



# Kısıtlı Mühendislik Problemlerinin Karşılaştırmalı Ağırlık ve Maliyet Optimizasyonu

Aybike Özyüksel Çiftçiöglü<sup>1</sup>

## ÖZ

Mühendislik alanındaki gerçek dünya problemleri genellikle doğrusal olmayan veya kısıtlı tasarım problemleridir. Pek çok nedenden ötürü, bir mühendis yalnızca uygun şekilde çalışan herhangi bir tasarımı değil, en iyi tasarımı elde etmek ister. En iyi tasarımı belirleme sürecine optimizasyon denir. Optimizasyon ile mevcut kısıtlayıcıları sağlayarak belirli bir amaç fonksiyonunu elde edecek şekilde problemin en iyi tasarımı belirlenir. Bu çalışmada çeşitli eşitlik ve eşitsizlik kısıtlamaları olan çekme/basınç yayı, kaynaklı giriş ve basınçlı kap tasarımları olmak üzere üç gerçek dünya mühendislik tasarım problemi optimize edilmeye çalışılmış, tasarım problemlerinin optimum değişkenleri belirlenmiştir. Optimizasyon sürecinde sekiz farklı algoritma kullanılmış, gerçek mühendislik problemlerine ait en iyi tasarımlar oluşturulmaya çalışılmıştır. Optimizasyon algoritmaları, meta-sezgisel algoritmaların alt dallarından olan evrimsel tabanlı, sürü tabanlı, matematik tabanlı ve fizik tabanlı algoritmalarından seçilmiştir. Bunların yanı sıra, algoritmaların sonuçları yakınsama eğrileri ve kutu grafikler yardımıyla birbirleri ile kıyaslanmıştır. Gri kurt algoritması her üç problemde de en başarılı performans gösteren algoritma olmuştur. Bunun yanı sıra, sürü tabanlı, fizik tabanlı ve matematik tabanlı algoritmalar gerçek mühendislik problemlerini optimize etmede diğer algoritmalarından daha iyi sonuç vermiştir.

**Anahtar Kelimeler:** Basınçlı kap tasarımı, çekme- germe yay tasarımı, kaynaklı giriş tasarımı, optimizasyon

## Comparative Weight and Cost Optimization of Constrained Engineering Problems

### ABSTRACT

Real-world problems in engineering are often nonlinear or constrained design problems. For many reasons, an engineer wants to get the best design, not just any that works properly. The process of determining the best design is called optimization. With optimization, the best design of the problem is determined to achieve a specific objective function by providing the current constraints. In this study, three real-world engineering design problems are tried to be optimized, namely tension/compression spring, welded beam, and pressure vessel designs with various equalities and inequality constraints. In the optimization process, eight different algorithms are used, the best designs are created, and the optimum variables of the problems are determined. Optimization algorithms are selected from evolutionary-based, swarm-based, mathematics-based, and physics-based algorithms, which are sub-branches of metaheuristic algorithms. In addition, the results of the algorithms are compared with each other with the help of convergence curves and box graphs. The grey wolf algorithm is the algorithm that showed the most successful performance in all three problems. Besides, swarm-based, physics-based, and math-based algorithms performed better than other algorithms in optimizing real engineering problems.

**Keywords:** Pressure vessel design, tension/compression spring design, welded beam design, optimization

Geliş/Received : 27.07.2021  
Kabul/Accepted : 30.08.2021

<sup>1</sup> Manisa Celal Bayar Üniversitesi, Mühendislik Fakültesi, İnşaat Mühendisliği Bölümü, Manisa  
aybike.ozyuksel@cbu.edu.tr, ORCID: 0000-0003-4424-7622



## EXTENDED ABSTRACT

### Introduction/ Background

Meta-heuristic methods are often inspired by simple concepts such as physical phenomena, animal behavior, evolutionary trends. They are easily adaptable to different types of problems. The optimization process starts with random solutions and they have a superior ability to avoid local optima compared to traditional optimization techniques. Meta-heuristic methods are generally divided into two main classes: single solution-based and population-based. The search process for single solution-based meta-heuristics starts with a candidate solution. This single candidate solution is developed through iterations. However, population-based metaheuristics perform the optimization using a set of solutions (populations). In this case, the search process starts with a random initial population (multiple solutions), and this population is developed throughout iterations. Population-based meta-heuristics have a broader knowledge of the search space by having more than one solution. In addition, they have advantages in terms of avoiding local optima more easily and having a wider research area compared to single solution-based methods. Some of the population-based meta-heuristics are: Swarm Intelligence, Physics-Based, Mathematics Based, Evolutionary based, etc. are algorithms. The main complexity in many real engineering problems comes from the size of the problem, its constraints, and the interaction of variables. In this study, three different real engineering problems with various constraints were optimized using eight different optimization algorithms, namely Gravitational Search Algorithm (GSA), Particle Swarm Optimization (PSO), Biogeography-Based Optimization (BBO), Genetic Algorithm (GA), Differential Evolution (DE), Salp Swarm Algorithm (SSA), Sine Cosine Algorithm (SCA), and Grey Wolf Optimizer (GWO), and optimum variables of the problems were obtained. Besides, the performance comparison of eight algorithms was made by drawing convergence curves and boxplots.

### Objectives/ Research Purpose

The study has two main objectives: First, to obtain the best variable values by optimizing three real engineering problems (welded beam design problem, tension-compression spring problem, pressure vessel design problem) with various equality and inequality constraints, and then to compare the performances of eight different metaheuristic algorithms over these problems using various methods like convergence curves and boxplots.

### Methods/ Methodology

Three real engineering design problems with various equality and inequality constraints are used: welded beam, tension/compression spring, and pressure vessel designs. Each design problem was solved with eight different optimization algorithms (GSA, PSO, BBO, GA, DE, SSA, SCA, and GWO). It has been tried to obtain optimum cost in welded beam and pressure vessel designs, and optimum weight in tension/compression spring design. During the optimization, 30 simulations were made for each algorithm so that the best, worst and average values of each algorithm were compared with each other. For each problem, the performances of the algorithms were compared within themselves with the help of convergence curves and box plots, and the best variable values giving the optimum objective functions were determined.

### Results/ Findings

Optimum results of engineering problems are illustrated in Table 8. According to the table, optimum values found for h (thickness of the weld), l (length of the welded joint), t (width of the beam), and b (thickness of the beam) values, which give the lowest cost in the welded beam design problem, were 0.21, 3.29, 9.04, and 0.21, respectively. The d (wire diameter), D (mean coil diameter), N (number of active coils) variable values giving the optimum weight of the tension-compression spring design problem were 0.05, 0.33, and 13.14, respectively. Optimum Ts (shell thickness), Th (thickness of the head), R (inner radius) and L (length of the cylindrical section of the vessel) variable values giving the optimum cost in the pressure vessel de-

sign problem were 0.74, 0.37, 40.32 and 200, respectively. While GWO was the algorithm that gave the best results in the welded beam and pressure vessel design problems, it was the second-best algorithm by finding a value very close to the PSO algorithm, which found the best value in the tension/compression spring design problem. However, in the same problem, GWO found the most successful result in terms of average and worst (highest) results. The GA, BBO, and DE algorithms were the most unsuccessful algorithms by finding the highest results in all three problems in the best, worst and average values.

**Table 8.** Optimum Results Of Engineering Problems

	Welded Beam Design	Spring Design	Pressure Vessel Design
$x_1$	0.21	0.05	0.74
$x_2$	3.29	0.33	0.37
$x_3$	9.04	13.14	40.32
$x_4$	0.21	-	200.00
$f(x)$	1.701	0.01269	5735.54

### Results/ Findings

Optimum results of engineering problems are illustrated in Table 8. According to the table, optimum values found for  $h$  (thickness of the weld),  $l$  (length of the welded joint),  $t$  (width of the beam), and  $b$  (thickness of the beam) values, which give the lowest cost in the welded beam design problem, were 0.21, 3.29, 9.04, and 0.21, respectively. The  $d$  (wire diameter),  $D$  (mean coil diameter),  $N$  (number of active coils) variable values giving the optimum weight of the tension-compression spring design problem were 0.05, 0.33, and 13.14, respectively. Optimum  $T_s$  (shell thickness),  $T_h$  (thickness of the head),  $R$  (inner radius) and  $L$  (length of the cylindrical section of the vessel) variable values giving the optimum cost in the pressure vessel de



## 1. GİRİŞ

Meta-sezgisel yöntemler genellikle fiziksel olaylar, hayvanların davranışları, evrimsel eğilimler gibi basit kavramlardan ilham alırlar. Farklı problem türlerine kolayca uyarlanabilirler. Optimizasyon süreci rastgele çözümlerle başlar ve geleneksel optimizasyon tekniklerine kıyasla yerel optimumdan kaçınma konusunda üstün yeteneklere sahiptirler. Bununla birlikte, her problem tipi için optimum sonucu veren bir meta-sezgisel yöntem yoktur [1]. Bazı meta-sezgiseller bir problem setinde en iyi sonucu verirken farklı bir problem setinde kötü sonuç verebilir. Her problem tipine göre en başarılı algoritmanın değişmesi, yeni meta-sezgisel algoritmaların oluşturulmasına ya da mevcutların geliştirilmesine imkan tanır. Literatürde optimizasyon algoritmalarının geliştirilmiş (iyileştirilmiş) versiyonları ile ilgili [2–5], algoritmaların birleştirilmesi ile oluşturulan hibrit algoritmalar [4–9] ile ilgili ya da yeni oluşturulan algoritmalarla ilgili çalışmalara rastlamak mümkündür.

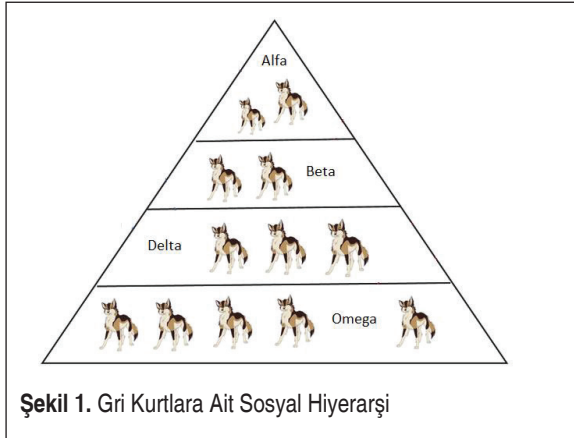
Meta-sezgisel yöntemler genel olarak tek çözüm tabanlı ve popülasyon tabanlı olmak üzere iki ana sınıfa ayrılabilir [10]. Tek çözüm tabanlı meta-sezgisellerin arama süreci bir aday çözümle başlar. Bu tek aday çözüm iterasyonlar boyunca geliştirilir. Bununla birlikte, popülasyon tabanlı meta sezgiseller, optimizasyonu bir dizi çözüm (popülasyon) kullanarak gerçekleştirir. Bu durumda arama süreci rastgele bir başlangıç popülasyonu (birden çok çözüm) ile başlar ve bu popülasyon iterasyonlar boyunca geliştirilir [11]. Popülasyon tabanlı meta-sezgiseller, birden fazla çözüme sahip olmaları ile arama uzayı hakkında daha geniş bilgiye sahip olurlar. Bunun yanı sıra yerel optima- lardan daha rahat kaçınabilmeleri ve tek çözüm tabanlı yöntemlere kıyasla daha geniş araştırma alanına sahip olmaları açısından üstünlüklere sahiptirler. Popülasyon tabanlı meta-sezgisel yöntemlerden bazıları: Sürü Zekalı, Fizik Tabanlı, Matematik Tabanlı, Evrimsel tabanlı vb. algoritmalarıdır. [5] Sürü zekası basit bireylerden ortaya çıkan kolektif grup zekasıdır. Sürü zekalı tekniklerden bazıları: gri kurt (GWO) [10], parçacık sürü (PSO) [12], salp sürüsü (SSA) [13], karınca kolonisi (ACO) [14], güve-alevi (MFO) [15], yarasa (BA) [16] algoritmalarıdır. Fizik tabanlı tekniklerden bazıları: yerçekimi arama (GSA) [17], büyük patlama- büyük çöküş (BBBC) [18] algoritmalarıdır. Matematik tabanlı algoritmalarından bazıları: sinüs kosinüs (SCA) [19], stokastik fraktal arama (SFS) [20] algoritmalarıdır. Evrimsel algoritmalarından bazıları ise: genetik algoritma (GA) [21], diferansiyel evrim (DE) [22], biyocoğrafya tabanlı (BBO) [23] algoritmalarıdır.

Çok amaçlı ve kısıtlayıcılara sahip problemlerin optimize edilmesi optimizasyon alanında hala önemli bir zorluk olarak kabul görür. Literatürde çok amaçlı ve kısıtlayıcılara sahip çeşitli problemlerin optimizasyonu ile ilgili yapılan çalışmalara her geçen gün bir yenisi eklenmektedir [24–27]. Algoritma performansını ölçmenin en önemli yolu algoritmaları benchmark problemleri üzerinden kıyaslamaktır. [28–31] çalışmalarında çeşitli benchmark problemleri üzerinden optimizasyon algoritmalarını kıyas-

lamışlardır. Birçok gerçek mühendislik problemlerindeki ana karmaşıklık ise problemin boyutu, kısıtlayıcılara sahip olması ve değişkenlerin birbirleri ile etkileşiminden kaynaklanır [32–36]. Bu çalışmada GSA, PSO, BBO, GA, DE, SSA, SCA ve GWO olmak üzere sekiz farklı optimizasyon algoritması kullanılarak çeşitli kısıtlayıcılara sahip üç farklı gerçek mühendislik problemi optimize edilmiş ve problemlere ait optimum değişkenler elde edilmiştir. Bunun yanı sıra sekiz algoritmanın performans kıyaslanması, yakınsama eğrileri ve kutu grafikleri çizilerek yapılmıştır.

## 2. GRİ KURT OPTİMİZASYON ALGORİTMASI

Gri kurt optimizasyon algoritması (GWO), Mirjalili [6] tarafından önerilen yeni bir meta sezgisel algoritmadır. GWO, doğadaki gri kurtların sosyal davranışlarını taklit eder. Şekil 1, doğadaki gri kurtların sosyal hiyerarşisi gösterilmektedir.



Şekil 1. Gri Kurtlara Ait Sosyal Hiyerarşi

Alfa, sürüdeki kararların çoğundan sorumlu olan grup lideridir. Beta ve Delta, Alfa'nın karar vermesine ve sürüyü yönetmesine yardımcı olur. Omega kurtları, sürüdeki diğer üç baskın kurda saygı duyar ve gri kurt sürüsünün en düşük seviyededirler. Gri kurtların avlanmaları için avını bulmaları ve çevrelemeleri gerekir. GWO'da gri kurtlar, aşağıdaki denklemleri kullanarak avın etrafındaki konumlarını günceller.

$$\vec{D} = |\vec{C} \cdot \vec{X}_p(t) - \vec{X}(t)| \quad (1)$$

$$\vec{X}(t + 1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D} \quad (2)$$

Burada  $\vec{C}$  ve  $\vec{A}$  katsayılarıdır.  $\vec{X}_p$  avın konumunu,  $\vec{X}$  ise gri kurtların konumunu gösterir. Optimizasyon işlemi sırasında arama yarıçapı optimize edilmelidir, bu amaçla yukarıdaki formülasyonlarda kullanılan iki katsayının değeri aşağıdaki gibi hesaplanır.



$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a} \quad (3)$$

$$\vec{C} = 2 \cdot \vec{r}_2 \quad (4)$$

Yukarıda belirtilen denklemler, gri kurtların av etrafındaki konumlarını güncellemelemlerini sağlar. Avlama işlemini gerçekleştirmek için aşağıdaki denklemler kullanılır.

$$\vec{D}_\alpha = |\vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}|, \vec{D}_\beta = |\vec{C}_2 \cdot \vec{X}_\beta - \vec{X}|, \vec{D}_\delta = |\vec{C}_3 \cdot \vec{X}_\delta - \vec{X}| \quad (5)$$

$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 \cdot \vec{D}_\alpha, \vec{X}_2 = \vec{X}_\beta - \vec{A}_2 \cdot \vec{D}_\beta, \vec{X}_3 = \vec{X}_\delta - \vec{A}_3 \cdot \vec{D}_\delta \quad (6)$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad (7)$$

GWO'ya ait sözde kod Şekil 2'de gösterildiği gibidir.

```

Gri kurt popülasyonunu başlat Xi (i = 1,2,...,n)
a, A ve C'yi başlat
Her avcının amaç fonksiyonu değerini hesapla
Xa=en iyi avcı
Xb=ikinci en iyi avcı
Xd=üçüncü en iyi avcı
while t < iterasyon sayısı
  for her avcı için
    mevcut avcının pozisyonunu güncelle
  end for
  a, A ve C'yi güncelle
  Bütün avcılarının amaç fonksiyonu değerlerini hesapla
  Xa, Xb ve Xd'yi güncelle
  t=t+1;
end while
return Xa

```

Şekil 2. GWO Algoritması Sözde Kodu

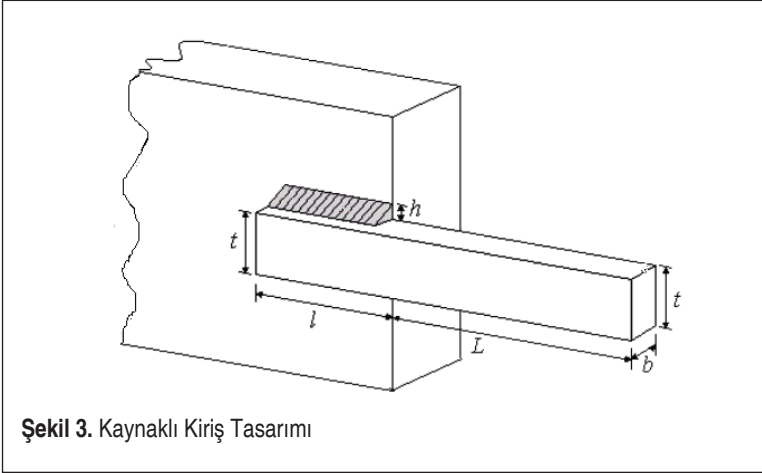
### 3. KISITLI MÜHENDİSLİK PROBLEMLERİNİN OPTİMUM BOYUTLANDIRILMASI

Bu bölümde çeşitli eşitlik ve eşitsizlik kısıtlamaları olan üç gerçek mühendislik tasarım problemi kullanılmıştır: kaynaklı kiriş, germe/sıkıştırma yay ve basınçlı kap tasarımları. Kaynaklı kirişler, kat döşemeleri, çalışma platformları, köprüler, vinç kirişleri ve diğer çelik yapılarda yaygın olarak kullanılmaktadır. Germe- sıkıştırma yayları otomobillerden (gövdeye eğilmek ve konforu artırmak için koltuklarda, araç hareket halindeyken yolun sert darbelerini emmek için süspansiyonda ve hatta otomotiv hortumlarına destek sağlamak amacıyla) uçaklara (hava türbinleri, yönlendirme sistemleri, motor kontrolleri, tekerlekler, frenler, sayaçlar, yakıt hücreleri ve dizel motorlar gibi bir çok yerde), tıbbi cihazlardan (cerrahi aletler ve ortopedik aletler

gibi tıbbi aletlerin yanı sıra kateterler, valfler, tekerlekli sandalyeler, endoskopik cihazlar ve diğer birçok tıbbi ekipman yapımında) ateşli silahlara pek çok alanda geniş bir kullanıma sahiptir. Basıncılı kap, gazları veya sıvıları ortam basıncından önemli ölçüde daha yüksek veya daha düşük bir basınçta tutmak için tasarlanmış kapalı bir kaptır. Sterilizatörler, sıkıştırılmış gaz silindirleri, kompresörler (soğutma dahil), vakum odaları ve özel olarak tasarlanmış laboratuvar kapları basınçlı kapların kullanım alanlarına örnek olarak gösterilebilir. Her tasarım problemi sekiz farklı optimizasyon algoritması (GSA, PSO, BBO, GA, DE, SSA, SCA ve GWO) ile çözülmüş, kaynaklı kiriş ve basınçlı kap tasarımlarında optimum maliyet, germe/sıkıştırma yay tasarımında ise optimum ağırlık elde edilmeye çalışılmıştır. Optimizasyon yapılırken her algoritma için 30 simülasyon yapılmış ([10]) böylece her algoritmanın en iyi, en kötü ve ortalama değerleri birbirleriyle kıyaslanmıştır. Her bir problem için algoritmaların performansları kendi içinde kıyaslanarak, optimum amaç fonksiyonlarını veren en iyi değişken değerleri belirlenmiştir.

### 3.1 Kaynaklı Kiriş Tasarım Problemi

Bu problemin amacı Şekil 3'te görülen kaynaklı kirişin üretim maliyetini en aza indirmektir. Problemden, kaynak kalınlığı ( $h$ ), kaynak boyu ( $l$ ), kiriş genişliği ( $t$ ) ve kiriş kalınlığı ( $b$ ) olmak üzere 4 adet değişken mevcuttur. Ayrıca tüm kiriş uzunluğu ( $l+L$ ) ile ifade edilebilir.



Problemin matematiksel tanımı aşağıdaki gibidir [10] :

Minimize

$$f(x) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2) \quad (8)$$



## Kısıtlayıcılar

$$g_1(x) = \tau(x) - \tau_{max} \leq 0 \quad (9)$$

$$g_2(x) = \sigma(x) - \sigma_{max} \leq 0 \quad (10)$$

$$g_3(x) = \delta(x) - \delta_{max} \leq 0 \quad (11)$$

$$g_4(x) = x_1 - x_4 \leq 0 \quad (12)$$

$$g_5(x) = P - P_c(x) \leq 0 \quad (13)$$

$$g_6(x) = 0.125 - x_1 \leq 0 \quad (14)$$

$$g_7(x) = 1.10471x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 5.0 \leq 0 \quad (15)$$

## Değişken aralıkları:

$$0.1 \leq x_1 \leq 2 \quad (16)$$

$$0.1 \leq x_2 \leq 10 \quad (17)$$

$$0.1 \leq x_3 \leq 10 \quad (18)$$

$$0.1 \leq x_4 \leq 2 \quad (19)$$

burada:

$$\tau(x) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2} \quad (20)$$

$$\tau' = \frac{P}{\sqrt{2x_1x_2}}, \tau'' = \frac{MR}{J}, M = P\left(L + \frac{x_2}{2}\right) \quad (21)$$

$$R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1+x_3}{2}\right)^2} \quad (22)$$

$$J = 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{4} + \left(\frac{x_1+x_3}{2}\right)^2\right]\right\} \quad (23)$$

$$\sigma(x) = \frac{6PL}{x_4x_3^2}, \delta(x) = \frac{6PL^3}{Ex_3^2x_4} \quad (24)$$

$$P_c(x) = \frac{4.013E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right) \quad (25)$$



$$P = 6000 \text{ lb}, L = 14 \text{ in.}, \delta_{max} = 0.25 \text{ in.} \quad (26)$$

$$E = 30 * 10^6 \text{ psi}, G = 12 * 10^6 \text{ psi} \quad (27)$$

$$\tau_{max} = 13600 \text{ psi}, \sigma_{max} = 30000 \text{ psi} \quad (28)$$

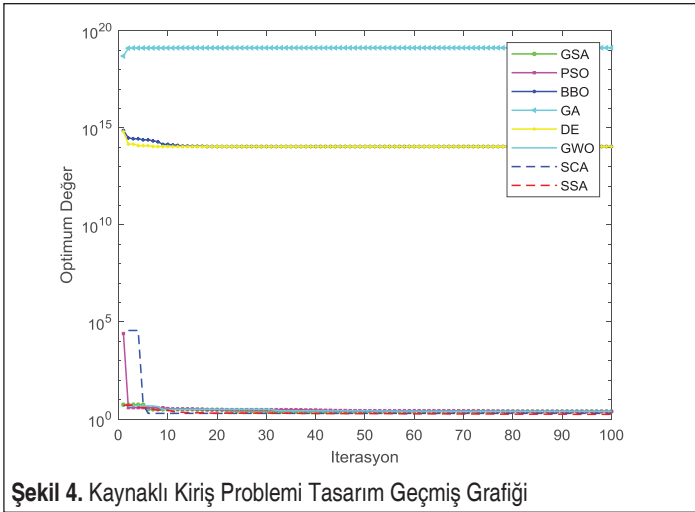
**Tablo 1.** Kaynaklı Kiriş Tasarımı Simülasyon Sonuçları

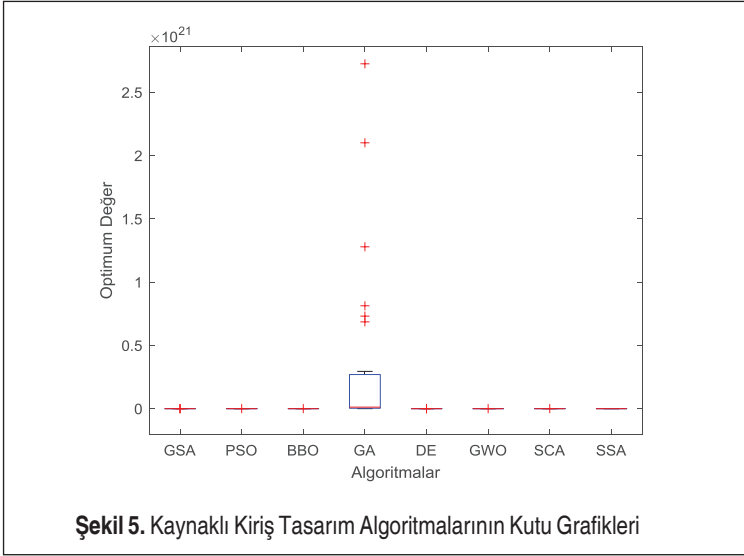
	GSA	PSO	BBO	GA	DE	SSA	SCA	GWO
1	2.429	2.490	1.1E+14	2.1E+21	1.1E+14	2.406	2.059	1.708
2	2.823	2.483	1.1E+14	2.35E+17	1.1E+14	1.995	2.289	1.708
3	4.130	2.648	1.1E+14	7.31E+20	1.1E+14	1.767	1.869	1.706
4	2.408	2.552	1.1E+14	4.44E+18	1.1E+14	3.779	2.207	1.705
5	4.177	2.483	1.1E+14	6.2E+18	1.1E+14	1.739	1.914	1.714
6	2.119	2.483	1.1E+14	6.86E+20	1.1E+14	1.822	1.944	1.706
7	2.348	2.561	1.1E+14	3.21E+18	1.1E+14	2.044	2.085	1.729
8	3.061	2.718	1.1E+14	4.91E+19	1.1E+14	2.044	2.044	1.728
9	3.006	2.483	1.1E+14	3.05E+18	1.1E+14	2.049	2.050	1.716
10	2.284	2.651	1.1E+14	2.7E+20	1.1E+14	2.004	1.884	1.726
11	2.235	2.483	1.1E+14	1.05E+19	1.1E+14	1.738	1.964	1.705
12	2.174	2.484	1.1E+14	1.08E+19	1.1E+14	2.029	2.257	1.710
13	2.620	2.483	1.1E+14	1.39E+18	1.1E+14	2.488	1.809	1.732
14	2.316	2.490	1.1E+14	1.28E+21	1.1E+14	2.447	2.006	1.751
15	2.221	2.614	1.1E+14	6.92E+17	1.1E+14	2.026	2.269	1.707
16	2.067	2.645	1.1E+14	8.13E+20	1.1E+14	2.367	1.890	1.734
17	2.640	2.483	1.1E+14	4.66E+17	1.1E+14	1.783	1.987	1.701
18	3.331	2.484	1.1E+14	4.97E+19	1.1E+14	2.252	1.893	1.718
19	2.248	2.531	1.1E+14	2.95E+20	1.1E+14	1.796	2.096	1.713
20	2.656	2.483	1.1E+14	1.56E+20	1.1E+14	1.806	1.783	1.711



21	2.553	2.568	1.1E+14	4.17E+19	1.1E+14	1.963	1.908	1.729
22	2.231	2.483	1.1E+14	9.19E+18	1.1E+14	2.357	2.532	1.712
23	2.926	2.484	1.1E+14	2.73E+21	1.1E+14	1.957	1.946	1.709
24	2.380	2.483	1.1E+14	2.6E+18	1.1E+14	2.555	1.895	1.720
25	5.660	2.487	1.1E+14	1.56E+20	1.1E+14	2.059	1.860	1.727
26	2.275	2.484	1.1E+14	7.75E+18	1.1E+14	2.086	1.845	1.716
27	2.454	2.483	1.1E+14	7.98E+16	1.1E+14	1.856	2.003	1.711
28	2.485	2.588	1.1E+14	1.4E+19	1.1E+14	1.826	1.912	1.707
29	2.241	2.483	1.1E+14	7.16E+17	1.1E+14	1.911	2.191	1.725
30	2.450	2.483	1.1E+14	1.33E+19	1.1E+14	2.522	1.959	1.751
en iyi	2.067	2.483	1.1E+14	7.98E+16	1.1E+14	1.738	1.783	1.701
en kötü	5.660	2.718	1.1E+14	2.73E+21	1.1E+14	3.779	2.532	1.751
ort.	2.698	2.525	1.1E+14	3.15E+20	1.1E+14	2.116	2.012	1.718

Problem sekiz farklı optimizasyon algoritması ile 30 defa simüle edilmiş, karşılaştırma sonuçları Tablo 1’de verilmiştir. Buna göre GWO’nun en iyi, en kötü ve ortalama sonuçlar açısından en küçük değeri bularak diğer algoritmalara kıyasla daha iyi performans gösterdiği görülmektedir. En iyi değeri bulma bakımından GWO’yu takip eden algoritmalar sırasıyla SSA, SCA, GSA ve PSO’dur. En büyük değer ve ortalama





Şekil 5. Kaynaklı Kiriş Tasarım Algoritmalarının Kutu Grafikleri

değer açılarından ise GWO'dan sonra en iyi performansı sırasıyla SCA, SSA, PSO ve GSA takip etmektedir. Bu problem için en kötü performansı GA göstermiştir. BBO ve DE aynı sonuçları bularak GA'dan sonra ikinci en kötü performans gösteren algoritmalar olmuştur. Şekil 4'te verilen yakınsama grafiğinden ise en iyi yakınsama performansının en iyi değeri bulan GWO, SSA, SCA, GSA ve PSO'ya ait olduğu görülür. Şekil 5'te algoritmaların kutu grafikleri verilmiştir. Buradan GWO'nun en iyi sonucu bulmanın yanı sıra dar ve ince bir kutu grafiğe sahip olarak kendi içinde de tutarlı bir algoritma olduğu sonucuna varılır. Aynı şekilden GA dışındaki diğer algoritmaların da kendi içerisinde tutarlı sonuçlar verdiği görülmektedir.

Kaynaklı kiriş probleminin değişkenleri için algoritmaların bulduğu optimum değerler Tablo 2'de sunulmuştur. Buna göre en düşük maliyeti bulan GWO'nun  $h$ ,  $l$ ,  $t$  ve  $b$  değerleri için bulduğu optimum değerler sırası ile, 0.21, 3.29, 9.04, 0.21 olmuştur.

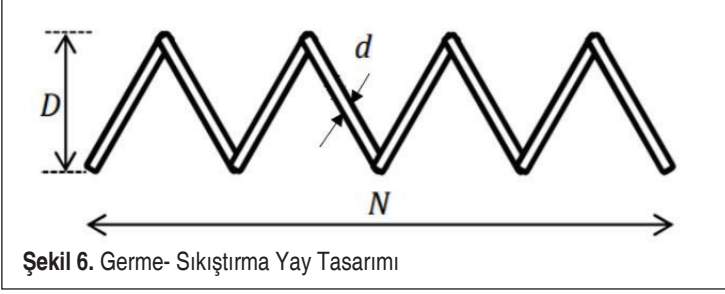
Tablo 2. Kaynaklı Kiriş Tasarımının Karşılaştırma Sonuçları

	GSA	PSO	BBO	GA	DE	SSA	SCA	GWO
$x_1(h)$	0.14	0.21	0.80	0.89	0.79	0.19	0.18	0.21
$x_2(l)$	6.14	9.04	1.96	4.43	2.00	3.61	3.89	3.29
$x_3(t)$	8.39	9.04	2.00	1.08	2.00	8.95	8.84	9.04
$x_4(b)$	0.24	0.21	2.00	0.47	2.00	0.21	0.22	0.21
$f(x)$	2.07	2.49	1E+14	8E+16	1E+14	1.74	1.79	1.70



### 3.2 Germe- Sıkıştırma Yay Tasarım Problemi

Bu problemin amacı, Şekil 6'da gösterilen germe- sıkıştırma yay ağırlığını en aza indirmektir. Problem, tel çapı ( $d$ ), ortalama bobin çapı ( $D$ ) ve aktif bobin sayısı ( $N$ ) olmak üzere üç adet tasarım değişkenine sahiptir.



Problemin matematiksel tanımı aşağıdaki gibidir [10]:

Minimize

$$f(x) = (x_3 + 2)x_2x_1^2 \quad (29)$$

Kısıtlayıcılar:

$$g_1(x) = 1 - \frac{x_2^3x_3}{71785x_1^4} \leq 0 \quad (30)$$

$$g_2(x) = \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} - 1 \leq 0 \quad (31)$$

$$g_3(x) = 1 - \frac{140.45x_1}{x_2^2x_3} \leq 0 \quad (32)$$

$$g_4(x) = \frac{x_1 + x_2}{1.5} - 1 \leq 0 \quad (33)$$

Değişken aralıkları:

$$0.05 \leq x_1 \leq 2 \quad (34)$$

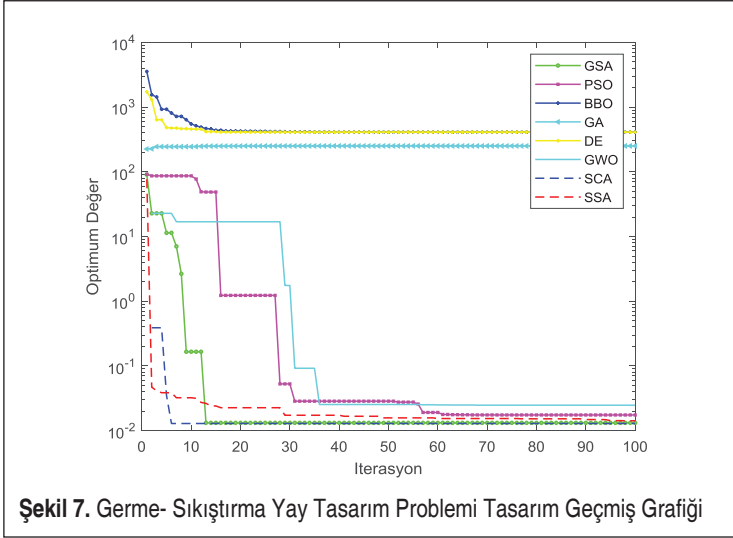
$$0.25 \leq x_2 \leq 1.3 \quad (35)$$

$$2 \leq x_3 \leq 15 \quad (36)$$

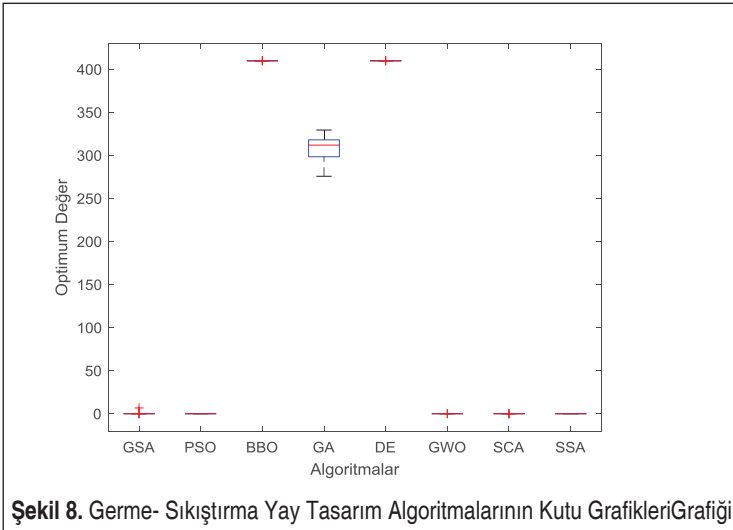
Germe- sıkıştırma yay tasarım problemi 8 farklı optimizasyon algoritması ile 30 defa simüle edilmiş, karşılaştırma sonuçları Tablo 3'te verilmiştir. Buna göre bu problem için en iyi (en düşük) değeri çok yakın değerlerle bulan PSO ve GWO'yu sırasıyla SSA, GSA, SCA algoritmaları takip etmiştir. Ortalama ve en kötü sonuçlar bakımın-

**Tablo 3.** Germe- Sıkıştırma Yay Tasarım Simülasyon Sonuçları

	GSA	PSO	BBO	GA	DE	SSA	SCA	GWO
1	0.01283	0.01770	409.77505	313.04340	409.77499	0.01274	0.01290	0.01271
2	0.01303	0.01268	409.79687	318.78718	409.77499	0.01272	0.01297	0.01270
3	0.01307	0.01285	409.78515	295.44667	409.77499	0.01278	0.01306	0.01269
4	0.01284	0.01770	409.78242	324.39039	409.80343	0.01282	0.01315	0.01269
5	0.01308	0.01340	409.77593	320.51704	409.77499	0.01283	0.01302	0.01271
6	0.01305	0.01770	409.77727	358.30356	409.77499	0.01272	0.01309	0.01282
7	0.01286	0.01565	409.77601	282.09296	409.77499	0.01302	0.01323	0.01272
8	0.01300	0.01770	409.77771	282.28586	409.77499	0.01277	0.01297	0.01272
9	0.01283	0.01275	409.78132	317.40720	409.77499	0.01281	0.01306	0.01269
10	0.01301	0.01331	409.77522	316.20756	409.77499	0.01298	0.01303	0.01272
11	0.01273	0.01297	409.77830	320.43729	409.80343	0.01276	0.01311	0.01271
12	0.01291	0.01315	409.77745	308.96701	409.77499	0.01281	0.01296	0.01272
13	0.01279	0.01270	409.77512	324.67059	409.77499	0.01304	0.01296	0.01270
14	0.01301	0.01624	409.77516	384.01078	409.77499	0.01290	0.01313	0.01272
15	0.01283	0.01370	409.79626	299.62341	409.77499	0.01276	0.01320	0.01267
16	0.01321	0.01266	409.77500	318.68800	409.77499	0.01269	0.01320	0.01272
17	0.01309	0.01281	409.77613	293.44965	409.77499	0.01288	0.01278	0.01273
18	0.01292	0.01321	409.79829	304.56746	409.80343	0.01273	0.01296	0.01272
19	0.01278	0.01707	409.77574	327.05307	409.77499	0.01293	0.01311	0.01269
20	0.01285	0.01280	409.78362	305.42083	409.80343	0.01272	0.01284	0.01271
21	0.01297	0.01310	409.77691	276.59454	409.77499	0.01279	0.01303	0.01279
22	0.01286	0.01283	409.77930	319.34484	409.77499	0.01284	0.01302	0.01272
23	0.01277	0.01346	409.77760	328.70629	409.77499	0.01275	0.01297	0.01272
24	0.01299	0.01267	409.78270	314.40699	409.77499	0.01301	0.01312	0.01272
25	0.01314	0.01770	409.77501	313.31879	409.77499	0.01283	0.01292	0.01271
26	0.01304	0.01268	409.77559	317.08702	409.80343	0.01301	0.01288	0.01272
27	0.01281	0.01433	409.78025	300.42008	409.77499	0.01302	0.01295	0.01270
28	0.01289	0.01712	409.82080	306.62959	409.80343	0.01272	0.01277	0.01272
29	0.01283	0.01308	409.77516	313.84367	409.77499	0.01285	0.01312	0.01272
30	0.01285	0.01271	409.77629	307.56736	409.77499	0.01289	0.01290	0.01268
en iyi	0.01273	0.01266	409.77500	276.59454	409.77499	0.01269	0.01277	0.01267
en kötü	0.01321	0.01770	409.82080	384.01078	409.80343	0.01304	0.01323	0.01282
ortalama	0.01293	0.01428	409.78112	313.77630	409.78068	0.01284	0.01301	0.01271



dan ise sırasıyla GWO, SSA, GSA, SCA, PSO en başarılı performansı sergilemiştir. Bu problem için tüm kriterler açısından en kötü sonuçları bulan algoritmalar birbirlerine çok yakın değerlerle BBO ve DE algoritmaları olmuş, GA da onları takip ederek en başarısız üçüncü algoritma olmuştur. Şekil 7’de verilen yakınsama eğrilerine bakıldığında GA dışındaki algoritmaların yakınsama performanslarının iyi olduğu söylenebilir. Şekil 8’deki kutu grafiklerinden ise GSA ve GA dışındaki algoritmaların da kendi içinde tutarlı olduğu gözlenir.



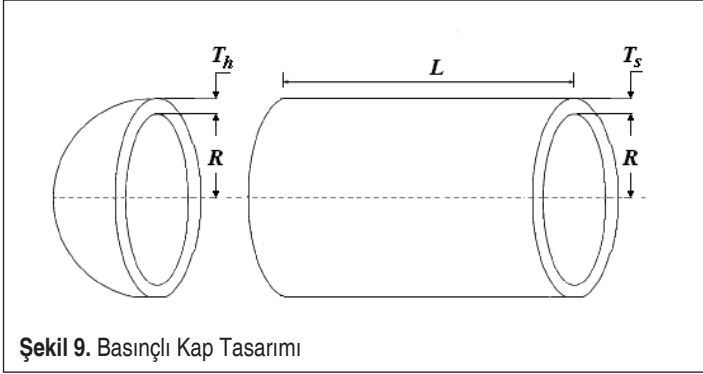
**Tablo 4.** Germe- Sıkıştırma Yay Tasarımı Karşılaştırma Sonuçları

	GSA	PSO	BBO	GA	DE	SSA	SCA	GWO
$x_1(d)$	0.05	0.05	2.00	1.88	2.00	0.05	0.05	0.05
$x_2(D)$	0.32	0.36	2.00	1.18	2.00	0.33	0.32	0.37
$x_3(N)$	14.05	10.94	2.00	14.71	2.00	13.14	13.99	10.71
$f(x)$	0.01273	0.01266	409.775	276.594	409.775	0.01269	0.01277	0.01267

Bunlara ek olarak germe-sıkıştırma yay tasarım probleminin optimum ağırlığı veren değişkenleri için algoritmaların bulduğu optimum değerler Tablo 4’te sunulmuştur.

### 3.3 Basıncı Kap Tasarım Problemi

Bu problemin amacı, Şekil 9’da gösterilen her iki ucu kapalı ve başı yarım küre şeklinde olan silindirik bir kabın malzemesi, biçimlendirilmesi ve kaynaklanmasından oluşan toplam maliyetini en aza indirmektir. Kabuk kalınlığı ( $T_s$ ), baş kalınlığı ( $T_h$ ), iç yarıçap ( $R$ ) ve baş kısmı dikkate alınmadan silindirik bölümün uzunluğu ( $L$ ) olmak üzere dört farklı değişken vardır.



Problemin matematiksel tanımı aşağıdaki gibidir [10]:

Minimize

$$f(x) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \quad (37)$$

Kısıtlayıcılar

$$g(x) = 0.0193x_3 - x_1 \leq 0 \quad (38)$$



$$g_2(x) = 0.00954x_3 - x_2 \leq 0 \quad (39)$$

$$g_3(x) = 1296000 - \pi x_3^2 x_4 - \frac{4}{3} \pi x_3^3 \leq 0 \quad (40)$$

$$g_4(x) = x_4 - 240 \leq 0 \quad (41)$$

Değişken aralıkları:

$$0 \leq x_1 \leq 99 \quad (42)$$

$$0 \leq x_2 \leq 99 \quad (43)$$

$$10 \leq x_3 \leq 200 \quad (44)$$

$$10 \leq x_4 \leq 200 \quad (45)$$

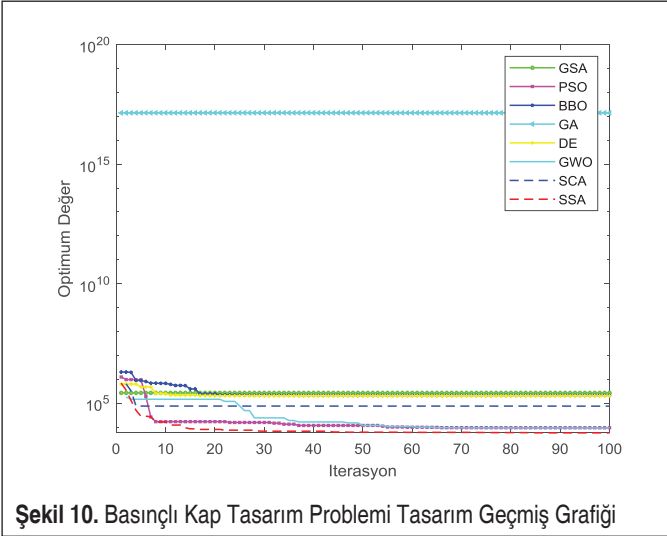
**Tablo 5.** Basıncılı Kap Tasarım Simülasyon Sonuçları

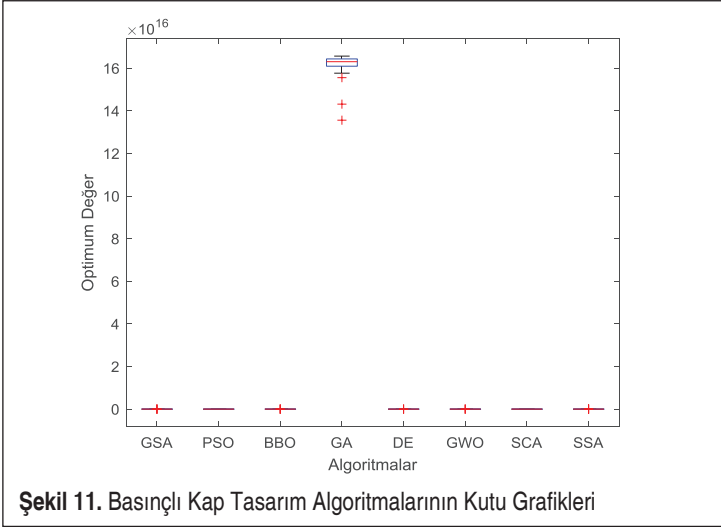
	GSA	PSO	BBO	GA	DE	SSA	SCA	GWO
1	53497.384	9441.233	204360.515	1.64E+17	204323.556	6.04E+03	6.16E+03	7.06E+03
2	7610.210	9441.233	205250.09	1.64E+17	204323.705	5.80E+03	1.98E+04	5.74E+03
3	33622.301	9441.233	204329.563	1.58E+17	204324.896	7.12E+03	1.98E+04	5.74E+03
4	7283.770	9441.233	204433.652	1.58E+17	204322.985	6.16E+03	1.97E+04	5.74E+03
5	7137.312	9441.233	205919.341	1.59E+17	204325.743	6.31E+03	7.58E+04	5.74E+03
6	7752.253	9441.233	204731.114	1.65E+17	204322.746	5.91E+03	1.96E+04	5.74E+03
7	26586.222	9441.233	209563.405	1.64E+17	204323.659	6.27E+03	1.95E+04	5.74E+03
8	120698.340	9441.233	207658.741	1.64E+17	204323.638	9.01E+03	2.04E+04	5.74E+03
9	7138.314	9441.233	208571.117	1.64E+17	204323.460	5.99E+03	7.56E+04	5.74E+03
10	6357.448	9441.233	204327.241	1.61E+17	204324.364	6.17E+03	7.54E+04	5.74E+03
11	6678.758	9441.233	211009.149	1.56E+17	204327.000	5.91E+03	7.54E+04	5.74E+03
12	6179.866	9441.233	205675.177	1.63E+17	204323.534	5.83E+03	2.00E+04	6.35E+03
13	7167.224	9441.233	204325.691	1.62E+17	204323.016	5.79E+03	7.54E+04	5.87E+03
14	138240.834	9441.233	205212.769	1.62E+17	204324.339	5.90E+03	1.97E+04	5.74E+03
15	76245.247	9441.233	204816.5	1.64E+17	204323.943	5.78E+03	1.99E+04	5.74E+03
16	6882.867	9441.233	206492.487	1.61E+17	204322.406	5.77E+03	2.02E+04	5.74E+03
17	26372.768	9441.233	207919.819	1.36E+17	204323.235	5.82E+03	7.54E+04	5.74E+03
18	11770.012	9441.233	204916.579	1.61E+17	204325.312	5.78E+03	7.54E+04	5.74E+03
19	7189.355	9441.233	204343.445	1.65E+17	204326.432	7.39E+03	6.24E+03	5.74E+03
20	6288.499	9441.233	204451.691	1.65E+17	204322.672	5.80E+03	7.54E+04	5.74E+03



21	7887.630	9441.233	204501.524	1.63E+17	204323.112	6.14E+03	7.54E+04	5.74E+03
22	151221.506	9441.233	204813.97	1.43E+17	204323.487	6.28E+03	1.99E+04	5.74E+03
23	6737.775	9441.233	218042.281	1.58E+17	204323.900	6.12E+03	7.54E+04	5.74E+03
24	6144.007	9441.233	204497.364	1.64E+17	204324.694	5.79E+03	1.98E+04	5.74E+03
25	6911.481	9441.233	204691.877	1.63E+17	204323.603	5.90E+03	7.54E+04	5.74E+03
26	7004.493	9441.233	204600.764	1.62E+17	204322.306	5.74E+03	1.98E+04	5.74E+03
27	8325.455	9441.233	205362.301	1.63E+17	204322.953	5.75E+03	7.54E+04	6.91E+03
28	8035.951	9441.233	206018.379	1.65E+17	204322.204	6.18E+03	1.96E+04	1.94E+04
29	7470.613	9441.233	204970.919	1.62E+17	204324.326	5.86E+03	6.88E+03	5.74E+03
30	7833.913	9441.233	204328.454	1.66E+17	204329.400	5.78E+03	1.98E+04	6.91E+03
en iyi	6144.007	9441.233	204325.69	1.356E+17	204322.2	5737.874	6157.699	5735.545
en kötü	151221.506	9441.233	218042.28	1.657E+17	204329.4	9008.867	75755.978	19440.810
ort.	26275.727	9441.233	206004.53	1.609E+17	204324.02	6136.329	40740.592	6341.371

Problem 8 farklı optimizasyon algoritması ile simüle edilmiş, karşılaştırma sonuçları Tablo 5’te verilmiştir. Buna göre en iyi (en düşük) sonucu bulan algoritmalar sırası ile GWO, SSA, GSA, SCA ve PSO olmuştur. En kötü (en yüksek) sonuç açısından bakıldığında SSA en iyi performansı göstermiş, PSO, GWO ve SCA onu takip etmiştir. Ortalama sonuç açısından ise SSA, GWO ve PSO sırasıyla en düşük değeri bularak en başarılı algoritmalar olmuştur. Bu problem için en iyi, en kötü ve ortalama değerlerin her üçü açısından da en kötü performansa sahip algoritma GA olurken BBO ve DE kötü performans açısından GA’yı takip eden diğer başarısız algoritmalar olmuştur.





Şekil 10'dan GA dışındaki algoritmaların yakınsama eğrilerinin başarılı olduğu söylenebilir. Şekil 11'deki kutu grafiklerinden ise GA dışındaki algoritmaların kendi içinde tutarlı olduğu sonucuna varılabilir.

Bunlara ek olarak basıncılı kap tasarım probleminin optimum maliyeti (2775.342) veren değişkenleri için algoritmaların bulunduğu optimum değerler Tablo 6'da sunulmuştur. Buna göre en başarılı performansa sahip olan SSA algoritmasının bulunduğu  $T_s$ ,  $T_h$ , R ve L değişken değerleri sırası ile 1.09, 0.06, 65.23 ve 10 olmuştur.

**Tablo 6.** Basıncılı Kap Tasarımı Karşılaştırma Sonuçları

	GSA	PSO	BBO	GA	DE	SSA	SCA	GWO
$x_1(T_s)$	0.93	1.03	10.00	74.02	10.00	0.74	0.79	0.74
$x_2(T_h)$	0.45	1.03	10.00	89.88	10.00	0.37	0.43	0.37
$x_3(R)$	49.66	56.13	53.53	28.71	53.66	40.40	41.22	40.32
$x_4(L)$	101.10	56.13	72.56	12.53	71.69	198.93	193.40	200.00
$f(x)$	6144.01	9441.23	204325.69	1.356E+17	204322.20	5737.87	6157.70	5735.54

#### 4. SONUÇLAR VE TARTIŞMA

Bu çalışmada üç kısıtlı gerçek mühendislik probleminin (kaynaklı kiriş, germe/sıkıştırma yay ve basıncılı kap) optimal tasarımları hedeflenmiştir. Bu amaçla GSA, PSO, BBO, GA, DE, SSA, SCA ve GWO olmak üzere sekiz farklı optimizasyon algoritması ele alınan mühendislik problemlerine uyarlanmıştır. Kaynaklı kiriş tasarım probleminin

de en düşük maliyeti (1.70) veren  $h$ ,  $l$ ,  $t$  ve  $b$  değerleri için bulunan optimum değerler sırası ile, 0.21, 3.29, 9.04, 0.21 olmuştur. Germe-sıkıştırma yay tasarım probleminin optimum ağırlığı (0.01269) veren  $d$ ,  $D$ ,  $N$  değişken değerleri sırası ile 0.05, 0.33 ve 13.14 olmuştur. Basıncılı kap tasarım probleminde optimum maliyeti (5735.54) veren optimum  $T_s$ ,  $T_h$ ,  $R$  ve  $L$  değişken değerleri ise sırası ile 0.74, 0.37, 40.32 ve 200 olmuştur. GWO kaynaklı kiriş ve basıncılı kap tasarımı problemlerinde de en iyi sonucu veren algoritma olurken germe- sıkıştırma yay tasarım probleminde en iyi değeri bulan PSO algoritmasına çok yakın değer bularak ikinci en iyi sonucu bulan algoritma olmuştur. Bununla birlikte aynı problemde ortalama ve en kötü (en yüksek) sonuçlar açısından en başarılı sonucu bulmuştur. GA, BBO ve DE algoritmaları her üç problemde de en iyi, en kötü ve ortalama değerlerin hepsinde en yüksek sonuçları bularak en başarısız algoritmalar olmuştur. Problemlerin bulunan optimum sonuçları Tablo 7'de literatür sonuçları ile kıyaslanmıştır. Buna göre her üç mühendislik probleminde de diğer çalışmalara kıyasla daha iyi sonuç elde edildiği görülmektedir. Çeşitli kısıtlayıcılara sahip üç gerçek mühendislik probleminin optimum tasarımından elde edilen sonuçlara bağlı olarak algoritmalar başarılarına göre gruplandırılacak olursa, evrimsel tabanlı algoritmaların (GA, BBO, DE) en kötü sonuçları verdiği, sürü tabanlı (GWO, PSO), matematik tabanlı (SSA, SCA) ve fizik tabanlı algoritmaların ise (GSA) ise en başarılı sonuçları bularak en iyi performans sergileyen algoritmalar olduğu savunulabilir. Bununla birlikte problem çeşidine göre bu gruplamanın değişebileceği de göz ardı edilmemelidir. Gelecek çalışmalar için optimizasyon algoritmalarının performansının algoritma parametrelerine bağlı oluşu göz önünde bulundurularak problem sayısı ve algoritma sayısı arttırılabilir ve elde edilen sonuçlar karşılaştırılabilir.

**Tablo 7.** Tasarım Problemleri Optimum Sonuç Kıyaslanması

		Optimum Sonuç
Kaynaklı Kiriş Tasarımı	GWO	1.701
	Lee ve Geem [37]however, reveal a limited approach to complicated real-world optimization problems. If there is more than one local optimum in the problem, the result may depend on the selection of an initial point, and the obtained optimal solution may not necessarily be the global optimum. This paper describes a new harmony search (HS)	2.3807
	Deb [38]	2.38
Germe- Sıkıştırma Yay Tasarım	PSO	0.01266
	He ve Wang [39]	0.01268
	Mahdavi [40]	0.01267
Basıncılı Kap Tasarımı	GWO	5735.545
	He ve Wang [39]	6061.0777
	Montes ve Coello [41]	6059.7456



## KAYNAKÇA

1. **Wolpert, D.H., W.G. Macready**, 1997. “No free lunch theorems for optimization,” IEEE Transactions on Evolutionary Computation, vol. 1, no. 1, p.67–82.
2. **Dogan, E., A.O. Ciftcioglu, F. Erdal**, 2017. Optimum design of cellular beams via bat algorithm with levy flights, In OPTIMA-2017 Conf., .
3. **Erdal, F., E. Dogan, M.P. Saka**, 2011. “Optimum design of cellular beams using harmony search and particle swarm optimizers,” Journal of Constructional Steel Research, vol. 67, no. 2, p.237–247.
4. **Khalilpourazari, S., H. Hashemi Doulabi, A. Özyüksel Çiftçiöglu, G.W. Weber**, 2021. “Gradient-based grey wolf optimizer with Gaussian walk: Application in modelling and prediction of the COVID-19 pandemic,” Expert Systems with Applications, vol. 177, .
5. **Khalilpourazari, S., B. Naderi, S. Khalilpourazary**, 2020. “Multi-Objective Stochastic Fractal Search: a powerful algorithm for solving complex multi-objective optimization problems,” Soft Computing, vol. 24, no. 4, p.3037–3066.
6. **Khalilpourazari, S., S. Khalilpourazary**, 2019. “An efficient hybrid algorithm based on Water Cycle and Moth-Flame Optimization algorithms for solving numerical and constrained engineering optimization problems,” Soft Computing, vol. 23, no. 5, p.1699–1722.
7. **Khalilpourazari, S., S. Khalilpourazary**, 2018. “SCWOA: an efficient hybrid algorithm for parameter optimization of multi-pass milling process,” Journal of Industrial and Production Engineering, vol. 35, no. 3, p.135–147.
8. **Khalilpourazari, S., S. Khalilpourazary**, 2016. “Optimization of production time in the multi-pass milling process via a Robust Grey Wolf Optimizer,” Neural Computing and Applications, vol. 29, no. 12, p.1321–1336.
9. **Rather, S.A., N. Sharma**, 2018. “Gsa-Bbo Hybridization Algorithm,” no. December,.
10. **Mirjalili, S., S.M. Mirjalili, A. Lewis**, 2014. “Grey Wolf Optimizer,” Advances in Engineering Software, vol. 69, p.46–61.
11. **Dogan, E.** 2010. Optimum design of rigid and semi-rigid steel sway frames including soil-structure interaction. PhD Thesis, Middle East Technical University, Ankara.
12. **Kennedy, J., R. Eberhart**, 1995. Particle Swarm Optimization, In IEEE Int. Conf. Neural Networks, IEEE Press, pp: 1942–1948.
13. **Mirjalili, S., A.H. Gandomi, S.Z. Mirjalili, S. Saremi, H. Faris, S.M. Mirjalili**, 2017. “Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems,” Advances in Engineering Software, vol. 114, p.163–191.
14. **Dorigo, M., M. Birattari, T. Stützle**, 2006. “Ant colony optimization: artificial ants as a computational intelligence technique,” IEEE Computational Intelligence Magazine, vol. 1, p.28–39.
15. **Mirjalili, S.** 2015. “Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm,” Knowledge-Based Systems, vol. 89, p.228–249.

16. **Yang, X.S.** 2010. A New Metaheuristic Bat-Inspired Algorithm, In Nat. Inspired Coop. Strateg. Optim., Springer, pp: 65–74.
17. **Rashedi, E., H. Nezamabadi-pour, S. Saryazdi,** 2009. “GSA: A Gravitational Search Algorithm,” Information Sciences, vol. 179, no. 13, p.2232–2248.
18. **Erol, O.K., I. Eksin,** 2006. “A new optimization method: Big Bang–Big Crunch,” Advances in Engineering Software, vol. 37, no. 2, p.106–111.
19. **Mirjalili, S.** 2016. “SCA: A Sine Cosine Algorithm for solving optimization problems,” Knowledge-Based Systems, vol. 96, p.120–133.
20. **Salimi, H.** 2015. “Stochastic Fractal Search: A powerful metaheuristic algorithm,” Knowledge-Based Systems, vol. 75, p.1–18.
21. **Bonabeau, E., M. Dorigo, G. Theraulaz,** 1999. Swarm intelligence: from natural to artificial systems: OUP.
22. **Storn, R., K. Price,** 1997. “Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces,” Journal of Global Optimization, vol. 11, no. 4, p.341–359.
23. **Ma, H., D. Simon,** 2010. Biogeography-Based Optimization with Blended Migration for Constrained Optimization Problems, In Proc. 12th Annu. Conf. Genet. Evol. Comput., Association for Computing Machinery, pp: 417–418.
24. **Güler, T., M. Kılıç,** 2019. “Klimatik Kontrollü Treyler İçerisindeki Hava Akışının Optimizasyonu Optimization of Air Flow in Refrigerated Semi-Trailer,” vol. 60, no. 697, p.289–302.
25. **Tekeliođlu, S., S. Eldek, H. Gümüş, A. Sarıgöl, Ş. Ayhan,** 2020. “Hidrolik Yüksek Basınç Hattı Filtre Gövdesinin Tasarımı , Optimizasyonu , Üretimi ve Test Edilmesi Design , Optimization and Fabrication of Body of Hydraulic High- Pressure Filter and Experimental Validation,” p.0–2.
26. **Alinaghian, M., E.B. Tirkolaee, Z.K. Dezaki, S.R. Hejazi, W. Ding,** 2021. “An augmented Tabu search algorithm for the green inventory-routing problem with time windows,” Swarm and Evolutionary Computation, vol. 60, no. November 2020, p.100802.
27. **Tirkolaee, E.B., A. Mardani, Z. Dashtian, M. Soltani, G.W. Weber,** 2020. “A novel hybrid method using fuzzy decision making and multi-objective programming for sustainable-reliable supplier selection in two-echelon supply chain design,” Journal of Cleaner Production, vol. 250, p.119517.
28. **Meneghini, I.R., M.A. Alves, A. Gaspar-Cunha, F.G. Guimarães,** 2020. “Scalable and customizable benchmark problems for many-objective optimization,” Applied Soft Computing Journal, vol. 90, p.106139.
29. **Varelas, K., O.A. El Hara, D. Brockhoff, N. Hansen, D.M. Nguyen, T. Tušar, A. Auger,** 2020. “Benchmarking large-scale continuous optimizers: The bbob-largescale testbed, a COCO software guide and beyond,” Applied Soft Computing Journal, vol. 97, p.106737.
30. **Shabani, A., B. Asgarian, M. Salido, S. Asil Gharebaghi,** 2020. “Search and rescue



optimization algorithm: A new optimization method for solving constrained engineering optimization problems,” *Expert Systems with Applications*, vol. 161, p.113698.

31. **Muthusamy, H., S. Ravindran, S. Yaacob, K. Polat**, 2021. “An improved elephant herding optimization using sine–cosine mechanism and opposition based learning for global optimization problems,” *Expert Systems with Applications*, vol. 172, no. October 2020, p.114607.
32. **Gucuyen, E., R.T. Erdem**, 2014. “Corrosion effects on structural behaviour of jacket type offshore structures,” *Gradjevinar*, vol. 66, no. 11, p.981–986.
33. **Dagli, B.Y., Y. Tuskan, D. Uncu**, 2019. “Investigation of fluid-structure interaction by using solidity ratio,” *Eurasian Journal of Civil Engineering and Architecture*, vol. 3, no. 2, p.41–47.
34. **Rather, S.A., P.S. Bala**, 2020. “Hybridization of Constriction Coefficient Based Particle Swarm Optimization and Gravitational Search Algorithm for Function Optimization,” *SSRN Electronic Journal*, p.1–10.
35. **Kanwal, S., A. Hussain, K. Huang**, 2021. “Novel Artificial Immune Networks-based optimization of shallow machine learning (ML) classifiers,” *Expert Systems with Applications*, vol. 165, no. September 2019, p.113834.
36. **Jin, X.-B., G.-S. Xie, K. Huang, A. Hussain**, 2018. “Accelerating Infinite Ensemble of Clustering by Pivot Features,” *Cogn. Comput.*, vol. 10, no. 6, p.1042–1050.
37. **Lee, K.S., Z.W. Geem**, 2005. “A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice,” *Computer Methods in Applied Mechanics and Engineering*, vol. 194, no. 36–38, p.3902–3933.
38. **Deb, K.** 2000. “An Efficient Constraint Handling Method for Genetic Algorithms,” *Computer Methods in Applied Mechanics and Engineering*, vol. 186, p.311–338.
39. **He, Q., L. Wang**, 2007. “An effective co-evolutionary particle swarm optimization for constrained engineering design problems,” *Engineering Applications of Artificial Intelligence*, vol. 20, no. 1, p.89–99.
40. **Mahdavi, M., M. Fesanghary, E. Damangir**, 2007. “An improved harmony search algorithm for solving optimization problems,” *Applied Mathematics and Computation*, vol. 188, no. 2, p.1567–1579.
41. **Mezura-Montes, E., C.A.C. Coello**, 2008. “An empirical study about the usefulness of evolution strategies to solve constrained optimization problems,” *International Journal of General Systems*, vol. 37, no. 4, p.443–473.