

Yapay Alg Algoritmasının Tasarım Optimizasyon Problemlerinde ki Performansı Üzerine Bir Çalışma: Basınç Yayısı Örneği

Araştırma Makalesi/Research Article

Ümit ATİLA¹, Murat DÖRTERLER², İsmail ŞAHİN*³

¹Bilgisayar Mühendisliği, Karabük Üniversitesi, Karabük, Türkiye

²Bilgisayar Mühendisliği, Gazi Üniversitesi, Ankara, Türkiye

³Endüstriyel Tasarım Mühendisliği, Gazi Üniversitesi, Ankara, Türkiye

umitatila@karabuk.edu.tr, dortlerler@gazi.edu.tr, isahin@gazi.edu.tr

(Geliş/Received:13.08.2018; Kabul/Accepted:05.10.2018)

DOI: 10.17671/gazibtd.452992

Özet — Makine elemanlarının optimum tasarımı mühendislikte yaygın olarak çalışılan bir araştırma konusudur. Basınç yaylarının minimum ağırlığa veya hacme göre tasarımını bu alanda en çok çalışılan problemlerden birisidir. Bu problem ayrıca optimizasyon yöntemleri için değerlendirme problemi olarak kullanılmaktadır. Yapay Alg Algoritması (YAA) bir optimizasyon yöntemidir ve besin üretmek için ihtiyaç duydukları maddelere erişmek üzere ortam şartlarına uyumda doğal bir yeteneğe sahip alglerin davranışlarından esinlenmiştir. Bu çalışmada, basınç yaylarının minimum hacme göre tasarımı YAA ile optimize edilmiştir ve YAA'nın problem üzerindeki başarımı incelenmiştir. YAA'nın başarımı daha önceki çalışmalarda probleme uygulanmış optimizasyon yöntemleri ile karşılaştırılmıştır. Deneysel çalışmalar YAA'nın tasarım optimizasyon problemini tutarlı ve düşük yakınsama oranıyla birlikte başarıyla çözmeye yeteneğinin olduğunu göstermiştir.

Anahtar Kelimeler —yapay alg algoritması, basınç yayı, tasarım optimizasyonu

A Study on the Performance of Artificial Algae Algorithm in Design Optimization Problems: Compressing Spring Example

Abstract — Optimal design of machine elements is a research field studying in engineering commonly. Design of compression springs according to minimum weight or volume is one of the most studied problems in this field. The problem is also used as a benchmark problem for the optimization methods. Artificial Algae Algorithm (AAA) is an optimization technique and inspired by the behaviors of algae, which have natural skill of adaptation to environmental conditions in order to obtain substances which they need to produce nutrients. In this study, the design of compression springs with minimum volume was optimized through AAA and performance of AAA on the problem was examined. Performance of AAA were compared with the results of the optimization methods applied to the problem in previous studies. Experimental results show that AAA is capable of solving the design optimization problem successively with consistency and low convergence rate.

Keywords —artificial algae algorithm, compression spring, design optimization

1. GİRİŞ (INTRODUCTION)

İyi tanımlanmış gereksinimleri karşılayan ürünler ortaya çıkarmak için hedef temelli, kısıtları olan karar verme

işlemi olarak karakterize edilen süreçler mühendislik tasarımı olarak adlandırılır. Ürünün tasarlanması sürecinde tasarımcı bazı matematiksel analizler, deneyimler ve sezgiler kullanarak tasarımı gerçekleştirmektedir.

Geliştirilen tasarımın kabul edilir olup olmadığına yönelik birtakım analizler yapılmaktadır [1]. Tasarım optimizasyonu, bir arama uzayı (uygun çözümler), belirli hedefler (amaç fonksiyon) ve arama sürecinden (optimizasyon yöntemleri) oluşmaktadır. Uygun çözümler, tasarım değişkenlerinin olası tüm değerlerinden oluşan tasarımlar topluluğudur. Optimizasyon yöntemleri ise uygun tasarımlar arasından en iyi (optimal) olanı bulmaya çalışmaktadır [2].

Mekanik tasarım, tasarımcının devamlı olarak dayanıklılık, eğilme, aşınma, ağırlık ve korozyon gibi amaçları ihtiyaçlar doğrultusunda optimize ettiği süreçleri içermektedir [2]. Mekanik tasarım optimizasyon problemlerinin birçoğunun geleneksel optimizasyon yöntemleri ile çözümü belirli kısıtlar içerdiğinden dolayı zordur [3]. Gerçek tasarım optimizasyonu problemlerinde tasarım parametrelerinin sayısının genellikle fazla olması ve bu parametrelerin amaç fonksiyonu üzerindeki etkisinin doğrusal olmayan karakteristik sergilemesi problem karmaşıklığını artırmaktadır ve amaç fonksiyonunun birçok yerel en iyilere sahip olma durumunu ortaya çıkarmaktadır. Bu da mekanik optimizasyon problemlerinde global en iyiyi yakalamak isteyen tasarımcıları klasik metotlar yerine daha etkili ve verimli çözümler sunan evrimsel optimizasyon metotları kullanmaya sevk etmektedir [2-4].

Evrimsel algoritmalar, popülasyon tabanlı ve doğadan ilham alan meta sezgisel algoritmalarlardır. İlhamın kaynağına bağlı olarak sürü zekasına, biyolojik sistemlere, fizik veya kimya prensiplerine dayanabilmektedir. Bu özellikleri sebebi ile özellikle son yirmi yıldır optimizasyon çalışmalarında yoğun olarak tercih edilmektedirler. Özellikle tavlama benzetimi [5], Genetik Algoritma (GA) [6-8], Parçacık Sürü Optimizasyonu (PSO) [9], Karınca Kolonisi Optimizasyonu (ACO) [10] ve Arı Kolonisi Optimizasyonu (ABC) [11] kendisini kanıtlamış popüler evrimsel algoritmalar arasında sayılabilir [12]. Meta sezgisel yaklaşımlar sadece bilgisayar bilimleri alanında değil diğer alanlarda da oldukça bilinen ve kullanılan optimizasyon yaklaşımlarıdır. Meta sezgisel yaklaşımların basitliği, esnekliği ve yerel en iyilerden kaçınması değişik alanlarda tercih edilmesinin önemli sebeplerindendir [4, 7, 8, 13].

Basınç yayı, kuvvet veya tork uygulandığında deforme olan ve eski haline döndüğünde ise depoladığı enerjiyi geri veren bir sistemdir. Mekanik tasarımlarda yaylar esneklik, titreşim izolasyonu, enerji emme ve hafifletici şok gibi fonksiyonlara sahip elemanlardır ve makinelerin en ağır yüklenen bileşenleri arasında yer almaktadır. Genel bir yay tasararken genellikle yük, sapma, aktif sarım sayısı ve yay telinin ortalama çapı gibi faktörleri belirlemek için iz ve kesim yöntemleri kullanılır [14, 15].

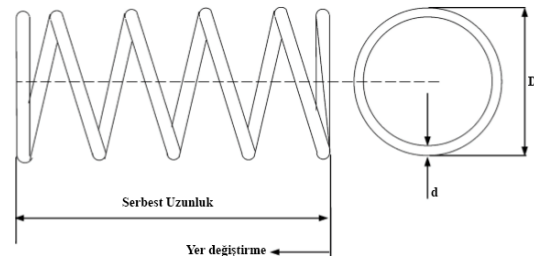
Yayların tasarım optimizasyonu, kullanıldıkları yere göre minimum ağırlık veya minimum hacim prensiplerine göre gerçekleştirilmektedir. Bu çalışmada, bir basınç yayının minimum hacme göre optimum tasarım parametrelerinin belirlenmesi hedeflenmektedir. İlk defa Sandgren

tarafından tanımlanan bu problem doğrusal olmayan bir optimizasyon problemidir. Sandgren tanımladığı problemin çözümü için doğrusal olmayan tamsayı programlama yöntemini önermiştir [16]. Deb ve Goyal yaptıkları çalışmada, mühendislik tasarım problemlerinde değişken değerlerinin sıfır-bir, ayrık ve sürekli değerler alabildiğinden bahsetmiş ve geleneksel yöntemler ile bu tür tasarım optimizasyonu problemlerini çözmenin zorluğunu belirtmişlerdir [17]. Çözüm için farklı değişken tiplerine göre ikilik ve gerçek değerli kodlama kullanılan birleşik bir Genetik Algoritma (GA) tekniği önermişlerdir. Lampinen ve arkadaşları yaptıkları çalışmada tam ayrık ve sürekli değerleri olan doğrusal olmayan problemlerin çözümü için Diferansiyel Gelişim (Differential Evolution DE) algoritmasını düzenlemişlerdir [18]. He ve arkadaşları ise yayın minimum hacme göre optimizasyonunu parçacık sürü optimizasyonu (PSO) yöntemi ile gerçekleştirmişlerdir [3]. Şahin ve arkadaşları ise Bozkurt Optimizasyonu (Grey Wolf Optimizer GWO) algoritmasını problemin çözümü için düzenlemişler ve algoritmanın problemi başarı ile çözebildiğini göstermişlerdir [19].

Bu çalışmada basınç yayının minimum hacme göre optimum tasarımı yeni bir meta sezgisel optimizasyon algoritması olan YAA [20] ile gerçekleştirilmiş ve elde edilen sonuçlar önceki çalışmalar ile karşılaştırılmıştır. Çalışmanın ikinci bölümünde basınç yayının minimum hacme göre optimizasyonu problemi tanıtılmaktadır. Üçüncü bölümünde YAA ayrıntılı olarak anlatılmaktadır. Çalışmanın dördüncü bölümünde yapılan deneysel çalışmalar ve karşılaştırmalı sonuçlar paylaşılmıştır. Son bölümde sonuç ve öneriler ile çalışma sonlanmaktadır.

2. BASINÇ YAYININ MİNİMUM HACME GÖRE OPTİMİZASYONU (OPTIMIZATION of COMPRESSION SPRING ACCORDING TO MINIMUM VOLUME)

İlk defa Sandgren [16] tarafından tanımlanan basınç yayının minimum hacme göre optimizasyonu probleminde amaç statik yük altındaki basınç yayının hacmini en aza indirmektir. Şekil 3'te görüldüğü üzere problemi tanımlamak için yayın tel çapı ($d=x_1$), ortalama dış çap değeri ($D=x_2$) ve aktif sarım sayısı ($N=x_3$) olmak üzere 3 tasarım değişkeni kullanılmaktadır.



Şekil 1. Basınç yayı tasarımı
(The design of compression spring)

Bu tasarım değişkenlerinden D sürekli, N tamsayı ve d ise Tablo-1'de verilen 42 değerden birini alabilen ayrık bir değişkendir.

Tablo 1. Muhtemel yay tel çapları
(Possible spring wire diameters)

0.009	0.0095	0.0104	0.0118	0.0128	0.0132	0.014
0.015	0.0162	0.0173	0.018	0.020	0.023	0.025
0.028	0.032	0.035	0.041	0.047	0.054	0.063
0.072	0.080	0.092	0.105	0.120	0.135	0.148
0.162	0.177	0.192	0.207	0.225	0.244	0.263
0.283	0.307	0.331	0.362	0.394	0.4375	0.500

Problemin amaç fonksiyonu Eşitlik 1'de verilmiştir.

$$f(x) = \frac{\pi^2 x_2 x_1^2 (x_3 + 2)}{4} \quad (1)$$

Problemde tanımlı kısıtlar ise Eşitlik 2-9 arasında verilmiştir.

$$g_1(x) = \frac{8C_f F_{max} x_2}{\pi x_1^3} - S \leq 0 \quad (2)$$

$$g_2(x) = l_f - l_{max} \leq 0 \quad (3)$$

$$g_3(x) = d_{min} - x_1 \leq 0 \quad (4)$$

$$g_4(x) = x_2 - D_{max} \leq 0 \quad (5)$$

$$g_5(x) = 3.0 - \frac{x_2}{x_1} \leq 0 \quad (6)$$

$$g_6(x) = \sigma_p - \sigma_{pm} \leq 0 \quad (7)$$

$$g_7(x) = \sigma_p + \frac{F_{max} - F_p}{K} + 1.05(x_3 + 2)x_1 - l_f \leq 0 \quad (8)$$

$$g_8(x) = \sigma_w - \frac{F_{max} - F_p}{K} \leq 0 \quad (9)$$

Kısıtlarda kullanılan parametrelerin formülleri ise Eşitlik 10-13 arasında verilmiştir.

$$C_f = \frac{4\left(\frac{x_2}{x_1}\right) - 1}{4\left(\frac{x_2}{x_1}\right) - 4} + \frac{0.615x_1}{x_2} \quad (10)$$

$$K = \frac{Gx_1^4}{8x_3x_2^3} \quad (11)$$

$$\sigma_p = \frac{F_p}{K} \quad (12)$$

$$l_f = \frac{F_{max}}{K} + 1.05(x_3 + 2)x_1 \quad (13)$$

Problemin diğer tanımlamaları şu şekildedir: Yayın maksimum serbest uzunluğu $l_{max} = 14.0$ in; maksimum çalışma yükü $F_{max} = 1000.0$ lb; sıkıştırma için uygulanan

ön yük $F_p = 300.0$ lb; müsaade edilebilir maksimum kayma direnci $S = 189000.0$ psi; yayın minimum tel çapı $d_{min} = 0.2$ in; yayın maksimum dış çapı $D_{max} = 3.0$ in.; malzemenin kayma modülü $G = 11.5 \times 10^6$ psi; Ön yük altında izin verilen maksimum sapma $\sigma_{pm} = 6.0$ in.; Ön yük konumundan maksimum yük konumuna kadar olan sapma $\sigma_w = 1.25$ in.

Tasarım değişkenlerine (x_1, x_2, x_3) ait değer aralıkları ise aşağıdaki şekilde sınırlandırılmıştır:

$$0.2 \leq x_1 \leq 1, \quad 0.6 \leq x_2 \leq 3, \quad 1 \leq x_3 \leq 70$$

3. YAPAY ALG ALGORİTMASI (ARTIFICIAL ALGAE ALGORITHM)

YAA, Uymaz ve arkadaşları tarafından geliştirilen foto sentetik ökaryot bir tür olan alglerin yaşam davranışlarını temel alan bir optimizasyon yöntemidir [20]. Algoritmada ilham alınan algler ayrı çekirdek zarları ve klorofilleri bulunan tek hücreli alglerdir (mikroalgler). Daha çok deniz ve tatlı sularla yaşayan alglerin büyük bir kısmı kendi organik besinlerini ve enerjilerini kendileri üreten ototrof canlılardır. Besinlerini çoğunlukla inorganik karbon ve sudan elde etmektedirler. Yapılarındaki klorofil-a ve klorofil-b pigmentleri ile güneş enerjisinden kendi enerjilerini üretmekte ve oksijen açığa çıkartmaktadır. Algler besin üretmek için ihtiyaç duydukları maddelere erişmede ortam şartlarına göre doğal bir adaptasyon yeteneğine sahiptirler.

Alg popülasyon dinamiği iki kısımda değerlendirilmektedir. İlki fotosentez alımını içeren büyüme evresi ikincisi ise rekabet, sıyrılmaya, asalaklık ve ölümü içeren kayıp evresidir. Algler gün boyunca büyümek için ışığa doğru hareket etmekte, geceleri ise besin üretmek için karanlık dip seviyelere inmektedirler. Suda ilerleyen alglerin hareketi içinde bulunduğu sıvının eylemsizlik dayanımı ve kendisinin karşılaştığı viskoz sürtünmedir.

Alglerin büyümesi Monod modeli ile tarif edilmektedir. Bu modele göre spesifik büyüme hızı substrat konsantrasyonunun (S) bir fonksiyonu olarak tanımlanmaktadır (Eşitlik 14).

$$\mu = \frac{\mu_{max} S}{K_s + S} \quad (14)$$

Burada, μ_{max} maksimum spesifik büyüme hızı (1/zaman) ve K_s substrat yarı doygunluk sabitidir (kütle/hacim).

Yapay alg, gerçek alg gibi fotosentez yapabilmek için ışık kaynağına doğru helisel yüzme şeklinde hareket etmektedir. Diğer yandan yapay alg ortama adapte olabilmekte, baskın türü değiştirebilmekte ve mitoz bölünme ile çoğalabilmektedir. Alglerin yaşam özellikleri

temel alınarak geliştirilen YAA üç ana fazdan oluşmaktadır. Bunlar:

1. Evrimsel süreç
2. Adaptasyon
3. Helisel hareket

YAA'da her bir çözüm yapay alg kolonisi ile temsil edilmektedir. Popülasyon alg kolonilerinden oluşmaktadır ve kolonideki hücreler tek hücre gibi davranmaktadır. Kolonideki hücreler birlikte hareket etmekte, mitoz bölünme ile çoğalmakta ve uygun olmayan koşullarda ise ölmektedirler. Kolonideki hücrelerin yaşaması için uygun ortam yok ise koloni dağılabilmektedir. Dağılan hücreler daha sonradan uygun koşul bulursa yeni bir koloni oluşturabilmektedirler. Optimum hücrelerden oluşan alg kolonisi ise optimum alg kolonisi olarak tanımlanmaktadır (Eş. 15).

$$x_i = x_i^1, x_i^2, \dots, x_i^D$$

$$\text{Alg Koloni Popülasyonu} = \begin{matrix} x_1^1 & \dots & x_1^D \\ \dots & \dots & \dots \\ x_N^1 & \dots & x_N^D \end{matrix} \quad (15)$$

Burada, x_i^j , i . alg kolonisinin j . boyuttaki hücresidir. Problem boyutuna göre hücre sayısı değişmektedir. Her x_i , çözüm uzayında uygun bir çözümü temsil etmektedir. Her x_i , içerisinde yer alan hücrelerin çözüm uzayında uygun bir yere birlikte hareket ettiği düşünülmektedir. En iyi çözüm bir alg kolonisinin ideal çözüme ulaşması ile gerçekleşmektedir [20].

3.1. Evrimsel Süreç (Evolutionary Process)

Yeterli ışık alan yapay alg hücresi büyüme ve mitoz bölünmeye benzer bir şekilde kendisine benzeyen iki yapay alg hücresi oluşturmaktadır. Bir alg kolonisinin büyümesi Eşitlik-1'de verilen Monod modeli temel alınarak hesaplanır. Buna göre $t+1$ anında i . alg kolonisinin büyüklüğü Eşitlik 16 ve Eşitlik 17 ile hesaplanır.

$$\mu_i^t = \frac{\mu_{max} f^t(x_i)}{\frac{G_i^t}{2} + f^t(x_i)} \quad (16)$$

$$G_i^{t+1} = G_i^t + \mu_i^t G_i^t \quad (17)$$

$$i = 1, 2, \dots, N$$

Burada μ_i^t , t anında i . alg kolonisinin özgül büyüme hızı, μ_{max} , maksimum büyüme hızı (1 kabul edilir), $f^t(x_i)$, t anında i . alg kolonisinin besin konsantrasyonu yani uygunluk değeri, G_i^t , t anında i . alg kolonisinin büyüklüğü, N ise popülasyondaki alg kolonisi sayısıdır. Monod modelindeki K yarı doygunluk sabiti olarak t anındaki alg kolonisinin büyüklüğünün yarısı kullanılmaktadır.

Başlangıçta tüm alg kolonilerinin büyüklükleri 1 olarak belirlenmektedir. Helisel hareket sonucu iyi çözüme ulaşan alg kolonileri büyümektedir. Popülasyondaki tüm alg kolonileri için bir çevrim tamamlandığında evrimsel süreç başlamaktadır. Evrimsel süreçte en küçük alg kolonisinin rasgele seçilen bir hücresinin yerine en büyük koloninin aynı boyutta yer alan bir hücresi kopyalanmaktadır. Bu süreç Eşitlik 18-20'deki gibi gösterilmektedir.

$$enbüyük^t = \max(G_i^t) \quad i = 1, 2, \dots, N \quad (18)$$

$$enküçük^t = \min(G_i^t) \quad i = 1, 2, \dots, N \quad (19)$$

$$enküçük_m^t = enbüyük_m^t \quad m = 1, 2, \dots, D \quad (20)$$

Burada, "enbüyük" en büyük alg kolonisini, "enküçük", en küçük alg kolonisini ve D ise problem boyutunu ifade etmektedir [20].

3.2. Adaptasyon (Adaptation)

Adaptasyon, kendi ortamında yeterince büyüemeyen alg kolonisinin hayatta kalabilmesi için aynı ortamdaki en büyük koloniye benzemeye çalıştığı süreçtir. Bu süreç alg kolonilerinin açlık seviyelerine göre yürütülmekte ve başlangıçta hepsi sıfır kabul edilmektedir. Bir çevrimde her alg kolonisi helisel hareketi gerçekleştirmektedir. Helisel hareket sonucu çözüm uzayında daha iyi çözüme varan alg kolonisinin açlık seviyesi değişmemektedir. Ancak daha kötü çözüme giden alg kolonisinin açlık seviyesi bir artmaktadır. Bir helisel hareket çevrimi sonucu açlık seviyesi en yüksek olan alg kolonisi adaptasyona tabi tutulmaktadır. Her çevrim sonucu adaptasyonun gerçekleşip gerçekleşmeyeceği A_p parametresi ile belirlenmektedir. Bu parametre [0,1] arasında seçilen bir sabittir ve her çevrimde rasgele 0 ile 1 arasında üretilen değer A_p parametresinden küçük ise adaptasyon gerçekleştirilmektedir (Eşitlik 21-22).

$$enAç^t = \max(A_i^t) \quad i = 1, 2, \dots, N \quad (21)$$

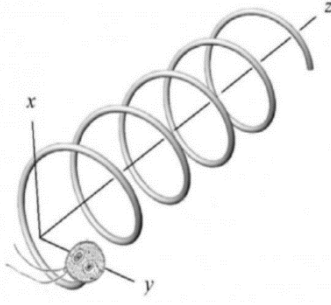
$$enAç^{t+1} = enAç + (enbüyük^t - enAç^t) \times rand \quad (22)$$

Burada, A_i^t , i . alg kolonisinin t anındaki açlık seviyesi, $enAç^t$ ise t anındaki en yüksek açlık seviyesine sahip alg kolonisi ve $rand$ ise 0 ile 1 arasında rasgele üretilen gerçek değerli bir sayıdır [20].

3.3. Helisel Hareket (Helical Movement)

Işığa yakın olmak isteyen algler yüzeyde daha fazla olan ışığa doğru hareket etmek istemektedirler. Algler kamçıları sayesinde helisel şekilde suda yüzerler (Şekil-2). Algin hareketi viskoz sürtünme ve yerçekimi olmak üzere iki etken tarafından engellenmektedir. Algler büyüdükçe sürtünme yüzeyleri artmakta ve suda daha yavaş hareket etmektedirler. Bu durumda algin yerel arama yeteneğini artırmaktadır (sömürü). Küçük alg hücrelerinin ise hareketi daha hızlıdır ve sıvı içerisindeki hareket mesafeleri daha büyüktür. Bu da çözüm uzayında adım büyüklüğünün fazla

olması demektir ki böylece algın global arama becerisi (keşif) daha da artmaktadır.



Şekil 2. Helisel hareket (Helical Movement)

Algler helisel hareket boyunca sahip oldukları enerjiyi kullanmaktadır. Her helisel hareket çevriminde alg kolonisinin çözüm uzayında kaç kez yer değiştireceği bu enerji ile belirlenmektedir. Her çevrim başında alg kolonilerinin enerjileri büyüklüklerinin [0, 1] aralığında normalleştirilmesiyle belirlenmektedir. Algoritmanın enerji kaybı parametresi (e) kullanılarak bir alg kolonisinin her hareket sonucu harcayacağı enerji miktarı belirlenmektedir. Her hareket sonunda eğer alg daha iyi bir çözüme ulaşırsa belirlenen enerji kaybı parametresinin yarısı kadar (e/2), eğer daha kötü bir çözüme varırsa enerji kaybı parametresi kadar enerji kaybına uğramaktadır.

Helisel hareketi engelleyen yerçekimi sıfır kabul edilmekte ve kesme kuvveti olarak viskoz sürtünme ele alınmaktadır. Viskoz sürtünme alg kolonisinin büyüklüğü (G) ile orantılıdır. Hesaplama yapılırken alg kolonisi küre şeklinde ve büyüklüğü de kürenin hacmi olarak kabul edilmektedir. Böylece sürtünme yüzeyi yarım kürenin yüzey alanı (τ) olmaktadır (Eşitlik 23).

$$\tau(x_i) = 2\pi \left(\sqrt[3]{\frac{3G_i}{4\pi}} \right)^2 \quad (23)$$

Burada, $\tau(x_i)$, *i*. alg kolonisinin sürtünme yüzeyidir.

Helisel hareket için rasgele 3 boyut belirlenmektedir. Boyutlardan biri doğrusal hareketi (Eşitlik-24) ve diğer iki boyut ise açısız hareketi sağlamaktadır (Eşitlik 25 ve 12). Eşitlik 24 tek boyutlu problemlerde kullanılmakta ve alg kolonisi doğrusal hareket gerçekleştirmektedir. Problem iki boyutlu ise hareket sinüzoidaldir ve Eşitlik-24 ve 25 kullanılmaktadır. Problem boyutunun 3 ve üzeri olması durumunda hareket heliseldir ve Eşitlik 24-26 kullanılmaktadır. Helisel hareketin adım büyüklüğü alg kolonisinin ışık kaynağına olan uzaklığı ve sürtünme yüzeyi ile belirlenmektedir.

$$x_{im}^{t+1} = x_{im}^t + (x_{jm}^t - x_{im}^t)(\Delta - \tau^t(x_i))p \quad (24)$$

$$x_{ik}^{t+1} = x_{ik}^t + (x_{jk}^t - x_{ik}^t)(\Delta - \tau^t(x_i)) \cos \alpha \quad (25)$$

$$x_{iz}^{t+1} = x_{iz}^t + (x_{jz}^t - x_{iz}^t)(\Delta - \tau^t(x_i)) \sin \beta \quad (26)$$

Burada x_{im}^t , x_{ik}^t , x_{iz}^t , *i*. alg kolonisinden rasgele seçilen 3 alg hücresi (*m*, *k* ve *z*. hücreler), x_{jm}^t , x_{jk}^t , x_{jz}^t ışık kaynağı olarak belirlenen popülasyondaki farklı bir alg hücresinin seçilen boyutlardaki alg hücreleri, $\alpha, \beta \in [0, 2\pi]$; $p \in [-1, 1]$; Δ kesme kuvveti katsayısı; $\tau^t(x_i)$ ise *i*. alg kolonisinin *t* anındaki sürtünme katsayısıdır [20]. Algoritmaya ait sözde kod Şekil 3'te verilmiştir.

```

Amaç fonksiyonu  $f(x)$ ,  $x = (x_1, x_2, \dots, x_d)$ 
Parametreleri belirle (kesme kuvveti  $\Delta$ , enerji kaybı  $e$  ve adaptasyon parametresi  $Ap$ )
n adet alg kolonisinden oluşan popülasyonu rastgele çözümler ile başlat
Her bir alg kolonisinin büyüklüğünü 1 ve açıklık değerini 0 olarak tanımla
Her bir alg kolonisinin amaç fonksiyonu değerini ve büyüklüğünü (G) hesapla
While (t < MaksimumHesaplama)
  Alg kolonilerinin enerji (E) ve sürtünme yüzeylerini ( $\tau$ ) hesapla
  For i=1:n
    Açlık = 1
    While ( $E(x_i) > 0$ )
      Amaç fonksiyon değerlerine göre turnuva yöntemi ile j alg kolonisini (ışık kaynağını) seç
      Helisel hareket için rastgele üç boyut (alg hücresi) (k, l ve m) seç
       $x_{im}^{t+1} = x_{im}^t + (x_{jm}^t - x_{im}^t)(\Delta - \tau^t(x_i))p$ 
       $x_{ik}^{t+1} = x_{ik}^t + (x_{jk}^t - x_{ik}^t)(\Delta - \tau^t(x_i)) \cos \alpha$ 
       $x_{iz}^{t+1} = x_{iz}^t + (x_{jz}^t - x_{iz}^t)(\Delta - \tau^t(x_i)) \sin \beta$ 
       $\alpha, \beta \in [0, 2\pi]$ ;  $p \in [-1, 1]$ ;
      Yeni çözümü hesapla
       $E(x_i) = E(x_i) - (e/2)$  hareket kaynaklı enerji kaybı
      if yeni çözüm daha iyi, i alg kolonisini güncelle ve  $iStarve = 0$ 
      else  $E(x_i) = E(x_i) - (e/2)$  metabolizma kaynaklı enerji kaybı end if
    end While
    if  $iAçlık=1, A(x_i)$  açıklık seviyesini bir arttır end if
  end For
  Alg kolonilerinin büyüklüklerini (G) hesapla
  Rastgele belirlenen r boyutunda en büyük alg kolonisinin hücresini en küçük alg kolonisinin hücresine kopyala
   $enküçük^t = enbüyük^t$ 
  En aç alg kolonisini seç
  if  $rand < Ap$ 
     $enAç^{t+1} = enAç^t + (enbüyük^t - enAç^t) \times rand$ 
  end if
  En iyi çözümü sakla
End While

```

Şekil 3. Yapay Alg Algoritması sözde kodu [20] (Artificial Algae Algorithm pseudo code)

4. SONUÇLAR VE TARTIŞMALAR (RESULTS AND DISCUSSIONS)

Bu bölümde, YAA'nın problemin çözümündeki başarımı karşılaştırmalı olarak ölçülmüş ve değerlendirilmiştir. Bu kapsamda YAA'dan elde edilen en iyi çözüme ait değerler önceki çalışmalarla elde edilen en iyi çözümlerle kıyaslanmıştır. Ancak önceki çalışmalarda yeterli istatistiki değerlendirmeye rastlanamamıştır. Bu sebeple, ayrıntılı ve adil bir değerlendirme için problem GWO ile de çözümlenerek sonuçlar kıyaslanmıştır. GWO, yakın zamanda optimizasyon problemlerine yaygın olarak uygulanan yöntemlerden birisi olması sebebiyle seçilmiştir [21].

YAA ve GWO'nun probleme uygulanması için gerekli kodlar MATLAB 2016b yazılımı üzerinde hazırlanmış ve yürütülmüştür. Ölçümler 64 bitlik Windows 10 işletim sistemi kullanan Intel(R) Core(TM) i7 2.66 GHz CPU ve 8 GB RAM kapasiteli bir bilgisayar üzerinde gerçekleştirilmiştir. Algoritmanın popülasyon büyüklüğü 30, yineleme sayısı 300 olarak belirlenmiştir. Bu bağlamda maksimum uygunluk hesaplama sayısı 9000 olarak sınırlandırılmıştır. YAA'nın kendi parametrelerinden olan kesme kuvveti 2, adaptasyon parametresi 0.5 ve enerji kaybı parametresi ise 0.3 olarak uygulanmıştır. Her iki algoritma 100 kez çalıştırılarak sonuçlar toplanmıştır.

Tablo 2'de elde edilen en iyi değerler önceki çalışmaların sonuçlarıyla kıyaslamalı olarak sunulmaktadır. Bu değerlere göre YAA problemin bilinen en iyi değerine

ulaşabilmektedir. Bunun yanında algoritmaların maksimum uygunluk değeri hesaplama (Function Evaluation FE) sayılarına bakıldığında YAA'nın maksimum uygunluk değeri hesaplama sayısı 7625 iken bu değer GWO algoritmasında 8979'dur. Bu da YAA'nın en

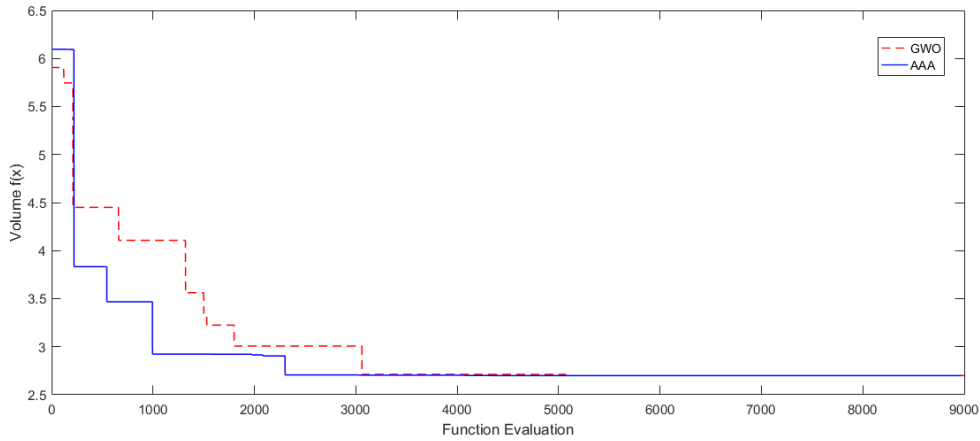
yakın rakibi GWO algoritmasından aynı sonucu bulmak için yaklaşık %24 oranında daha az değerlendirme yaptığını göstermektedir. Dolayısıyla YAA'nın GWO algoritmasına göre çözüme daha erken yakınsadığı açıkça görülmektedir.

Tablo 2. Basınç yayı için bulunan en iyi çözümler (Optimum solutions for compression spring)

Tasarım Değişkenleri	Sandgren (1990)	Deb ve Goyal (1997) GeneAS	Lampinen ve Zelinka (1999) DE	He ve ark. (2004) PSO	GWO	YAA
$x_1 (d)$	0.283	0.283	0.283	0.283	0.283	0.283
$x_2 (D)$	1.180701	1.226	1.223041	1.223041	1,223042604	1,22304101
$x_3 (N)$	10	9	9	9	9	9
$g_1(x)$	-54309	-713.510	-1008.8114	-1008.8114	-1008,64452	-1008,8114
$g_2(x)$	-8.8187	-8.933	-8.9456	-8.9456	-8,94562873	-8,94563571
$g_3(x)$	-0.08298	-0.083	-0.083	-0.083	-0,083	-0,083
$g_4(x)$	-1.8193	-1.491	-1.777	-1.777	-1,7769574	-1,77695899
$g_5(x)$	-1.1723	-1.337	-1.3217	-1.3217	-1,32170531	-1,32169968
$g_6(x)$	-5.4643	-5.461	-5.4643	-5.4643	-5,46428362	-5,46428571
$g_7(x)$	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
$g_8(x)$	0.0000	-0.009	0.0000	0.0000	-4,8863E-06	-1,1102E-15
$f(x)$	2.7995	2.665	2.65856	2.65856	2,65856263	2,658559166
Uygunluk Değeri Hesaplama Sayısı	N/A	N/A	26000	15000	8979	7265

Şekil 4'te GWO ve YAA'nın yakınsama eğrileri verilmiştir. Şekilde YAA'nın GWO'ya göre daha hızlı

yakınsadığı ve daha yüksek yakınsama oranına sahip olduğu görülmektedir.



Şekil 4. YAA ve GWO yöntemlerinin performans eğrileri
(Performance curves of AAA and GWO methods)

Tablo 3'te elde edilen sonuçların istatistiksel değerlendirmesi sunulmuştur. YAA, gerçekleştirilen 100 çalışmanın tamamında geçerli sonuç üretirken, GWO 98 geçerli çözüm üretebilmiştir. GWO'ya ait geçersiz sonuçlar istatistiksel değerlendirmeye katılmamıştır. YAA en iyi ve en kötü uygunluk değerleri açısından GWO'ya nazaran daha iyi sonuçlar vermiştir. Algoritmanın tutarlılığı açısından ortalama ve standart sapma metrikleri

önem arz etmektedir. YAA ortalama değer açısından GWO'nun gerisinde kalsa da aradaki fark göz ardı edilebilecek boyuttadır. Diğer taraftan YAA'ya ait standart sapma değeri GWO'dan daha iyi sonuç vermiştir. Genel olarak bakıldığında, YAA'nın daha tutarlı olduğu değerlendirilebilir.

Tablo 3. Deneysel çalışmaya ait istatistiksel sonuçlar
(Statistical results of experimental work)

	GWO	YAA
Elde edilen Geçerli Sonuç Oranı	0,98	1,00
En İyi Uygunluk Değeri (Best Fitness)	2,65856263	2,65855916
En Kötü Uygunluk (Worst Fitness)	5,745442158	3,95978115
Ortalama Uygunluk (Average Fitness)	2.730815938	2.82657160
Standart Sapma (Standard Deviation)	0.313048907	0.26194741
Ortalama Çalışma Zamanı (ms)	1050	1247

Algoritmaların çalışma zamanlarının ortalamasına bakıldığında ise GWO'nun daha hızlı olduğu görülmektedir. Ancak çalışma zamanı, en iyi değeri bulma süresi açısından değerlendirildiğinde, YAA'nın daha düşük uygunluk değerlendirme sayısı ile en iyi sonuca daha hızlı ulaştığı görülmektedir.

5. DEĞERLENDİRME (CONCLUSION)

Bu çalışmada YAA, basınç yayının minimum hacme göre tasarımının optimizasyonu problemine ilk defa uygulanmıştır. Elde edilen sonuçlar değerlendirildiğinde YAA'nın bilinen en iyi sonuca erişebildiği görülmüştür. Ayrıca YAA'nın yüksek yakınsama oranı ve tutarlılığa sahip olduğu görülmüştür. Bununla birlikte çalışma zamanı açısından görece zafiyetinin olduğu saptanmıştır. YAA, çözüm bulmaya yönelik istatistiksel başarımı, tutarlılığı ve yakınsama oranlarıyla tasarım optimizasyonu problemlerinin çözümlerine başarıyla uygulanabilecek bir yöntem olduğunu göstermektedir.

KAYNAKLAR (REFERENCES)

- [1] J. S. Arora, "Introduction to Optimum Design", *Waltham: Elsevier*, 2004.
- [2] R. V. Rao, V. J. Savsani, D. P. Vakhaira, "Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems", *Computer Aided Design*, 43, 303-315, 2011.
- [3] S. He, E. Prempain, Q. H. Wu, "An improved particle swarm optimizer for mechanical design optimization problems", *Engineering Optimization*, 36(5), 585-605, 2004.
- [4] M. Dörterler, İ. Şahin, H. Gökçe, "A grey wolf optimizer approach for optimal weight design problem of the spur gear", *Engineering Optimization*, 1-15, 2018
- [5] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, "Optimization by simulated annealing", *Science*, 220 (4598), 671-680, 1983.
- [6] E. Bonebeau, M. Dorigo, G. Theraulaz, **Swarm intelligence: From natural to artificial systems**, Oxford university press, USA, 1999.
- [7] M. H. Calp, M. A. Akcayol, "Optimization of Project Scheduling Activities in Dynamic CPM and PERT Networks Using Genetic Algorithms", *Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 22, 615-627, 2018.
- [8] M. Dener, M. H. Calp, "Solving the exam scheduling problems in central exams with genetic algorithms", *Mugla Journal of Science and Technology*, 4, 102-115, 2018.
- [9] J. Kennedy, R. Eberhart, "Particle swarm optimization, in Neural Networks", **Proc. IEEE International Conf. on Neural Networks**, Perth, Australia, 1942-1948, 1995.
- [10] M. Dorigo, M. Birattari, T. Stutzle, "Ant colony optimization". *Comput Intell Magaz.*, 1, 28-39, 2006.
- [11] D. Karaboga, B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm", *J Glob Optim.*, 39, 459-471, 2007.
- [12] I. Fister Jr., X. S. Yang, I. Fister, J. Brest, D. Fister, "A Brief Review of Nature-Inspired Algorithms for Optimization", *Elektrotehnikski Vestnik/Electrotechnical Review*, 80(3), 1-7, 2013.
- [13] S. Mirjalili, S. M. Mirjalili, A. Lewis, "Grey Wolf Optimizer", *Advances in Engineering Software*, 69, 46-61, 2014
- [14] H. Trabelsi, P. A. Yvars, J. Louati, M. Haddar, "Interval computation and constraint propagation for the optimal design of a compression spring for a linear vehicle suspension system", *Mechanism and Machine Theory*, 84, 67-89, 2015.
- [15] T. Yokota, T. Taguchi, M. Gen, "A solution method for optimal weight design problem of helical spring using genetic algorithms", *Computers Ind. Engineering*, 33, 71-76, 1997.
- [16] E. Sandgren, "Nonlinear integer and discrete programming in mechanical design optimization" *Journal of Mechanical Design*, 112, 223-229, 1990.
- [17] K. Deb, M. Goyal, "Optimizing engineering designs using a combined genetic", **In Seventh International Conference on Genetic Algorithms**, Ed. I. T. Back, Michigan State University, East Lansing, 512-528, 1997.
- [18] J. Lampinen, I. Zelinka, "Mixed integer-discrete-continuous optimization by differential evolution". **5th International Mendel Conference on Soft Computing**, Czech Rep., 71-76, 1999.
- [19] İ. Şahin, M. Dörterler, H. Gökçe, "Optimum Design of Compression Spring According to Minimum Volume Using Grey Wolf Optimization Method", *Gazi Journal of Engineering Sciences*, 3(2), 21-27, 2017.
- [20] S. A. Uymaz, G. Tezel, E. Yel, "Artificial Algae Algorithm (AAA) for nonlinear global optimization", *Applied Soft Computing*, 31, 153-171, 2015.
- [21] H. Faris, I. Aljarah, M. A. Al-Betar, S. Mirjalili, "Grey wolf optimizer: a review of recent variants and applications", *Neural Computing and Applications*, 20(2), 413-435, 2018