

## **MARTI OPTİMİZASYON ALGORİTMASININ KISITLI MÜHENDİSLİK TASARIM PROBLEMLERİ İÇİN PERFORMANS ANALİZİ**

**Feyza ALTUNBEY ÖZBAY<sup>1\*</sup>, Erdal ÖZBAY<sup>2</sup>**

<sup>1</sup>Fırat Üniversitesi, Mühendislik Fakültesi, Yazılım Mühendisliği Bölümü, Elazığ, 23119, Türkiye

<sup>2</sup>Fırat Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Elazığ, 23119, Türkiye

Geliş Tarihi/Received Date: 03.09.2021 Kabul Tarihi/Accepted Date: 30.09.2021 DOI: 10.54365/adyumbd.990708

### **ÖZET**

Metasezgisel arama algoritmaları, birçok uygulama alanında farklı optimizasyon problemlerini çözmek için yaygın bir biçimde kullanılmaktadır. Özellikle, son yıllarda, karmaşık optimizasyon problemlerini etkin bir biçimde çözebilmek için fiziksel, kimyasal veya biyolojik olaylardan esinlenilerek birçok farklı metasezgisel algoritma geliştirilmiştir. Doğadaki martıların göç ve saldırı davranışlarından ilham alınarak geliştirilen Martı Optimizasyon Algoritması (MOA), maliyetli optimizasyon problemlerinin çözümü için etkili biyoloji tabanlı metasezgisel bir yöntemdir. Bu çalışmada, MOA'nın performansını değerlendirmek için, MOA amaç fonksiyonları, kısıtları ve karar değişkenleri farklı beş kısıtlı mühendislik tasarım problemine uygulanmıştır. MOA ile elde edilen sonuçlar, farklı metasezgisel algoritmalar ile karşılaştırılmıştır. Elde edilen deney sonuçlarına göre, MOA, karşılaştırılan diğer optimizasyon yöntemlerine göre oldukça iyi sonuçlar vermiştir.

**Anahtar Kelimeler:** Martı optimizasyon algoritması, Metasezgisel algoritmalar, Mühendislik tasarım problemi

## **PERFORMANCE ANALYSIS OF SEAGULL OPTIMIZATION ALGORITHM FOR CONSTRAINED ENGINEERING DESIGN PROBLEMS**

### **ABSTRACT**

Metaheuristic search algorithms are widely used in many application areas to solve different optimization problems. In particular, in recent years, many different metaheuristic algorithms have been developed, inspired by physical, chemical, or biological phenomena, in order to effectively solve complex optimization problems. The Seagull Optimization Algorithm (MOA), inspired by the migration and attack behavior of seagulls in nature, is an effective biology-based metaheuristic method for solving costly optimization problems. In this study, MOA objective functions, constraints, and decision variables have been applied to five different constrained engineering design problems to evaluate the performance of MOA. The results obtained with MOA have been compared with different metaheuristic algorithms. According to the experimental results obtained, MOA achieved quite good results compared to other optimization methods.

**Keywords:** Seagull optimization algorithm, Metaheuristic algorithms, Engineering design problem

### **1. Giriş**

Optimizasyon son yıllarda oldukça büyük ilgi görmekte ve her geçen gün birçok yeni optimizasyon yöntemi geliştirilmektedir. Geliştirilen bu yeni yöntemler gerçek dünya problemlerini çözmek için kullanılmaktadır. Optimizasyon, belirli kısıtlamalar altında mevcut arama alanında, belirli

\*1 e-mail: [faltunbey@firat.edu.tr](mailto:faltunbey@firat.edu.tr) ORCID ID: <https://orcid.org/0000-0003-0629-6888> (Sorumlu Yazar),

<sup>2</sup> e-mail: [erdalozbay@firat.edu.tr](mailto:erdalozbay@firat.edu.tr) ORCID ID: <https://orcid.org/0000-0002-9004-4802>

bir problemin en uygun çözümünü araştıran aktif bir araştırma alanıdır. Optimizasyonun temel amacı, bir amaç fonksiyonu minimize veya maksimize etmek için karar değişkenlerini belirleme sürecidir.

Gerçek dünya problemlerinin çoğu, doğrusal olmayan kısıtlamalara, yüksek hesaplama maliyetine, karmaşık ve çok sayıda çözüm alanına sahiptir. Bu nedenle, çok sayıda değişken ve kısıtlama içeren bu tür problemleri çözmek oldukça karmaşıktır. Klasik optimizasyon tekniklerinin çoğu, son zamanlarda geliştirilen karmaşık optimizasyon problemlerine uygun çözümler sağlayamayan klasik matematik ve olasılık ilkelerine dayalı olarak formüle edilmiştir. Matematiksel programlama gibi klasik optimizasyon yöntemleri genellikle, basit optimizasyon problemleri için etkili çözümler sunabilmektedir. Ancak, gerçek dünya mühendislik optimizasyon problemlerini klasik optimizasyon yöntemleri ile çözülmesi nispeten zordur. Bu nedenle birçok araştırmacı, karmaşık optimizasyon problemlerini makul bir hesaplama süresi içinde kabul edilebilir bir maliyet ile çözmek için metasezgisel algoritmalar adı verilen yeni çözüm teknikleri önermeye başlamıştır.

Metasezgisel yöntemler, klasik optimizasyon tekniklerine göre birçok avantaja sahip olduğu için son yıllarda araştırmacılar arasında daha popüler hale gelmiştir. Bu yöntemlerin, klasik optimizasyon yöntemlerine göre en önemli avantajı, optimizasyon problemlerini çözmek için gradyan bilgisine ihtiyaç duymamasıdır. Diğer önemli bir avantaj ise, farklı çalışma alanlarında uygulanmalarının kolay olmasıdır. Metasezgisel optimizasyon algoritmalarının en önemli avantajlarından biri de, iyi bilinen kısıtlı matematiksel ve mühendislik tasarım problemlerine daha iyi çözümler sağlayabilme konusundaki etkinlikleri ve başarılarıdır. Metasezgisel algoritmalar, zor optimizasyon problemleri için iyi çözümler bulabilir, ancak optimal çözümlere ulaşılabilmesinin garantisi bulunmamaktadır [1]. Araştırmacılar metasezgisel algoritmaları geliştirirken doğadaki fiziksel, biyolojik, kimyasal olaylardan ya da doğadaki canlıların yaşam biçimlerinden ilham almışlardır. Elektromanyetik kuvvet, eylemsizlik kuvveti ve yerçekimi gibi fizik kanunlarını kullanarak farklı yöntemler önermişlerdir. Fizik kanunlarından esinlenerek geliştirilen algoritmalarından popüler olanları: Büyük-Patlama Büyük-Çöküş (Big-Bang Big-Crunch) [2], Kara Delik (Black Hole) [3], Merkezi Kuvvet Optimizasyonu (Central Force Optimization) [4], Yer Çekimi Arama Algoritması (Gravitational Search Algorithm) [5], Su Döngüsü Algoritması (Water Cycle Algorithm) [6]. Metasezgisel yöntemler önerilirken ilham alınan diğer önemli bir esin kaynağı ise doğadaki canlıların sürü şeklindeki yaşayış biçimleridir. Parçacık Sürüsü Optimizasyonu (Particle Swarm Optimization) [7], Karınca Koloni Optimizasyonu (Ant Colony Optimization) [8], Yarasa Algoritması (Bat Algorithm) [9], Ateşböceği Algoritması (Firefly Algorithm) [10], Yapay Arı Kolonisi (Artificial Bee Colony) [11] gibi birçok algoritma sürü tabanlı metasezgisel yöntemlere örnek olarak verilebilir.

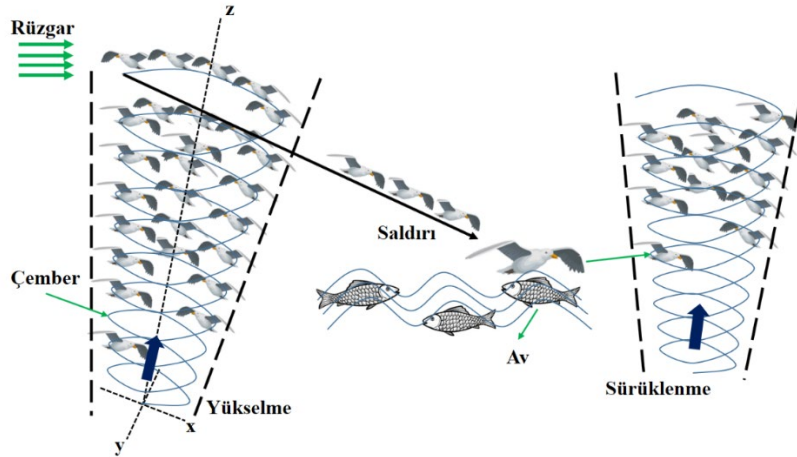
Literatürde birçok optimizasyon algoritması önerilmesine rağmen bütün optimizasyon problemlerine en uygun çözümü sunabilecek bir algoritma bulunmamaktadır [12]. Martı Optimizasyon Algoritması (MOA), doğadaki martıların göç ve saldırı davranışlarından esinlenerek geliştirilmiş yeni biyoloji tabanlı bir metasezgisel yöntemdir [13]. Bu yöntemde, martıların bu davranışları, belirli bir arama uzayında keşif ve sömürüyü vurgulamak için matematiksel olarak modellenmiştir [14]. Bu çalışmada, yeni geliştirilmiş bir metasezgisel yöntem olan MOA'nın farklı mühendislik tasarım problemleri üzerindeki performansları ölçülmüştür. Bu amaçla MOA, kaynaklı kiriş, germe/sıkıştırma yay, basınçlı kap, hız düşürücü ve üç çubuklu kafes olmak üzere amaç fonksiyonları, kısıtları ve karar değişkenleri farklı beş kısıtlı mühendislik tasarım problemine uygulanmıştır. Biyolojiden esinlenerek geliştirilen MOA ile edilen sonuçlar, literatürdeki, kaynaklı kiriş, germe/sıkıştırma yay, basınçlı kap, hız düşürücü ve üç çubuklu kafes problemine çözüm bulan farklı metasezgisel algoritmalar ile karşılaştırılarak performans analizi yapılmıştır.

Çalışmanın geri kalanı aşağıdaki gibi ilerlemektedir: İkinci bölümde, MOA'nın ilham kaynağı ve algoritmanın matematiksel modeli incelenmiştir. MOA'nın performansı, beş farklı mühendislik tasarım problemi üzerinde test edilmiş ve elde edilen sonuçlar iyi bilinen birçok farklı algoritma ile karşılaştırılmıştır. Sonuçlar ve gelecekteki çalışmalar dördüncü bölümde tartışılmıştır.

## 2. Martı Optimizasyon Algoritması

### 2.1. Biyolojik Paradigma

Laridae ismiyle bilinen martıgiller familyasının birer üyesi olan martılar, türlerinin çoğunluğunun Larus olarak adlandırıldığı deniz kuşlarının ortak adıdır. Martılar, hem tatlı hem de tuzlu su içebilen nadir hayvanlardan biridir. Bu bakımdan martılar, çöller ve kutupların donmuş kesimleri haricinde dünyanın birçok bölgesindeki su çevrelerinde koloniler halinde yaşama olanağı bulabilen, genellikle iri gövdeli hantal kıyı kuşlarıdır. Çeşitli martı türleri, farklı kütle ve uzunluğa sahip olabilmektedir. Türlerinin çoğu uzun bir gagaya ve perdeli ayaklara sahiptir, büyük çoğunluğunun gövdeleri beyaz ya da gri tonlarındadır ve baş kısımlarında siyah renkte izler bulunabilmektedir. Martı türleri, gaga renkleri ve kanat desenlerinin yanı sıra bacaklarına da bakılarak ayrıştırılabilmektedir. Martılar omnivordur, çoğunlukla besinleri, yengeç, küçük balık ve yumuşakçalardan oluşmaktadır. Bunların dışında böcek, sürüngen, amfibi ve solucan gibi hemen hemen her şeyi yiyebilen cesur ve zeki kuşlardır. Örneğin; balık avlarken onları çekmek için ekme kırıntılarını kullanırlar, benzer şekilde toprak altındaki solucanları yukarıya çekmek için ayaklarıyla yağmuru anımsatan sesler çıkarabilmektedirler. İletişimlerinde ise gelişmiş bir sosyal yapının karmaşık metotlarını sergilemektedirler. Örneğin; birçok martı türü, farklı türden kuş ya da diğer davetsiz misafirlere karşı saldırı eğilimindedir.



Şekil 1. Martıların ava saldırma ve göç davranışları

Martılar için en önemli iki davranışı, göç etme ve ava saldırı planlamasıdır. Martıların bu iki önemli davranışı Şekil 1’de örneklendirilmiştir. Martılar av bulma ve saldırı konularında zekâsını ortaya koymaktadırlar. Martılar için göç, kendilerine yeterli enerjiyi sağlayabilecek, bol ve en zengin besin kaynaklarına ulaşabilmek için mevsimsel olarak bir yerden başka bir yere hareket etmeleri olarak nitelendirilmektedir [15]. Martıların göç ve ava saldırı davranışlarına ait bazı özellikler şu şekilde açıklanmaktadır;

- Martılar göçe hazırlanırken birbirleriyle çarpışmalarını için farklı başlangıç konumlarından harekete geçerler ve göç esnasında sürekli grup halinde seyahat ederler.
- Grup halindeki martılar, hayatta kalmaya en uygun olan martı (popülasyonu değerlendiren ve çözümün kalitesini ölçen bir fonksiyon olan uygunluk değeri diğerlerine kıyasla düşük olan) yönünde seyahati tercih edebilirler.
- Diğer martılar en uygun martıya göre başlangıç konumlarını güncelleyebilmektedir.

Martılar, göç esnasında göç eden diğer kuşlara saldırı yapma eğilimindedir [16]. Saldırı hücumu sırasında doğal spiral şeklindeki hareketleri sergileyebilirler. Bu davranışın yansıtıldığı kavramsal bir model Şekil 1’de verilmiştir.

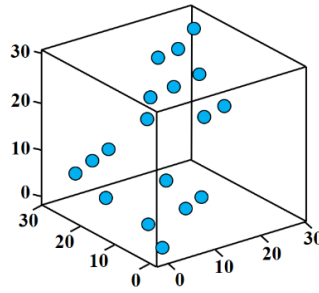
## 2.2. Matematiksel Model

MOA, martıların göç ve ava saldırı esnasında sergilediği doğal davranışlardan esinlenerek formüle edilen bir algoritmadır. Bu bakımdan bu davranışlar, optimize edilecek olan amaç fonksiyonu ile ilişkilendirilebilecek şekilde formüle edilmeye imkân sağlamaktadır.

### 2.2.1. Göç (Keşif)

MOA'da, bir grup martının bir konumdan diğerine geçiş esnasında nasıl hareket ettiği benzeştirilmektedir. Bir martının bu aşamada sağlaması zorunda olduğu üç koşul bulunmaktadır. Bunlar:

- **Çarpışmalardan kaçınma:** Şekil 2'de gösterildiği üzere birbirleri (komşu martılar) arasında gerçekleşebilecek çarpışmaları önlemek adına, yeni arama ajanı konumunun hesaplanması için denklem (1)'de ek bir  $A$  değişkeni kullanılmaktadır [17].



Şekil 2. Arama ajanları arasında çarpışmadan kaçınma

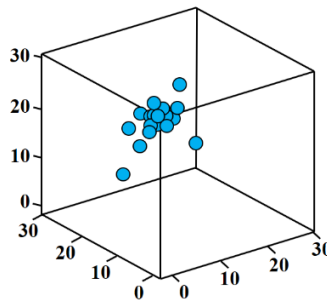
$$\vec{C}_s = A \times \vec{P}_s(x) \quad (1)$$

Denklem 1'deki  $\vec{C}_s$ , diğer arama ajanları ile çarpışmayan bir aday arama ajanının konumunu temsil etmektedir,  $\vec{P}_s$ , mevcut arama ajanının pozisyonunu temsil etmektedir. Buradaki  $x$ , mevcut yinelemeyi göstermektedir.  $A$  ile bir arama ajanının, verilen arama uzayındaki hareketine ait davranışı temsil edilmektedir [17].

$$A = f_c - (x \times (f_c / Mak_{yineleme})) \quad (2)$$

Burada  $x = 0, 1, 2, \dots, Mak_{yineleme}$  aralığındadır.  $f_c$ ,  $f_c$ 'den 0'a doğru lineer olarak azaltılan  $A$  değişkeninin kullanım sıklığını kontrol etmek için tanımlanmıştır.

- **En iyi komşu yönünde hareket:** Komşular arasındaki çarpışmalardan kaçınmakla birlikte Şekil 3'te gösterildiği üzere arama ajanları en iyi komşu yönüne doğru hareket etmektedirler [17].



Şekil 3. Arama ajanlarının en iyi komşu yönünde hareketi

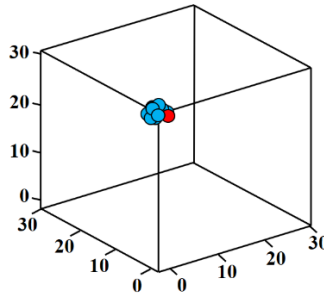
$$\vec{M}_s = B \times (\vec{P}_{bs}(x) - \vec{P}_s(x)) \quad (3)$$

Buradaki  $\vec{M}_s$ ,  $\vec{P}_{bs}$ 'nin en uygun arama ajanı olan  $\vec{P}_s$ 'ye (en uygun martı) yönelik konumlarını temsil etmektedir.  $B$ , keşif ve sömürü arasındaki dengeyi niteleyen rastgele bir davranışı nitelendirmektedir ve denklem (4) ile hesaplanmaktadır:

$$B = 2A^2 \times rd \quad (4)$$

Burada, [0, 1] aralığında yer alan rastgele bir sayıdır.

• **En iyi arama ajanına yakın kalma:** Göç (keşif) davranış modelinin son koşulu olarak, arama ajanı, Şekil 4'te gösterildiği üzere en iyi arama ajanına yönelimli olarak konumunu güncelleyebilmektedir.

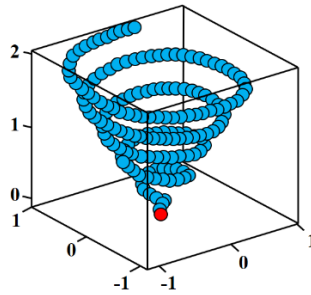


Şekil 4. En iyi arama ajanına yönelik yakınsama hareketi

$$\vec{D}_s = |\vec{C}_s + \vec{M}_s| \quad (5)$$

Buradaki  $\vec{D}_s$ , arama ajanı ile en uygun arama ajanı (uygunluk değeri daha düşük olan en iyi martı) arasındaki mesafeyi temsil etmektedir.

### 2.2.2. Ava Saldırı (Sömürü)



Şekil 5. Martının avına saldırıdaki doğal spiral hareketi

Martılar için ava saldırı, geçmişteki arama süreçlerinden edinilen tecrübe ve deneyimlerin bütününden yararlanmayı hedeflemektedir. Göç esnasında martılar, hızlarının dışında zaman zaman saldırı açılarını da değiştirebilmektedir. Bununla birlikte yüksek irtifalarını korumak adına uzun kanat açıklığı ve gövde ağırlıklarından faydalanmaktadırlar. Martılar belirledikleri bir ava saldırı düzenledikleri sırada havada spiral hareket davranışları sergilemektedirler. Bu hareket şekil 5'te temsili

olarak gösterilmektedir. Bu davranış koordinat ekseninin  $x$ ,  $y$  ve  $z$  düzlemlerinde denklem (6)-(9) ile tanımlanmaktadır [17].

$$x' = r \times \cos(k) \quad (6)$$

$$y' = r \times \sin(k) \quad (7)$$

$$z' = r \times k \quad (8)$$

$$r = u \times e^{kv} \quad (9)$$

Burada  $r$ , spiral şeklindeki her bir dönüşün yarıçapını ifade etmektedir.  $k$ ,  $[0 \leq k \leq 2\pi]$  aralığında rastgele üretilen bir katsayıdır.  $u$  ve doğal logaritmik taban olan  $e$  üzerindeki  $v$ , spiral şekli tanımlayan sabit değerlerdir. Arama ajanının güncel konumları denklem (5)-(9) ile hesaplanmaktadır.

$$\vec{P}_s(x) = \vec{D}_s \times x' \times y' \times z' + \vec{P}_{bs}(x) \quad (10)$$

Buradaki  $\vec{P}_s(x)$  ile hesaplanan en iyi çözüm saklanmakta ve diğer arama ajanlarının konumu güncellenmektedir [17].

Biyolojiden ilham alınarak geliştirilen bir optimizasyon algoritması olan MOA, çalışmaya rastgele oluşturulmuş bir popülasyon ile başlamaktadır. Konum yineleme işlemleri sırasında, arama ajanları konumlarını en iyi arama ajanına göre güncelleyebilmektedir.  $A$  değişkenin değeri, her bir yinelemede,  $f_c$ 'den 0'a doğru lineer olarak azaltılmaktadır.  $B$  değişkeni, göç (keşif) ve ava saldırı (sömürü) arasındaki geçiş dengesinin yumuşaklığından sorumludur. Bu bakımdan MOA, daha iyi bir göç (keşif) sunması ve kullanım kabiliyetinin kolaylığı nedeniyle küresel bir optimize edici olarak kabul edilebilmektedir. MOA'nın sözde kodu Algoritma-1 ile verilmiştir.

---

#### Algoritma-1: Martı Optimizasyon Algoritması (MOA)

---

**Girdi:**  $\vec{P}_s$ , martı popülasyonu

**Çıktı:**  $\vec{P}_{bs}$ , en iyi arama ajanı

1: **başla** (MOA prosedürü)

2:  $A$ ,  $B$  ve  $Max_{yineleme}$  başlangıç parametrelerini ayarla

3:  $f_c \leftarrow 2$ ,  $u \leftarrow 1$ ,  $v \leftarrow 1$ 'e ayarla

4: **while** ( $x < Max_{yineleme}$ ) **do**

5:  $\vec{P}_{bs} \leftarrow FitnessHes(\vec{P}_s)$ , FitnessHes ile her arama ajanı için uygunluk değeri hesaplanır.

*Göç (keşif) davranışı-*

6:  $rd \leftarrow Rast(0, 1)$ ,  $[0, 1]$  aralığında rastgele sayı üretilir.

7:  $k \leftarrow Rast(0, 2\pi)$ ,  $[0, 2\pi]$  aralığında rastgele sayı üretilir.

*Ava saldırı (sömürü) davranışı-*

8:  $r \leftarrow u \times e^{kv}$ , spiral davranışın oluşturulması için.

9: Denklem (5) kullanılarak  $\vec{D}_s$  mesafesi hesaplanır.

10:  $P \leftarrow x' \times y' \times z'$  (6)-(9) Denklemleri kullanılarak  $x$ ,  $y$ ,  $z$  düzlemleri hesaplanır.

11:  $\vec{P}_s(x) \leftarrow (\vec{D}_s \times P) + \vec{P}_{bs}(x)$

12:  $x \leftarrow x + 1$

13: **end while**

14: return  $\vec{P}_{bs}$

15: **bitir**

---

1: **başla** FitnessHes ( $\vec{P}_s$ ) prosedürü

2: **for**  $i \leftarrow 1 : n$  **do**, burada  $n$ , verilen bir problemin boyutunu temsil etmektedir.

3:  $Fit_s[i] \leftarrow FitnessFunction(\vec{P}_s(i, :))$ , Her bireyin uygunluk değerini hesapla.

4: **end for**

---

---

```

5:  $Fit_{s_{best}} \leftarrow Best(Fit_s[])$ ,  $Best$  fonksiyonu kullanılarak en iyi uygunluk değeri hesaplanır.
6: return  $Fit_{s_{best}}$ 
7: bitir

```

---

```

1: başla  $Best(Fit_s[])$  prosedürü
2:  $Best \leftarrow Fit_s[0]$ 
3: for  $i \leftarrow 1 : n$  do
4:   if ( $Fit_s[i] < Best$ ) then
5:      $Best \leftarrow Fit_s[i]$ 
6:   end if
7: end for
8: return  $Best$ , En iyi uygunluk değeri döndürülür.
9: bitir

```

---

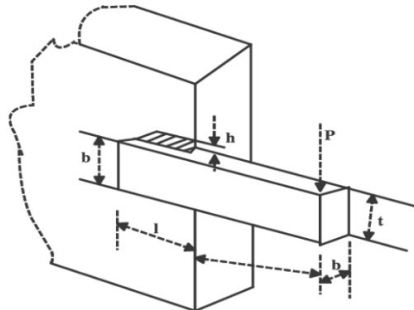
### 3. Deneysel Sonuçlar

Bu bölümde, MOA'nın etkinliğini test etmek için gerçek hayattaki mühendislik tasarım problemlerine uygulanmıştır. Gerçekleştirilen tüm uygulamalar Matlab R2019b yazılım ortamında yapılmıştır. MOA, kaynaklı kiriş, germe/sıkıştırma yayı, basınçlı kap, hız düşürücü ve üç çubuklu kafes problemi olmak üzere iyi bilinen beş adet kısıtlı mühendislik tasarım problemine uygulanmıştır. Ele alınan tüm problemler, farklı amaç fonksiyonlarına, kısıtlamalarına ve karar değişkenlerine sahiptir. Bu problemler, gerçek hayat problemlerinde MOA geçerliliğini ve etkinliğini göstermek için seçilmiştir. MOA'nın etkinliğini göstermek için elde edilen sonuçlar birçok farklı güncel metasezgisel algoritma ile karşılaştırılmıştır.

#### 3.1. Kaynaklı Kiriş Tasarım Problemi

Kaynaklı tasarım problemi ilk kez Coello tarafından önerilmiştir [18]. Bu tasarım probleminin temel amacı, kesme gerilimi ( $t$ ), kirişteki eğilme gerilimi ( $\sigma$ ), çubuktaki burkulma yükü ( $P_c$ ), kirişin uç sapması ( $\delta$ ) ve yan kısıtlamalara bağlı olarak kaynaklı kirişin üretim maliyetini minimize etmektir. Şekil 6'da görseli verilen kaynaklı kiriş tasarım probleminin aşağıda tanımlanan dört adet tasarım değişkeni bulunmaktadır;

- $h(x_1)$ : kaynak kalınlığı
- $l(x_2)$ : kaynak bağlantı uzunluğu
- $t(x_3)$ : kirişin genişliği
- $b(x_4)$ : kiriş kalınlığı



Şekil 6. Kaynaklı kiriş tasarım problemi

$x = [x_1, x_2, x_3, x_4] = [h, l, t, b]$  değişkenlerini göz önüne alınarak kaynaklı kiriş probleminin matematiksel olarak denklem (11) ile modellenabilir:

$$\begin{aligned}
f(x) &= 1.10471x_1^1x_2 + 0.04811x_3x_4(14 + x_2) \\
g_1(x) &= \tau(x) + \tau_{maks} \leq 0, \\
g_2(x) &= \sigma(x) + \sigma_{maks} \leq 0, \\
g_3(x) &= \delta(x) + \delta_{maks} \leq 0, \\
g_4(x) &= x_1 - x_4 \leq 0, \\
g_5(x) &= P - P_c(x) \leq 0, \\
g_6(x) &= 0.125 - x_1 \leq 0, \\
g_7(x) &= 0.10471x_1^2 + 0.04811x_3x_4(14 + x_2) - 5 \leq 0,
\end{aligned} \tag{11}$$

Burada;

$$\tau(x) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x^2}{2R} + (\tau'')^2}, \quad \tau' = \frac{P}{\sqrt{2}x_1x_2}, \quad \tau'' = \frac{MR}{J} \tag{12}$$

$$M = P(L + \frac{x_2}{2}), \quad R = \sqrt{\frac{x_2^2}{4} + (\frac{x_1+x_3}{2})^2}, \tag{13}$$

$$J = 2 \left\{ \sqrt{2}x_1x_2 \left[ \frac{x_2^3}{4} + (\frac{x_1+x_3}{2})^2 \right] \right\}$$

$$\sigma(x) = \frac{6PL}{x_4x_3^2}, \quad \delta(x) = \frac{4PL^3}{Ex_4x_3^2}, \quad P_c(x) = \frac{4.013E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2} \left\{ 1 - \frac{x_3}{2L} \sqrt{\frac{E}{4G}} \right\} \tag{14}$$

Burada;  $P = 6000lb$ ,  $L = 14in$ ,  $E = 30 \times 10^6psi$ ,  $G = 12 \times 10^6psi$ ,  $\tau_{maks} = 13,600psi$ ,  $\sigma(x) = 30,000psi$ ,  $\delta(x) = 0.25in$ . Değişkenlerin aralıkları sırasıyla;  $0.1 \leq x_1 \leq 2$ ,  $0.1 \leq x_2 \leq 10$ ,  $0.1 \leq x_3 \leq 10$ ,  $0.1 \leq x_4 \leq 2$ 'dir.

Bu problem literatürde birçok farklı araştırmacı tarafından farklı metasezgisel optimizasyon yöntemleri ile çözülmüştür. MOA, kaynaklı kiriş tasarım problemine uygulanırken popülasyon sayısı 30 ve maksimum iterasyon sayısı 1000 olarak belirlenerek 30 kez çalıştırılmıştır. Elde edilen sonuçlar Genetik Algoritma (Genetic Algorithm-GA3) [18], Karga Arama Algoritması (Crow Search Algorithm-CSA) [19], Eş-Evrimsel Parçacık Sürü Optimizasyonu (Co-evolutionary Particle Swarm Optimization-CPSO) [20], Mayın Patlatma Algoritması (Mine Blast Algorithm-MBA) [21], Hibrit Parçacık Sürü Optimizasyonu (Hybrid Particle Swarm Optimization-HPSO) [22], Eş-Evrimsel Diferansiyel Evrim (Co-Evolutionary Differential Evolution-CDE) [23], Yapay Arı Kolonisi (Artificial Bee Colony-ABC) [24], Sinüs Cosinüs Algoritması (Sine Cosine Algorithm-SCA) [25], Çoklu Evren Optimizasyonu (Multi Verse Optimization-MVO) [26] algoritmaları ile karşılaştırılmıştır.

**Çizelge 1.** Kaynaklı kiriş tasarım problemi için sonuçların karşılaştırılması.

Yöntem	Karar Değişkenleri				$f_{maliyet}$
	$x_1$	$x_2$	$x_3$	$x_4$	
GA3	0.20880	3.42050	8.99750	0.21000	1.74830
CSA	0.20570	3.47050	9.03660	0.20570	1.72480
CPSO	0.20240	3.54420	9.04820	0.20570	1.72800
MBA	0.20570	3.47049	9.03663	0.20573	1.72480
HPSO	0.20573	3.47049	9.03662	0.20573	1.72485
CDE	0.20313	3.54299	9.03349	0.20618	1.73346
ABC	0.20573	3.47049	9.03662	0.20573	1.72485
SCA	0.20469	3.53629	9.00429	0.21002	1.75917
MVO	0.20561	3.47210	9.04093	0.20571	1.72547
<b>MOA</b>	<b>0.21760</b>	<b>3.23450</b>	<b>9.03941</b>	<b>0.20590</b>	<b>1.71500</b>

Çizelge 1'de MOA ve diğer çözüm yöntemlerinin kaynaklı kiriş tasarım problemi için elde ettikleri en iyi maliyet değeri ve ilgili karar değişkenlerinin değerleri verilmiştir. Elde edilen sonuçlar



incelendiğinde, MOA'nın  $x_{1-4} = (0.2176, 3.2345, 9.03941, 0.2059)$  için 1.715 maliyet fonksiyonu değeri ile diğer yöntemlere göre oldukça iyi bir sonuç verdiği görülmektedir.

**Çizelge 2.** Kaynaklı kiriş tasarım problemi için istatistiksel sonuçlarının karşılaştırılması.

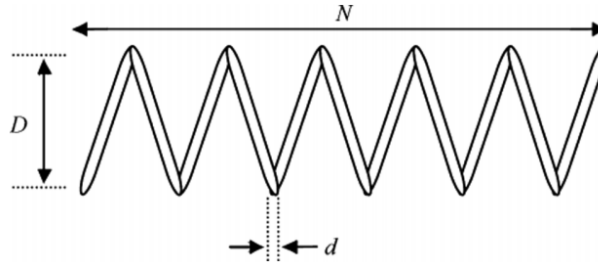
Yöntem	En Kötü	Ortalama	En İyi	Standart S.
GA3	1.78583	1.77197	1.74830	1.12E-2
CSA	1.72480	1.72480	1.72480	1.19450
CPSO	1.78210	1.74880	1.72800	1.29E-2
MBA	1.72485	1.72485	1.72485	6.94E-1
HPSO	1.81429	1.74904	1.72485	4.01E-0
CDE	1.82410	1.76816	1.73346	2.22E-0
ABC	-	1.74191	1.72485	3.1E-02
SCA	1.87341	1.81766	1.75917	2.28E-0
MVO	3.04823	2.23031	1.82039	0.32452
<b>MOA</b>	<b>1.72495</b>	<b>1.72181</b>	<b>1.71500</b>	2.48E-0

Çizelge 2, MOA ve diğer algoritmalar tarafından elde edilen istatistiksel sonuçlarının karşılaştırmasını göstermektedir. Çizelgede görüldüğü gibi, en iyi, en kötü ve ortalama maliyetin elde edilmesi açısından MOA, diğer çözüm yöntemlerinden daha iyi performans göstermiştir.

### 3.2. Germe/Sıkıştırma Yay Tasarım Problemi

Germe/sıkıştırma yay tasarım problemi, Arora [27] tarafından tanımlanan ve minimum ağırlığa sahip bir yay tasarımı oluşturmayı amaçlayan bir problemdir. Şekil 7'de şematik gösterimi verilen bu minimizasyon problemi, kesme gerilimi, dalgalanma frekansı ve minimum sapma olmak üzere belirli kısıtlamalara sahiptir. Germe/sıkıştırma yayı probleminin üç adet karar değişkeni bulunmaktadır.

- $d(x_1)$ : Tel çapı
- $D(x_2)$ : Ortalama bobin çapı
- $N(x_3)$ : Aktif bobin sayısı



**Şekil 7.** Germe/sıkıştırma yay tasarım problemi

$x = [x_1, x_2, x_3] = [d, D, N]$  değişkenlerini göz önüne alınarak germe/sıkıştırma yay tasarım probleminin matematiksel gösterimi denklem (15) ile modellenabilir:

$$\begin{aligned}
 f(x) &= (x_3 + 2)x_2x_1^2 \\
 g_1(x) &= 1 - \frac{x_2^3x_3}{71785x_1^4} \leq 0, \\
 g_2(x) &= \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} \leq 0, \\
 g_3(x) &= 1 - \frac{140.45x_3}{x_2^2x_3} \leq 0, \\
 g_4(x) &= \frac{x_1 + x_2}{1.5} \leq 0,
 \end{aligned} \tag{15}$$

Değişkenlerin aralıkları sırasıyla;  $0.05 \leq x_1 \leq 2$ ,  $0.25 \leq x_2 \leq 1.3$ ,  $2 \leq x_3 \leq 15$ 'dir.

MOA, germe/sıkıştırma yay tasarım problemine uygulanırken popülasyon sayısı 30 ve maksimum iterasyon sayısı 1000 olarak belirlenerek 30 kez çalıştırılmıştır. Germe/sıkıştırma yay tasarım problemi için MOA, karar değişkenleri ve bunlara bağlı olarak maliyet fonksiyonun değeri bakımından Çizelge 3'te verildiği gibi literatürdeki farklı çözüm yöntemleri ile karşılaştırılmıştır. Çizelge 3 incelendiğinde, MOA'nın  $x_{1-3} = (0.05092, 0.34125, 12.10954)$  değerleri için 0.01253 maliyet fonksiyonu değeri ile diğer yöntemlere göre oldukça iyi bir sonuç verdiği görülmektedir. Çizelge 4, germe/sıkıştırma yay tasarımı problemi için MOA ile elde edilen uygunluk fonksiyonu değerlerinin istatistiksel olarak farklı optimizasyon yöntemleri ile karşılaştırılmasını göstermektedir. Çizelge 4'e göre, MOA'nın, en kötü, ortalama ve en iyi değerler açısından diğer algoritmalara göre daha başarılı olduğu gözlemlenmektedir.

**Çizelge 3.** Germe/sıkıştırma yay tasarım problemi için sonuçların karşılaştırılması.

Yöntem	Karar Değişkenleri			$f_{maliyet}$
	$x_1$	$x_2$	$x_3$	
GA3	0.05148	0.35166	11.63220	0.01270
CSA	0.05169	0.35672	11.28901	0.01266
CPSO	0.05179	0.35764	11.24454	0.01267
MBA	0.05165	0.35594	11.34466	0.01266
HPSO	0.05171	0.35713	11.26508	0.01266
CDE	0.05161	0.35471	11.41083	0.01267
ABC	0.05175	0.35818	11.20376	0.01266
SCA	0.05078	0.33478	12.72269	0.01271
MVO	0.05000	0.31596	14.22623	0.01282
<b>MOA</b>	<b>0.05092</b>	<b>0.34125</b>	<b>12.10954</b>	<b>0.01253</b>

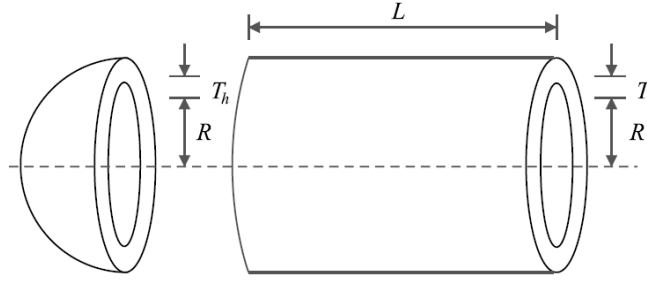
**Çizelge 4.** Germe/sıkıştırma yay tasarım problemi istatistiksel sonuçlarının karşılaştırılması.

Yöntem	En Kötü	Ortalama	En İyi	Standart S.
GA3	0.01282	0.01277	0.01270	3.94E-5
CSA	0.01267	0.01266	0.01266	1.35707
CPSO	0.01292	0.01273	0.01267	5.20E-4
MBA	0.01290	0.01271	0.01266	6.3E-5
HPSO	0.01272	0.01271	0.01266	1.58E-5
CDE	0.01279	0.01270	0.01267	2.70E-5
ABC	-	0.01271	0.01266	0.01281
SCA	0.01299	0.01284	0.01271	0.00008
MVO	0.01784	0.01446	0.01281	0.00162
<b>MOA</b>	<b>0.01266</b>	<b>0.01257</b>	<b>0.01253</b>	2.24587

### 3.3. Basınçlı Kap Tasarım Problemi

Kannan ve Kramer tarafından önerilen basınçlı kap tasarımı probleminde amaç, malzeme, şekillendirme ve kaynak maliyetini içeren toplam maliyeti en aza indirmektir [28]. Şekil 8, her iki ucu yarım küre başlıklarla kapatılan basınçlı kabın şematik görünümü göstermektedir. Bu tasarım probleminin dört adet karar değişkeni bulunmaktadır:

- $T_s(x_1)$ : Kabuğun kalınlığı
- $T_h(x_2)$ : Kafanın kalınlığı
- $R(x_3)$ : İç yarıçap
- $L(x_4)$ : Kafa göz alınmadan silindirik bölümün uzunluğu



Şekil 8. Basıncılı kap tasarım problemi

$x = [x_1, x_2, x_3, x_4] = [T_s, T_h, R, L]$  değişkenlerini göz önüne alınarak germe/sıkıştırma yay tasarım probleminin matematiksel gösterimi denklem (16) ile modellenebilir:

$$\begin{aligned}
 f(x) &= 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \\
 g_1(x) &= -x_1 + 0.0193x, \\
 g_2(x) &= -x_2 + 0.00954x_3 \leq 0, \\
 g_3(x) &= -\pi x_3^2x_4 - \left(\frac{4}{3}\right)\pi x_3^3 + 1296000 \leq 0, \\
 g_4(x) &= x_4 - 240 \leq 0,
 \end{aligned} \tag{16}$$

Değişkenlerin aralıkları sırasıyla;  $0.0625 \leq x_1, x_2 \leq 99 \times 0.0625$ ,  $10 \leq x_3, x_4 \leq 200$ 'dür.

Çizelge 5. Basıncılı kap tasarım problemi için sonuçların karşılaştırılması.

Yöntem	Karar Değişkenleri				$f_{\text{maliyet}}$
	$x_1$	$x_2$	$x_3$	$x_4$	
GA3	0.81250	0.43750	40.32390	200.00000	6288.74450
CSA	0.81250	0.43750	42.09845	176.63659	6059.71436
CPSO	0.81250	0.43750	42.09127	176.74650	6061.07770
MBA	0.78020	0.38560	40.42920	198.49640	5889.32160
HPSO	0.81250	0.43750	42.09840	176.74650	6061.07770
CDE	0.81250	0.43750	42.09841	176.63769	6059.73400
ABC	0.81250	0.43750	42.09845	176.63659	6059.71434
SCA	0.81758	0.41793	41.74939	183.57270	6137.37240
MVO	0.84572	0.41856	43.81627	156.38164	6011.51480
<b>MOA</b>	<b>0.71610</b>	<b>0.35840</b>	<b>40.34700</b>	<b>200.00000</b>	<b>5430.06150</b>

Çizelge 6. Basıncılı kap tasarım problemi için istatistiksel sonuçlarının karşılaştırılması

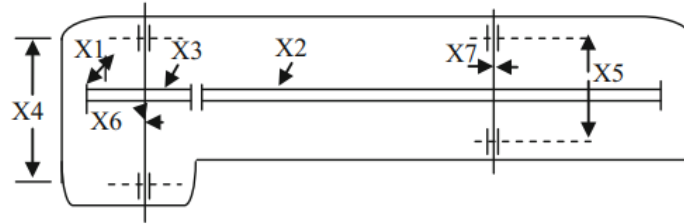
Yöntem	En Kötü	Ortalama	En İyi	Standart S.
GA3	6308.49700	6293.8432 0	6288.74450	7.41330
CSA	7332.84100	6342.4991 0	6059.7143 0	384.94540
CPSO	6363.80410	6147.13320	6061.07770	86.45450
MBA	6392.50620	6200.64765	5889.32160	160.34000
HPSO	6288.67700	6099.93230	6059.71430	86.20220
CDE	6371.04550	6085.23030	6059.73400	43.01300
ABC	-	6245.30814	6059.71474	205
SCA	6512.35410	6326.76060	6137.37240	126.60900
MVO	7250.91700	6477.30500	6011.51480	327.00700
<b>MOA</b>	<b>5927.51930</b>	<b>5589.80825</b>	<b>5430.06150</b>	156.00904

MOA, basınçlı kap tasarım problemine uygulanırken popülasyon sayısı 30 ve maksimum iterasyon sayısı 1000 olarak belirlenerek 30 kez çalıştırılmıştır. Çizelge 5'te, MOA ve literatürde basınçlı kap tasarım problemine uygulanan farklı çözüm yöntemlerinden elde edilen sonuçlar, karar değişkenleri ve bunlara bağlı olarak maliyet fonksiyonun değeri bakımından karşılaştırmalı olarak sunulmuştur. Çizelge 5 incelendiğinde, MOA'nın  $x_{1-4} = (0.7161, 0.3584, 40.3470, 200.0000)$  değerleri için 5430.0615 maliyet fonksiyonu değeri ile diğer yöntemlere göre oldukça iyi bir sonuç verdiği görülmektedir. Çizelge 6, basınçlı kap tasarım problemi için MOA ile elde edilen uygunluk fonksiyonu değerlerinin istatistiksel olarak farklı optimizasyon yöntemleri ile karşılaştırılmasını göstermektedir. Çizelge 6 incelendiğinde, MOA'nın, en kötü, ortalama ve en iyi değerler açısından diğer algoritmalara göre daha başarılı olduğu gözlemlenmektedir.

### 3.4. Hız Düşürücü Tasarım Problemi

Hız düşürücü tasarım problemi şekil 9'da görüldüğü gibi yedi tasarım değişkenine sahip olduğundan dolayı zorlu bir problemdir [29]. Bu problemin temel amacı, hız düşürücünün ağırlığını 11 farklı kısıta bağlı olarak en aza indirmektir [30].

- $b(x_1)$ : Yüz genişliği
- $m(x_2)$ : Diş modülü
- $z(x_3)$ : Pinyondaki diş sayısı
- $l_1(x_4)$ : Yataklar arasındaki ilk milin uzunluğu
- $l_2(x_5)$ : Yataklar arasındaki ikinci milin uzunluğu
- $d_1(x_6)$ : Birinci milin çapı
- $d_2(x_7)$ : İkinci milin çapı



Şekil 9. Hız düşürücü tasarım problemi

$x = [x_1, x_2, x_3, x_4, x_5, x_6, x_7] = [b, m, z, l_1, l_2, d_1, d_2]$  değişkenlerini göz önüne alınarak hız düşürücü probleminin matematiksel gösterimi denklem (17) ile modellenenabilir:

$$\begin{aligned}
 f(x) &= 0.785x_1x_2^2(3.333x_3^2 + 14.9334x_3 - 42.0934) \\
 &\quad - 1.508x_1(x_6^2 + x_7^2) + 7.4777x_1(x_6^3 + x_7^3) + 1.508x_1(x_4x_6^2 + x_5x_7^2) \\
 g_1(x) &= \frac{27}{x_1x_2^2x_3} - 1 \leq 0, \\
 g_2(x) &= \frac{397.5}{x_1x_2x_3^2} - 1 \leq 0, \\
 g_3(x) &= \frac{1.93x_4^3}{x_1x_3x_6^4} - 1 \leq 0, \\
 g_4(x) &= \frac{1.93x_5^3}{x_1x_3x_7^4} - 1 \leq 0, \\
 g_5(x) &= \frac{1}{110x_6^3} \sqrt{\left(\frac{745x_4}{x_2x_3}\right)^2 + 16.9 \times 10^6} - 1 \leq 0, \\
 g_6(x) &= \frac{1}{85x_7^3} \sqrt{\left(\frac{745x_5}{x_2x_3}\right)^2 + 157.5 \times 10^6} - 1 \leq 0,
 \end{aligned} \tag{17}$$

$$g_7(x) = \frac{x_2 x_3}{40} - 1 \leq 0,$$

$$g_8(x) = \frac{5x_2}{x_1} - 1 \leq 0,$$

$$g_9(x) = \frac{x_1}{12x_2} - 1 \leq 0,$$

$$g_{10}(x) = \frac{1.5x_6 + 1.9}{x_4} - 1 \leq 0,$$

$$g_{11}(x) = \frac{1.1x_7 + 1.9}{x_5} - 1 \leq 0,$$

Değişkenlerin aralıkları sırasıyla;  $2.6 \leq x_1 \leq 3.6$ ,  $0.7 \leq x_2 \leq 0.8$ ,  $17 \leq x_3 \leq 28$ ,  $7.3 \leq x_4 \leq 8.3$ ,  $7.8 \leq x_5 \leq 8.3$ ,  $2.9 \leq x_6 \leq 3.9$ ,  $5.0 \leq x_7 \leq 5.5$ 'dir.

MOA, hız düşürücü problemine uygulanırken popülasyon sayısı 30 ve maksimum iterasyon sayısı 1000 olarak belirlenerek 30 kez çalıştırılmıştır. Literatürde hız düşürücü problemine çözüm bulmak için kullanılan birçok farklı yöntem bulunmaktadır. Bu çalışmada, MOA hız düşürücü problemi için Modifiyeli Diferansiyel Evrim (Modified Differential Evolution-MDE) [31], Toplum ve Medeniyet (Society and Civilization SC) [32], Su Döngüsü Algoritması (Water Cycle Algorithm-WCA) [33], Hibrit Evrimsel Algoritması ve Adaptif Kısıtlama İşleme (Hybrid Evolutionary Algorithm and Adaptive Constraint-Handling Technique-HEAA [34], MBA, ABC, SCA ve MVO olmak üzere sekiz farklı yöntem ile karşılaştırılmıştır. Bu yöntemler ile elde edilen sonuçların, karar değişkenlerine ve bunlara bağlı olarak elde edilen maliyet fonksiyonu değerleri çizelge 7'de verilmiştir.

**Çizelge 7.** Hız düşürücü tasarım problemi için sonuçların karşılaştırılması.

Yöntem	Karar Değişkenleri							$f_{\text{maliyet}}$
	X1	X2	X3	X4	X5	X6	X7	
MDE	3.49999	0.69999	17	7.30000	7.80000	3.35021	5.28668	2996.34809
SC	3.50001	0.70000	17	7.32760	7.71532	3.35027	5.28665	2994.74424
WCA	3.50000	0.70000	17	7.30000	7.71532	3.35021	5.28665	2994.47107
HEAA	3.50002	0.70000	17	7.30043	7.71538	3.35023	5.28666	2994.49911
MBA	3.50000	0.70000	17	7.30003	7.71577	3.35022	5.28665	2994.48245
ABC	3.49999	0.70000	17	7.30000	7.80000	3.35021	5.28780	2997.05841
SCA	3.50875	0.70000	17	7.30000	7.80000	3.46102	5.28921	3030.56300
MVO	3.50850	0.70000	17	7.39284	7.81603	3.35807	5.28678	3002.92800
<b>MOA</b>	<b>3.32680</b>	<b>0.70000</b>	<b>17</b>	<b>7.39580</b>	<b>7.53170</b>	<b>3.31190</b>	<b>5.23040</b>	<b>2901.14200</b>

**Çizelge 8.** Hız düşürücü tasarım problemi için istatistiksel sonuçlarının karşılaştırılması.

Yöntem	En Kötü	Ortalama	En İyi	Standart S.
MDE	2996.39014	2996.36722	2996.35669	0
SC	3009.96474	3001.75826	2994.74424	4.00914
WCA	2994.50558	2994.47439	2994.47107	2994.47106
HEAA	2994.75231	2994.61337	2994.49911	7.0E-02
MBA	2999.65244	2996.76902	2994.48245	1.56
ABC	-	2997.05841	2997.05841	0
SCA	3104.77900	3065.91700	3030.56300	18.07420
MVO	3060.95800	3028.84100	3002.92800	13.01860
<b>MOA</b>	<b>2979.67020</b>	<b>2931.88600</b>	<b>2901.14200</b>	<b>19.23320</b>

Çizelge 7 incelendiğinde, MOA'nın  $x_{1-7} = (3.3268, 0.7, 17, 7.3958, 7.5317, 3.3119, 5.2304)$  değerleri için 2901.1420 maliyet fonksiyonu değeri ile diğer yöntemlere göre oldukça iyi bir sonuç verdiği görülmektedir. Çizelge 8, hız düşürücü problemi için MOA ile elde edilen uygunluk fonksiyonu değerlerinin istatistiksel olarak farklı optimizasyon yöntemleri ile karşılaştırılmasını göstermektedir.

Çizelge 8 incelendiğinde, MOA'nın, en kötü, ortalama ve en iyi değerler açısından diğer algoritmalara göre daha başarılı olduğu gözlemlenmektedir.

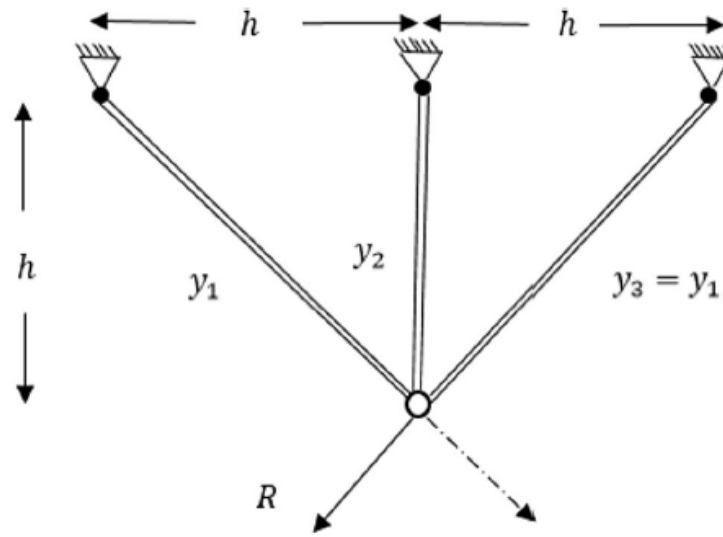
### 3.5. Üç Çubuklu Kafes Tasarım Problemi

Üç çubuklu kafes tasarımı, inşaat mühendisliği alanında yapısal bir optimizasyon problemidir. Nowacki tarafından tanıtılan bu problemin amacı çapraz kesit alanlarını ( $x_1$  ve  $x_2$ ) ayarlayarak kafes elemanlarının her biri üzerindeki gerilimi ( $\sigma$ ) göz önüne alarak üç çubuklu kafesin hacmini en aza indirmektir [35]. Şekil 10'da şematik gösterimi verilen üç çubuklu kafes probleminin matematiksel gösterimi denklem (18)'de verilmiştir.

$$\begin{aligned} f(x) &= L \times (2\sqrt{2x_1} + x_2) \\ g_1(x) &= \frac{\sqrt{2x_1+x_2}}{\sqrt{2x_1^2+2x_1x_2}} P - \sigma \leq 0, \\ g_2(x) &= \frac{x_2}{\sqrt{2x_1^2+2x_1x_2}} P - \sigma \leq 0, \\ g_3(x) &= \frac{1}{\sqrt{2x_2+x_1}} P - \sigma \leq 0, \end{aligned} \quad (18)$$

Değişkenlerin aralıkları sırasıyla;  $0 \leq x_i \leq 1, i = 1,2$ 'dir.

Burada;  $l = 100$  cm,  $P = 2$  kN/cm<sup>2</sup>,  $\sigma =$  kN/cm<sup>2</sup>'dir.



Şekil 10. Üç çubuklu kafes tasarım problemi

MOA, üç çubuklu kafes problemine uygulanırken popülasyon sayısı 30 ve maksimum iterasyon sayısı 1000 olarak belirlenerek 30 kez çalıştırılmıştır. Bu probleme çözüm bulabilmek için literatürde birçok farklı çalışma yapılmıştır. Bu çalışmada, MOA üç çubuklu kafes tasarımı problemi için SC, CSA, WCA, MBA, HEAA, SCA ve MVO olmak üzere yedi farklı yöntem ile karşılaştırılmıştır. Bu yöntemler ile elde edilen sonuçların, karar değişkenlerine ve bunlara bağlı olarak elde edilen maliyet fonksiyonu değerleri çizelge 9'da verilmiştir. Çizelge 9 incelendiğinde, MOA'nın  $x_{1-2} = (0.7775, 0.4014)$  değerleri için 260.7186 maliyet fonksiyonu değeri ile diğer yöntemlere göre oldukça iyi bir sonuç verdiği görülmektedir. Çizelge 10, üç çubuklu kafes problemi için MOA ile elde edilen uygunluk fonksiyonu değerlerinin istatistiksel olarak farklı optimizasyon yöntemleri ile karşılaştırılmasını göstermektedir. Çizelge 10 incelendiğinde, MOA'nın, en kötü, ortalama ve en iyi değerler açısından diğer algoritmalara göre daha başarılı olduğu gözlemlenmektedir.

**Çizelge 9.** Üç çubuklu kafes tasarım problemi için sonuçların karşılaştırılması.

Yöntem	Karar Değişkenleri		
	$x_1$	$x_2$	$f_{maliyet}$
SC	0.78862	0.40840	263.89585
CSA	0.78867	0.40824	263.89584
WCA	0.78865	0.40831	263.89584
MBA	0.78856	0.40855	263.89585
HEAA	0.78868	0.40823	263.89584
SCA	0.78906	0.40716	263.89838
MVO	0.78899	0.40735	263.89594
<b>MOA</b>	<b>0.77750</b>	<b>0.40140</b>	<b>260.71860</b>

**Çizelge 10.** Üç çubuklu kafes tasarım problemi istatistiksel sonuçlarının karşılaştırılması.

Yöntem	En Kötü	Ortalama	En İyi	Standart S.
SC	263.96975	263.90330	263.89580	0.01258
CSA	263.89584	263.89584	263.89584	1.01225
WCA	263.89620	263.89590	263.89584	8.71E-5
MBA	2999.65244	2996.76902	2994.48245	1.56
HEAA	263.89609	263.89586	263.89584	4.9E-05
SCA	263.99511	263.93564	263.89838	0.02884
MVO	263.89709	263.89614	263.89594	0.00025
<b>MOA</b>	<b>260.76120</b>	<b>260.72920</b>	<b>260.71860</b>	0.01158

#### 4. Sonuçlar

Birçok araştırmacı doğadaki fiziksel, biyolojik, kimyasal olaylar ya da doğadaki canlıların yaşam biçimlerinden ilham alarak birçok farklı metasezgisel optimizasyon algoritması geliştirmiştir. Doğadaki süreçler, kavramlar, kurallar ve olaylar etkili ve verimli arama ve optimizasyon yöntemlerinin geliştirilmesi için modellenmiştir.

Bu çalışmada, doğadaki martıların göç ve saldırı davranışlarından ilham alarak önerilen biyoloji tabanlı yeni bir yöntem olan MOA detaylı olarak incelenmiş ve tanıtılmıştır. Martıların bu davranışları belirli bir arama uzayında keşif ve sömürüyü vurgulamak için matematiksel olarak modellenmiştir. MOA'nın etkinliğini test etmek için kaynaklı giriş, germe/sıkıştırma yay, basınçlı kap, hız düşürücü ve üç çubuklu kafes olmak üzere amaç fonksiyonları, kısıtları ve karar değişkenleri farklı beş kısıtlı mühendislik tasarım problemine uygulanmış ve elde edilen sonuçlar literatürdeki farklı optimizasyon yöntemleri ile kıyaslanarak performans analizi yapılmıştır. Performans kıyaslamaları için algoritmaların temel versiyonları kullanılmıştır. Elde edilen sonuçlar, MOA'nın mevcut diğer tekniklerden daha iyi optimizasyon sağladığını ortaya koymaktadır. Bununla birlikte, istatistiksel karşılaştırmaların sonuçları, MOA'nın çalışılan kısıtlama problemlerinin çoğunda daha iyi ve oldukça rekabetçi performans elde ettiğini göstermektedir.

MOA algoritması çok yeni bir algoritmadır ve farklı yönlerden geliştirilebilecek bir yöntemdir. Elde edilen umut verici sonuçlar göz önüne alındığında MOA, farklı metasezgisel yöntemler veya diğer global optimizasyon algoritmaları ile birleştirilebilir ve farklı karmaşık arama ve optimizasyon problemleri için verimli sonuçlar elde etmek amacıyla hibrit versiyonları önerilebilir.

**Kaynaklar**

- [1] Altunbey F, Alataş B. Review of social-based artificial intelligence optimization algorithms for social network analysis. *International Journal of Pure and Applied Sciences* 2015; 1(1): 33-52.
- [2] Erol OK, Eksin I. A new optimization method: big bang–big crunch. *Advances in Engineering Software* 2006; 37(2): 106-111.
- [3] Hatamlou A. Black hole: A new heuristic optimization approach for data clustering. *Information Sciences* 2013; 222: 175-184.
- [4] Formato RA. Central force optimization: A new deterministic gradient like optimization metaheuristic. *Opsearch* 2009; 46(1): 25–51.
- [5] Rashedi E, Nezamabadi-Pour H, Saryazdi S. GSA: a gravitational search algorithm. *Information Sciences* 2009; 179(13): 2232-2248.
- [6] Eskandar H, Sadollah A, Bahreininejad A, Hamdi M. Water cycle algorithm–A novel metaheuristic optimization method for solving constrained engineering optimization problems. *Computers & Structures* 2012; 110: 151-166.
- [7] Kennedy J, Eberhart R. Particle swarm optimization. Paper presented at: ICNN'95-IEEE International Conference on Neural Networks, 1995.
- [8] Dorigo M, Birattari M, Stutzle T. Ant colony optimization. *IEEE Computational Intelligence Magazine* 2006; 1(4): 28-39.
- [9] Yang XS. A new metaheuristic bat-inspired algorithm. *Nature Inspired Cooperative Strategies For Optimization (NICSO 2010)*, Springer, Berlin, Heidelberg, 2010: 65-74.
- [10] Yang XS. Firefly algorithm, stochastic test functions and design optimisation. *International Journal of Bio-Inspired Computation* 2010; 2(2):78–84.
- [11] Karaboga D, Basturk, BA powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of global optimization* 2007; 39(3): 459-471.
- [12] Dhiman G, Singh KK, Soni M, Nagar A, Dehghani M, Slowik A, Cengiz K. MOSOA: a new multi-objective seagull optimization algorithm. *Expert Systems with Applications* 2021; 167, 114150.
- [13] Dhiman G, Kumar V. (2019). Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems. *Knowledge-Based Systems* 2019; 165: 169-196.
- [14] Xu L, Mo Y, Lu Y, Li J. Improved Seagull Optimization Algorithm Combined with an Unequal Division Method to Solve Dynamic Optimization Problems. *Processes* 2021; 9(6): 1037.
- [15] Del Hoyo J, Elliott A, Sargatal J. *Handbook of the Birds of the World*. Barcelona: Lynx edicions 1992: 1(8),
- [16] Macdonald SM, Mason CF. Predation of migrant birds by gulls. *British Birds* 1973; 66: 361-363.
- [17] Dhiman G, Kumar V. Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems. *Knowledge-Based Systems* 2019; 165: 169-196.
- [18] Coello CAC. Use of a self-adaptive penalty approach for engineering optimization problems, *Computers in Industry* 2000; 41: 113–127.
- [19] Askarzadeh A. A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm. *Computers & Structures* 2016; 169: 1-12.
- [20] He Q, Wang L. An effective co-evolutionary particle swarm optimization for constrained engineering design problems. *Engineering applications of artificial intelligence* 2007; 20(1): 89-99.
- [21] Sadollah A, Bahreininejad A, Eskandar H, Hamdi M. Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems. *Applied Soft Computing* 2013; 13(5): 2592-2612.
- [22] He Q, Wang L. A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization. *Applied Mathematics and Computation* 2007; 86: 1407–1422.



- [23] Huang FZ, Wang L, He Q. An effective co-evolutionary differential evolution for constrained optimization. *Applied Mathematics and computation* 2007; 186(1): 340-356.
- [24] Akay B, Karaboga, D. Artificial bee colony algorithm for large-scale problems and engineering design optimization. *Journal of intelligent manufacturing* 2012; 23(4): 1001-1014.
- [25] Mirjalili S. SCA: a sine cosine algorithm for solving optimization problems. *Knowledge-based systems* 2016; 96: 120-133.
- [26] Mirjalili S, Mirjalili SM, Hatamlou A. Multi-verse optimizer: a nature-inspired algorithm for global optimization. *Neural Computing and Applications* 2016; 27(2): 495-513.
- [27] Arora JS. *Introduction to Optimum Design*. McGraw-Hill, New York, 1989.
- [28] Kannan BK, Kramer SN. An augmented lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. *Journal of Mechanical Design* 1994; 16: 405–411.
- [29] Gandomi AH, Yang XS. (2011). Benchmark problems in structural optimization. In *Computational optimization, methods and algorithms*, Springer, Berlin, Heidelberg; 2011: 259-281.
- [30] Mezura-Montes E, Coello CAC. (2005, November). Useful infeasible solutions in engineering optimization with evolutionary algorithms. In *Mexican international conference on artificial intelligence*, Springer, Berlin, Heidelberg 2005: 652-662.
- [31] Mezura-Montes E, Velázquez-Reyes J, Coello CC. Modified differential evolution for constrained optimization. Paper presented at: 2006 IEEE International Conference on Evolutionary Computation 2006: 25-32.
- [32] Ray T, Liew KM. Society and civilization: an optimization algorithm based on the simulation of social behavior. *IEEE Transactions on Evolutionary Computation* 2003; 7(4): 386-396.
- [33] Eskandar H, Sadollah A, Bahreininejad A, Hamdi M. Water cycle algorithm—A novel metaheuristic optimization method for solving constrained engineering optimization problems. *Computers & Structures* 2012; 110: 151-166.
- [34] Wang Y, Cai Z, Zhou Y, Fan Z. (2009). Constrained optimization based on hybrid evolutionary algorithm and adaptive constraint-handling technique. *Structural and Multidisciplinary Optimization* 2009; 37(4): 395-413.
- [35] Fujita Y, Lind K, Williams TJ. (1974). Computer applications in the automation of shipyard operation and ship design. Paper presented at: IFIP/IFAC/JSNA joint conference 1973; 2: 28-30.