

Nümerik Optimizasyon için Kaotik Altın Sinüs Algoritması

Erkan TANYILDIZI^{1*}, Gökhan DEMİR²

¹ Fırat Üniversitesi Teknoloji Fakültesi Yazılım Mühendisliği Bölümü, Elazığ

² Fırat Üniversitesi Teknoloji Fakültesi Yazılım Mühendisliği Bölümü, Elazığ

*¹ etanyildizi@firat.edu.tr, ² gokhan.demir.dg@gmail.com

(Geliş/Received: 06/07/2018;

Kabul/Accepted: 28/12/2018)

Özet: Rastgele sayı dizileri, sezgisel optimizasyon algoritmalarının stokastik yapılarından dolayı sıkça kullanılmaktadır. Sezgisel optimizasyon algoritmaları global optimuma yakınsamalarında rastgele üretilen sayı dizilerinin dağılımları oldukça etkili olmaktadır. Rastgele üretilen sayıların belli bir alanda toplanması veya aynı değerler üretilmesi ile algoritmanın lokal optimumlara takılma riski artabilmektedir. Bu nedenle üretilen sayıların aynı olmaması, yayılmış spektruma sahip olması sezgisel algoritmaların performansını etkileyen durumlardır. Kaotik haritalar ayrık zamanlı sistemlerdir ve kaotik haritalarla üretilen sayılar geniş bir spektruma sahip olup periyodik değildir. Bu nedenle, sezgisel algoritmalarda kaotik harita kullanarak optimum noktalardan kaçmak ya da yerel optimuma takılma riskini azaltmak mümkün olabilmektedir. Bu çalışmada kaotik haritaların yeni optimizasyon yöntemlerinden olan Altın Sinüs Algoritması üzerindeki etkisi incelenmiştir. **Anahtar kelimeler:** Kaotik haritalar, kaos, Altın Sinüs Algoritması, sezgisel optimizasyon.

Chaotic Gold Sine Algorithm for Numerical Optimization

Abstract: Random number sequences are frequently used for stochastic structures of metaheuristic optimization algorithms. The distributions of randomly generated sequences of numbers are quite effective in the convergence of heuristic optimization algorithms to the global optimum. By gathering randomly generated numbers in a certain area or by generating the same values, the risk of the algorithm to be localized may be increased. Therefore, the numbers produced are not the same, the spread spectrum has the effect of the performance of heuristic algorithms. Chaotic maps are discrete-time systems, and the numbers produced by chaotic maps have a broad spectrum and are not periodic. Therefore, it is possible to avoid the optimum points by using the chaotic map in heuristic algorithms or to reduce the risk of local optimization. In this study, the effect of chaotic maps on the Golden Sine Algorithm, which is one of the new optimization methods, has been investigated.

Key words: Chaotic maps, chaos, Golden Sine Algorithm, metaheuristic optimization.

1. Giriş

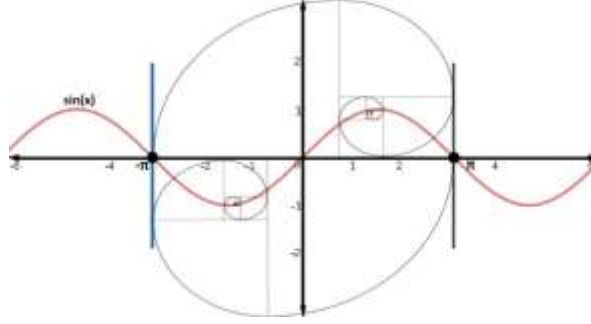
Sezgisel yöntemlerin büyük bir kısmında rastgele sayı dizileri önemli bir yer tutmaktadır. Rastgele üretilen sayıların belli bir alanda toplanması veya aynı değerler üretilmesi ile algoritmanın lokal optimumlara takılma riski artabilmektedir. Bu nedenle üretilen sayıların aynı olmaması, yayılmış spektruma sahip olması sezgisel algoritmaların performansını etkileyen durumlardır.

Kaos teorisi, kaotik dinamik sistemlerin çalışmasını ifade eder. Kaotik sistemler, başlangıç koşullarına duyarlı doğrusal olmayan dinamik sistemlerdir. Başlangıç koşullarındaki küçük değişiklikler, sistemin sonucunda yüksek varyasyonlara neden olur. Kaotik sistemler rastgele davranışlar sergilese bile, bir sistemin kaotik davranışlar sergilemesi için mutlaka rastgele olması gerekmez. Başka bir deyişle, deterministik sistemler de kaotik davranışlar gösterebilir. Sezgisel algoritmalarda kaotik harita kullanarak optimum noktalardan kaçmak ya da yerel optimuma takılma riskini azaltmak mümkün olabilmektedir. Bu nedenle, optimizasyon algoritmalarının performanslarını arttırmak için rastgele sayı dizileri yerine kaotik haritalar yaygın şekilde kullanılmıştır. Bu çalışmalara örnek olarak doğadan esinlenmiş olan Kaotik Parçacık Sürü Optimizasyon Algoritması [1], Kaotik Arı Koloni Algoritması[2], Kaotik Genetik Algoritmalar [3], Kaotik Diferansiyel Gelişim [4], Kaotik Benzetilmiş Tavlama [5] ve Kaotik Ateşböceği Algoritması [6], Kaotik Kril Sürü Algoritması [7], Kaotik Balina Optimizasyon Algoritması [8] ve Kaotik Biyocoğrafya Tabanlı Optimizasyon [9] dur. Bu çalışmaların sonuçları, kaosun metasezgisel algoritmalara başarıyla uygulanabilirliğini teyit etmektedir. Bu çalışmanın amacı literatüre yeni kazandırılmış matematik tabanlı bir metasezgisel algoritma olan Altın Sinüs Algoritması (ASA)[10]'nın kaotik versiyonları oluşturularak kaotik haritaların sezgisel algoritma üzerindeki etkisi incelenmiştir. ASA'nın seçilme sebebi ise kısıtsız kıyaslama fonksiyonlarında global optimumu bulmada diğer optimizasyonlardan daha başarılı sonuçlar vermesi ve daha az başlangıç parametresine sahip olmasıdır.

* Sorumlu yazar: etanyildizi@firat.edu.tr. Yazarların ORCID Numarası: ¹ 0000-0003-2973-9389

2. Altın Sinüs Algoritması

Sezgisel yöntemler deterministik yöntemlerden farklı olarak stokastik tabanlı operatörler kullanırlar. Bu operatörlerin oluşturulmasında farklı kaynaklardan (biyoloji, fizik, müzik, spor, matematik vb.) ilham alınmaktadır. Önerilen matematik tabanlı algoritmanın ilham kaynağı sinüs fonksiyonudur. Sinüs fonksiyonu değerlerini düzenli aralıklarla tekrar eden bir periyodik bir fonksiyondur. Sinüs fonksiyonun tüm değerlerinin birim çemberini taraması optimizasyon problemlerindeki arama uzayının taranmasına benzemektedir. Bu benzerlik ASA'nın geliştirilmesine ilham kaynağı olmuştur. ASA, sinüs fonksiyonun ve altın oran yöntemini birleştirerek aramayı geliştirmeyi amaçlamaktadır (Şekil 1) [10].



Şekil 1. Altın Sinüs Algoritması [10]

ASA'nın en iyi çözümü bulmak için Denklem 1'de verilen yapıyı kullanır.

$$V(i, j) = V(i, j) \cdot |\sin(r_1)| - p \cdot \sin(r_1) \cdot |x_1 \cdot D(j) - x_2 \cdot V(i, j)| \quad (1)$$

Burada $V(i, j)$; i . çözümdeki j . boyutun mevcut değerini göstermektedir. D , belirlenen hedef değerdir. $r_1 = [0, 2\pi]$ aralığında rastgele bir sayıdır. $p = [0, \pi]$ aralığında rastgele bir sayıdır. x_1 ve x_2 ise altın kesit yöntemi ile elde edilen katsayılardır. Sezgisel yöntemlerin temel hedefi arama uzayının en iyi sonuç vereceği düşünülen alanlarını keşfetmek ve bu alanların da mümkün olduğunca tamamının taranmasını sağlamaktır. Arama uzayının geniş olması problemlerin çözümü için büyük bir sorun oluşturmaktadır. Arama uzayının daraltılması işleminin nasıl yapıldığı problemlerin çözümünde sonuçları önemli ölçüde etkilemektedir. ASA bu işlemi en iyi şekilde yapmak için altın kesit yöntemini kullanmaktadır. ASA'nın sözde kodu Şekil 2'de gösterilmektedir.

```

1. Başlangıç popülasyonunu ( $V_{pop}$ ), arama alanı sayısı ( $N$ ) ve altın oran ( $\tau$ : 0,618033)
ve maksimum iterasyon sayısını ( $t_{max}$ ) tanımla
2. Arama ajanlarının uygunluğunu hesapla
3. En iyi arama ajanını bul ve hedef değer olarak ata
4. while ( $t_{max}$ )
5.   for 1 to  $V_{pop}$ 
6.      $r \leftarrow \text{rand}(0,1)$ 
7.      $r_1 \leftarrow 2\pi * r$ 
8.      $p \leftarrow \pi * r$ 
9.     for 1 to  $N$ 
10.       $V(i, j) \leftarrow V(i, j) * |\sin(r_1)| - p * \sin(r_1) * |x_1 * D(j) - x_2 * V(i, j)|$ 
11.    end for
12.  end for
13. En iyi çözümü (arama ajanı) bul ve  $D(j)$ 'ye hedef değer olarak ata
14. if  $V(i, j) < D(j)$ 
15.   then  $b \leftarrow x_2$ ,  $x_2 \leftarrow x_1$ 
16.      $x_1 \leftarrow a * \tau + b * (1 - \tau)$ 
17.   else  $a \leftarrow x_1$ ,  $x_1 \leftarrow x_2$ 
18.      $x_2 \leftarrow a * (1 - \tau) + b * \tau$ 
19.   if  $x_1 == x_2$ 
20.     then  $a \leftarrow \text{rand}(0, \pi)$ ,  $b \leftarrow \text{rand}(0, \pi)$ 
21.        $x_1 \leftarrow a * \tau + b * (1 - \tau)$ 
22.        $x_2 \leftarrow a * (1 - \tau) + b * \tau$ 
23.   end while
24. return en iyi çözüm kümesi ve elde edilen global optimum sonuç

```

Şekil 2. Altın Sinüs Algoritmasının sözde kodu.

3. Kaotik Haritalı ASA Yöntemleri

Literatürde birçok kaotik harita tanımlanmıştır. Bu çalışmada kullanılan kaotik haritalar Tablo 1 de verilmiştir. ASA'da rastgele değerler kullanan kısımlar kaotik harita denklemleri ile değiştirilerek ASA'nın on farklı kaotik versiyonu oluşturulmuştur.

Tablo 1. Kaotik Haritalar

	Kaotik Harita	Foksiyon	
K1	Chebyshev	$x_{i+1} = \cos(\cos^{-1}(x_i))$	(2)
K2	Çember	$x_{i+1} = \text{mod}\left(x_i + b - \left(\frac{a}{2\pi}\right) \sin(2\pi x_i), 1\right), a = 0.5, b = 0.2$	(3)
K3	Gauss/Mouse	$x_{i+1} = \begin{cases} 1 & x_i = 0, \\ \frac{1}{\text{mod}(x_i, 1)} & \text{aksi takdirde} \end{cases}$	(4)
K4	İteratif	$x_{i+1} = \sin\left(\frac{a\pi}{x_i}\right) a = 0.7$	(5)
K5	Lojistik	$x_{i+1} = ax_i(1 - x_i), a = 4$	(6)
K6	Parçalı	$x_{i+1} = \begin{cases} \frac{x_i}{P} & 0 \leq x_i < P \\ \frac{x_i - P}{0.5 - P} & P \leq x_i < 0.5, \\ \frac{1 - P - x_i}{0.5 - P} & 0.5 \leq x_i < 1 - P, 1 - P \leq x_i < 1 \\ \frac{1 - x_i}{P} & \end{cases}$	(7)
K7	Sinüs	$x_{i+1} = \frac{a}{4} \sin(\pi x_i) a = 4,$	(8)
K8	Singer	$x_{i+1} = \mu(7.86x_i - 23.31x_i^2 + 28.75x_i^3 - 13.302875x_i^4), \mu = 2.3$	(9)
K9	Sinüzoidal	$x_{i+1} = ax_i^2 \sin(\pi x_i), a = 2.3$	(10)
K10	Tent	$x_{i+1} = \begin{cases} \frac{x_i}{0.7} x_i < 0.7 \\ \frac{10}{3} (1 - x_i)x_i \geq 0.7 \end{cases}$	(11)

Altın Sinüs Algoritması'nda p parametresi her arama ajanı için $[0, \pi]$ aralığında rastgele olarak belirlenmektedir. Rastgele üretilen sayılar bazen arama uzayının tamamının sömürülmesini engellemektedir. Seydedali Mirjalili ve Amir H. Gandomi [11], kaotik haritaları Denklem 12 ve Denklem 13 ile normalize ederek başarılı sonuçlar elde etmişlerdir. Bu çalışmada normalize edilmiş kaotik haritalara entegre olarak adaptif olarak azalan Denklem 1'de gösterilen r_1 deseni ilk kez kullanılarak arama uzayının mümkün olduğunca tamamının taranması hedeflenmektedir.

$$V(t) = \text{Max} - \frac{t}{T}(\text{Max} - \text{Min}) \quad (11)$$

$$\text{Normalize } (K_i(t); [a, b]'dan [0, V(t)]'ye): K_i^{\text{norm}}(t) = \frac{(K_i(t) - a) \times (V(t) - 0)}{(b - a)} + 0 = \frac{(K_i(t) - a) \times (V(t))}{(b - a)} \quad (12)$$

burada, K : Kaotik harita, i : Kaotik haritanın indeksi, t : Mevcut iterasyon, T : Maksimum iterasyon sayısı, $[\text{Max} - \text{Min}]$: Uyarlanabilir aralığı temsil eder. $[a, b]$: Kaotik haritaların aralığını gösterir. $V(t)$ her iterasyonda azaltılırken her iterasyonda $[a, b]$, $[0, V(t)]$ ile eşleşir. Bu, adaptif normalizasyon sürecini ifade eder. Sonuç olarak p katsayısının değeri Denklem 13 kullanılarak güncelleştirilir.

$$p(t) = K_i^{\text{norm}}(t) + c - t \times ((c)/T) \quad (13)$$

p 'ye kaotik haritalar eklemek, hem adaptif p 'nin hem de kaotik haritaların rastgele davranışına aynı anda katkıda bulunur. Kaotik haritalar, p için farklı değerler sağlamaktadırlar. p 'nin değerini aniden değiştirerek yerel minimumdan kurtulmaya yardımcı olarak, daha iyi yakınsama hızı sağlamaktadır. Uyarlamalı normalleştirme yaklaşımı ise kaos tabanlı ASA algoritmalarının arama evresinden sömürü evresine yavaş yavaş geçmesine yardımcı olur.

4. Deneysel Sonuçlar

Global optimizasyon algoritmalarının performanslarını incelemek için farklı boyutlarda test fonksiyonlarının sayısal örneklerine göre değerlendirmek daha uygundur. Geliştirilen yeni yöntem literatürde yaygın olarak kullanılan beş farklı kıyaslama fonksiyonuna uygulanmıştır (Tablo 2) [12]. F1-F5, kalite testi fonksiyonlarına ait sonuçlar Tablo 3'te gösterilmektedir.

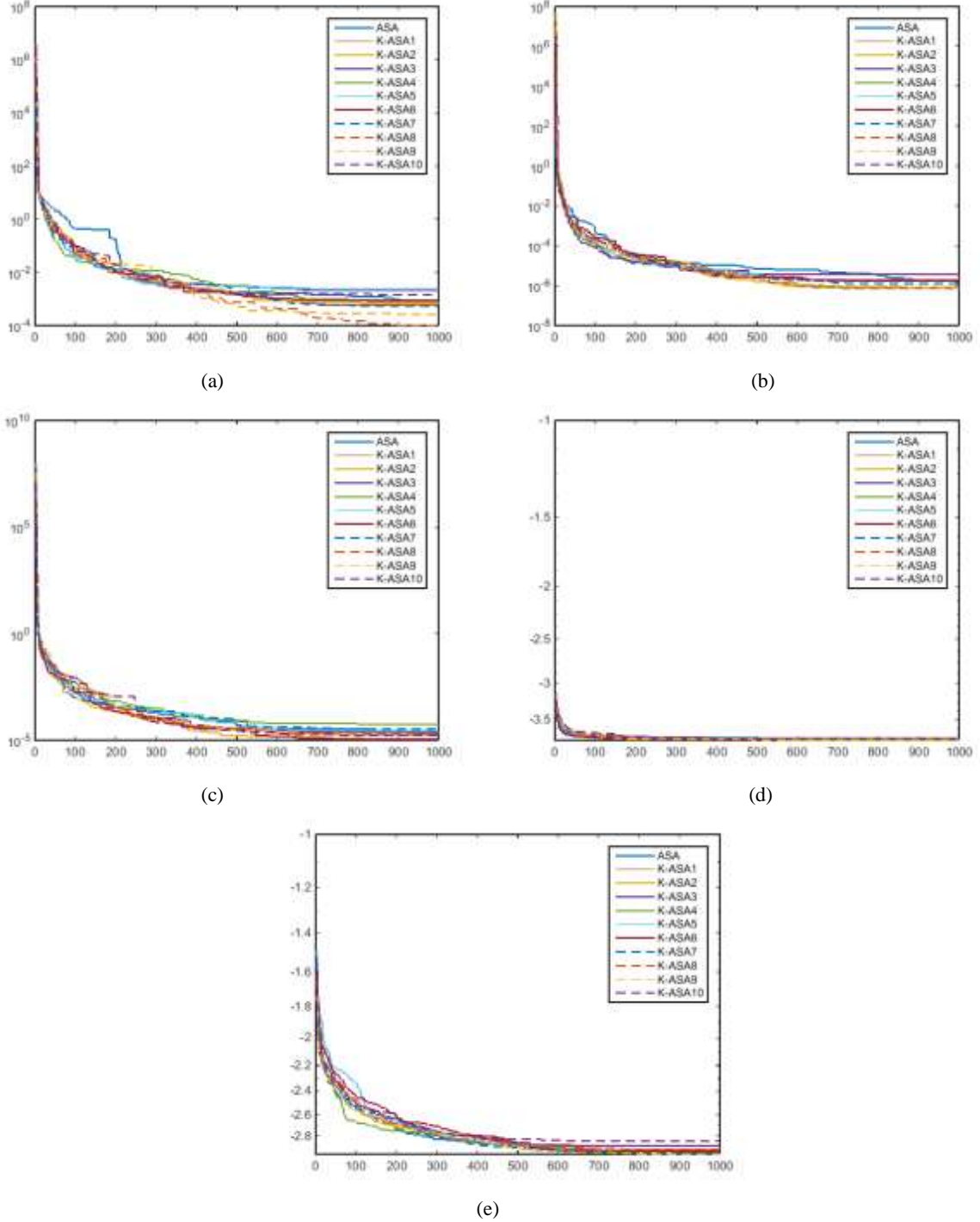
Tablo 2. Kalite test fonksiyonları

	Foksiyon	Boyut	Aralık	Optimum
F1	Rosenbrock	30	[-5, 10]	0
F2	Penalty(01)	30	[-50, 50]	0
F3	Penalty(02)	30	[-50, 50]	0
F4	Hartmann (H3,4)	6	[0, 1]	- 3.86278
F5	Hartmann (H6,4)	6	[0, 1]	- 3.32237

Tablo 3. Kalite testi fonksiyonları sonuçları (*F*:fonksiyon, *İ*: istatistikler)

<i>F</i>	<i>İ</i>	F1	F2	F3	F4	F5
ASA	<i>Ortalama</i>	7.6412e-04	1.5886e-06	1.9802e-05	-3.7841	-2.9592
	<i>Std</i>	0.0011	3.0044e-06	6.0729e-05	0.0865	0.3052
	<i>En iyi</i>	1.1344e-06	1.0691e-08	7.5857e-08	-3.8627	-3.3036
	<i>En kötü</i>	0.0042	1.3794e-05	3.3367e-04	-3.5980	-1.4908
K1-ASA	<i>Ortalama</i>	5.9770e-04	1.8438e-06	1.8587e-05	-3.7973	-2.9281
	<i>Std</i>	0.0017	4.6076e-06	4.9544e-05	0.0678	0.3884
	<i>En iyi</i>	1.3246e-08	4.5515e-11	2.4121e-09	-3.8624	-3.3114
	<i>En kötü</i>	0.0090	2.0640e-05	2.1019e-04	-3.6667	-1.6320
K2-ASA	<i>Ortalama</i>	7.1666e-04	7.3447e-07	1.1229e-05	-3.8036	-2.9376
	<i>Std</i>	0.0025	1.2626e-06	2.5533e-05	0.0766	0.2701
	<i>En iyi</i>	6.9253e-08	1.9028e-09	2.6391e-09	-3.8626	-3.2262
	<i>En kötü</i>	0.0139	4.3577e-06	1.2592e-04	-3.6089	-1.7379
K3-ASA	<i>Ortalama</i>	0.0021	3.5520e-06	2.3832e-05	-3.7973	-2.8948
	<i>Std</i>	0.0074	1.4880e-05	4.5442e-05	0.0745	0.4178
	<i>En iyi</i>	5.6950e-08	3.8111e-09	3.5240e-09	-3.8622	-3.2817
	<i>En kötü</i>	0.0403	8.1773e-05	1.9561e-04	-3.6044	-1.1704
K4-ASA	<i>Ortalama</i>	8.4300e-04	8.0683e-07	5.7996e-05	-3.7849	-2.9479
	<i>Std</i>	0.0028	1.3490e-06	1.7730e-04	0.0818	0.3961
	<i>En iyi</i>	4.6070e-08	1.7830e-09	1.0332e-08	-3.8627	-3.2552
	<i>En kötü</i>	0.0145	5.0969e-06	9.5103e-04	-3.6018	-1.4650
K5-ASA	<i>Ortalama</i>	0.0023	1.6635e-06	2.7116e-05	-3.7810	-2.9439
	<i>Std</i>	0.0072	4.6690e-06	6.1372e-05	0.0761	0.2922
	<i>En iyi</i>	5.8983e-08	4.6439e-10	4.7801e-10	-3.8625	-3.2723
	<i>En kötü</i>	0.0306	2.4306e-05	2.7017e-04	-3.6007	-1.5972
K6-ASA	<i>Ortalama</i>	8.9644e-04	1.7741e-06	1.1126e-05	-3.8007	-2.9447
	<i>Std</i>	0.0014	5.0282e-06	1.8043e-05	0.0714	0.3346
	<i>En iyi</i>	2.9675e-08	2.5383e-10	3.1385e-09	-3.8628	-3.3052
	<i>En kötü</i>	0.0052	2.6333e-05	6.6479e-05	-3.5950	-1.3301
K7-ASA	<i>Ortalama</i>	5.2884e-04	1.2585e-06	3.3608e-05	-3.7955	-2.9658
	<i>Std</i>	0.0010	3.6219e-06	6.8010e-05	0.0649	0.1243
	<i>En iyi</i>	8.4680e-08	4.3978e-10	5.7179e-09	-3.8627	-3.1448
	<i>En kötü</i>	0.0052	1.5491e-05	3.1941e-04	-3.6213	-2.6929
K8-ASA	<i>Ortalama</i>	1.0084e-04	7.2342e-07	2.0493e-05	-3.7863	-2.9263
	<i>Std</i>	1.7375e-04	1.1624e-06	3.6663e-05	0.0795	0.3267
	<i>En iyi</i>	4.3724e-08	4.7235e-10	4.6726e-09	-3.8614	-3.2193
	<i>En kötü</i>	8.3789e-04	3.8443e-06	1.2470e-04	-3.6099	-1.3824
K9-ASA	<i>Ortalama</i>	2.7427e-04	8.4349e-07	1.7158e-05	-3.8101	-2.9757
	<i>Std</i>	9.3238e-04	2.6673e-06	4.4625e-05	0.0575	0.2004
	<i>En iyi</i>	2.8276e-07	4.3278e-10	8.5489e-09	-3.8622	-3.2979
	<i>En kötü</i>	0.0051	1.4665e-05	2.0350e-04	-3.6634	-2.3267
K10-ASA	<i>Ortalama</i>	0.0015	1.7372e-06	1.7069e-05	-3.7870	-2.8510
	<i>Std</i>	0.0042	3.6962e-06	3.7362e-05	0.0823	0.4274
	<i>En iyi</i>	2.2575e-09	3.0208e-09	4.6515e-08	-3.8623	-3.1764
	<i>En kötü</i>	0.0199	1.3951e-05	1.3582e-04	-3.6016	-1.1698

Tablo 3'deki test sonuçlarından F1 fonksiyonunun çözümünde K10-ASA algoritmasının, F2 fonksiyonunun çözümünde K1-ASA algoritmasının, F3 fonksiyonunun çözümünde K5-ASA algoritmasının, F4 ve F5 fonksiyonlarının çözümünde K6-ASA algoritmasının optimum noktaya en yakın sonuçlar verdikleri görülmektedir. Kaotik haritalı ASA algoritmasının uygulandığı F1-F5 fonksiyonlarına ait ortalama yakınsama eğrileri Şekil 3'te verilmiştir.



Şekil 3. Ortalama yakınsama eğrileri; a) F1 fonksiyonu, b) F2 fonksiyonu, c) F3 fonksiyonu, d) F4 fonksiyonu, e) F5 fonksiyonu

Ortalama uygunluk eğrilerine bakıldığında F1 ve F2 fonksiyonu için K8-ASA algoritmasının iyi optimum değere sahip olduğu (Şekil 3a- Şekil 3b), F3 fonksiyonu için K8-ASA algoritmasının iyi optimum değere sahip olduğu (Şekil 3c), F4 ve F5 fonksiyonu için ise K9-ASA algoritmasının iyi optimum değere sahip olduğu (Şekil 3d-Şekil 3e) görülmektedir. Temel algoritma ile kaotik algoritmalar arasındaki farkların istatistiki anlamlılık düzeyini gözlemlenmek amacıyla Wilcoxon işaretli sıralar testi uygulanmıştır. Bu test her bir kaotik versiyon için en iyi algoritma ile temel algoritma çiftleri arasında uygulanmıştır (Tablo 4). Tablo 4'te *p*-değeri anlamlılık düzeyini R^+ kaotik algoritmaların ve R^- ise temel algoritmanın rank toplamlarını göstermektedir.

Tablo 4. Wilcoxon işaretli sıralar testi karşılaştırma sonuçları

F	İ	F5	F12	F13	F19	F20
ASA /K1-ASA	<i>p</i> -değeri	0.1915	0.5999	0.6583	0.5038	0.7343
	R^+	169	207	211	200	216
	R^-	296	258	254	265	249
ASA /K2-ASA	<i>p</i> -değeri	0.0285	0.1109	0.1986	0.2802	0.5999
	R^+	126	155	170	180	258
	R^-	339	310	295	285	207
ASA /K3-ASA	<i>p</i> -değeri	0.4653	0.2989	0.8774	0.5857	0.4779
	R^+	197	182	240	206	267
	R^-	268	283	225	259	198
ASA /K4-ASA	<i>p</i> -değeri	0.0428	0.2289	0.1986	0.8612	0.6583
	R^+	134	174	295	241	211
	R^-	331	291	170	224	254
ASA /K5-ASA	<i>p</i> -değeri	0.6143	0.1650	0.7189	0.7499	0.5577
	R^+	208	165	250	248	261
	R^-	257	300	215	217	204
ASA /K6-ASA	<i>p</i> -değeri	0.9918	0.1064	0.4284	0.6143	0.7655
	R^+	233	154	194	208	247
	R^-	232	311	271	257	218
ASA /K7-ASA	<i>p</i> -değeri	0.5038	0.1306	0.5440	0.9918	0.6143
	R^+	200	159	262	233	257
	R^-	265	306	203	232	208
ASA /K8-ASA	<i>p</i> -değeri	0.0068	0.0428	0.7971	0.9426	0.4165
	R^+	101	134	245	236	272
	R^-	364	331	220	229	193
ASA /K9-ASA	<i>p</i> -değeri	0.0148	0.0196	0.3709	0.0519	0.7971
	R^+	114	119	189	138	220
	R^-	351	346	276	327	245
ASA /K10-ASA	<i>p</i> -değeri	0.3493	0.4284	0.4779	0.7499	0.1306
	R^+	187	194	198	248	306
	R^-	278	271	267	217	159

p-değeri <0.05

Wilcoxon işaretli sıralar testi karşılaştırma sonuçlarına göre Tablo 4 incelendiğinde algoritmalar arasında $p > 0.05$ olduğundan anlamlı bir fark görülmemiştir. Ancak rank değerlerine bakıldığında K1-ASA, ASA'ya 5/0, K2-ASA, ASA'ya 4/1, K3-ASA, ASA'ya 3/2, K4-ASA ASA'ya 3/2, K6-ASA, ASA'ya 3/2, K7-ASA, ASA'ya 3/2, K9-ASA, ASA'ya 5/0, K10-ASA, ASA'ya 3/2 üstünlük sağlamıştır. ASA ise, K5-ASA'e 3/2, K8-ASA'e 3/2 üstünlük sağlamıştır.

5. Sonuçlar

Bu çalışmada ASA'nın performansını iyileştirmek için on kaotik harita kullanılmıştır. Kalite testi fonksiyonları üzerinde testler yapılarak sonuçlar Wilcoxon işaretli sıralar testi ile desteklenmiştir. Sonuçlar, kaotik haritaların ASA'nın hem arama hem de sömürme evrelerini iyileştirme kabiliyetine sahip olduğunu kanıtlamaktadır. Ayrıca sonuçlar, Chebyshev ve Sinüzoidal kaotik haritanın en iyi olduğunu göstermektedir.

Kaotik haritaların ASA'nın, keşif aşamasını güçlendirir ve yerel minimuma sıkışmaktan kurtulmaya yardımcı olur. Önerilen yeni yaklaşımla keşif ve sömürünün adaptif olarak dengelenmesine izin verir. Başka bir

deyişle, K-ASA'nın keşif evresinden sömürü evresine aşamalı olarak geçmesine yardımcı olur. Gelecek çalışmalarda, gerçek dünya mühendislik problemlerinin çözümü için K-ASA algoritmalarının kullanılması amaçlanmaktadır. Ayrıca, diğer kaotik haritaların da ASA'ya uygulanması hedeflenmektedir.

Kaynaklar

- [1] Alatas B, Akin E, Ozer AB. Chaos embedded particle swarm optimization algorithms. *Chaos, Solitons & Fractals* 2009; 40(4):1715-1734.
- [2] Alatas B. Chaotic bee colony algorithms for global numerical optimization. *Expert Syst Appl* 2010; 37(8): 5682-5687.
- [3] Jun-feng Y, Chi M, Xiao-qi P, Zhi-kun H, Jun H. A new optimization approach-chaosgenetic algorithm, *Syst Eng* 2001; 1:1-5.
- [4] Zhenyu G, Bo C, Min Y, Binggang C. Self-adaptive chaos differential evolution. *Adv Nat Comput* 2006; 972-975.
- [5] Mingjun J, Huanwen T. Application of chaos in simulated annealing. *Chaos Solitons Fractals* 2004; 21: 933-941.
- [6] Gandomi AH, Yang XS, Talatahari S, Alavi AH. Firefly algorithm with chaos. *Commun Nonlinear Sci* 2012; 18(1):89-98
- [7] Wang GG, Guo L, Gandomi AH, Hao GS, Wang H. Chaotic krill herd algorithm. *Inform Sciences* 2014; 274:17-34.
- [8] Tanyıldızı E, Cigal T. Kaotik Haritalı Balina Optimizasyon Algoritması. *Fırat Üniversitesi Mühendislik Bilimleri Dergisi* 2017; 29(1):309-319.
- [9] Saremi S, Mirjalili S, Lewis A. Biogeography-based optimisation with chaos. *Neural Comput Appl* 2014; 25:1077-1097.
- [10] Tanyıldızı E, Demir G. Golden Sine Algorithm: A Novel Math-Inspired Algorithm. *Adv Electr Comput En* 2017; 17: 71-78
- [11] Mirjalili S, Gandomi AH. Chaotic gravitational constants for the gravitational search algorithm. *Appl Soft Comput* 2017; 53:407-419.
- [12] Yao X, Liu Y, Lin G. Evolutionary programming made faster. *IEEE T Evolut Comput* 1999; 3(2): 82-102.