

## Kaotik Haritalı Balina Optimizasyon Algoritmaları

Erkan TANYILDIZI<sup>1\*</sup>, Tuncay CİĞAL<sup>2</sup>

<sup>1</sup>Fırat Üniversitesi, Yazılım Mühendisliği Bölümü, 23100, Elazığ, TÜRKİYE  
\*etanyildizi@gmail.com

<sup>2</sup>Besni İlçe Milli Eğitim Müdürlüğü, 02300, Adıyaman, TÜRKİYE

(Geliş/Received: 22.02.2017; Kabul/Accepted:02.03.2017)

### Özet

Bu çalışmada, balinaların davranışlarından esinlenerek geliştirilmiş optimizasyon yöntemi olan balina optimizasyon algoritması (BOA) detaylı olarak açıklanmış ve kaotik global yakınsama özelliğinin artırılması amaçlanmıştır. BOA’da kullanılan rastgele sayı dizilerinin yerine kaotik sistemler kullanılarak oluşturulan, “kaotik haritalı BOA” adı altında beş farklı algoritma önerilmiştir. Kaotik haritalı algoritmaların ve temel algoritmanın performansı, kalite testi fonksiyonları ve bir adet kısıtlı gerçek mühendislik problemi üzerinde test edilmiştir. Test sonuçları karşılaştırmalı tablolar ve grafikler aracılığıyla sunulmuş ve yorumlanmıştır.

**Anahtar kelimeler:** Optimizasyon, Metasezgisel algoritmalar, Balina optimizasyon algoritması, Kaotik haritalar

## Whale Optimization Algorithms With Chaotic Mapping

### Abstract

In this study, whale optimization algorithm (WOA) which imitates the behaviors of whales has been explained in detail and it is aimed to increase the global convergence feature with chaos. We have suggested five different WOA by using chaotic maps instead of random number sequences used in the algorithm called “WOA with chaotic maps”. In addition, we applied base algorithm and suggested algorithms to test functions and a constained real engineering problem to analyze their performance. The results of the analysis are presented and interpreted through comparative tables and graphs.

**Keywords:** Optimization, Metaheuristic algorithms, Whale optimization algorithm, Chaotic maps

### 1. Giriş

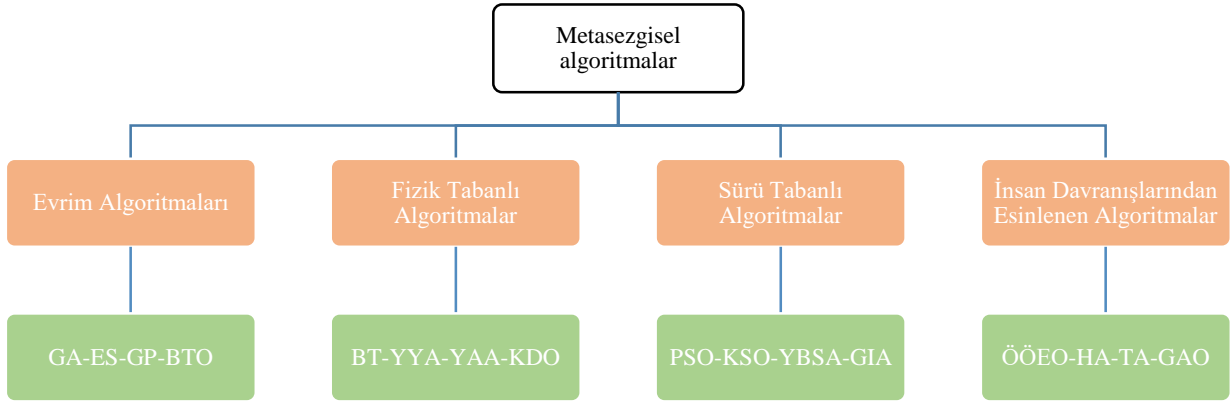
Metasezgisel optimizasyon algoritmaları, dönüşümleri kolay olduğundan, türev gerektirmediğinden, yerel minimumları aşabildiğinden ve farklı alanlardaki geniş problem yelpazesine cevap verebildiğinden dolayı mühendislik uygulamalarında sıkça rastlanmaya başlanmıştır.

Metasezgisel algoritmalar olayları, fiziksel veya biyolojik davranışları taklit ederek problemleri çözerler. Bu algoritmalara genel olarak doğa tabanlı algoritmalar diyebiliriz. Şekil 1.1’de görüldüğü gibi evrim tabanlı, fizik tabanlı ve sürü tabanlı olmak üzere 3 ana kategoriye ayrılabilir [1].

Evrim tabanlı metotlar doğal gelişim kurallarını taklit ederler. Arama işlemi, sonraki iterasyonlar boyunca geliştirilecek olan rastgele

bir popülasyon oluşturarak başlar. Bu metotları güçlü kılan özelliği, sonraki bireyleri oluşturacak olan bireylerin en iyilerden seçilmesidir. Bu da sonucun iterasyonlar boyunca optimuma yakınsamasını sağlar. En popüler evrim algoritması olan Genetik Algoritma (GA) [2] Darwin’in evrim teorisini taklit eder. Diğer popüler evrim algoritmalarına örnek olarak, Evrim Stratejisi (ES) [3], Olasılık Tabanlı Artımsal Öğrenme (OTAÖ) [4], Genetik Programlama (GP) [5] ve Biyoloji Tabanlı Optimizasyon (BTO) [6] verilebilir.

Fizik tabanlı metotlar doğanın fizik kurallarını taklit ederler. Benzetimli Tavlama (BT) [7,8], Yerçekimsel Yerel Arama (YYA) [9], Büyük Patlama Büyük Çöküş Algoritması (BPBÇ) [10], Yerçekimsel Arama Algoritması (YAA) [11], Kara Delik Optimizasyonu (KDO)



Şekil 1.1 Metasezgisel algoritmaların sınıflandırılması

[12] ve Işın Algoritması (IA) [13] fizik tabanlı algoritmalara örnek olarak verilebilir. Doğa tabanlı metotlardan üçüncü kategori canlı gruplarının davranışlarını taklit eden sürü tabanlı algoritmalarıdır. En popüler olan Parçacık Sürü Optimizasyonu (PSO) [14] örnek olarak verilebilir. PSO kuş sürülerinin davranışlarından ilham alır ve en iyi çözümü bulmak için arama uzayında hareket eden parçacıkları kullanır. Yine popüler olan Karınca Koloni Algoritması (KKO) [15] sürü tabanlı algoritmalara örnek olarak verilebilir. Bu algoritma kolonideki karınca davranışlarını taklit eder. Karıncaların yuvaya ve yiyecek kaynağına en yakın yolu bulma yetenekleri bu algoritmaya ilham kaynağı olmuştur.

Genel olarak bakarsak, sürü tabanlı algoritmaların evrim tabanlı algoritmalara göre bazı avantajları vardır. Örneğin evrim tabanlı algoritmalar sadece bulunan iterasyondaki arama uzayı bilgilerini saklarken, sürü tabanlı algoritmalar iterasyonlar boyunca arama uzayı bilgilerini saklarlar. Genellikle sürü tabanlı algoritmaların, evrim tabanlı algoritmalara göre daha az operatörü vardır ve farklı problemlere uygulamak daha kolaydır.

Literatürde ayrıca insanlardan ilham alan algoritmalar da mevcuttur. Bunlara Öğretme-Öğrenme Esaslı Optimizasyon (ÖÖEO) [16,17], Harmoni Arama (HA) [18], Tabu Arama (TA) [19,20] ve Grup Arama Optimizasyonu (GAO) [21,22] örnek olarak verilebilir.

Popülasyon tabanlı algoritmalar, ilham kaynakları ne olursa olsun bazı ortak özelliklere

sahiptir [23,24]. Popülasyon tabanlı bir algortmada genel olarak,

- Arama işlemi yerel ve global arama olmak üzere iki kısımdan oluşur.
- Optimizasyon algoritmasının global arama için operatörleri olmalıdır. Bu kısımda bireylerin hareketleri mümkün olduğunca rastgele olmalıdır.
- Global aramadan sonra yerel arama yapılır ve bu arama dar bölgede detaylı arama olarak da ifade edilebilir. Global aramada muhtemel optimum nokta bulunduktan sonra yerel arama ile devam edilerek küçük adımlarla arama devam eder.

Metasezgisel optimizasyon algoritmalarının stokastik yapısından dolayı algoritma çalışma sürecinde, global ve yerel aramalardan hangisinin yapılacağına karar vermek te önemli bir işlemdir [25].

Bu çalışmada bahsedilen BOA, kambur balinaların avlanma davranışlarını taklit etmiştir. BOA'nın ve kaotik harita ile oluşturulan 5 farklı versiyonunun performansı, seçilen 5 test fonksiyonu ve 1 adet gerçek mühendislik problemi üzerinde test edilerek sonuçları incelenmiştir.

Bu makalenin ikinci kısmında BOA'nın yapısı anlatılmış, üçüncü kısmında BOA'nın kaotik versiyonları gösterilmiş, dördüncü kısmında algoritmalar test fonksiyonları ve gerçek probleme uygulanarak sonuçları ele alınmış ve beşinci kısmında ise sonuç ve yorumlara yer verilmiştir.

## 2. Balina Optimizasyon Algoritması

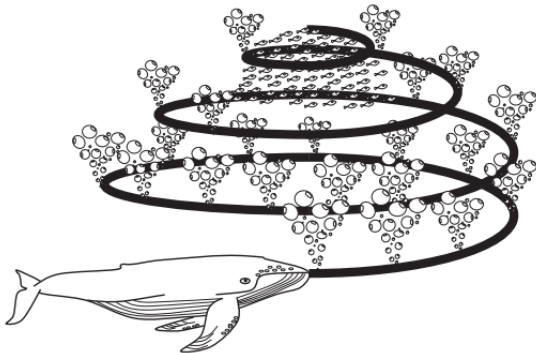
Bu kısımda BOA'nın ilham kaynağından bahsedilerek matematiksel modeli gösterilmiştir.

### 2.1 Algoritmanın ilham kaynağı

Balinalar dünyanın en büyük memelileri olarak bilinirler. Yetişkin bir balina 30 metre uzunluğuna ve 180 ton ağırlığına erişebilir. Katil, Minke, Kuzey, Kambur, Buzul, Oluklu ve mavi balina olmak üzere 7 türü vardır. Balinalar okyanus yüzeylerinden nefes almak zorunda oldukları için hiçbir zaman uyumazlar. Sadece beyinlerinin yarısı uyur. Balinalar ile ilgili diğer bir gerçek ise duygularının olduğudur.

Hof ve Van Der Gucht'a göre [26], balinalar ile insanların beyinlerinin belli bölgelerinde benzer hücreler bulunmaktadır. Bu hücreler insanların sosyal davranışlarındaki yargılama ve duyguları ile ilgilidir ve insanları diğer canlılardan ayırır. Balinalarda bu hücrelerden insanlara oranla iki kat daha fazladır. Balinaların düşünebildiği, öğrenebildiği, iletişim kurabildiği ve hatta duygularının olduğu kanıtlanmıştır. Ama insanlara göre çok daha az zekidirler. Balinalar genellikle grup olarak bazen de yalnız hareket ederler. Bazı balinalar bütün hayatları boyunca aileleri ile yaşarlar.

Bu çalışmada esas alınan tür kambur balınadır. Kambur balinalar ortalama bir otobüs büyüklüğündedir ve küçük balık sürülerini avlarlar. Şekil 2.1 [1]'de kambur balinaların avlanması gösterilmiştir.



Şekil 2.1 Kambur balinaların su kabarcığı yöntemi ile avlanması

Kambur balinaların hava kabarcığı davranışı denilen kendilerine özgü bir beslenme davranışı vardır. Suyun altında soluk vererek hava

kabarcığı bulutları ve sütunları meydana getirirler. Bu geniş, birbirleri ile bağlantılı hava kabarcıkları kümesi avları bir araya toplar. Daha sonra, balina su kabarcıklarının içerisinde yüze doğru yükselir. Yükselmeye devam ederken soluk alıp vererek kabarcık oluşturmaya devam eder ve avına yaklaştıkça kabarcık çemberini daraltarak hedef küçültür. Bu davranışın, avın bulunmasında veya avın hareketsiz hale getirilmesi ve şaşırtılması yoluyla ele geçirilmesinde yardımcı olduğu görülmektedir. Ayrıca bu davranış, avcuyu avdan gizleyebilir [27].

### 2.2 Matematiksel model ve algoritma

Bu kısımda kambur balinanın avlanma stratejisi avın etrafını sarma, avı doğru hareket etme ve av arama olarak 3 ayrı kısımda modellenmiştir.

#### 2.2.1 Avın etrafını sarma

Kambur balinalar avlarının konumunu tahmin edebilirler. Bundan dolayı avlarının etrafını hava kabarcıkları ile sarmalayabilirler. BOA'da balinanın avı, ulaşılacak olan optimum nokta olarak kabul edilir. Optimizasyon problemlerinde optimum çözüm bilinmediğinden, optimum nokta ulaşılan en iyi çözüm ya da onun etrafında bir nokta olarak kabul edilir. En iyi arama ajanı belirlendikten sonra diğer arama ajanlarının konumları en iyi arama ajanı kullanılarak güncellenir. Avın etrafını sarma davranışının matematiksel modeli Denklem 2.1 ve Denklem 2.2'de gösterilmiştir.

$$\vec{D} = |\vec{C} \cdot \vec{X}^*(t) - \vec{X}(t)| \quad (2.1)$$

$$\vec{X}(t+1) = |\vec{X}^*(t) - \vec{A} \cdot \vec{D}| \quad (2.2)$$

Denklem 2.1 ve 2.2'de  $t$  bulunan iterasyonu,  $\vec{A}$  ve  $\vec{C}$  yakınsama vektörlerini,  $\vec{X}^*$  elde edilmiş en iyi çözüm vektörünü ve '.' ise matris çarpımını ifade etmektedir.

$\vec{A}$  ve  $\vec{C}$  Denklem 2.3 ve Denklem 2.4 'te gösterildiği gibi hesaplanmaktadır.

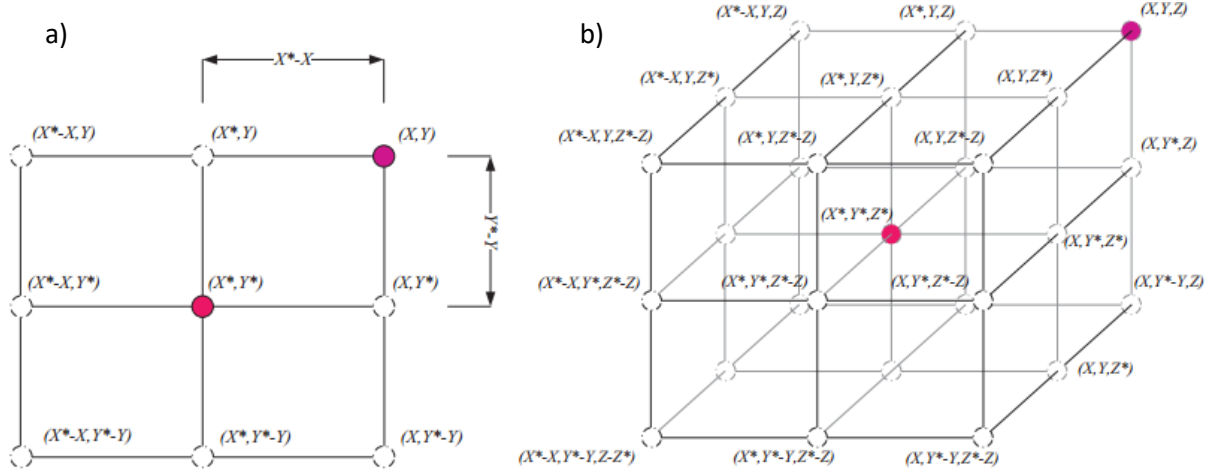
$$\vec{A} = 2\vec{a} \cdot \vec{r} - \vec{a} \quad (2.3)$$

$$\vec{C} = 2 \cdot \vec{r} \quad (2.4)$$

Denklem 2.3 ve 2.4'te  $r$  rastgele vektörü ve  $\vec{a}$  iterasyonlar boyunca 2'den sıfıra doğru lineer olarak azalan vektörü göstermektedir.

Şekil 2.2 (a), 2 boyutlu problem için Denklem 2.2 'nin gerekçesini göstermektedir. Herhangi bir arama ajanı  $(X,Y)$  pozisyonunu en

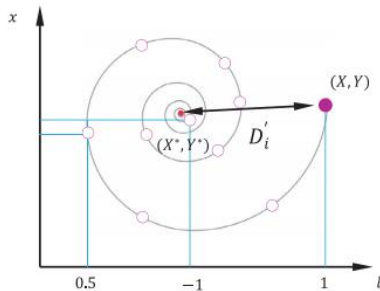
iyi arama ajanına  $(X^*,Y^*)$  göre belirleyebilir. En iyi ajanın etrafındaki noktalara ise  $\vec{A}$  ve  $\vec{C}$  sabitleri ayarlanarak ulaşılabilir. Şekil 2.1 (b)' de ise aynı şekilde 3 boyutlu arama uzayı gösterilmiştir.



Şekil 2.2 [1]. Arama ajanlarının 2 ve 3 boyutlu pozisyon vektörleri ve olası konumları

### 2.2.2 Ava doğru hareket etme

Bu davranış avın etrafındaki çemberi daraltma ve spiral hareket olmak üzere iki kısımda modellenmiştir. Avın etrafındaki çemberi daraltma, Denklem 2.3'teki  $\vec{a}$ 'nın değerini azaltmakla mümkün olabilmektedir. Şekil 2.3'te arama ajanının spiral hareketi ve en iyi ajanın konumu gösterilmiştir. Gösterilen spiral hareket için hedef konum (en iyi ajan) ve arama ajanı arasındaki mesafe hesaplanarak Denklem 2.5 oluşturulmuştur.



Şekil 2.3 [1]. Spiral hareket

$$\vec{X}(t+1) = \vec{D}^i \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}^*(t) \quad (2.5)$$

Denklem 2.5'te  $\vec{D}^i = \vec{X}^*(t) - \vec{X}(t)$ 'dir. Bu ifade arama ajanı ile bilinen en iyi nokta arasındaki mesafeyi vermektedir.  $b$  logaritmik

spiral sabiti,  $l$  ise  $[-1,1]$  arasında rastgele sayıdır. Kambur balinalar su yüzeyindeki ava doğru yükselirken eş zamanlı olarak spiral harekete devam ederler. Algoritmada spiral hareket ve doğrusal hareketten hangisinin yapılacağı Denklem 2.6'da gösterildiği gibi  $\frac{1}{2}$  olasılıkla belirlenir.

$$\vec{X}(t+1) = \begin{cases} \vec{X}^x(t) - \vec{A} \cdot \vec{D}, & p < 0,5 \\ \vec{D}^i \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}^*(t), & p \geq 0,5 \end{cases} \quad (2.6)$$

Denklem 2.6'da  $p$ ,  $[0,1]$  aralığında rastgele sayıdır.

### 2.2.3 Av arama

Global arama için arama ajanlarının yeni konumları, bilinen en iyi nokta yerine, rastgele seçilen bir arama ajanı etrafında belirlenir. Matematiksel modeli Denklem 2.7 ve Denklem 2.8'de gösterilmiştir.

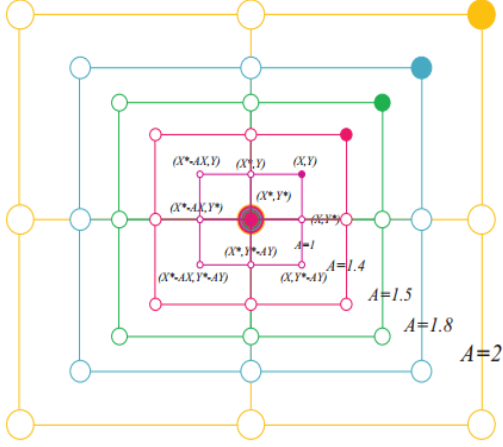
$$\vec{D}^i = \vec{C} \cdot \vec{X}_{rand} - \vec{X} \quad (2.7)$$

$$\vec{X}(t+1) = \vec{X}_{rand} - \vec{A} \cdot \vec{D} \quad (2.8)$$

Denklem 2.7 ve 2.8'de  $\vec{X}_{rand}$  rastgele seçilen bir arama ajanını göstermektedir.

Global veya yerel aramalardan hangisinin yapılacağına ise  $\vec{A}$ 'nın değerine göre karar

verilir. Şekil 2.4'te gösterildiği gibi,  $\vec{A} > 1$  veya  $\vec{A} < -1$  olduğunda en iyi noktadan daha uzak nokta seçilebileceğinden, bu durumlar global arama olarak kabul edilir ve Denklem 2.7 ve Denklem 2.8 uygulanır.



Şekil 2.4 [1]. BOA'nın global arama davranışı

Tablo 2.1'de BOA'nın sözde kodu gösterilmiştir.

### 3. Kaotik Haritalar

Sezgisel yöntemlerin çoğunda olduğu gibi bu algoritmada da uzun periyotlu rastgele sayı dizileri önemli bir yer tutmaktadır. Oluşturulan sayıların aynı olmaması, yayılmış spektruma sahip olması, sayılar oluşturulurken fazla zamana gereksinim duyulmaması, saklanma maliyetinin düşük olması istenen durumlardandır. Rastgele üretilen sayıların belli bir alanda toplanması veya aynı değerlerin yinelenmesi ile algoritmanın yerel optimumlara takılma riski artabilmektedir.

Kaotik harita kullanarak optimum noktalardan kaçmak ya da yerel optimuma takılma riskini azaltmak mümkün olabilmektedir. Kaotik sayı dizileri yapay sinir ağları [28], Kaotik Haritalı Hibrit Parçacık Sürü Optimizasyonu Algoritması [29], Kaotik Beyin Fırtınası Optimizasyonu [30], Kaotik Evrimsel Yaklaşım İle Çok Amaçlı Transformatör Tasarımı [31] gibi birçok uygulamada kullanılmıştır.

Kaotik sayı dizisinin oluşturulması ve depolanması kolay ve hızlıdır. Basit bir fonksiyon ve birkaç başlangıç parametresi ile istenilen uzunlukta dizileri elde edebilmek çok kolaydır [32]. Kaotik haritalarla üretilen sayıların

tahmin edilemez, ergodik, yayılmış spektrumlu karakteristiğe sahip olduğu ve periyodik olmadığı teorik olarak kanıtlanmıştır [33].

Bu çalışmada kaosu en iyi anlatan ve uygulaması basit olan kaotik haritalardan lojistik harita [34] BOA'ya eklenerek kaotik haritalı BOA (KHBOA) versiyonları oluşturulmuştur.

Tablo 2.1. BOA algoritmasının sözde kodu

```

Başlangıç popülasyonu ayarla  $X_i$  ( $i = 1, 2, \dots, n$ )
Her bir arama ajanının uygunluk değerini hesapla
 $X^*$ =Bilinen en iyi arama ajanı
while ( $t <$  maksimum iterasyon sayısı)

for (her bir arama ajanı için)
Güncelle  $a, A, C, l$ , ve  $p$ 

if ( $p < 0.5$ )
if ( $|A| < 1$ )
Denklem 2.1 ile arama ajanı konumu güncelle
else if ( $|A| < 1$ )
Rastgele bir arama ajanı seç ( $\vec{X}_{rand}$ )
Denklem 2.8 ile arama ajanını güncelle
end if
else if ( $p \geq 0.5$ )
Denklem 2.5 ile arama ajanı konumu güncelle
end if

end for

Kısıt dışına çıkan bireylere sınır değerini ver
amaç fonksiyon değerlerini hesapla
Daha iyi çözüm bulunmuşsa en iyi ajanı güncelle
 $t=t+1$ 
end while
Sonuç  $X^*$ 

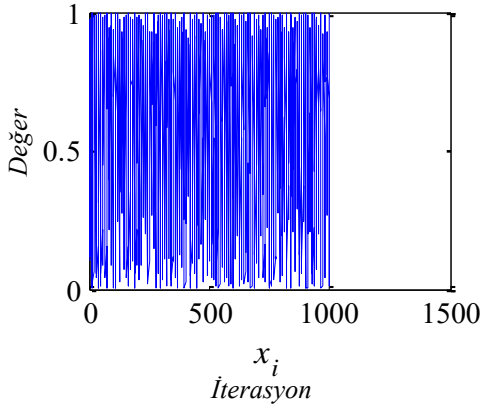
```

### 3.1 Lojistik harita

Lojistik harita kaosu gösterilmesi için ayrık zamanlı dinamik bir sistemdir. Tek boyutludur ve doğrusal değildir. Lojistik harita formülasyonu Denklem 3.1'de verilmiştir.

$$X_{n+1} = aX_n(1 - X_n) \quad (3.1)$$

Denklem 3.1'de  $n$  iterasyon sayısı,  $X_n$   $n$ .kaotik sayı,  $a$  ise lojistik harita sabitidir.  $a$ 'nın alabileceği değerlere göre sistem kaotik ya da kaotik olmayan davranış sergiler.  $a$  sabiti [3.57,4] aralığında değer aldığı zaman lojistik harita kaotik olmaktadır [35]. Bu çalışmada  $a = 4$  olarak belirlenmiştir. 0.1 başlangıç noktası ile 1000 iterasyon sonunda lojistik haritanın verdiği değerler Şekil 3.1'deki grafikte gösterilmiştir. Grafikten, sistemin yayılmış spektruma sahip olduğu görülebilmektedir.



Şekil 3.1  $x_0 = 0.1$  ve  $a=4$  için 100 iterasyon ile çalıştırılan lojistik harita grafiği

### 3.2 Kaotik haritalı balina optimizasyon algoritmaları

KHBOA1: Denklem 2.3'te  $r$  parametresi kaotik harita fonksiyonu ile değiştirilerek Denklem 3.2 elde edilir. Denklem 3.2'deki  $kh$  değişkeni  $[0,1]$  aralığında değer alan lojistik harita tabanlı fonksiyondur ve sonraki denklemlerde de bu ifade kullanılmıştır.

$$\vec{A} = 2\vec{a}.kh - \vec{a} \quad (3.2)$$

KHBOA2: Denklem 2.4'te  $r$  parametresi kaotik harita fonksiyonu ile değiştirilerek Denklem 3.3 elde edilir.

$$\vec{C} = 2.kh \quad (3.3)$$

KHBOA3: Denklem 2.5'te  $l$  parametresi kaotik harita fonksiyonu ile değiştirilerek Denklem 3.4 ile elde edilir.  $kh$   $[-1,1]$  değişkeni aralığında değer alan lojistik harita tabanlı fonksiyondur.

$$\vec{X}(t+1) = \vec{D}^t . e^{b(khl)}. \cos(2\pi l) + \vec{X}^*(t) \quad (3.4)$$

KHBOA4: Denklem 2.6'daki  $p$  parametresi kaotik harita fonksiyonu ile değiştirilerek Denklem 3.5 ile elde edilir.

$$\vec{X}(t+1) = \begin{cases} \vec{X}^*(t) - \vec{A} . \vec{D}, & kh < 0.5 \\ \vec{D}^t . e^{bl}. \cos(2\pi l) + \vec{X}^*(t), & kh \geq 0.5 \end{cases} \quad (3.5)$$

KHBOA5: Başlangıç popülasyonu değerleri kaotik haritanın yinelenmesi ile oluşturulur.

## 4. Kalite Test Fonksiyonları

Optimizasyon algoritmalarının performans testi ve karşılaştırılmalarında kalite testi

fonksiyonları kullanılmaktadır. Literatürde yerel optimum sayıları ve sıklıklarına göre sınıflandırılan birçok kalite test fonksiyonu tanımlanmıştır.

Bu çalışmada farklı yapılarıdaki 5 adet kalite test fonksiyonu seçilmiştir. Tablo 4.1'de seçilen fonksiyonların grafikleri verilmiştir. Bu bölümde test fonksiyonlarının matematiksel ifadeleri de verilmiştir.

### 4.1 Branin fonksiyonu (F1)

Üç adet global minimum noktası olan kalite test fonksiyonudur. Denklem 4.1'de amaç fonksiyonu gösterilmektedir.

Amaç fonksiyonu:

$$f(x) = An + \sum_{i=1}^n [(x_i^2 - A \cos(2\pi x_i))] \quad (4.1)$$

Değişken aralığı:  $x_1 \in [-5,0], x_2 \in [10,15]$

Optimum nokta:  $f(x) = 0,397887$

Boyut: 2

### 4.2 Sphere fonksiyonu (F2)

Sphere Fonksiyonu, global minimum dışında boyut ( $d$ ) sayısı kadar yerel minimuma sahip olan, dış bükey, sürekli ve tek modlu fonksiyon türüdür. Denklem 4.2'de amaç fonksiyonu gösterilmektedir.

$$\text{Amaç fonksiyonu: } f(x) = \sum_{i=1}^d x_i^2 \quad (4.2)$$

Değişken aralığı:  $x_i \in [(-5,12), (5,12)]$

Optimum nokta:  $f(x) = 0$

Boyut: 30

### 4.3 Schwefel fonksiyonu (F3)

Karmaşık ve birçok yerel minimum noktası bulunan tek modlu bir fonksiyondur. Denklem 4.3'te amaç fonksiyonu gösterilmektedir.

Amaç fonksiyonu:

$$f(x) = 418.9829d - \sum_{i=1}^d x_i \sin(\sqrt{|x_i|}) \quad (4.3)$$

Değişken aralığı:  $x_i \in [-500,500]$

Optimum nokta:  $f(x) = 0$ , Boyut: 30

### 4.4 Ackley fonksiyonu (F4)

Sıklıkla optimizasyon problemlerini test etmek için kullanılan çok modlu bir test fonksiyonudur. Tablo 4.1'deki grafikte görüldüğü gibi geniş bir dış düzlük ve merkezde keskin bir iniş vardır. Özellikle tepe tırmanışı algoritmalarının zorlandığı bir fonksiyondur.



Denklem 4.4'te amaç fonksiyonu gösterilmektedir.

$$\text{Amaç fonk.: } f(x) = a \exp\left(-b \sqrt{\frac{1}{d} \sum_{i=1}^d x_i^2}\right) - \exp\left(\frac{1}{d} \sum_{i=1}^d \cos(cx_i^2)\right) + a + \exp(1) \quad (4.4)$$

Değişken aralığı:  $x_i \in [-50,50]$   
Optimum nokta:  $f(x) = 0$   
Boyut: 30,  $a=-20$

#### 4.5 De Jong fonksiyonu (F5)

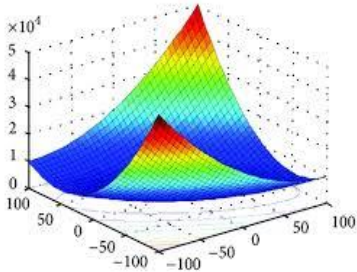
De Jong fonksiyonu dik iniş çıkışlı fonksiyon türüdür. Keskin inişlere ve birçok düzlüğe sahiptir. Bu fonksiyonun global

minimumu '1' olmaktadır. Fonksiyonun bir çok yerel minimumu olduğu için çoğu optimizasyon algoritmaları yerel minimuma takılmaktadır. Denklem 4.5'te amaç fonksiyonu gösterilmektedir.

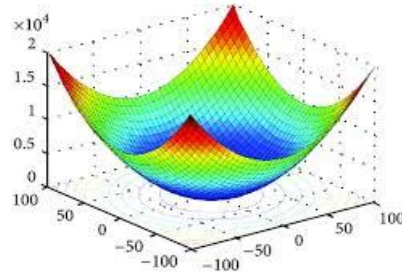
$$\text{Amaç fonksiyonu: } f(x) = (0.002) + \sum_{i=1}^{25} \frac{1}{i + (x_1 - a_{1i})^6 + (x_2 - a_{2i})^6} - 1 \quad (4.5)$$

Değişken aralığı:  $x_i \in [(-65,536), (65,536)]$   
Optimum nokta:  $f(x) = 1$ , Boyut: 2  
 $a = \begin{pmatrix} -32 & 16 & 0 & 16 & 32 & -32 & \dots & 9 & 16 & 32 \\ -32 & -32 & -32 & -32 & -32 & -16 & \dots & 32 & 32 & 32 \end{pmatrix}$

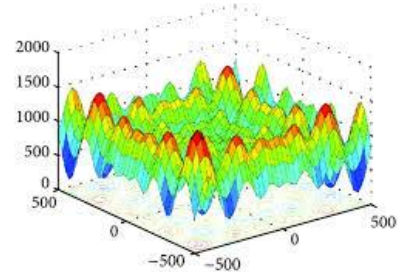
**Tablo 4.1** Seçilen test fonksiyonlarının grafikleri



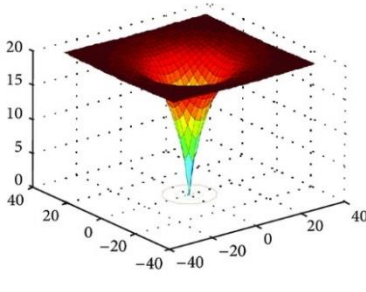
a) Branin Fonksiyonu



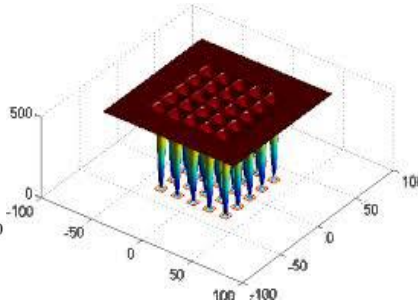
b) Sphere Fonksiyonu



c) Schfewel Fonksiyonu



d) Ackley Fonksiyonu



e) De Jong Fonksiyonu

#### 4.6 Kalite Test Fonksiyonlarının Sonuçları

Algoritmalar her test fonksiyonu için 500 iterasyon ile 30'ar kez çalıştırılarak ortalamaları alınmıştır. Sonuçlar Tablo 4.2'de verilmiştir. Ayrıca Tablo 4.3'te, test fonksiyonları ile çalıştırılan algoritmaların optimuma yakınsama grafikleri verilmiştir. Grafiklerden kaotik harita kullanan algoritmaların temel algoritmaya göre

çoğu zaman daha erken yakınsadıkları görülebilmektedir. Tablo 4.4'te Wilcoxon sıra toplam testi sonuçları görülmektedir. Bu tabloda "-" ile gösterilen alanlar sonuçlar arasında anlamlı fark olmadığını "+" ile gösterilen alanlar ise anlamlı bir fark olduğunu göstermektedir.

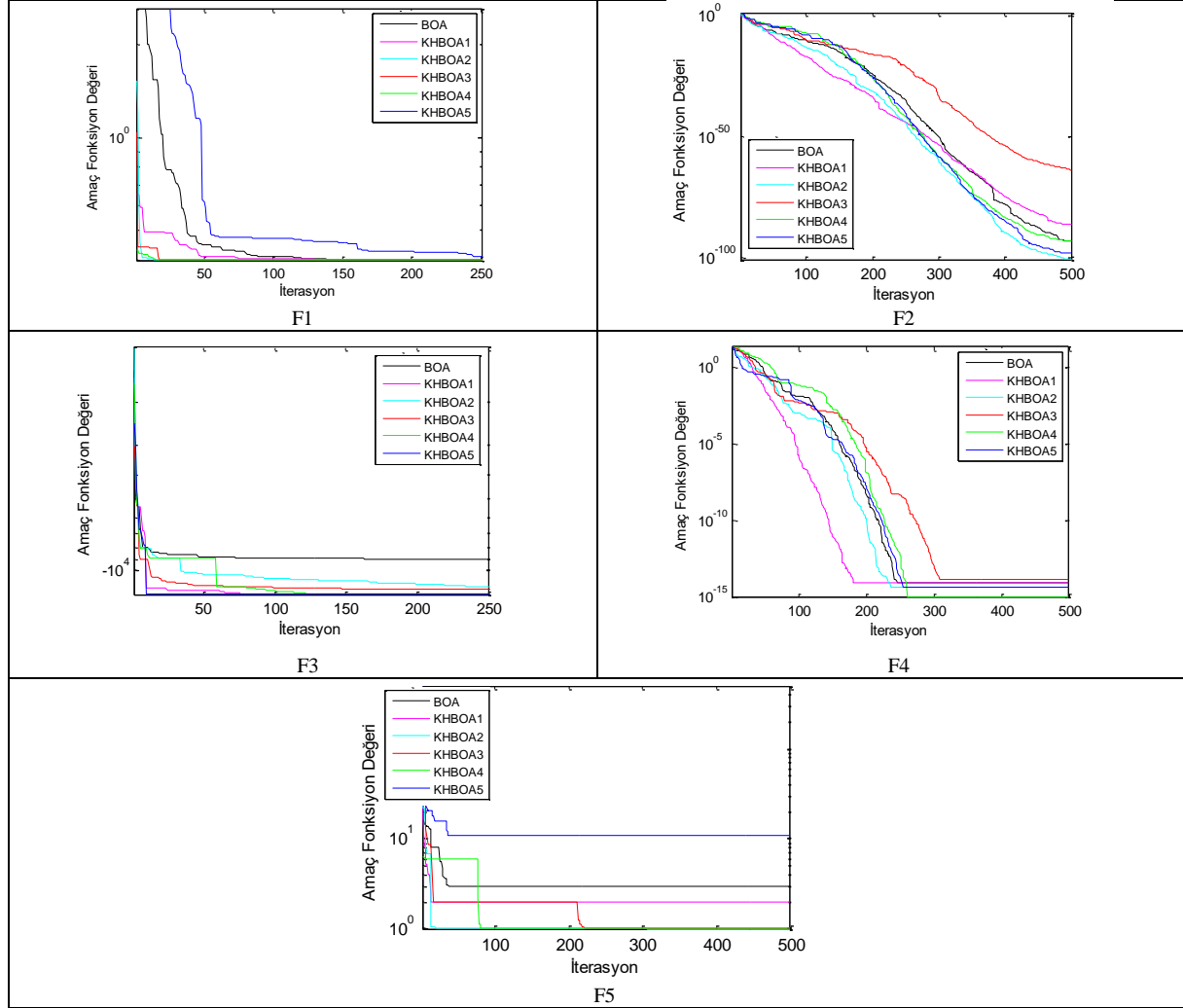
Bu test sadece gözlenen değerler arasında manidar bir farkın olup olmadığını anlaşılması için yapılmıştır. Bu nedenle çift yönlü Wilcoxon sıra toplam testi seçilmiştir. Ayrıca testin güven aralığı %95 olarak değerlendirilmiştir.

## Kaotik Haritalı Balina Optimizasyon Algoritmaları

**Tablo 4.2.** Test fonksiyonları sonuçları

<b>F</b>	<b>BOA</b>	<b>KHBOA1</b>	<b>KHBOA2</b>	<b>KHBOA3</b>	<b>KHBOA4</b>	<b>KHBOA5</b>
<b>F1</b>	0,397894	0,3979	0, 3979	0, 39789	0, 39789	0, 39789
<b>F2</b>	3,87E-84	7,12E-85	3,76E-94	4,6805E-85	6,51E-85	1,299E-87
<b>F3</b>	-10321,3456	-11551.9496	-10942,7321	-12484,2086	-9318,2718	-9434,3438
<b>F4</b>	4,44E-15	4,56E-15	3,26E-15	5,03E-15	4,68E-15	4,20E-15
<b>F5</b>	3,077910	1,1635	1,8256	2,6402	2,469756	3,965

**Tablo 4.3.** Algoritmaların yakınsama grafikleri



**Tablo 4.4.** Wilcoxon sıra toplam testi sonuçları

	BOA-KHBOA1		BOA-KHBOA2		BOA-KHBOA3		BOA-KHBOA4		BOA-KHBOA5	
	<i>p</i>	durum	<i>p</i>	durum	<i>p</i>	durum	<i>p</i>	durum	<i>p</i>	durum
<b>F1</b>	0,357	-	0,197	-	0,07	-	0,131	-	0,008	+
<b>F2</b>	0,00398	+	0,001	+	0,77	-	0,27	-	0,267	-
<b>F3</b>	0,00362	+	0,696	-	0,001	+	0,246	-	0,652	-
<b>F4</b>	0,05	-	0,01	+	0,610	-	0,003	+	0,158	-
<b>F5</b>	0,0083	+	0,57	-	0,1416	-	0,3125	-	0,1031	-

Tablo 4.2’de F1 fonksiyonu için temel algoritma optimuma yakın bir sonuç vermektedir, kaotik haritalar ise sonucu çok

küçük miktarda optimuma yaklaştırmıştır ama Tablo 4.4’te görüldüğü gibi Wilcoxon testi F1 fonksiyonu için anlamlı bir fark göstermemiştir.

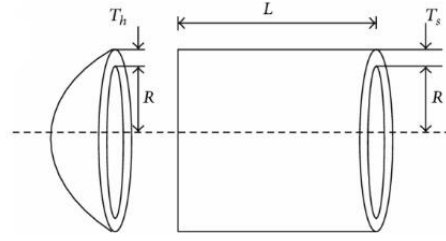


F2 fonksiyonunda KHBOA3 dışında, kaotik harita kullanan algoritmalar özellikle KHBOA2 sonuçları iyileştirebilmiştir. Wilcoxon testi sonucunda KHBOA1 ve KHBOA2 için anlamlı bir fark olduğu görülmüştür. F3 fonksiyonu için boyut sayısı 30 olduğundan, optimum nokta  $30 \times -418,9829 = -12569,487$  olarak alınmıştır. F3 fonksiyonunda özellikle KHBOA1 ve KHBOA3 sonuçları iyileştirebilmiştir. Bu durumu Wilcoxon testi de doğrulamaktadır. F4 fonksiyonunda kayda değer bir iyileşme görülmemiştir. F5 fonksiyonunda KHBOA1 ve KHBOA2 yöntemlerinin diğerlerine göre daha iyi sonuç verdiği görülebilmektedir ama Wilcoxon testinde KHBOA2 algoritmasının sonuçları için anlamlı fark olmadığı görülmektedir.

Sezgisel algoritmaların stokastik yapısından dolayı, iterasyonlar sırasında nadiren de olsa optimum noktadan çok uzak değerler verebiliyor. Örneğin F5 fonksiyonu için 30 çalıştırmanın 28'sinde değerler optimum nokta olan 1'e yakın, fakat 2 sonuç çok uzaktır. 30 çalıştırmanın ortalamasını sonuç olarak kabul ettiğimiz için, bu iki değer ortalamaya çok büyük etki etmemekte ama optimumdan uzaklaştırmaktadır. Kaotik harita kullanan algoritmalarda bu sapma azalmış sonuçlar daha istikrarlı olmuştur.

#### 4.7 Basınçlı Kap Tasarım Problemi

Literatürde optimizasyon problemlerinin performansını ölçmek için gerçek mühendislik uygulamaları sıkça kullanılmaktadır. Bunlardan biri de basınçlı kap tasarım problemidir. Basınçlı kap şekil 4.1'de görüldüğü gibi iki ucu yarı küresel başlıklarla kapalı olan bir silindirdir. Bu problemde amaç maliyeti (malzeme, kaynak, şekillendirme) minimize etmektir. Problemin tasarım değişkenleri de şekil 4.1 üzerinde görülmektedir.  $T_s$  ( $x_1$ ) kabuk kalınlığı,  $T_h$  ( $x_2$ ) başlık kalınlığı,  $R$  ( $x_3$ ) iç yarıçap ve  $L$  ( $x_4$ ) başlık haricindeki uzunluk olmak üzere 4 adettir [36]. Değişkenlerin değer aralıkları Denklem 4.6'da gösterilmiştir.



Şekil 4.1. Basınçlı kap tasarımı

$$\begin{aligned} 0 &\leq x_1 \leq 99, \\ 0 &\leq x_2 \leq 99, \\ 0 &\leq x_3 \leq 200, \\ 0 &\leq x_4 \leq 200 \end{aligned} \quad (4.6)$$

Problemin amaç fonksiyonu Denklem 4.7'de gösterilmiştir. Tasarım Denklem 4.8'de matematiksel ifadeleri gösterilen 4 adet kısıtı ihlal etmemelidir. Algoritmalar kısıtlı problem üzerinde doğrudan çalıştırılmamaktadır. Kısıtlı optimizasyon problemlerinin çözümü için literatürde birçok yöntem bulunmaktadır [37]. Bu çalışmada ceza metodu kullanılmış ve algoritmaya entegre edilmiştir. Ceza fonksiyon sabiti ise 1000000 olarak algoritmalar alınmıştır.

$$f(\vec{x}) = 0,6224x_1x_3x_4 + 1,7781x_2x_3^2 + 3,1661x_1^2x_4 + 19,84x_1^2x_3 \quad (4.7)$$

#### Kısıtlar

$$\begin{aligned} g_1(\vec{x}) &= -x_1 + 0,0193x_3 \leq 0, \\ g_2(\vec{x}) &= -x_2 + 0,00954x_3 \leq 0, \\ g_3(\vec{x}) &= -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 + 1,296000 \leq 0, \\ g_4(\vec{x}) &= x_4 - 240 \leq 0, \end{aligned} \quad (4.8)$$

BOA ve Kaotik Haritalı Balina Optimizasyon algoritmaları basınçlı kap tasarım problemi üzerinde karşılaştırmalı olarak test edilmiştir. Algoritmalar 200 popülasyon ve 10000 iterasyon ile 30'ar kez çalıştırılarak elde edilen sonuçların ortalamaları alınmıştır. Elde edilen değerler Tablo 4.6'da gösterilmiştir. Tabloda görüldüğü gibi KHBOA5 dışındaki kaotik harita kullanan algoritmalar, özellikle KHBOA1 optimum maliyeti düşürebilmiştir. Bu testte sadece temel algoritma üzerinde kaotik haritaların etkisi incelenmiştir.

**Tablo 4.6** Basınçlı kap tasarımı problemi test sonuçları

Algoritma	Optimum değerler				Optimum Maliyet
	$T_s$	$T_h$	$R$	$L$	
WOA	0,90049	0,4514	46,5828	139,6198	6215,8866
KHBOA1	0,79059	0,40527	41,1396	189,2508	5940,211
KHBOA2	0,87241	0,43459	45,1905	147,9899	6110,7279
KHBOA3	0,8805	0,44425	45,5427	144,2143	6153,7201
KHBOA4	0,86314	0,43793	44,8383	152,3472	6102,132
KHBOA5	0,97072	0,49234	50,3386	112,4768	6447,2273

## 6.Sonuç

Optimizasyon alanında algoritmaların performansını karşılaştırmak için global optimizasyon olarak bilinen matematiksel fonksiyonları kullanmak ortak bir fikirdir. Bu fonksiyonlar gerçek arama alanlarına son derece benzerdir ve algoritma karşılaştırması için uygundur.

Bu çalışmada kambur balinaların avlanma davranışından esinlenen Balina Optimizasyon Algoritması incelenmiş ve kaotik versiyonları oluşturulmuştur. Algoritmaların performansları farklı boyutlardaki tek modlu, çok modlu kalite test fonksiyonları ve bir adet gerçek mühendislik problemi üzerinde incelenerek elde edilen sonuçlar Wilcoxon sıra toplam testi ile anlamlandırılmış, tablo ve grafikler ile karşılaştırılmıştır. Kaotik algoritmalar, temel algoritmaya göre büyük bir performans artışı sağlayamasa da yerel optimumdan kaçarak optimum noktaya yakınsamayı attrabilmektedir. Özellikle KHBOA1'in çözüm kalitesini arttırdığı görülmüştür. Kaotik haritaların aramadaki sert iniş çıkışları engellemesi, sonucu olumlu olarak etkilemiştir. Kaosun bazı optimizasyon problemleri için istenen bir süreç olabileceği görülmüştür.

## Kaynaklar

1. Mirjalili S., Lewis A. (2016), The Whale Optimization Algorithm, *Advances in Engineering Software*,**95**:51,67
2. EMEL G.G.,TAŞKIN Ç. (2002), Genetik Algoritmalar ve Uygulama Alanları, Uludağ Üniversitesi İktisadi ve İdari Bilimler Fakültesi, Cilt XXI, Sayı 1,129-152
3. Rechenberg I.( 1978), *Evolutions strategien*. Springer Berlin Heidelberg, 83–114.
4. Dasgupta D., Zbigniew M.(2013), *Evolutionary algorithms in engineering applications*. Springer Science & Business Media.
5. Koza J. R. (1992),*Genetic Programming*,
6. Simon D.(2008), Biogeography-based optimization. *IEEE Trans Evol Comput*;**12**:702–13.
7. Kirkpatrick S, Gelatt CD, Vecchi MP. (1983), Optimization by simulated annealing. *Science*;**220**:671–80.
8. Cerny V.(1985), Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm. *J Opt Theory Appl*;**45**:41–51.
9. Webster B, Bernhard PJ.(2003), A local search optimization algorithm based on natural principles of gravitation, *Proceedings of the international conference on information and knowledge engineering (IKE'03)*, 255–61.
10. Erol O.K., Eksin I.. (2006), A new optimization method: big bang–big crunch. *Adv Eng Softw*;**37**:106–11.
11. Rashedi E, Nezamabadi-Pour H, Saryazdi S. (2009), GSA: a gravitational search algorithm. *Inf Sci*;**179**:2232–48.
12. Hatamlou A.(2013), Black hole: a new heuristic optimization approach for data clustering. *Inf Sci*;**222**:175–84.
13. Kaveh A, Khayatazad M.(2012), A new meta-heuristic method: ray optimization. *Comput Struct*;**112**:283–94.
14. Kennedy J, Eberhart R. (1995) , Article swarm optimization, *Proceedings of the IEEE international conference on neural networks*, 1942–8.
15. Dorigo M, Birattari M, Stutzle T. (2006), Ant colony optimization. *IEEE Comput Intell*;**1**:28–39.
16. Rao R.V, Savsani V.J, Vakharia D.P. (2012), Teaching–learning-based optimization: an optimization method for continuous non-linear large scale problems. *Inf Sci*;**183**:1–15.
17. Rao V., Savsani V., Vakharia D.P.(2011), Teaching–learning-based optimization: a novel method for constrained mechanical design

- optimization problems. *Computer-Aided Des*;43:303–15.
18. Geem Z.W, Kim J.H., Loganathan G. (2001), A new heuristic optimization algorithm: harmony search. *Simulation*;76:60–8.
  19. Glover F.(1989), Tabu search – Part I. *ORSA J. Comput.*;1:190–206.
  20. Glover F.(1990), Tabu search – Part II. *ORSA J. Comput.*;2:4–32.
  21. He S., Wu Q.H., Saunders J.R.(2006),A novel group search optimizer inspired by animal behavioural ecology, *Proceedings of the 2006 IEEE congress on evolutionary computation, CEC*; 1272–8.
  22. He S, Wu Q.H, Saunders J.(2009), Group search optimizer: an optimization algorithm inspired by animal searching behavior. *IEEE Trans Evol Comput*;13:973– 90.
  23. Olorunda O., Engelbrecht A.P.(2008), Measuring exploration/exploitation in particle swarms using swarm diversity. *Proceedings of the 2008 IEEE congress on evolutionary computation, CEC (IEEE world congress on computational intelligence)*, 1128–34.
  24. Lin L., Gen M.(2009), Auto-tuning strategy for evolutionary algorithms: balancing between exploration and exploitation. *Soft Comput*;13:157–68.
  25. Mirjalili S., Mirjalili S.M., Lewis A.(2014), Grey wolf optimizer. *Adv Eng Softw*;69:46–61.
  26. Hof P.R., Gucht V.D. E.(2007), Structure of the cerebral cortex of the humpback whale, *Megaptera novaeangliae* (Cetacea, Mysticeti, Balaenopteridae). *Anat Rec*;290:1–31.
  27. Goldbogen J.A., Friedlaender A.S., Calambokidis J., Mckenna M.F., Simon M, Nowacek D.P.(2013), Integrative approaches to the study of baleen whale diving behavior, feeding performance, and foraging ecology. *BioScience*;63:90–100.
  28. Nozawa H.,(1992) A neural network model as globally coupled map and application based on chaos, *Chaos*, 377–386
  29. Tan Y. Tan G. Deng S.(2014), Hybrid particle swarm optimization with chaotic search for solving integer and mixed integer programming problems, *Journal of Central University*, Volume 21, Issue 7, 2731-2742
  30. Yang Z., Shi Y.(2015), Brain storm optimization with chaotic operation, *Advanced Computational Intelligence (ICACI), 2015 Seventh International Conference*, 978-1-4799-7259-3
  31. 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
  32. Alataş B. Akın E., Özer B. (2007). Kaotik Haritalı Parçacık Sürü Optimizasyon Algoritmaları, Eskişehir Osmangazi Üniversitesi Elektrik Elektronik Bilgisayar Biyomedikal Mühendisliği Ulusal Kongresi.
  33. 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
  34. Alpar, O., Öztürk, A. A., & Eren, Ö. (2011). Kaotik Markov zincirleri. *İşletme İktisadi Enstitüsü Yönetim Dergisi*, 69
  35. Özkaynak F., Özer A.B. (2015). An Analysis Study on Role of Chaos in Symmetric Encryption Algorithm, *The 8th Chaotic Modeling and Simulation International Conference, Paris France*, 26 - 29 May, 551-662
  36. Nasser SH, Alizadeh Z, Taleshian F.(2012). Optimized solution of pressure vessel design using geometric programming. *The Journal of Mathematics and Computer Science Vol. 4*, 3:344 – 349
  37. Coello C.A.C.(2002) Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Comput Methods Appl Mech Eng*; 191:1245–87