



Düzce Üniversitesi Bilim ve Teknoloji Dergisi

Araştırma Makalesi

Test Fonksiyonları için Kaos Tabanlı Yerçekimsel Arama Algoritmaları (CbGSA-X)

 Serdar ÖZYÖN^{a,*},  Celal YAŞAR^a,  Hasan TEMURTAŞ^b

^a Elektrik Elektronik Müh. Bölümü, Mühendislik Fakültesi, Dumlupınar Üniversitesi, Kütahya, TÜRKİYE

^b Bilgisayar Müh. Bölümü, Mühendislik Fakültesi, Dumlupınar Üniversitesi, Kütahya, TÜRKİYE

* Sorumlu yazarın e-posta adresi: serdar.ozyon@dpu.edu.tr

DOI: 10.29130/dubited.598619

ÖZET

Optimizasyon problemlerinin çözümünde kullanılan sezgisel algoritmalar farklı tasarımlarından dolayı, her problem için en iyi sonuca kararlı bir şekilde ulaşamayabilir. Bu nedenle literatürde bu sezgisel algoritmalar bazı geliştirici yapıların eklendiği çalışmalara sıklıkla rastlanmaktadır. Benzer şekilde bu çalışmada sezgisel algoritmalarından biri olan yerçekimsel arama algoritmasının (GSA) performansının geliştirilmesine çalışılmıştır. Çalışmada algoritmanın yakınsama hızının artırılması amaçlanarak GSA'ya bazı kaotik haritalama metodları entegre edilerek, yeni bir algoritma ortaya çıkarılmıştır. Bu yeni algoritmaya Kaos tabanlı yerçekimsel arama algoritması (CbGSA-X) adı verilmiştir. Çalışmada CbGSA-X'deki ilk popülasyondaki ajanlar oluşturulurken ilk ajan arama uzayında rastgele konumlandırılırken, diğer ajanlar ise bu ajana bağlı olarak 5 farklı ($X=1, 2, 3, 4, 5$) kaotik haritalama yöntemi kullanılarak konumlandırılmıştır. Her haritalama metodu için performans değerlendirilmesi yapılabilmesi için literatürde GSA ile çözümü yer alan test fonksiyonları ele alınmış ve çözüm yapılarak sonuçlar değerlendirilmiştir.

Anahtar Kelimeler: Optimizasyon, Sezgisel algoritmalar, Yerçekimsel arama algoritması, Kaotik haritalar, Test fonksiyonları.

Chaos Based Gravitational Search Algorithms (CbGSA-X) for Benchmark Functions

ABSTRACT

The heuristic algorithms used in the solution of optimization problems may not be able to reach the best result for each problem resolutely due to their different designs. As a consequence, literature denotes some developing structures which are added to these heuristic algorithms. Within this scope, we evaluated the performance of gravitational search algorithm (GSA), which is one of the most prominent heuristic algorithm. In the study, aiming to increase the convergence speed of the algorithm, a new algorithm has been created by integrating some chaotic mapping methods to GSA. This new algorithm has been called chaos-based gravitational search algorithm (CbGSA-X). In order to make a performance evaluation for each mapping method, we utilized test functions which were solved with GSA in literature. Finally we introduce our solutions and performed evaluations.

Keywords: Optimization, Meta-heuristic algorithms, Gravitational search algorithm, Chaotic maps, Benchmark functions.

I. GİRİŞ

Son yıllarda karmaşık ve sayısal yöntemlerle çözümü zor ya da imkânsız olan problemlerin çözümünde sezgisel algoritmaların kullanımı giderek yaygınlaşmaktadır. Bunun nedeni, büyük arama uzaylarına sahip çok boyutlu problemlerin sayısal yöntemlerle çözümlerinin çok uzun süreler almasıdır. Bu tür problemlerin daha hızlı çözümüne yönelik birçok sezgisel algoritma geliştirilmiştir. Bu algoritmaların başlıcaları yerçekimsel arama algoritması (gravitational search algorithm-GSA) [1,2], çarpışan cisimler optimizasyonu (colliding bodies optimization-CBO) [3], büyük patlama - büyük çöküş yöntemi (big bang-big crunch method-BB-BC) [4], kara delik algoritması (black hole algorithm-BH) [5], tavlama benzetimi (simulated annealing-SA) [6], genetik algoritma (genetic algorithms-GA) [7], diferansiyel gelişim (differential evolution-DE) [8,9], parçacık sürü optimizasyonu (particle swarm optimization-PSO) [10], armoni arama (harmony search-HS) [11], yapay arı kolonisi (artificial bee colony-ABC) [12], yüklü sistem arama (charged system search-CSS) [13], su dalgası optimizasyonu (water-wave optimization-WWO) [14], balina optimizasyonu algoritması (the whale optimization algorithm-WOA) [15], guguk kuşu optimizasyon algoritması (cuckoo optimization algorithm-COA) [16], güvercin optimizasyonu (crow search algorithm-CSA) [17], optik tabanlı optimizasyon (optics inspired optimization-OIO) [18], bozkurt optimizasyonu (grey wolf optimizer-GWO) [19] ve girdap arama algoritmaları (vortex search-VS) [20] şeklinde belirtilebilir.

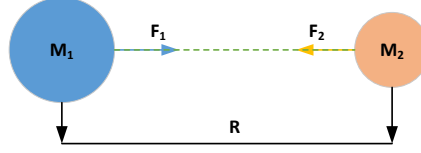
Sezgisel algoritmalar yapıları gereği geliştirilmeye açık algoritma tipleridir. Bu algoritmaların ilk tasarım aşamalarında yapılarında birçok eksik bulunmakta ve bu durum algoritma performansını etkilemektedir. Bu yüzden farklı araştırmacılar tarafından algoritmalar bazı yapılar ilave edilerek performanslarının iyileştirildiği çalışmalara literatürde sıklıkla rastlanmaktadır. Bu metodların başında algoritmaya zıt konumlu öğrenme yapısı, artırımlı sosyal öğrenme yapısı ve kaotik haritalama yöntemlerinin eklenmesi gelmektedir. Çalışmada Newton'un yerçekimi ve hareket kanunlarından esinlenilerek Rashedi ve arkadaşları tarafından 2009 yılında yayınlanmış olan yerçekimsel arama algoritması (GSA) geliştirilmek üzere seçilmiştir. Çalışmadaki amaç, GSA'ya bu çalışmada kaotik haritalama metodlarının entegre edilmesiyle performansının, kararlılığının ve yakınsama hızının iyileştirilmesi hedeflenmektedir.

Literatür taramalarında şimdiye kadar birçok algoritmanın kaotik yapılarla birlikte kullanıldığı görülmüştür. Bunlar kaotik biyocoğrafya tabanlı optimizasyon (chaotic biogeography-based optimisation -CBBO) [21], kaotik yüklü sistem arama (chaotic charged system search -CCSS) [22], kaotik meyve sineği (chaotic fruit fly optimization algorithm-CFOA) [23], kaotik yarası (chaotic bat algorithm-CBA) [24], kaotik kril sürü (chaotic krill herd algorithm-CKHA) [25], kaotik armoni arama (chaotic harmony search algorithm-CHS) [26], kaotik parçacık sürü optimizasyonu (chaotic particle swarm optimization-CPSO) [27] ve kaotik yapay arı kolonisi (chaotic artificial bee colony algorithm-CABC) [28] algoritmalarıdır. Bu çalışmalar incelendiğinde kaotik haritalama metodlarının, seçilen algoritmalar üzerinde olumlu etkilerinin olduğu belirlenmiştir.

Çalışmada GSA'ya kaos tabanlı haritalama metodları birlikte kullanıldığından ortaya çıkan algoritmaya Kaos tabanlı yerçekimsel arama algoritması (CbGSA-X) adı verilmiştir. CbGSA-X'de ilk popülasyondaki ajanlar oluşturulurken ilk ajan arama uzayında rastgele konumlandırılır, diğer ajanlar ise rastgele oluşturulan ajana bağlı olarak üçüncü bölümde detayları verilen 5 farklı (X=1, 2, 3, 4, 5) haritalama yöntemi kullanılarak konumlandırılmıştır. Bunun sonucunda ilk popülasyondaki ajanların arama uzayındaki dağılımları rastgele değil belirli bir düzene göre oluşturulmuştur. Elde edilen sonuçlar değerlendirilmiştir. Çalışmada iki farklı amaç öne çıkmaktadır, bunlardan birincisi GSA'nın iyileştirilmesi, diğeri ise 5 farklı kaotik haritalama metodundan daha iyi olanlarının belirlenmesidir.

II. YERÇEKİMSSEL ARAMA ALGORİTMASI

Fizikte, kütleyle sahip nesnelere birbirine doğru hızlanma eğilimi gösterir. Newton'un evrensel çekim kanununu özetleyen grafik Şekil 1'de, kütlelerin birbirlerine uyguladıkları kuvvetler Denklem (1)'de verilmiştir.

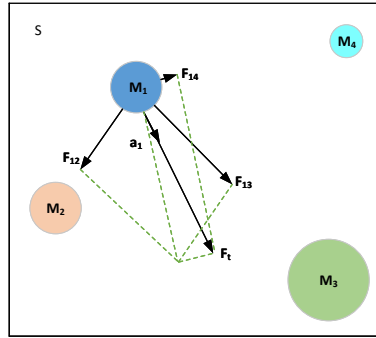


Şekil 1. Kütleler arası çekim kuvveti

$$F_1 = F_2 = G \frac{M_1 M_2}{R^2} \quad (1)$$

Kanuna göre; her bir noktasal kütle diğer noktasal kütleyle, ikisini birleştiren çizgi doğrultusundaki bir kuvvet ile çeker. Bu kuvvet bu iki kütlelerin çarpımıyla doğru orantılı, aralarındaki mesafenin karesi ile ters orantılıdır. Bu çekimin etkisiyle küçük olan kütle, büyük olan kütleyle doğru ivmelenecektir [1,2].

Yerçekimsel arama algoritması (GSA), Newton'un yerçekimi ve hareket kanunlarından esinlenilerek Rashedi ve arkadaşları tarafından 2009 yılında ortaya atılmıştır. GSA'da kütleler olarak adlandırılan bir dizi ajan Newton'un yerçekimi ve hareket kanunlarının simülasyonu ile optimum çözümü bulmak üzere tanımlanır. S arama uzayında konumlandırılan ajanlar (N) ve bir ajana (M_j) etkileyen kuvvetlerin gösterimi Şekil 2'de verilmiştir. Şekilde yer alan her bir kütlelerin bulunduğu konum değerleri, mevcut problemin birer aday çözümüdür. Kütlelerin büyüklüğü çözüme ne kadar yakın olduklarının göstergesidir [1,2].



Şekil 2. Arama uzayındaki bir ajana etkileyen kuvvetler

GSA, i . kümenin konumunun denklem (2)'de verildiği gibi tanımlandığı N ajanlı bir yapıyla aramaya başlar. Bu ajanlar klasik GSA'da ilk popülasyon için S arama uzayında rastgele konumlandırılırlar [1,2].

$$X_i = (x_i^1, \dots, x_i^d, \dots, x_i^n), \quad i = 1, 2, \dots, N \quad (2)$$

Burada x_i^d , d . boyuttaki i . kümenin konumudur, n ise arama alanının boyutudur.

Her bir ajanın kütlesi, uygunluğu (çözüme yakınlığı) ile temsil edilir ve popülasyondaki diğer bireylerin uygunluğuna bağlı olarak denklem (3) ve (4)'e göre hesaplanır [1], [2].

$$q_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)} \quad (3)$$

$$M_i(t) = \frac{q_i(t)}{\sum_{j=1}^s q_j(t)} \quad (4)$$

Bu denklemlerde $q_i(t)$, $M_i(t)$ ve $fit_i(t)$ sırasıyla bağıl kütle, kütle ve t anındaki i . ajanın uygunluk değerini göstermektedir.

Bir minimizasyon problemi için en iyi $best(t)$ ve en kötü $worst(t)$ çözüm değerleri denklem (5) ve (6)'ya göre tanımlanır.

$$best(t) = \min_{j \in \{1, \dots, s\}} fit_j(t) \quad (5)$$

$$worst(t) = \max_{j \in \{1, \dots, s\}} fit_j(t) \quad (6)$$

Problem maksimizasyon problemi en iyi $best(t)$ ve en kötü $worst(t)$ çözüm değerleri yer değiştirilerek aşağıdaki gibi tanımlanır.

$$best(t) = \max_{j \in \{1, \dots, s\}} fit_j(t) \quad (7)$$

$$worst(t) = \min_{j \in \{1, \dots, s\}} fit_j(t) \quad (8)$$

Bir ajanın ivmesi hesaplanırken, öncelikle diğer kütleler (ajanlar) tarafından onun üzerine uygulanan toplam güç, yerçekimi kanununa bağlı olarak hesaplanır. Söz konusu ajana etkiyen toplam güç denklem (9)'daki gibidir [1,2].

$$F_i^d(t) = \sum_{j \in k_{best}, j \neq 1} rand_j G(t) \frac{M_j(t) M_i(t)}{R_{ij}(t) + \varepsilon} [x_j^d(t) - x_i^d(t)] \quad (9)$$

Burada $rand_i$ ve $rand_j$ $[0,1]$ aralığında dağıtılmış rastgele iki sayıdır. ε ise tanımsızlığı ortadan kaldırmak için kullanılan çok küçük bir değerdir. $R_{ij}(t)$, i ve j ajanları arasındaki öklid mesafesidir ve $R_{ij}(t) = \|X_i(t), X_j(t)\|_2$ olarak tanımlanır. k_{best} , en iyi uygunluk değerli dolayısıyla en büyük kütleyle sahip ajanların bir dizisidir, ki bu da başlangıçta K_0 'da başlatılan ve zamanla azaltılan bir fonksiyondur. Burada K_0 , toplam ajan sayısı (N) kümesidir ve doğrusal olarak 1 'e düşürülür. Bunun anlamı başlangıçta bütün ajanlar birbirlerine kuvvet uygularken, zaman geçtikçe kuvvet uygulayan ajan sayısı azalacak ve sonunda sistemde diğer kütlelere kuvvet uygulayan tek bir ajan kalacaktır.

Toplam güç hesabından sonra hareket kanunları kullanılarak denklem (10)'dan ajanın ivmesi hesaplanır [1,2].

$$a_i^d(t) = \frac{F_i^d(t)}{M_i(t)} = \sum_{j \in k_{best}, j \neq 1} rand_j G(t) \frac{M_j(t)}{R_{ij}(t) + \varepsilon} [x_j^d(t) - x_i^d(t)] \quad (10)$$

Ajana ait denklem (10)'dan hesaplanan ivme değeri, ajanın o andaki mevcut hızına eklenerek denklem (11)'de gösterildiği gibi yeni hız vektörü elde edilir.

$$v_i^d(t+1) = rand_i \times v_i^d(t) + a_i^d(t) \quad (11)$$

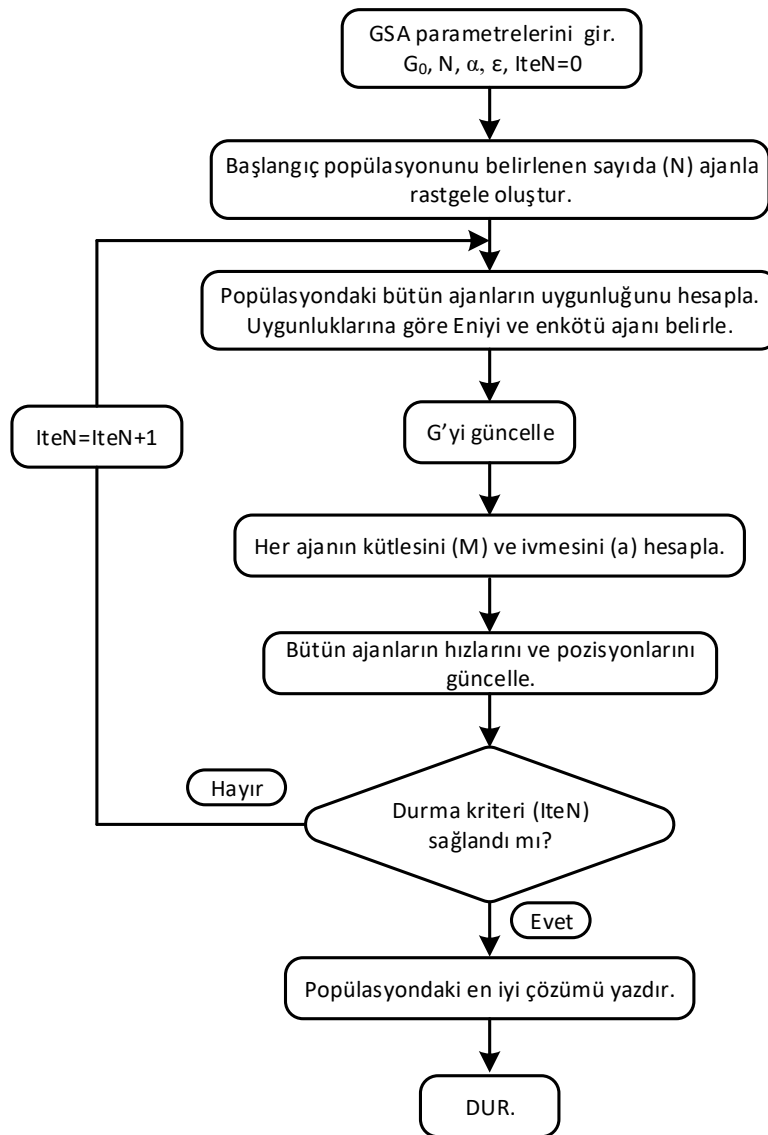
Son olarak, ajanın bir sonraki popülasyondaki konumu, denklem (12)'ye göre belirlenir.

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \quad (12)$$

Kuvvet ve ivme denklemlerinde yer alan yerçekimsel sabit $G(t)$, bir başlangıç değeri G_0 , α sabit katsayı, t şu anki iterasyon sayısı T ise bitiş iterasyon sayısını göstermek üzere denklem (13)'te verilmiştir. Belirlenen G_0 ve α değerlerinin en iyi çözümü elde etme ve algoritmanın yakınsama hızı üzerine etkileri büyüktür. Dolayısıyla her problemin özelliğine göre bu değerlerin yeniden belirlenmesi gerekmektedir [1,2].

$$G(t) = G_0 e^{-\alpha(t/T)} \quad (13)$$

Yukarıda çözüm aşamaları anlatılan GSA algoritmasının akış diyagramı Şekil 3'te verilmiştir.



Şekil 3. GSA akış diyagramı

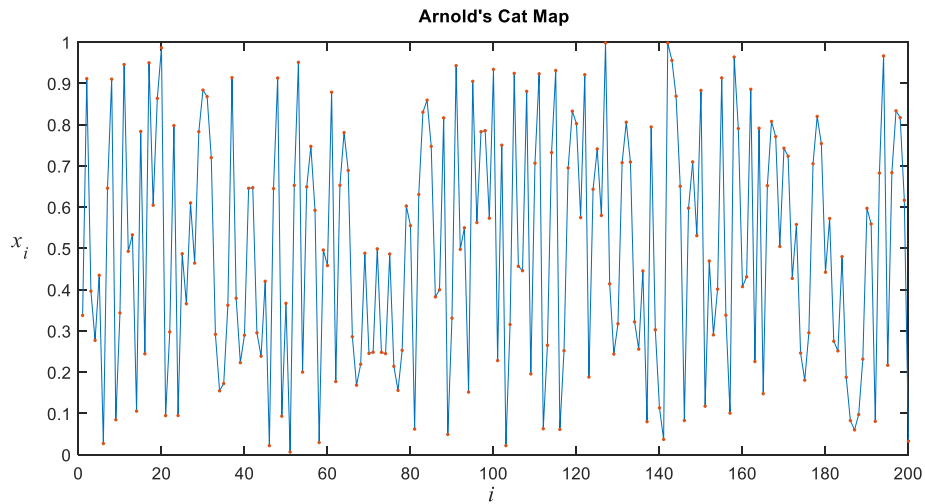
III. KAOTİK HARİTALAMA METOTLARI

Kaos, başlangıç şartlarına bağlı olarak ortaya çıkan, sınırları belli, kararsız ve dinamik bir davranıştır. Son yıllarda rastgele seçim işlemi yerine kaos tabanlı sistemleri kullanma fikri oldukça yaygınlaşmaktadır. Özellikle rastgele seçim temeline sahip optimizasyon algoritmalarında, kaotik haritalar, rastgele seçim rolünü üstlenmektedirler. Rastgele işaretler yerine kaotik işaretlerin kullanılmasının avantajı deneysel çalışmalarda daha fazla ortaya çıkmaktadır. Özellikle sayısal hesaplama, karar verme ve sezgisel optimizasyon kullanımlarında rastgele seçim denklemleri, uzun bir periyoda ve dengeli olmayan düzgün dağılımlara ihtiyaç duyarlar. Oysa kaotik yapılar rastgele benzeri işlemlerin bulunduğu deterministik bir dinamik sistemdir. Doğası gereği rastgele gibi görünür ve öngörülemez. Ama bir düzen elemanı gibi davranır ve bir düzende işlem görür. Bir düzeni izleyen kaotik haritalar rastgele sayı üretiminin kaynağı olarak ta kullanılırlar [29-31]. Bu çalışmada aşağıda detayları verilen 5 farklı kaotik haritalama metodu incelenmiştir.

A. ARNOLD'UN KEDİ KAOTİK HARİTALAMASI (ARNOLD'S CAT MAP-ACM)

Matematikte Arnold'un kedi kaotik haritalaması (ACM) olarak bilinen tanımı Vladimir Arnold tarafından 1968 yılında yapılmıştır [32]. ACM'nin tanımlanması çalışmaları bir kedi resmi üzerinde yapıldığı için bu ismi almıştır. Bu metodun matematiksel gösterimi denklem (14)'te, 200 ajan için arama uzayı haritalaması Şekil 4'te, metodun matlab kodu ise Şekil 5'te verilmiştir.

$$\begin{bmatrix} X_{k+1} \\ Y_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} X_k \\ Y_k \end{bmatrix} \pmod{1}, \quad \text{burada } X_k, Y_k \in [0,1] \quad (14)$$



Şekil 4. Arnold'un kedi kaotik haritalaması (ACM) - 200 ajan

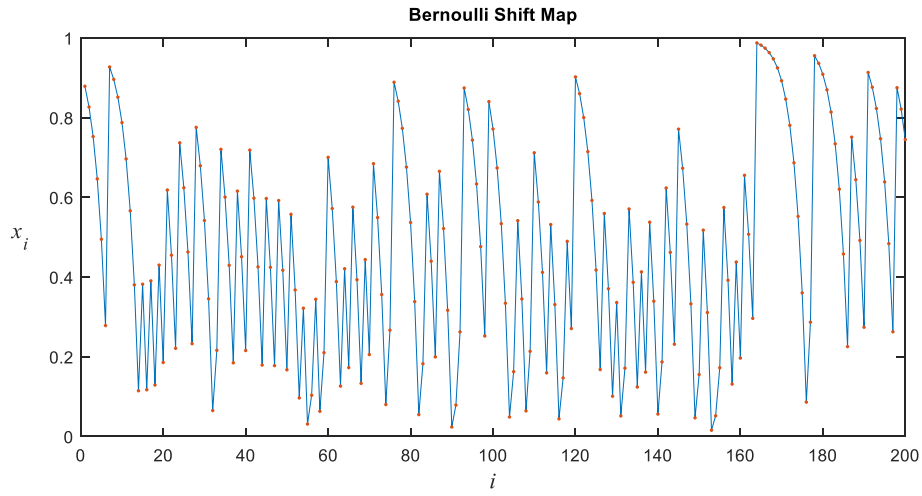
```
case 1, % Arnold's Cat Map
x = rand;
y = rand;
for k = 1 : ChaoK
    gec = mod(x+y,1);
    y = mod(x+2*y,1);
    x = gec;
    ChaoArray(k) = x;
end
```

Şekil 5. Arnold'un kedi kaotik haritalaması metodunun matlab kodu

B. BERNOULLİ KAYDIRMA KAOTİK HARİTASI (BERNOULLI SHIFT MAP-BSM)

Bernoulli kaydırma kaotik haritası iki çizgiden oluşmaktadır. Bu çizgilerden biri 0 ile λ arasındaki eğimi $1/\lambda$ olan bir çizgidir. Diğeri ise λ ile 1 arasında eğimi $1/(1-\lambda)$ olan bir çizgidir. BSM tüm kaotik sistemler gibi başlangıç parametresi ile başlar, bu parametreyi belirli değerlere dönüştürür [33]. $\lambda=0,7$ alınarak elde edilen BSM'nin gösterimi Şekil 6'da verilmiştir. Bu haritalama metodunun özel bir durumu denklem (15)'te, çalışmada kullanılan matlab kodu Şekil 7'de verilmiştir.

$$X_{k+1} = \begin{cases} \frac{X_k}{1-\lambda} & 0 < X_k \leq 1-\lambda \\ \frac{X_k - (1-\lambda)}{\lambda} & 1-\lambda \leq X_k < 1 \end{cases} \quad (15)$$



Şekil 6. Bernoulli kaydırma kaotik haritası (BSM) - 200 ajan

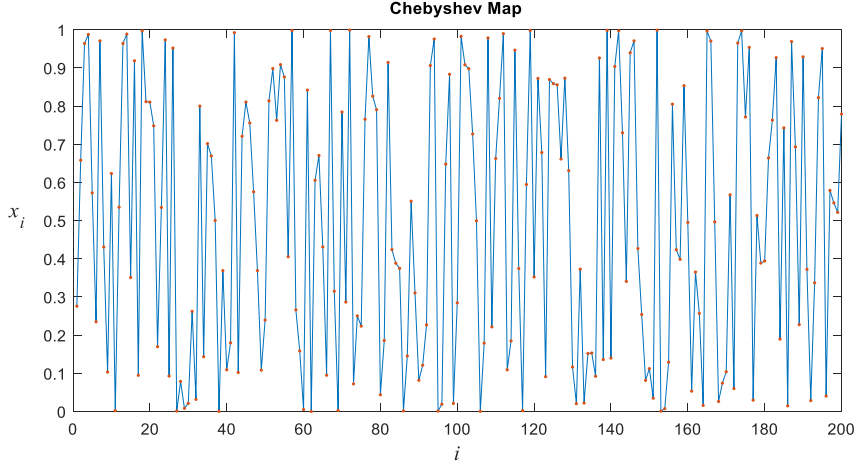
```
case 2, % Bernoulli Shift Map
lamda = 0.7;
x = rand;
for k = 1 : ChaoK
    if x < 1-lamda
        x = x/(1-lamda);
    else
        x = (x-(1-lamda))/lamda;
    end
    ChaoArray(k) = x;
end
```

Şekil 7. Bernoulli kaydırma kaotik haritası metodunun matlab kodu

C. CHEBYSHEV KAOTİK HARİTASI (CHEBYSHEV MAP-CM)

Chebyshev kaotik haritası ile önerilen sözde rasgele bit üretici, iki Chebyshev polinomuna dayanmaktadır. Bu metodun matematiksel gösterimi Denklem (16)'da verilmiştir [34], [35]. Bu çalışmada $f=50$ alınmıştır. Bu metodun 200 ajan için arama uzayı haritalaması Şekil 8'de, metodun çalışmada kullanılan Matlab kodu ise Şekil 9'da verilmiştir.

$$X_{k+1} = \cos(f \cos^{-1} X_k), \quad burada \quad f \in [2, \infty) \quad (16)$$



Şekil 8. Chebyshev kaotik haritalaması (CM) - 200 ajan

```

case 3, % Chebyshev Map
f = 50;
x = rand;
for k = 1 : ChaoK
    x = cos(f*acos(x));
    ChaoArray(k) = (x+1)/2;
end

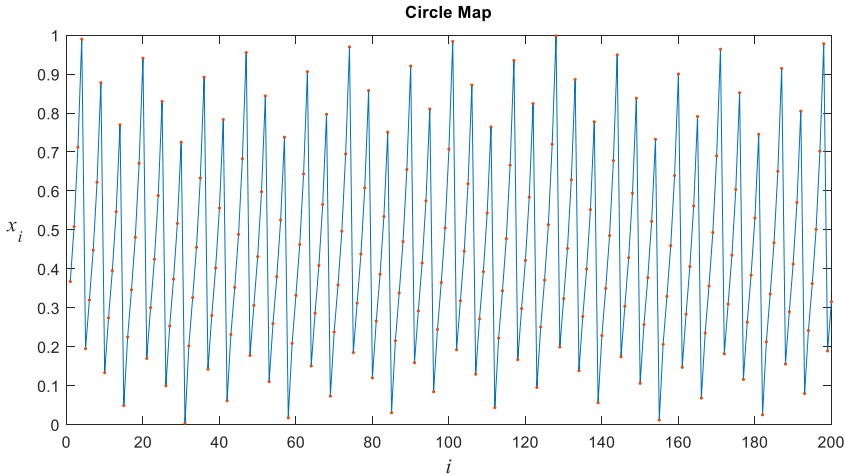
```

Şekil 9. Chebyshev kaotik haritalama metodunun matlab kodu

D. DAİRE KAOTİK HARİTASI (CIRCLE MAP-CIRCLEM)

Daire kaotik haritası, ilk olarak Andrey Kolmogorov tarafından tanımlanmıştır. Bu harita, daire üzerinde yer alan bir dinamik sistem ailesinin üyesidir. Daire kaotik haritası denklemleri, elektronikte faz kilitlemeli döngünün basitleştirilmiş bir modelini de açıklamaktadır. (0,1) aralığında kaotik dizilimler üreten bir kaotik haritalama metodu olan daire kaotik haritasının matematiksel ifadesi denklem (17)'deki gibidir [36]. Bu metodun 200 ajan için arama uzayı haritalaması Şekil 10'da, çalışmada kullanılan matlab kodu ise Şekil 11'de verilmiştir.

$$X_{k+1} = \left\{ X_k + b - \left(\frac{a}{2\pi} \right) \sin(2\pi X_k) \right\} \text{mod}(1), \quad a = 0.5, \quad b = 0.2 \quad (17)$$



Şekil 10. Daire kaotik haritası (CircleM) - 200 ajan


```

case 4, % Circle Map
a = 0.5;
b = 0.2;
x = rand;
for k = 1 : ChaoK
    x = mod(x+b-sin(2*pi*x)*a/(2*pi),1);
    ChaoArray(k) = x;
end

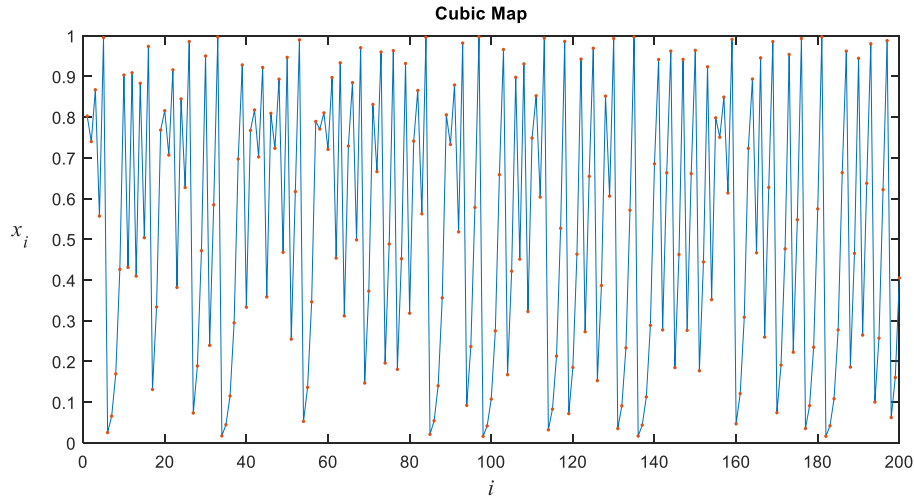
```

Şekil 11. Daire kaotik haritası metodunun matlab kodu

E. KÜBİK KAOTİK HARİTASI (CUBIC MAP-CUBICM)

Kübik kaotik haritası, çeşitli uygulamalarda kaotik dizilerin üretilmesinde en yaygın kullanılan haritalardan biridir. (0,1) aralığında kaotik dizilimler üreten bir kaotik haritalama metodu olan kübik kaotik haritanın matematiksel tanımı denklem (18)'deki gibidir [37,38]. Bu metodun 200 ajan için arama uzayı haritalaması Şekil 12'de, bu çalışmada kullanılan matlab kodu ise Şekil 13'te verilmiştir.

$$X_{k+1} = pX_k(1 - X_k^2), \quad X_k \in (0,1), \quad p = 2.59 \quad (18)$$



Şekil 12 Kübik kaotik haritalaması (CubicM) - 200 ajan

```

case 5, %Cubic Map
a = 2.59;
x = rand;
for k = 1 : ChaoK
    x = a*x*(1-x^2);
    ChaoArray(k) = x;
end

```

Şekil 13. Kübik kaotik haritalaması metodunun matlab kodu

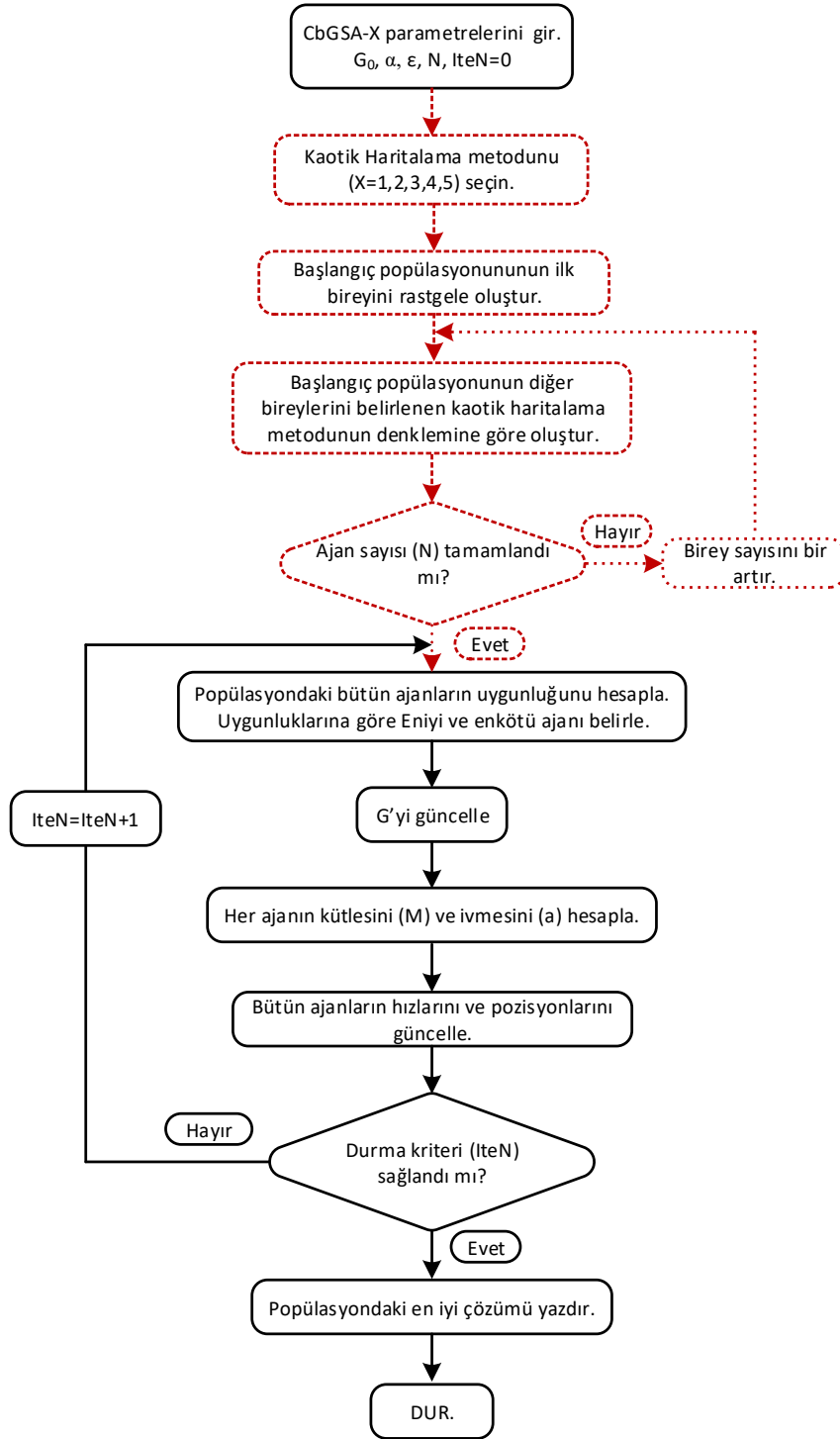
IV. KAOS TABANLI YERÇEKİMSSEL ARAMA ALGORİTMALARININ GELİŞTİRİLMESİ

Kaos tabanlı yerçekimsel arama algoritmasında (CbGSA-X) ilk popülasyondaki ajanlar oluşturulurken ilk ajan arama uzayında rastgele konumlandırılmış, diğer ajanlar ise rastgele oluşturulan ajana bağlı olarak üçüncü bölümün alt bölümlerinde tanımları yapılan 5 farklı ($X=1, 2, 3, 4, 5$) haritalama yöntemleri kullanılarak konumlandırılmıştır. Çalışmada bu kaotik haritalama yapıları test fonksiyonlarına uygulanmıştır. Belirlenen bütün kaotik haritaların arama uzayları $[0-1]$ arama aralığına haritalanmıştır. CbGSA-X'e ait akış diyagramı Şekil 14'te verilmiştir. CbGSA-X'in GSA'dan farklılıkları Şekil 14 üzerinde kırmızı renkte çizgilerle belirtilmiştir. Çalışmada CbGSA-X, ($X=1, 2, 3, 4, 5$) olarak adlandırılan algoritmalar 30 boyutlu test fonksiyonlarına ayrı ayrı uygulanmış ve sonuçlar değerlendirilmiştir.

Arama algoritmalarının ilkendirme aşamasında başlangıç havuzunun oluşumunda bireylerin konumları denklem (19)'a göre rastgele oluşturulmaktadır. Çalışmada ise bireyler, bu rastgele konumlandırma yerine kaotik haritalardan üretilecek sayı değerlerine göre başlangıç havuzuna denklem (20)'ye göre yerleştirilecektir. Bu şekilde başlangıç havuzu rastlantısal değil daha düzenli bir dağılım ile oluşturulmuş olacaktır [29-31]. Önerilen metodun amaçları, yerel arama hızını artırma, yakınsama hızını artırma ve ajanların sistematik davranmasını sağlamak olarak sayılabilir.

$$x_{i,j}^{(o)} = x_{i,\min} + rand_{ij} \cdot (x_{i,\max} - x_{i,\min}), \quad i = 1, 2, \dots, N \quad (19)$$

$$x_{i,j}^{(o)} = x_{i,\min} + CM_{ij} \cdot (x_{i,\max} - x_{i,\min}), \quad i = 1, 2, \dots, N \quad (20)$$



Şekil 14. CbGSA-X'in akış diyagramı

V. TEST FONKSİYONLARI

Önerilen algoritmalar, performanslarının değerlendirilebilmesi için literatürde daha önce farklı algoritmalar ile çözümü yapılmış 23 adet test fonksiyonuna uygulanmıştır. Bu fonksiyonlar üç grup halinde Tablo 1, 2 ve 8'de verilmiştir. Tablolarda (n) değeri fonksiyonun boyutunu, (S) arama uzayını, f_{\min} ise fonksiyonun minimum değerini göstermektedir [1,2]. Tablo 1 ve 2'deki $f_1 - f_{13}$ arasındaki

fonksiyonlar yüksek boyutlu veya geniş arama uzayına sahip fonksiyonlardır. Bu özelliklerinin yanında Tablo 1'deki $f_1 - f_7$ arasındaki fonksiyonlar tek bir optimum noktaya sahip unimodal fonksiyonlar iken, Tablo 2'deki $f_8 - f_{13}$ arasındaki fonksiyonlar ise birçok lokal minimum noktaları olan multimodal fonksiyonlardır. Tablo 8'de yer alan $f_{14} - f_{23}$ arasındaki fonksiyonlar ise düşük ve sabit boyutlu, az sayıda lokal minimum noktaları olan multimodal fonksiyonlardır.

Tablo 1. Unimodal test fonksiyonları

Formül	Adı	D (n)	Aralık (S)	f_{\min}
$f_1(x) = \sum_{i=1}^n x_i^2$	Sphere	30	$[-100,100]^n$	0
$f_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	Schwefel's No: 2.22	30	$[-10,10]^n$	0
$f_3(x) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2$	Schwefel's No: 1.2	30	$[-100,100]^n$	0
$f_4(x) = \max\{ x_i , 1 \leq i \leq n\}$	Schwefel's No: 2.21	30	$[-100,100]^n$	0
$f_5(x) = \sum_{i=1}^{n-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right]$	Rosenbrock	30	$[-30,30]^n$	0
$f_6(x) = \sum_{i=1}^n ([x_i + 0.5])^2$	Step	30	$[-100,100]^n$	0
$f_7(x) = \sum_{i=1}^n ix_i^4 + \text{random}[0,1)$	Quartic	30	$[-1.28,1.28]^n$	0

Tablo 2. Multimodal test fonksiyonları

Formül	Adı	D (n)	Aralık (S)	f_{\min}
$f_8(x) = \sum_{i=1}^n -x_i \sin(\sqrt{ x })$	Schwefel's No: 2.26	n	$[-500,500]^n$	(-418.9829) x n
$f_9(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	Rastrigin	n	$[-5.12,5.12]^n$	0
$f_{10}(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$	Ackley	n	$[-32,32]^n$	0
$f_{11}(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	Griewank	n	$[-600,600]^n$	0
$f_{12}(x) = \frac{\pi}{n} \left\{ 10 \sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] \right\} + \sum_{i=1}^{n-1} u(x_i, a, k, m)$ $y_i = 1 + \frac{x_i + 1}{4}, a = 10, k = 100, m = 4$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	Penalized No: 1	n	$[-50,50]^n$	0
$f_{13}(x) = 0.1 \left\{ \sin^2(3\pi x_1) + \sum_{i=1}^n (x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2 [1 + \sin^2(2\pi x_n)] \right\} + \sum_{i=1}^n u(x_i, a, k, m), a = 5, k = 100, m = 4$	Penalized No: 2	n	$[-50,50]^n$	0

Tablo 8'de yer alan fonksiyonlardan $f_{14,15}$ ve f_{19-22} 'e ait $a_i, b_i, c_i, a_{ij}, b_{ij}$ ve P_{ij} katsayılarının değerleri sırasıyla Tablo 3-7'de verilmiştir.

Tablo 3: $f_{14}(x)$ fonksiyonundaki a_{ij} katsayıları

j	1	2	3	4	5	...	21	22	23	24	25
$a_{ij} \quad i = 1, 2$	-32	-16	0	16	32	...	-32	-16	0	16	32
	-32	-32	-32	-32	-32	...	32	32	32	32	32

Tablo 4: $f_{15}(x)$ fonksiyonundaki a_i ve b_i katsayıları

i	1	2	3	4	5	6	7	8	9	10	11
a_i	0,1957	0,1947	0,1735	0,1600	0,0844	0,0627	0,0456	0,0342	0,0323	0,0323	0,0246
b_i^{-1}	0,25	0,5	1	2	4	6	8	10	12	14	16

Tablo 5: $f_{19}(x)$ fonksiyonundaki a_{ij} , c_i ve P_{ij} katsayıları

i	$a_{ij}, j=1,2,3$			c_i	$P_{ij}, j=1,2,3$		
1	3	10	30	1	0,3689	0,1170	0,2673
2	0,1	10	35	1,2	0,4699	0,4387	0,7470
3	3	10	30	3	0,1091	0,8732	0,5547
4	0,1	10	30	3,2	0,03815	0,5743	0,8828

Tablo 6: $f_{20}(x)$ fonksiyonundaki a_{ij} , c_i ve P_{ij} katsayıları

i	$a_{ij}, j=1,2,3,4,5,6$						c_i	$P_{ij}, j=1,2,3,4,5,6$					
1	10	3	17	3,5	1,7	8	1	0,131	0,169	0,556	0,012	0,828	0,588
2	0,05	10	17	0,1	8	14	1,2	0,232	0,413	0,830	0,373	0,100	0,999
3	3	3,5	1,7	10	17	8	3	0,234	0,141	0,352	0,288	0,304	0,665
4	17	8	0,05	10	0,1	14	3,2	0,404	0,882	0,873	0,574	0,109	0,038

Tablo 7: $f_{21}(x)$ $f_{22}(x)$ ve $f_{23}(x)$ fonksiyonlarındaki a_{ij} ve c_i katsayıları

i	$a_{ij}, j=1,2,3,4$				c_i
1	4	4	4	4	0,1
2	1	1	1	1	0,2
3	8	8	8	8	0,2
4	6	6	6	6	0,4
5	3	7	3	7	0,4
6	2	9	2	9	0,6
7	5	5	3	3	0,3
8	8	1	8	1	0,7
9	6	2	6	2	0,5
10	7	3,6	7	3,6	0,5

Tablo 8. Sabit boyutlu multimodal test fonksiyonları

Formül	Adı	$D(n)$	Aralık (S)	f_{\min}
$f_{14}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6} \right)^{-1}$	Shekel's Foxholes	2	$[-65.53, 65.53]^2$	0.998
$f_{15}(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_i(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]$	Kowalik	4	$[-5, 5]^4$	0.0003
$f_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	Six-Hump Camel-Back	2	$[-5, 5]^2$	-1.0316
$f_{17}(x) = \left(x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6 \right)^2 + 10 \left(1 - \frac{1}{8\pi} \right) \cos x_1 + 10$	Branin	2	$[-5, 10] \times [0, 15]$	0.397887
$f_{18}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times \dots [30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	Goldstein-Price	2	$[-2, 2]^2$	3
$f_{19}(x) = -\sum_{i=1}^4 c_i \exp\left(-\sum_{j=1}^3 a_{ij}(x_j - p_{ij})^2\right)$	Hartman3	3	$[0, 1]^3$	-3.86278
$f_{20}(x) = -\sum_{i=1}^4 c_i \exp\left(-\sum_{j=1}^6 a_{ij}(x_j - p_{ij})^2\right)$	Hartman6	6	$[0, 1]^6$	-3.32237
$f_{21}(x) = -\sum_{i=1}^5 \sum_{j=1}^4 [(x_j - a_{ij})(x_j - a_{ij})^T + c_i]^{-1}$	Shekel5	4	$[0, 10]^4$	-10.1532
$f_{22}(x) = -\sum_{i=1}^7 \sum_{j=1}^4 [(x_j - a_{ij})(x_j - a_{ij})^T + c_i]^{-1}$	Shekel7	4	$[0, 10]^4$	-10.4029

$$f_{25}(x) = -\sum_{i=1}^{10} \sum_{j=1}^4 [(x_j - a_{ij})(x_j - a_{ij})^T + c_i]^{-1}$$

Shekel10

4

[0,10]⁴

-10.5364

VI. SAYISAL SONUÇLAR

Literatürde daha önce GSA ile çözülmüş olan test fonksiyonları, bu çalışmada CbGSA-X (X=1, 2, 3, 4, 5) ile çözülmüş, sonuçları literatürdeki GSA sonuçlarıyla karşılaştırılmıştır. Geliştirilen metotlar yakınsama hızı ve performans yönünden değerlendirilmiştir. Test fonksiyonlarının çözümü için MATLAB R2015b’de geliştirilen program Intel Xeon E5-2637 v4 3.50 GHz işlemcili ve 128 GB RAM bellekli iş istasyonunda Tablo 1 ve 2’deki fonksiyonlar için 1000 iterasyon (50000 fonksiyon çağırımı, FCall) ve Tablo 8’deki fonksiyonlar için 500 iterasyon (25000 fonksiyon çağırımı, FCall) çalıştırılmıştır. Çalışmada Tablo 1, 2 ve 8’deki test fonksiyonlarının, performans açısından karşılaştırmalarının yapılabilmesi için literatürdeki GSA ile aynı şartlar, boyutlar ve parametrelerle çözülmüş olmasına dikkat edilmiştir. CbGSA-X için parametre değerleri Tablo 9’da verilmiştir. GSA’nın geliştirilen beş farklı kaotik tabanlı yapıları olan CbGSA-X (X=1, 2, 3, 4, 5) kullanılarak f_{1-7} unimodal fonksiyonlar için $n=30$ (30-D) iken elde edilen değerleri Tablo 10’da verilmiştir.

Tablo 9. CbGSA-X parametre değerleri

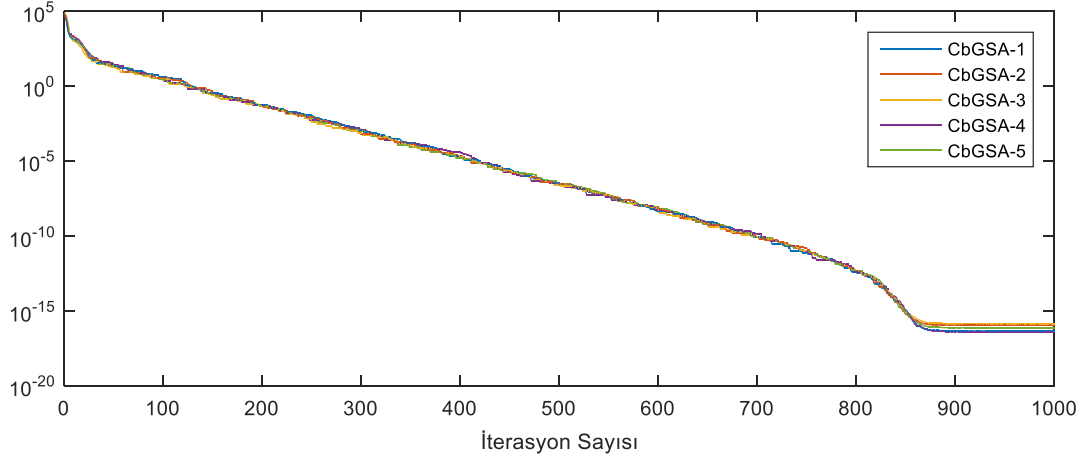
<i>Fonksiyon</i>	<i>İterasyon Sayısı (IteN)</i>	<i>Ajan Sayısı (N)</i>	<i>Fonksiyon Çağırımı (fCall)</i>	<i>Yerçekimsel Sabit Başlangıç Değeri (Go)</i>	<i>Sabit Sayı (a)</i>	<i>Tanımsızlık Sabiti (ε)</i>
f_{1-13}	1000	50	50000	100	20	10 ⁻⁶
f_{14-23}	500	50	25000	100	20	10 ⁻⁶

Tablo 10. 30-D için elde edilen veriler (Tablo 1 - 30 çalışma - 1000 iterasyon)

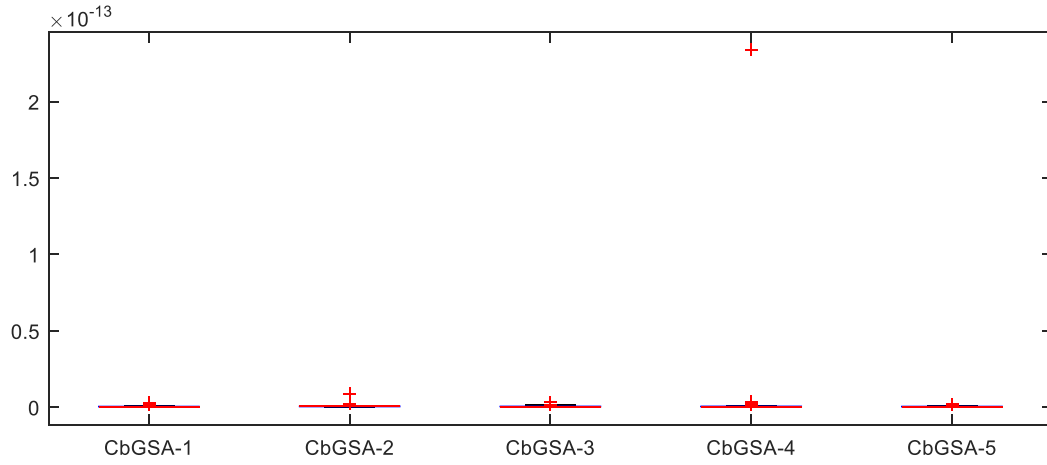
		<i>CbGSA-1</i>	<i>CbGSA-2</i>	<i>CbGSA-3</i>	<i>CbGSA-4</i>	<i>CbGSA-5</i>
f_1	En kötü	2,659902e-15	8,718485e-15	3,102217e-15	2,339266e-13	1,890799e-15
	Ortalama	5,068916e-16	8,503192e-16	5,950063e-16	8,345003e-15	4,832201e-16
	En İyi	4,401041e-17	1,091713e-16	1,364454e-16	3,892299e-17	7,274784e-17
	St. Sapma	5,288955e-16	1,514634e-15	5,591161e-16	4,189458e-14	4,087875e-16
	Süre (s)	5,05177	4,70398	5,33752	4,25085	5,81777
f_2	En kötü	1,281331e-02	2,118761e-06	3,094142e-03	3,830644e-06	3,653049e-07
	Ortalama	5,819513e-04	2,479872e-07	1,170670e-04	2,688439e-07	1,058782e-07
	En İyi	3,749932e-08	2,185942e-08	3,031872e-08	2,376645e-08	3,004552e-08
	St. Sapma	2,418987e-03	5,160319e-07	5,552978e-04	7,421988e-07	7,168612e-08
	Süre (s)	4,85296	4,96636	4,71208	5,22266	4,54053
f_3	En kötü	1,402426e+03	1,105040e+03	1,516259e+03	6,106312e+00	5,784289e+01
	Ortalama	1,344038e+02	7,397278e+01	2,028870e+02	4,569989e-01	1,166923e+01
	En İyi	1,513509e-05	6,530554e-06	2,236669e-02	2,535218e-05	1,284255e-05
	St. Sapma	2,494661e+02	2,110143e+02	3,779615e+02	1,257213e+00	1,914248e+01
	Süre (s)	5,46154	4,25937	4,45371	4,45712	4,50141
f_4	En kötü	8,100296e+00	3,126170e+00	5,023718e+00	8,931477e+00	2,288748e+00
	Ortalama	4,481255e-01	1,051494e-01	2,304151e-01	4,468437e+00	1,457643e-01
	En İyi	1,920559e-07	4,510397e-08	1,096698e-07	2,351702e-07	1,073174e-07
	St. Sapma	1,491426e+00	5,609979e-01	9,119072e-01	3,185851e+00	4,853315e-01
	Süre (s)	3,80374	4,02210	4,07426	3,77101	3,82654
f_5	En kötü	3,843815e+02	2,320016e+02	5,954874e+02	3,303691e+02	8,548088e+02
	Ortalama	5,141849e+01	5,032405e+01	6,283391e+01	5,001535e+01	1,211890e+02
	En İyi	2,141220e+01	2,266581e+01	2,355737e+01	2,245741e+01	2,354076e+01
	St. Sapma	6,783763e+01	5,192926e+01	1,075680e+02	6,047400e+01	1,758623e+02
	Süre (s)	4,07221	4,37168	4,7035	4,5398	4,30898
f_6	En kötü	1,509689e-15	2,414345e-15	7,496921e-13	2,596010e-15	2,888467e-15
	Ortalama	5,042535e-16	6,108282e-16	2,542011e-14	6,388628e-16	6,449536e-16
	En İyi	5,875281e-17	6,902870e-17	1,728119e-17	6,100148e-17	5,184519e-17
	St. Sapma	4,423187e-16	6,259308e-16	1,344946e-13	6,266236e-16	5,954833e-16
	Süre (s)	5,54664	5,04064	5,01527	4,89886	4,98583
f_7	En kötü	4,991181e-02	5,643973e-02	4,029613e-02	4,676078e-02	4,389277e-02
	Ortalama	1,972576e-02	2,025446e-02	1,762027e-02	1,821690e-02	1,700346e-02

En İyi	2,622054e-03	5,125597e-03	6,492990e-03	4,362321e-03	5,481061e-03
St. Sapma	1,142993e-02	1,168216e-02	8,221138e-03	1,058315e-02	9,088907e-03
Süre (s)	5,92263	6,05422	5,737	5,91528	5,7297

Tablo 10'da yer alan f_1 fonksiyonu için 30 çalışmada elde edilen en iyi çözümlere ait, iterasyon sayısına göre yakınsamayı gösteren grafik Şekil 15.a'da verilmiştir. Aynı fonksiyon için 30 çalışmadaki en iyi değerlerin yayılımını gösteren kutu grafiği ise Şekil 15.b'de gösterilmiştir. Benzer grafikler tablodaki bütün fonksiyonlar için elde edilmiş fakat çok fazla yer kaplayacağı düşüncesiyle burada gösterilmemiştir. Ancak sonuç ve algoritmalar arasındaki performans farklılıklarının değerlendirmesinde f_{1-7} grafiklerinin tümü dikkate alınmıştır.



(a)



(b)

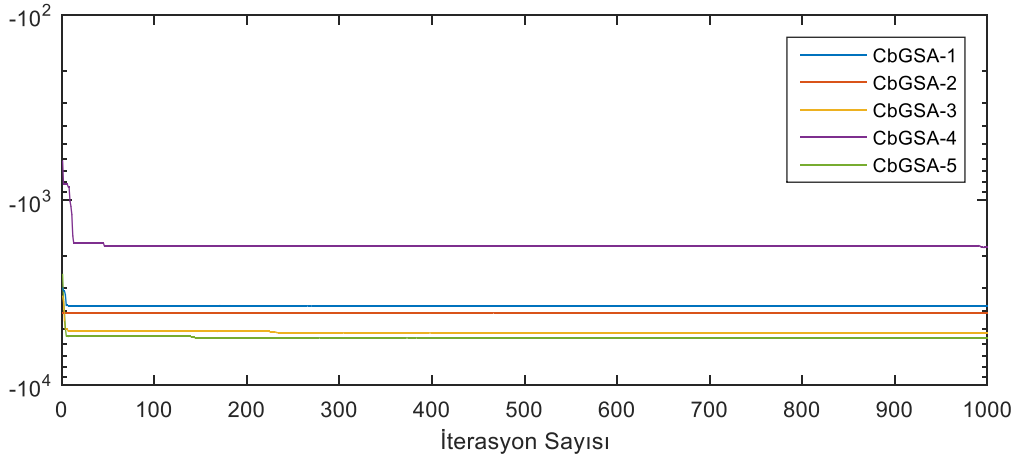
Şekil 15. (a) f_1 için yakınsama eğrileri, (b) f_1 kutu grafikleri (30-D)

CbGSA-X (X=1, 2, 3, 4, 5) kullanılarak f_{8-13} multimodal fonksiyonları için $n=30$ (30-D) iken elde edilen değerleri Tablo 11'de verilmiştir.

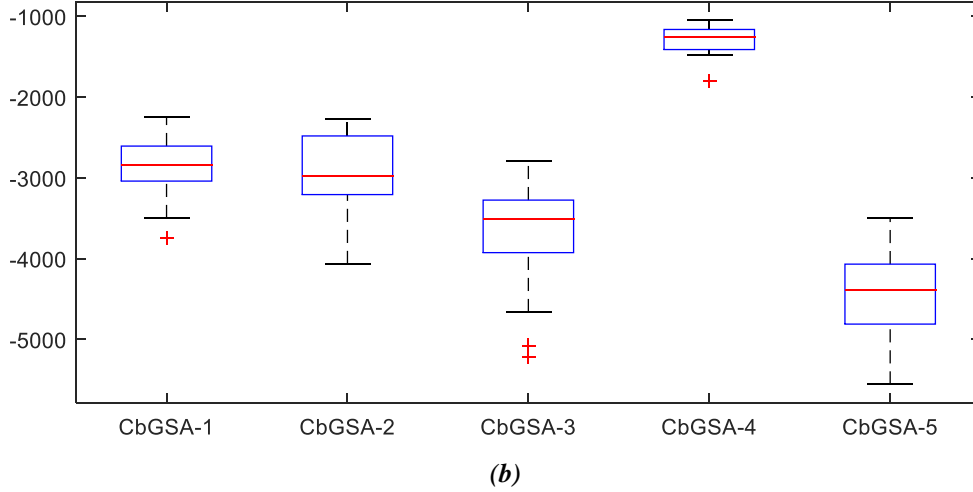
Tablo 11'de yer alan f_8 fonksiyonu için 30 çalışmada elde edilen en iyi çözümlere ait, iterasyon sayısına göre yakınsamayı gösteren grafik Şekil 16.a'da gösterilmiştir. 30 çalışma için en iyi değerlerin yayılımını gösteren kutu grafiği ise Şekil 16.b'de verilmiştir.

Tablo 11. 30-D için elde edilen veriler (Tablo 2 - 30 çalışma - 1000 iterasyon)

		CbGSA-1	CbGSA-2	CbGSA-3	CbGSA-4	CbGSA-5
f_8	En kötü	-2,246590e+03	-2,271637e+03	-2,794737e+03	-1,044421e+03	-3,495461e+03
	Ortalama	-2,864673e+03	-2,925225e+03	-3,660012e+03	-1,282017e+03	-4,436686e+03
	En İyi	-3,742098e+03	-4,065358e+03	-5,221234e+03	-1,803722e+03	-5,556769e+03
	St. Sapma	3,486994e+02	4,468600e+02	5,862463e+02	1,601838e+02	4,823358e+02
	Süre (s)	4,99471	4,9904	4,85818	4,94747	4,98457
f_9	En kötü	2,066964e+02	1,955765e+02	2,024743e+02	2,044469e+02	2,056468e+02
	Ortalama	1,439041e+02	1,412595e+02	1,656351e+02	1,559720e+02	1,616797e+02
	En İyi	1,494993e+01	1,492470e+01	5,875823e+01	1,791010e+01	2,145592e+01
	St. Sapma	5,796978e+01	6,342737e+01	2,792505e+01	4,987319e+01	4,609940e+01
	Süre (s)	5,04903	4,82213	5,29665	5,12035	5,01842
f_{10}	En kötü	2,220113e+00	1,899744e+00	1,646224e+00	2,018376e+01	1,899744e+00
	Ortalama	3,408786e-01	2,634716e-01	3,317829e-01	1,405533e+01	2,856075e-01
	En İyi	3,063004e-08	2,936748e-08	4,190281e-08	2,687728e-08	3,296918e-08
	St. Sapma	6,492835e-01	5,508271e-01	5,231255e-01	9,052879e+00	5,384041e-01
	Süre (s)	5,02363	5,01851	5,03054	4,72566	5,03869
f_{11}	En kötü	3,036992e+00	3,680517e+01	1,406973e+01	5,024235e+00	9,288011e+00
	Ortalama	1,194779e+00	1,139132e+01	4,757490e+00	3,928531e+00	5,152052e+00
	En İyi	0,000000e+00	3,594005e+00	1,372521e+00	2,832146e+00	1,233191e+00
	St. Sapma	6,823204e-01	6,923728e+00	2,613485e+00	5,777034e-01	2,041643e+00
	Süre (s)	5,99594	6,15017	6,06579	6,03872	6,09298
f_{12}	En kötü	1,364771e+00	8,053929e+00	8,310685e+00	1,220888e+00	5,504424e+00
	Ortalama	1,696554e-01	3,672167e-01	9,150427e-01	7,656638e-02	7,409670e-01
	En İyi	2,017161e-18	2,101359e-19	2,656574e-18	5,092642e-19	2,578093e-18
	St. Sapma	3,428089e-01	1,446381e+00	2,044815e+00	2,283642e-01	1,548794e+00
	Süre (s)	5,65237	5,61829	5,55018	5,53492	5,64545
f_{13}	En kötü	1,029545e+01	1,532520e-01	1,544735e+01	2,634513e+00	3,341909e+01
	Ortalama	3,535006e-01	8,962101e-03	1,761536e+00	9,326125e-02	3,416213e+00
	En İyi	2,052928e-17	3,062214e-17	3,319051e-17	1,764978e-17	1,193406e-17
	St. Sapma	1,846417e+00	2,840209e-02	3,977167e+00	4,720422e-01	8,388308e+00
	Süre (s)	5,39299	5,23693	5,51071	5,10457	5,64419



(a)



Şekil 16. a) f_8 için yakınsama eğrileri, b) f_8 kutu grafikleri (30-D)

CbGSA-X (X=1, 2, 3, 4, 5) kullanılarak f_{14-23} sabit boyutlu multimodal fonksiyonlar için elde edilen değerleri ise Tablo 12’de verilmiştir.

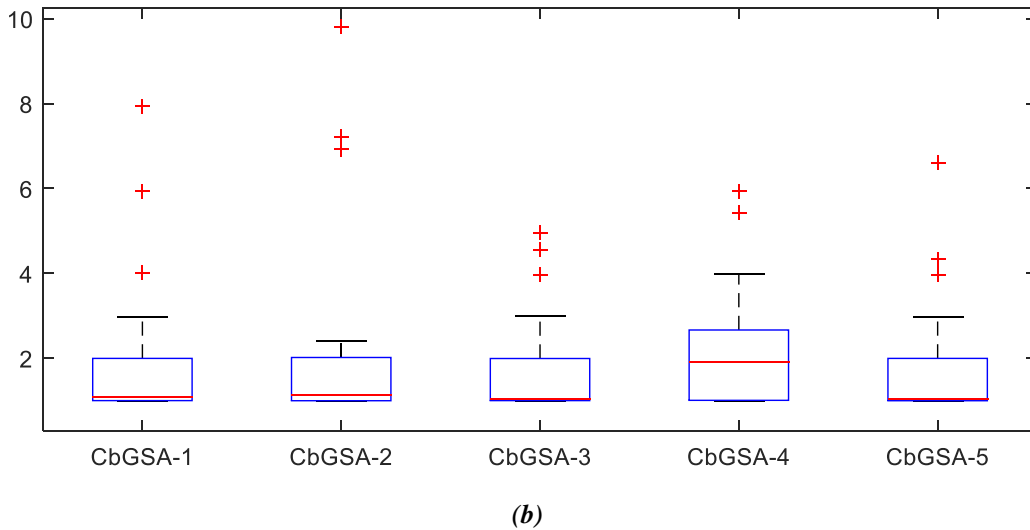
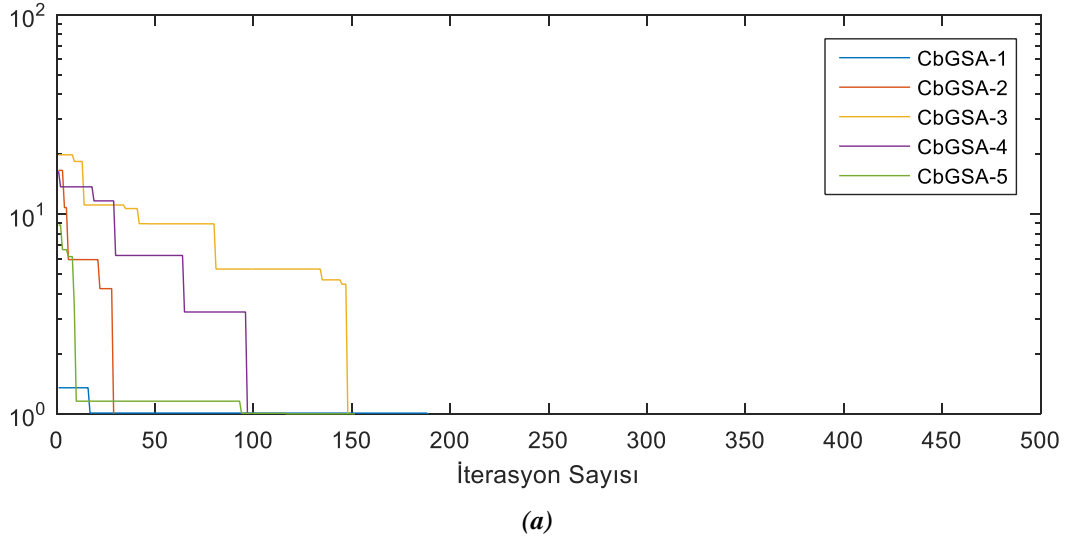
Tablo 12. Sabit boyutlu fonksiyonlar için elde edilen veriler (Tablo 8 - 30 çalışma - 500 iterasyon)

		CbGSA-1	CbGSA-2	CbGSA-3	CbGSA-4	CbGSA-5
f_{14}	En kötü	7,932507e+00	9,803931e+00	4,959948e+00	5,928883e+00	6,597868e+00
	Ortalama	1,822495e+00	2,049525e+00	1,713823e+00	2,193733e+00	1,743580e+00
	En İyi	9,980038e-01	9,980038e-0	9,980051e-01	9,980042e-01	9,980048e-01
	St. Sapma	1,550294e+00	2,075367e+00	1,237338e+00	1,429304e+00	1,265061e+00
	Süre (s)	2,24155	2,2321	2,23695	2,23259	2,22178
f_{15}	En kötü	2,070887e-01	2,061863e-01	2,066049e-01	9,295968e-01	2,070909e-01
	Ortalama	1,876181e-01	2,010658e-01	1,962543e-01	2,281077e-01	1,975176e-01
	En İyi	2,989233e-02	1,905240e-01	7,505161e-02	1,917118e-01	1,518690e-01
	St. Sapma	4,282906e-02	4,777818e-03	2,380650e-02	1,302906e-01	1,036453e-02
	Süre (s)	2,01466	2,00399	2,1276	2,04043	2,3858
f_{16}	En kötü	-1,028851e+00	-1,031555e+00	-1,031486e+00	-1,031617e+00	-1,031425e+00
	Ortalama	-1,031452e+00	-1,031626e+00	-1,031621e+00	-1,031628e+00	-1,031621e+00
	En İyi	-1,031628e+00	-1,031628e+00	-1,031628e+00	-1,031628e+00	-1,031628e+00
	St. Sapma	5,794842e-04	1,318461e-05	2,719249e-05	2,112437e-06	3,659054e-05
	Süre (s)	2,29617	1,99134	2,07227	2,097	2,06956
f_{17}	En kötü	3,989258e-01	4,205583e-01	4,557857e-01	3,978874e-01	3,978874e-01
	Ortalama	3,979263e-01	3,986431e-01	3,998173e-01	3,978874e-01	3,978874e-01
	En İyi	3,978874e-01	3,978874e-01	3,978874e-01	3,978874e-01	3,978874e-01
	St. Sapma	1,863614e-04	4,069559e-03	1,039307e-02	0,000000e+00	9,872780e-15
	Süre (s)	2,01428	2,05743	2,01727	2,02422	2,11437
f_{18}	En kötü	3,000000e+00	3,000000e+00	3,000000e+00	3,229076e+01	3,000000e+00
	Ortalama	3,000000e+00	3,000000e+00	3,000000e+00	4,190410e+00	3,000000e+00
	En İyi	3,000000e+00	3,000000e+00	3,000000e+00	3,000000e+00	3,000000e+00
	St. Sapma	4,787025e-08	5,874235e-09	9,688063e-10	5,285372e+00	5,724273e-09
	Süre (s)	2,5277	2,46902	2,41526	2,48502	2,56533
f_{19}	En kötü	-3,862782e+00	-3,855495e+00	-3,860223e+00	-3,862782e+00	-3,860617e+00
	Ortalama	-3,862782e+00	-3,862461e+00	-3,862616e+00	-3,862782e+00	-3,862640e+00
	En İyi	-3,862782e+00	-3,862782e+00	-3,862782e+00	-3,862782e+00	-3,862782e+00
	St. Sapma	6,351074e-13	1,320087e-03	5,064566e-04	2,544906e-12	4,478730e-04
	Süre (s)	2,28449	2,26694	2,44393	2,26939	2,33728
f_{20}	En kötü	-3,193713e+00	-3,159257e+00	-3,192574e+00	-3,087380e+00	-2,948575e+00
	Ortalama	-3,257361e+00	-3,247689e+00	-3,245574e+00	-3,184927e+00	-3,253387e+00
	En İyi	-3,321996e+00	-3,321996e+00	-3,321996e+00	-3,203102e+00	-3,321996e+00
	St. Sapma	6,049256e-02	6,122544e-02	5,819161e-02	3,161347e-02	8,186710e-02
	Süre (s)	2,74336	2,56604	2,35184	2,35178	2,22699

Tablo 12. (devam) Sabit boyutlu fonksiyonlar için elde edilen veriler (Tablo 8 - 30 çalışma - 500 iterasyon)

f_{21}	En kötü	2,659902e-15	8,718485e-15	3,102217e-15	2,339266e-13	1,890799e-15
	Ortalama	5,068916e-16	8,503192e-16	5,950063e-16	8,345003e-15	4,832201e-16
	En İyi	4,401041e-17	1,091713e-16	1,364454e-16	3,892299e-17	7,274784e-17
	St. Sapma	5,288955e-16	1,514634e-15	5,591161e-16	4,189458e-14	4,087875e-16
	Süre (s)	2,05177	2,70398	2,33752	2,25085	2,81777
f_{22}	En kötü	-5,087672e+00	-3,778269e+00	-5,087672e+00	-3,724300e+00	-3,724300e+00
	Ortalama	-9,694238e+00	-9,434991e+00	-7,922482e+00	-7,345509e+00	-8,408563e+00
	En İyi	-1,040294e+01	-1,040294e+01	-1,040294e+01	-1,040294e+01	-1,040294e+01
	St. Sapma	1,806844e+00	2,179118e+00	2,651722e+00	2,684529e+00	2,631848e+00
	Süre (s)	2,42245 sn	2,57602	2,54128	2,63266	2,34781
f_{23}	En kötü	-2,427335e+00	-2,427335e+00	-2,553415e+00	-3,835427e+00	-2,427335e+00
	Ortalama	-9,663447e+00	-9,852689e+00	-1,027031e+01	-1,031304e+01	-9,787232e+00
	En İyi	-1,053641e+01	-1,053641e+01	-1,053641e+01	-1,053641e+01	-1,053641e+01
	St. Sapma	2,278164e+00	2,105002e+00	1,432991e+00	1,202863e+00	2,255206e+00
	Süre (s)	2,41452	2,76267	2,4236	2,47475	2,54648

Tablo 12’de yer alan sabit boyutlu f_{14} fonksiyonu için 30 çalışma ile elde edilen en iyi çözümlere ait, iterasyon sayısına göre yakınsamayı gösteren grafik Şekil 17.a’da, aynı fonksiyona ait 30 çalışma için en iyi değerlerin yayılımını gösteren kutu grafiği ise Şekil 17.b’de verilmiştir.



Şekil 17. (a) f_{14} için yakınsama eğrileri, (b) f_{14} kutu grafikleri (2-D)

CbGSA-X (X=1, 2, 3, 4, 5) kullanılarak çalışmada bütün fonksiyonlar için elde edilen sonuçlar ile GSA'nın kaynak [1]'de yer alan 30-D için verilen en iyi sonuçlarla karşılaştırılması Tablo 13'te verilmiştir. Literatürden alınan sayısal değerler, karşılaştırmaların daha kolay anlaşılabilmesi için çalışmadaki formatla aynı şekilde yazılmıştır. Çalışmada geliştirilen CbGSA-X (X=1, 2, 3, 4, 5) algoritmaları için kendi aralarında elde edilen en iyi sonuçlar tabloda koyu renkle verilmiştir.

Tablo 13. 30-D için CbGSA-X ile GSA'nın literatür karşılaştırması

	GSA [1]	CbGSA-1	CbGSA-2	CbGSA-3	CbGSA-4	CbGSA-5
f_1	7,300000e-11	4,401041e-17	1,091713e-16	1,364454e-16	3,892299e-17	7,274784e-17
f_2	4,030000e-05	3,749932e-08	2,185942e-08	3,031872e-08	2,376645e-08	3,004552e-08
f_3	0,160000e+03	1,513509e-05	6,530554e-06	2,236669e-02	2,535218e-05	1,284255e-05
f_4	3,700000e-06	1,920559e-07	4,510397e-08	1,096698e-07	2,351702e-07	1,073174e-07
f_5	2,516000e+01	2,141220e+01	2,266581e+01	2,355737e+01	2,245741e+01	2,354076e+01
f_6	8,300000e-11	5,875281e-17	6,902870e-17	1,728119e-17	6,100148e-17	5,184519e-17
f_7	0,018000e+00	2,622054e-03	5,125597e-03	6,492990e-03	4,362321e-03	5,481061e-03
f_8	-2,800000e+03	-3,742098e+03	-4,065358e+03	-5,221234e+03	-1,803722e+03	-5,556769e+03
f_9	1,532000e+01	1,494993e+01	1,492470e+01	5,875823e+01	1,791010e+01	2,145592e+01
f_{10}	6,900000e-06	3,063004e-08	2,936748e-08	4,190281e-08	2,687728e-08	3,296918e-08
f_{11}	0,290000e+00	0,000000e+00	3,594005e+00	1,372521e+00	2,832146e+00	1,233191e+00
f_{12}	0,010000e+00	2,017161e-18	2,101359e-19	2,656574e-18	5,092642e-19	2,578093e-18
f_{13}	3,200000e-32	2,052928e-17	3,062214e-17	3,319051e-17	1,764978e-17	1,193406e-17
f_{14}	3,700000e+00	9,980038e-01	9,980038e-01	9,980051e-01	9,980042e-01	9,980048e-01
f_{15}	8,000000e-03	2,989233e-02	1,905240e-01	7,505161e-02	1,917118e-01	1,518690e-01
f_{16}	-1,031600e+00	-1,031628e+00	-1,031628e+00	-1,031628e+00	-1,031628e+00	-1,031628e+00
f_{17}	0,397900e+00	3,978874e-01	3,978874e-01	3,978874e-01	3,978874e-01	3,978874e-01
f_{18}	3,000000e+00	3,000000e+00	3,000000e+00	3,000000e+00	3,000000e+00	3,000000e+00
f_{19}	-3,735700e+00	-3,862782e+00	-3,862782e+00	-3,862782e+00	-3,862782e+00	-3,862782e+00
f_{20}	-2,056900e+00	-3,321996e+00	-3,321996e+00	-3,321996e+00	-3,203102e+00	-3,321996e+00
f_{21}	-6,074800e+00	4,401041e-17	1,091713e-16	1,364454e-16	3,892299e-17	7,274784e-17
f_{22}	-9,339900e+00	-1,040294e+01	-1,040294e+01	-1,040294e+01	-1,040294e+01	-1,040294e+01
f_{23}	-9,454800e+00	-1,053641e+01	-1,053641e+01	-1,053641e+01	-1,053641e+01	-1,053641e+01

Tablo 13 incelendiğinde çalışmada geliştirilen tüm algoritmaların klasik GSA'ya göre daha iyi sonuçlar elde ettiği görülmüştür. Geliştirilen beş farklı kaotik tabanlı algoritma olan CbGSA-X (X=1, 2, 3, 4, 5) kendi aralarında değerlendirildiğinde ise CbGSA-1 yaklaşımı diğerlerine göre daha fazla sayıda fonksiyon için en iyi değeri yakalamıştır.

CbGSA-X (X=1, 2, 3, 4, 5) algoritmaları 23 adet test fonksiyonu için 30 kez çalıştırılmıştır. Elde edilen en iyi değerlerin 30 çalışma için kararlılıklarını gösteren örnek grafikler, f_1 , f_{12} ve f_{16} fonksiyonları için sırasıyla Şekil 18.a-c'de gösterilmiştir.

Literatürde bu tür çalışmaları istatistiki olarak karşılaştırabilmek için Wilcoxon testi uygulanmaktadır [39]. Veri sayısının az olması durumunda parametrik olmayan Wilcoxon testi uygulanarak daha hassas ve doğru sonuç elde edilmektedir. Bu nedenle bu çalışmada Tablo 13’te verilen 23 adet test fonksiyonuna ait GSA, CbGSA-X ($X=1, 2, 3, 4, 5$) algoritmalarına ait 30 çalışmanın en iyi değerleri, Kaynakça [39]’da tanımı yapılan Wilcoxon istatistiki değerlendirme testlerine tabi tutulmuş ve elde edilen sonuçlar Tablo 14’te verilmiştir. Verilerin analizinde anlamlılık düzeyi 0,05 olarak alınmıştır.

Tablo 14. f_1-f_{23} fonksiyonlarına ait 30 çalışma ve 30-D için elde edilen en iyi değerlerin Wilcoxon testleri

	Wilcoxon (p-değeri)
	f_1-f_{23}
GSA - CbGSA-1	0,0022
GSA - CbGSA-2	0,0666
GSA - CbGSA-3	0,0495
GSA - CbGSA-4	0,2113
GSA - CbGSA-5	0,0459

Şekil 18’de görüldüğü gibi CbGSA-1 algoritması, aynı parametrelerle 30 çalışma değeri için diğer algoritmalara göre daha fazla sayıda minimum sonuca yakınsamıştır. Tablo 13’te verilen değerler baz alındığında ise CbGSA-1 algoritmasının diğer kaotik tabanlı algoritmalara göre daha fazla sayıda en iyi değeri yakaladığı görülmektedir. Tablo 14’te verilen p değerlerine göre ise, çalışmada geliştirilen CbGSA-X algoritmalarının, GSA algoritmasından anlamlı farklılıkları oldukları görülmektedir. En fazla anlamlı farklılık CbGSA-1 metodu lehine gerçekleşmiştir ($p=0.0022$).

IV. SONUÇ

Çalışmada GSA algoritmasına beş farklı kaotik haritalama metoduyla kaotik yapılar entegre edilerek CbGSA-X ($X=1, 2, 3, 4, 5$) algoritmaları geliştirilmiştir. Her beş yaklaşımla da 23 adet test fonksiyonu çözülmüştür. CbGSA-X ($X=1, 2, 3, 4, 5$) algoritmaları ile bulunan sonuçlar, tablolardan görüldüğü üzere GSA’dan daha iyi sonuçlar vermiştir. Yapılan çalışmalar sonucunda geliştirilen yeni algoritmalar olan CbGSA-X, çalışmanın amacına uygun olarak test fonksiyonlarında GSA algoritmasına göre daha iyi değerler elde etmişlerdir. Kendi aralarında yapılan karşılaştırmalar sonucunda ise CbGSA-1 algoritmasının klasik GSA ve diğer algoritmalara göre daha kararlı ve daha iyi başlangıç koşullarını sağlayarak aramayı daha iyi bölgelere sevk etme konusunda daha başarılı olduğu sonucuna varılmıştır. Çalışmada iki farklı amaç öne çıkmaktadır, bunlardan birincisi GSA’nın iyileştirilmesi, diğeri ise 5 farklı ($X=1, 2, 3, 4, 5$) kaotik haritalama metodundan daha iyi olanlarının belirlenmesidir. Araştırmacılar tarafından daha sonra yapılacak olan çalışmalarda, bu çalışmada önerilen yaklaşımlar diğer sezgisel algoritmalara da entegre edilerek performansları gözlemlenebilir.

V. KAYNAKLAR

- [1] E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, “GSA: A gravitational search algorithm,” *Information Sciences*, c. 179, s. 13, ss. 2232-2248, 2009.
- [2] U. Güvenç, F. Katırcıoğlu, “En İyi Ajana Özel Davranış: Geliştirilmiş Yerçekimi Arama Algoritması,” *El-Cezeri Journal of Science and Engineering*, c. 3, s. 1, ss. 143-153, 2016.
- [3] A. Kaveh, VR. Mahdavi, “Colliding bodies optimization: A novel meta-heuristic method,” *Computers and Structures*, c. 139, ss. 18-27, 2014.
- [4] OK. Erol, I. Eksin, “A new optimization method: Big Bang–Big Crunch,” *Advances in Engineering Software*, c. 37, s. 2, ss. 106-111, 2006.

- [5] A. Hatamlou, "Black hole: A new heuristic optimization approach for data clustering," *Information Sciences*, c. 222, ss. 175-184, 2013.
- [6] S. Kirkpatrick, CD. Gelatt, MP. Vecchi, "Optimisation by simulated annealing," *Science*, c. 220, ss. 671-680, 1983.
- [7] DE. Goldberg, "Genetic Algorithms in Search, Optimization, and Machine Learning," *Addison-Wesley Publishing Company, Inc.*, 1989.
- [8] R. Storn, K. Price, "Differential evolution - A simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, c. 11, ss. 341-359, 1997.
- [9] Aİ. Çanakoğlu, AG. Yetgin, H. Temurtaş, M. Turan, "Induction motor parameter estimation using metaheuristic methods," *Turkish Journal of Electrical Engineering & Computer Sciences*, c. 22, ss. 1177-1192, 2014.
- [10] J. Kennedy, R. Eberhart, "Particle Swarm Optimization," *Proceedings of IEEE International Conference on Neural Networks*, c. 4, ss. 1942-1948, 1995.
- [11] ZW. Geem, JH. Kim, GV. Loganathan, "A new heuristic optimization algorithm: Harmony search," *Simulation*, c. 76, s. 2, ss. 60-68, 2001.
- [12] D. Karaboğa, B. Baştürk, "A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, c. 39, s. 3, ss. 459-471, 2007.
- [13] A. Kaveh, S. Talahatari, "A novel heuristic optimization method: Charged system search," *Acta Mechanica*, c. 213, s. 3-4, ss. 267-289, 2010.
- [14] YJ. Zheng, "Water wave optimization: A new nature-inspired metaheuristic," *Computers and Operations Research*, c. 55, ss. 1-11, 2015.
- [15] S. Mirjalili, A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, c. 95, ss. 51-67, 2016.
- [16] R. Rajabioun, "Cuckoo optimization algorithm," *Applied Soft Computing*, c. 11, ss. 5508-5518, 2011.
- [17] A. Askarzadeh, "A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm," *Computers and Structures*, c. 169, ss. 1-12, 2016.
- [18] AH. Kashan, "A new metaheuristic for optimization: Optics inspired optimization (OIO)," *Computers and Operations Research*, c. 55, ss. 99-125, 2015.
- [19] S. Mirjalili, SM. Mirjalili, A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, c. 69, ss. 46-61, 2014.
- [20] B. Doğan, T. Ölmez, "A new metaheuristic for numerical function optimization: Vortex search algorithm," *Information Sciences*, c. 293, ss. 125-145, 2015.
- [21] S. Saremi, S. Mirjalili, A. Lewis, "Biogeography-based optimisation with chaos," *Neural Computing and Applications*, c. 25, ss. 1077-1097, 2014.

- [22] B. Nouhi, S. Talatahari, H. Kheiri, C. Cattani, "Chaotic charged system search with a feasible-based method for constraint optimization problems," *Mathematical Problems in Engineering*, c. 2013, ss. 1-8, 2013.
- [23] M. Mitic, N. Vukovic, M. Petrovic, Z. Miljkovic, "Chaotic fruit fly optimization algorithm," *Knowledge-Based Systems*, c. 89, ss. 446-458, 2015.
- [24] AH. Gandomi, XS. Yang, "Chaotic bat algorithm," *Journal of Computational Science*, c. 5, ss. 224-232, 2014.
- [25] GG. Wang, L. Guo, AH. Gandomi, GS. Hao, H. Wang, "Chaotic krill herd algorithm," *Information Sciences*, c. 274, ss. 17-34, 2014.
- [26] B. Alataş, "Chaotic harmony search algorithms," *Applied Mathematics and Computation*, c. 216, ss. 2687-2699, 2016.
- [27] B. Alataş, E. Akın, O. Bedri, "Chaos embedded particle swarm optimization algorithms," *Chaos, Solitons and Fractals*, c. 40, s. 4, ss. 1715-1734, 2009.
- [28] B. Alataş, "Chaotic bee colony algorithms for global numerical optimization," *Expert Systems with Applications*, c. 37, s. 8, ss. 5682-5687, 2010.
- [29] H. Peitgen, H. Jurgens, D. Saupe, "Chaos and Fractals: New frontiers of science." *Springer-Verlag*, Berlin, 1992.
- [30] A. Törn, A. Zilinskas, "Global Optimization." *Springer-Verlag*, Berlin, 1989.
- [31] X. Yao, Y. Liu, G. Lin, "Evolutionary programming made faster," *IEEE Transactions on Evolutionary Computation*, c. 3, s. 2, ss. 82-102, 1999.
- [32] VI. Arnold, A. Avez, "Ergodic problems in classical mechanics," *Benjamin*, New York, 1968.
- [33] F. Ge, L. Tan, Y. Wang, "Traffic modeling with Bernoulli Shift Map," 2009 Fifth International Joint Conference on INC, IMS and IDC, Seoul, ss. 449-452, 2009.
- [34] B. Stoyanov, "Pseudo-random bit generation algorithm based on Chebyshev Polynomial and Tinkerbell Map," *Applied Mathematical Sciences*, c. 8, s. 125, ss. 6205-6210, 2014.
- [35] B. Stoyanov, K. Kordov, "Novel image encryption scheme based on Chebyshev Polynomial and Duffing Map," *The Scientific World Journal*, c. 2014, ss. 1-11, 2014.
- [36] WM. Zheng, "Kneading plane of the circle map," *Chaos, Solitons and Fractals*, c. 4, s. 7, ss. 1221-1233, 1994.
- [37] R. Caponetto, L. Fortuna, S. Fazzino, MG. Xibilia, "Chaotic sequences to improve the performance of evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, c. 7, s. 3, ss. 289-304, 2003.
- [38] H. Lu, X. Wang, Z. Fei, M. Qiu, "The effects of using chaotic map on improving the performance of multiobjective evolutionary algorithms," *Mathematical Problems in Engineering*, c. 2014, ss. 1-16, 2014.
- [39] S. García, D. Molina, M. Lozano, F. Herrera, "A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 Special Session on Real Parameter Optimization," *Journal of Heuristics*, c. 15, ss. 617-644, 2009.