_1^2 + x_2^2}}{\pi} + 1\right \right)\cos(x_1)\cos(x_2)\right]^{-1}$
Himmelblau ( $f7$ )	$f(x) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2$
Schweffel ( $f8$ )	$f(x) = 418.9829d - \sum_{i=1}^d x_i \sin\sqrt{ x_i }$
Six-hump Camel ( $f9$ )	$f(x) = \left(4 - 2.1x_1^2 + \frac{x_1^4}{3}\right)x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2$
Testtubeholder ( $f10$ )	$f(x) = -4 \left \exp\left(\left \cos\left(\frac{1}{200}x_1^2 + \frac{1}{200}x_2^2\right)\right \right)\sin(x_1)\cos(x_2)\right $

**Tablo 2.** Rössler Tabanlı KFGA Optimizasyon Sonuçları (50 defa koşma, NP=20, F=1.2 ve CR=0.7)

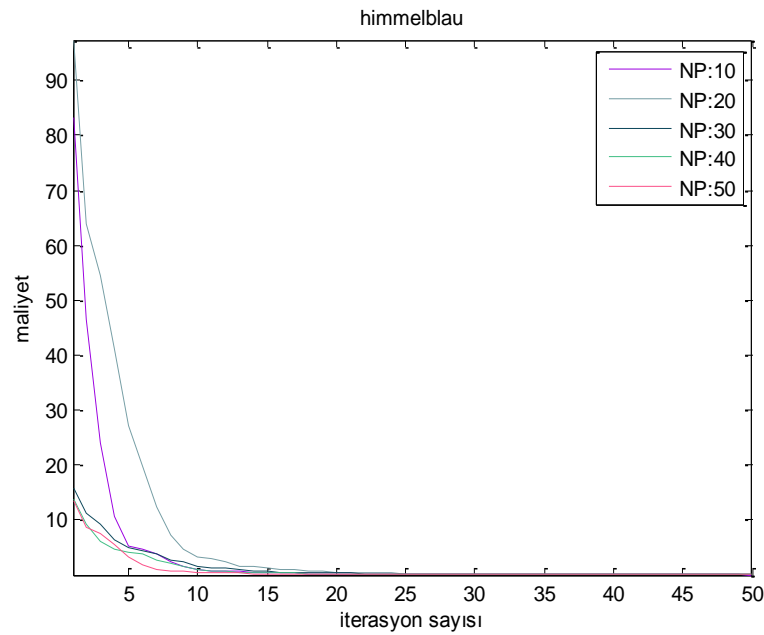
FN	Ortalama Sonuç	Ortalama Çözüm(x1;x2)	Ortalama İterasyon	Ortalama Süre (sn)	Global Sonuç	MSE
f1	0,000001	-0,000000;-0,000000	127,36	0,114062	0	1E-12
f2	-16,4118	-0,912011;-0,849705	199	0,165994	-19,20850257	7,821596
f3	0,318387	-0,159192;-0,159189	89,1	0,071072	0	0,10137
f4	0	1,000025;1,000043	67,6	0,055979	0	0
f5	0,06447	0,467320;0,467320	199	0,1732	0,060247218	1,78E-05
f6	-0,84998	0,435369;-1,107981	199	0,165885	-0,96353	0,012895
f7	0	1,543623;0,606838	94,38	0,072846	0	0
f8	-691,097	275,507328;275,507328	199	0,175157	-837,9658	21570,44
f9	-1,03163	-0,014375;0,114025	199	0,159007	-1,031628	0
f10	-10,8679	-0,376771;-0,313977	199	0,162556	-10,8723	1,9E-05

Tablo 2'deki MSE değerlerine bakıldığında f2, f3 ve f8 test fonksiyonlarının dışında, diğer test fonksiyonları ile yapılan koşturuların başarımının yüksek olduğunu görülmektedir. En yüksek başarımlar f4, f7 ve f9 test fonksiyonu için yapılan koşturular sonunda sağlanmıştır. f1, f5 ve f10 test fonksiyonlarıyla yapılan koşturularda da global sonuca yakın değerler elde edilmiştir. Şekil 3’de Holdertable (f2) optimizasyon test problemi için Rössler tabanlı KFGA’nın 10 tekrar sonucu elde edilen maliyetleri ve ortalama maliyet değişim grafiği gösterilmiştir.



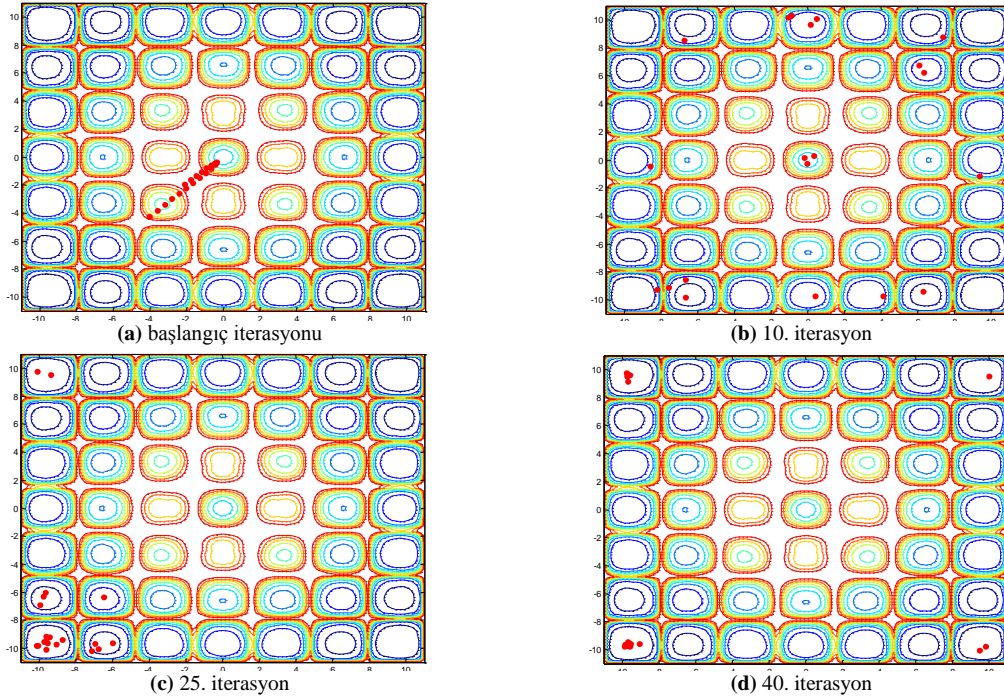
Şekil 3. Holdertable optimizasyon test problemi için Rössler tabanlı KFGA'nın 10 tekrar sonucu (TS1-TS10) maliyet değişimleri (a) ve ortalama maliyet değişimi (b)

Holdertable (f2) test fonksiyonu için iki tekrarda Rössler tabanlı KFGA yapısının lokal minimuma takıldığı, diğer sekiz tekrar için başarılı performans gösterdiği görülmektedir. Ortalama maliyet grafiği incelendiğinde, önerilen Rössler tabanlı KFGA yapısının global minimum noktasına başarı ile gittiği görülmektedir. Şekil 4'de Himmelblau (f7) optimizasyon test problemi için Rössler tabanlı KFGA'nın farklı popülasyon değerleri için maliyet değişim grafikleri gösterilmiştir. Şekilde popülasyon büyüklüğünün 10 ve 20 olduğu değerlerde başarımının diğer değerlere göre az olduğu görülmektedir. Diğer popülasyon büyüklüklerinde 20. iterasyondan sonra başarımın arttığı ve yaklaşık olarak aynı olduğu anlaşılmaktadır.



Şekil 4. Himmelblau optimizasyon test problemi için Rössler tabanlı KFGA'nın farklı popülasyon değerleri için maliyet değişimleri (F=1.2, CR=0.7, tekrar sayısı=20)

Optimizasyon problemlerinin çözümünde, olurlu çözüm bölgesinde, adayların birbirine çok yakın olmaması çeşitliliği (*diversity*) gösteren önemli bir unsurdur. Çeşitlilik her iterasyon sonrası üretilen yeni aday çözümler yardımı ile lokal optimum çözümlerden kurtulmayı sağlamaktadır. Dolayısıyla çözümün belirli noktalarda takılmasını engeller. Çeşitlilik, optimizasyonun başarımını gösteren faktörlerden birisidir. Şekil 5'de Penholder (f6) optimizasyon test problemi için Rössler tabanlı KFGA'nın çalıştırılması sırasında başlangıç iterasyonunda, 10, 25 ve 40. iterasyonlarda popülasyon içerisindeki aday çözümlerin dağılımları görülmektedir. Başlangıç iterasyonları ve 10. iterasyonlarda adayların dağılımlarının çeşitli noktalarda yer aldığı ve bir nokta üzerinde toplanmadığı görülmektedir. 25. iterasyonda aday çözümlerin global çözüm noktalarına doğru hareket ettiği; ama bunun yanı sıra çeşitliliğin de devam ettiği görülmektedir. 40. iterasyon sonunda ise adayların belirli noktalarda kümelenmesine rağmen kısmen çeşitliliğin sağlandığı görülmektedir.



Şekil 5. Penholder optimizasyon test problemi için Rössler tabanlı KFGA'nın farklı iterasyonlardaki popülasyonlarının değişimi (NP=20, F=1.2, CR=0.7, itermax=200)

## V. SONUÇLAR

Yapılan bu çalışmada optimizasyon problemlerinin çözümünde en çok kullanılan evrimsel algoritmalar olan FGA'nın temelini oluşturan rastgele sayı üretim süreci yerine, kaotik Rössler sistemine dayanan sayı üretici geliştirilmiştir. Önerilen Rössler tabanlı KFGA'nın performansı literatürden alınan on optimizasyon test problemi için koşutularak benzetim çalışmaları yapılmıştır. Önerilen yapının optimizasyon problemlerinin çözümünde başarımı ve çeşitliliği sağladığı görülmüştür.

## KAYNAKLAR

- [1] Goldberg D. E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, New York, 1989.
- [2] Michalewicz Z., *Genetic algorithms + Data structures = Evolution Programs*, AI Series, Springer-Verlag, New York, 1994.
- [3] Holland J. H., *Adaptation in Natural and Artificial Systems*, Cambridge, MA:MIT Press. Second edition, 1992.
- [4] Pham D.T. and Karaboga D., *Intelligent Optimization Techniques*, Springer Verlag, London, 2000.
- [5] Rossler O.E., "An equation for continuous chaos", *Phys. Lett. A*, 57, pp:397-398, 1976.
- [6] Rossler O.E., "An equation for hyperchaos", *Phys. Lett. A*, 71, pp:155-157, 1979.
- [7] Price K., & Storn R., "Differential evolution: A simple evolution strategy for fast optimization", *Dr. Dobb's J. Software Tools*, 22 (4), pp:18-24, 1997.

- [8] Storn R., and Price K., "Differential evolution-A simple and efficient heuristic for global optimization over continuous spaces", *J. Global Optimization*, 11, pp:341-359,1997.
- [9] Price K., *An Introduction to Differential Evolution*, D. Corne, M. Dorigo, and F. Glover, Eds. London, U.K.: McGraw-Hill, pp. 79–108, 1999.
- [10] Eser M. Yüzgeç U., "Kaotik Tabanlı Diferansiyel (Farksal) Gelişim Algoritması", *2nd International Symposium on Innovative Technologies in Engineering and Science, ISITES 2014*, s.201-210, Karabük, 18-20 Haziran 2014.
- [11] Yüzgeç U., *Kendinden uyarlanabilir karışıklık tabanlı farksal gelişim algoritması*, Bilecik Üniversitesi, BAP Sonuç Raporu, Bilecik, 2011.
- [12] Yüzgeç U., "Performance comparison of differential evolution techniques on optimization of feeding profile for an industrial scale fed-batch baker's yeast fermentation process", *ISA Transactions*, 49(1), pp: 167-176, January 2010.

