All Sciences Proceedings
http://as-proceeding.com/

2nd International Conference on Engineering, Natural and Social Sciences

April 4-6, 2023 : Konya, Turkey

https://www.icensos.com/

Meta-Heuristik Optimizasyon Algoritmalarının Sistem Tanımlama Problemine Uygulanması ve Performans Karşılaştırması

Metin ZALOĞLU[1*], Şehmus FİDAN [2] ve Emre ERKAN [3]

[1]Elektrik Elektronik Mühendisliği / Lisansüstü Eğitim Enstitüsü, Batman Üniversitesi, Türkiye
[2]Teknik Bilimler MYO/Elektronik Programı, Batman Üniversitesi, Türkiye
[3] Teknik Bilimler MYO/Elektronik Programı, Batman Üniversitesi, Türkiye

[*](metinzaloglu@gmail.com)

Özet **–** Son yıllarda meta-heuristik optimizasyon algoritmalarının sayısında önemli oranda bir artış yaşanmıştır. Çok çeşitli problemlerin optimal çözümünde kullanılan meta-heuristik algoritmaların evrimsel, sürü, fiziksel, insan, biyolojik, matematik ve sistem tabanlı olmak üzere çeşitli türleri bulunmaktadır. Bir sistemin matematiksel modelini elde etmek çoğu zaman zahmetli, birçok ihmale dayanan ve karmaşık bir süreçtir. Giriş çıkış verilerinin analiz edilerek bir matematiksel modelin elde edilmesi süreci olarak ifade edilen ve kontrol sistemleri mühendisliğinin önemli alt dallarından biri olan sistem tanımlama yöntemleri sayesinde basit deneyler ile belirtilen zorlukları aşmak mümkündür. Bu çalışmada; yapay ekosistem (AEO), çiçek tozlaşma (FPA), güve-alev (MFO), halat çekme (TWO) ve karınca aslanı algoritması (ALO) gibi çeşitli meta heuristik optimizasyon algoritmaları sistem giriş çıkışından toplanan verileri kullanarak bir transfer fonksiyonunu elde etmek için kullanılmıştır. Böylece meta heuristik algoritmalarının sistem tanımlama problemlerinden biri olan transfer fonksiyonlarının da elde edilebilmesi için kullanılabileceği ilk defa gösterilmiştir. Belirtilen algoritmalar yaygın bir geliştirici ekosistemine sahip olması ve düşük özellikli donanımlarda bile başarılı bir şekilde çalıştırılabilmesinden dolayı Python programlama dili kullanılarak gerçekleştirilmiştir. Önerilen algoritmaların performans karşılaştırmasını yapabilmek için zaman sınırlılığı, maksimum jenerasyon kıstası, erken durdurma kriteri ve maksimum fonksiyon hesaplama sınırlılıkları ele alınmış ve sonuçları sunulmuştur. Yapılan analizler sonucunda, önerilen meta heuristik algoritmalarının sistem tanımlama problemlerine kolaylıkla ve başarıyla uygulanabileceği görülmüştür.

Anahtar Kelimeler – Sistem Tanımlama, Meta-Heuristik Algoritmalar, Optimizasyon

I. GİRİŞ

Gerçek dünya problemleri için optimizasyon alanı giderek daha zorlu hale gelmekte ve onlarca yıldır araştırmacıların ilgisini çekmektedir. Bu problemlerin üstesinden gelmek için en sık kullanılan yöntemler sayısal yöntemlerdir, ancak bu yöntemler gradyan bilgisi gerektirir ve başlangıç noktalarına duyarlıdır, bu da global optimum aramayı zor ve kararsız hale getirmektedir. Son zamanlarda, geleneksel sayısal yöntemlerin etkili bir şekilde çözemediği, çoklu karar değişkenleri, karmaşık doğrusal olmayan kısıtlamalar ve amaç fonksiyonları olan karmaşık optimizasyon problemleri ortaya çıkmıştır. Ancak doğa, bu karmaşık sorunların üstesinden gelebilecek yapay zekâ yöntemleri geliştirmek için ilham vermektedir [1]. Meta-heuristik algoritmalar, çok çeşitli problemleri çözme amacıyla kullanılmaktadır ve çeşitli uygulamalar için tasarlandıklarından, geniş bir kullanım alanına sahiptirler. Bir şirketin üretim sürecini optimize etme, yapay zekâ veya veri madenciliği gibi birçok alanda kullanımları mümkündür. Bu noktada makine öğrenmesi matematiksel modellerin elde

edilmesi için kullanılmakla birlikte yapılan literatür çalışmalarında meta heuristik algoritmaların bir sistemin matematiksel modelini elde etmek içinde kullanılabileceği görülmüştür. Kontrol sistemlerinde bir sistemin giriş/çıkış arasındaki ilişkiyi matematiksel olarak göstermek için transfer fonksiyonlarının kullanımı mümkündür. Bunun için Ziegler-Nichols'un açık döngü veya kapalı döngü metotlarını kullanarak yaklaşık olarak transfer fonksiyonlarını elde etmek mümkündür. Böylece transfer fonksiyonu belirlendikten sonra bir kontrolör tasarlamak daha kolay olmaktadır [2][3].

Sistem tanımlama yöntemleri, gerçek sistem veya simülasyon üzerinden elde edilen veriler kullanarak sistemin matematiksel modelini üretmeyi amaçlar. Bu yöntemler, sistemlerin doğrusal veya doğrusal olmayan modellerini tanımlayabilirler ve kontrol teorisi, istatistik, sinyal işleme gibi bilim dallarında sıklıkla kullanılmaktadırlar. Sistem tanımlama için parametrik ve non-parametrik yöntemler geliştirilmiştir. Parametrik yöntemler, sistemin belirli bir matematiksel modele sahip olduğu varsayımıyla çalışır. Bu yöntemler, sistemin parametrelerini tahmin etmek için giriş-çıkış verilerini kullanır. [4]

Yapay ekosistem tabanlı optimizasyon (AEO), canlı organizmaların üretim, tüketim ve ayrışma davranışlarını taklit ederek optimizasyon problemlerini çözmek için kullanılan bir yöntemdir. Bu yöntem, matematiksel problemleri ve gerçek dünya mühendislik problemlerini çözmek için kullanılabilir. AEO algoritması PV modül parametrelerinin hesaplanmasında kullanılmış ve başarılı sonuçlar elde edilmiştir [5] Omotoso vd. (2022) Hibrid enerji sistemlerinin en uygun boyutlandırması için AEO algoritmasını kullanmış ve diğer yöntemlerle karşılaştırmalar yapmıştır [6].

Abdel-Basset vd. bitkilerdeki çiçeklerin yayılma rolünden esinlenilerek geliştirilmiş bir hesaplama zekası olan Çiçek Tozlaşma Algoritması (FPA) önermiştir. Bu algoritmayı Non-parametrik Friedman testi ile istatistiksel olarak analiz edilerek performansının daha üstün olduğunu belirtmiştir [7]. FPA, çoğu evrimsel algoritma gibi, erken yakınsama sorunu yaşamaktadır. Bu nedenle, makalede adalar modeli kullanılarak FPA'nın çeşitliliği korunacak şekilde geliştirilmesi önerilmektedir [8]. Kopciewicz, P. vd. (2020), meta-sezgisel arama stratejilerinin son zamanlardaki büyümesinin, hesaplama optimizasyonu alanında büyük bir ilerleme getirdiğini belirtmektedir. Yaptıkları çalışmada, çiçek tozlaşması sürecini taklit eden bir modifiye edilmiş bir optimizasyon tekniği sunulmaktadır [9]. Karınca aslanı optimizasyonu (ALO) çok amaçlı optimizasyon problemlerini ele almak için iki sorunu çözmesi gerektiği belirtilmiştir. Pareto eşitliği ve bireylerin mesafe bilgilerinin birleştirilmesiyle yeni bir ölçüm sunulmuş ve bu ölçüm ilk sorunu çözmek için kullanılmıştır [10]. Diğer bir çalışmada ALO adlı algoritma, fonksiyon optimizasyonu, kısıtlamalı mühendislik problemleri ve çok amaçlı fonksiyon optimizasyon problemleri gibi birçok alanda test edilmiş ve hızlı yakınsama ve yüksek hassasiyetle üstün performans göstermiştir [11]. Ayrıca ALO algoritmalarını makine öğrenimi, ağ uygulamaları, mühendislik uygulamaları, yazılım mühendisliği ve görüntü işleme gibi alanlara uygulayan çalışmalar mevcuttur [12].

Güve-Alev Optimizasyon (MFO - Moth-Flame Optimizasyon) algoritmasının erken olgunlaşma sorununa odaklanan çalışmalarda mevcuttur. Bir çalışmada keşif davranışını geliştirmek için alev füzyon mekanizması var olan yönteme entegre edilmiş ve algoritmayı geliştirmişlerdir [13] Kaur vd. (2020) doğadan esinlenen MFO algoritmasının keşif yapma kapasitesinin zayıf olduğunu, bu nedenle performansını artırmak için bir Cauchy dağılım fonksiyonu eklenmiş, en iyi alevin etkisi artırılmış, adaptif adım boyutu ve yineleme bölünmesi gibi değişiklikler önerilmiştir [14]. Diğer bir çalışmada, yazarlar hiperspektral görüntüleme (HSI) veri kümesi için MFO algoritmasını kullanarak bant seçimi yapmayı önermektedir [15]

Bu çalışmada ele alınan algoritmalardan biri de halat çekme optimizasyon (TWO) algoritmasıdır. Ghelichi vd. (2020) sivil yapılar için Halat Çekme Optimizasyonu (Tug of War Optimization - TWO) yöntemine dayanan yeni bir aktif nöro-bulanık optimize edilmiş kontrol algoritması önermektedir. Önerilen yöntem, hidrolik aktüatörler ve izolasyon mesnetleri ile donatılmış doğrusal olmayan bir karayolu köprüsünün tasarımı için kullanmıştır [16]. Kaveh ve Zolghadr (2016), TWO algoritmasını Newton mekanik yasalarını kullanarak aday çözümleri bir ip çekme yarışması takımı olarak ele alır ve çözüm

kalitesine göre takımların yeni konumlarını belirlemişlerdir [17].

Özet olarak;

1- Önerilen meta-heuristik algoritmalar ilk defa bir sistemi tanımlamak için kullanılan transfer fonksiyonları üretmek için kullanılmıştır.

2- Yapılan karşılaştırmalarda AEO algoritması daha iyi performans değerine ulaşmıştır.

3- Yaygın geliştirici ekosistemine sahip olması ve küçük donanımlarda bile başarıyla çalışabildiği için Python programlama dili tercih edilmiştir.

4- Erken durdurma sınırlılığı, maksimum jenerasyon sınırlılığı, zaman sınırlılığı ve fonksiyon hesaplama sınırlılığı kıyaslamaları yapılmış ve performansları değerlendirilmiştir.

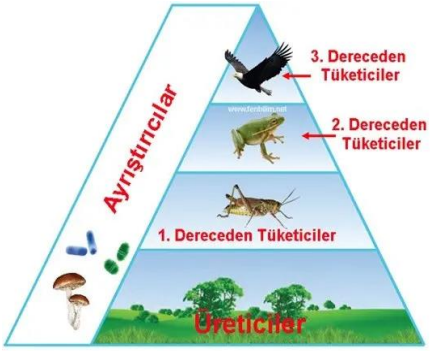
5- Ele alınan sistem için parametre değişimleri ve güvenilirlikleri incelenmiştir.

II.MATERYAL VE YÖNTEM

Alt bölümler boyunca bu makalenin konusu olan sistem tanımlama ve optimizasyon algoritmalarından bahsedilmiştir.

A. YAPAY EKOSISTEM OPTIMIZASYONU (AEO)

Yapay ekosistem tabanlı optimizasyon (AEO) adlı doğadan ilham alan yeni bir meta-sezgisel optimizasyon algoritmasıdır. Doğal ekosistemdeki enerji akışından motive edilen popülasyona dayalı bir optimize edicidir ve bu algoritma, üretim, tüketim ve ayrışma dahil olmak üzere canlı organizmaların üç benzersiz davranışını taklit etmektedir. Üreticiler genellikle otobur tüketiciler ve omnivor tüketiciler için temel gıda sağlamaktadır. Tüketiciler, yiyeceklerini yapamayan hayvanlardır; bu nedenle enerji ve besin elde etmek için üreticilerden veya diğer tüketicilerden beslenmek zorundadırlar. Tüketiciler etobur, otobur ve omnivor olarak sınıflandırılabilir. Sadece üreticileri (bitkileri) yiyen hayvanlara otobur denir. Hem üreticileri hem de diğer hayvanları yiyen hayvanlara omnivorlar denir. Sadece diğer hayvanları yiyen hayvanlara etobur denir. Ayrıştırıcılar, hem ölü bitkiler (üreticiler) hem de hayvanlar (tüketiciler) veya canlı organizmaların atıklarıyla beslenen bir organizmadır.[1] Ayrıştırıcılar bakteri ve mantardan oluşur. Bir organizma öldüğünde, ayrıştırıcılar kalıntıları parçalar ve onları karbondioksit, su ve mineraller gibi basit moleküllere dönüştürmektedir. AEO'ya dayalı bir ekosistemin temsili Şekil 1'de gösterilmektedir.



Şekil-1. Üretici, Tüketici ve Ayrıştırıcılar

Üretim operatörünün matematiksel modeli aşağıdaki gibi temsil edilir:

$$x_1(t+1) = (1-a)x_n(t) + ax_{rand}(t) \quad (1)$$

$$a = \left(1 - {}^{t}/_{T}\right)r_1 \quad (2)$$

$$x_{rand} = r(U-L) + L \quad (3)$$

Tüketimi modelleyen denklem ise aşağıdaki gibidir:

$$x_i(t+1) = x_i(t) + C\left(x_i(t) - x_j(t)\right), \quad i \in [3, \ldots\ldots, n] \quad j = randi([2i-1]) \quad (4)$$

Bir omnivorun tüketim davranışını modelleyen matematiksel denklem şu şekilde ifade edilebilir:

$$x_i(t+1) = x_i(t) + C\left(r_2\left(x_i(t) - x_1(t)\right) + (1-r_2)\left(x_i(t) - x_j(t)\right)\right), \quad i \in [3, \ldots\ldots, n] \quad (5)$$

Burada C, [0,1] aralığında rastgele bir sayıdır. Bu tüketim operatörü için, AEO, bir popülasyondaki en kötü bireye veya rastgele seçilen bir bireye veya her ikisine göre arama yapan bireyin konumunu güncellemektedir. Ayrışma, bir ekosistemin işleyişi açısından çok hayati bir süreçtir ve denklemdeki gibi gösterilir.

$$x_i(t+1) = x_n(t) + D\left(ex_n(t) - hx_i(t)\right), \quad i = 1, \ldots\ldots, n \quad (6)$$

$$D = 3u, \quad u \sim N(0,1) \quad (7)$$

$$e = r_3 randi([1\ 2]) - 1 \quad (8)$$

$$h = 2r_3 - 1 \quad (9)$$

B. ÇIÇEK TOZLAŞMAS ALGORITMASI (FPA)

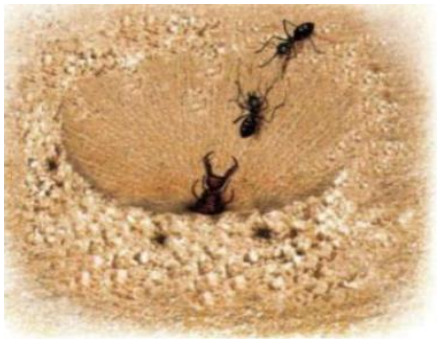
Çiçek tozlaşma algoritması FPA, çiçekli bitkilerin üremesiyle ilgili doğadan ilham alan bir optimizasyon yöntemidir. FPA'nın temel matematiksel ifadesi denklem 10'da gösterilmiştir. [7]

$$x^{t+1} = x_i^t + \gamma L(\lambda)(g_* - x_i^t) \quad (10)$$

Burada x^{t+1} çözüm vektörü, g* mevcut en iyidir. γ adım boyutunu ayarlama faktörüdür.

C. KARINCA ASLANI ALGORİTMASI (ALO)

Karınca aslanları, Myrmeleontidae ailesinden olan yırtıcı bir böcek türüdür. Larva evresindeki beslenme davranışlarından dolayı isimlerini almışlardır. Karıncaların bulunduğu bölgelere koni şeklinde tuzaklar oluşturarak karıncaları beklerler. Karıncalar tuzaktan çıkmasını engellemek için kum fırlatırlar ve tuzağın dibine çektikleri karıncaları büyük çeneleri ile yutarlar. Her avdan sonra tuzaklarını yeniden hazırlarlar[18]. Bu mekanizma Şekil 2'de gösterilmektedir.



Şekil-2. Karınca aslanı avlanma stratejisi.

Bu ilginç avlanma mekanizmasına ait matematiksel model rastgele yürüyüşlerle başlar ve aşağıdaki denklemler ile gösterilir.

$$c_i^t = Antlion_i^t + c^t \quad (11)$$

$$d_i^t = Antlion_i^t + d^t \quad (12)$$

$$c^t = c^t . I^{-1} \quad (13)$$

$$d^t = d^t . I^{-1} \quad (14)$$

Burada I kaydırma oranını göstermektedir ve optimizasyon sırasında belirli oranlarda arttırılır. Bu mekanizmanın ayrıntıları Mirjalili'nin çalışmasında [19] bulunabilir.

D. GÜVE-ALEV ALGORİTMASI (MFO)

MFO algoritmasının ana fikri, birçok güve türünün ışık kaynaklarının etrafında yaptığı hareketleri taklit etmektedir. Bu hareketler, güvelerin ışık kaynağına olan uzaklıklarını en aza indirmeyi amaçlamaktadır. Bu amaç doğrultusunda, MFO algoritması, popülasyon içerisindeki güvelerin pozisyonlarını değiştirerek en iyi çözüme ulaşmayı hedeflemektedir [20]. MFO algoritmasının denklemi şöyle ifade edilmektedir.

$$S(M_i, F_j) = D_i . e^{bt} . Cos(2\pi t) + \frac{a * F_j + (1-a) * en\ iyi\ alev}{2} \quad (15)$$

Burada "a" parametresi lineer azalan fonksiyondur ve en iyi ve karşılık gelen alevin etkisini kontrol eden kontrol faktörü olarak kullanılır.

$$a = 0.6 - 0.4 * \frac{L}{T} \quad (16)$$

E. HALAT ÇEKME OPTİMİZASYONU (TWO)

Halat çekme, iki rakip takımın zıt uçlarını kavrayarak bir güç yarışması yapmasıdır. Oyunun temel kuralları, takımların karşı taraftaki oyuncuları kendi tarafına çekmesini ve ipi belirli bir mesafeye kadar çekmesini içerir. Genellikle iki takım arasında oynanır ve her takımın eşit sayıda oyuncusu olur. Yarışma, belirli bir alanda gerçekleştirilir ve belirli bir süre içinde kazanan takım, rakibinin yer değiştirmesini sağlar. Halat çekme, dünya genelinde popüler bir etkinlik olarak kabul edilir ve özellikle okul etkinlikleri, şirket piknikleri ve aile etkinlikleri gibi grup etkinliklerinde yaygın olarak oynanır [21]. Şekil 3' te çekişmede yarışan takımlardan birini göstermektedir.

Şekil-3. Halat çekmede yarışan bir takım

TWO algoritmasında bileşke kuvvet şu şekilde hesaplanabilir:

$$F_r = F_p - W_i \mu_k \quad (17)$$

Sonuç olarak, i nesnesi j nesnesine doğru Newton'un ikinci kanununa göre saniyesine hızlanır:

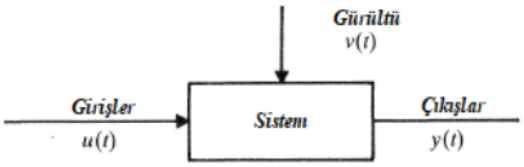
$$a = \frac{F_r}{\left(W_i/g\right)} \tag{18}$$

i nesnesi sıfır hızdan başladığı için, yeni konumu şu şekilde belirlenebilir:

$$X_i^{yeni} = \frac{1}{2}at^2 + X_i^{eski} \tag{19}$$

F. Sistem Tanımlama

Sistem tanımlama, matematiksel modelini tam olarak çıkaramadığımız sistemlerde deneysel verileri kullanarak sistemi modelleme işlemidir. Sistem tanımlama uygulamaları kontrol mühendisliği alanının önemli bir alt dalıdır. Bir sistemin girişler ve gürültülerle etkileşime girerek bir çıkış ürettiğini ve bu süreci hesaplamak için sistemin bir modelinin olması gerekmektedir. Sistemin girişleri kontrol edilebilir ancak gürültüler kontrol edilemez. Bu nedenle, sistemi hesaplamak için bir matematiksel model oluşturulması gerekmektedir [4]. Şekil 4 'te gösterilen sistem, bu matematiksel ifadenin gerçek sistemi temsil ettiği alanı ifade etmektedir.



Şekil-4. Temel sistem tasarımı

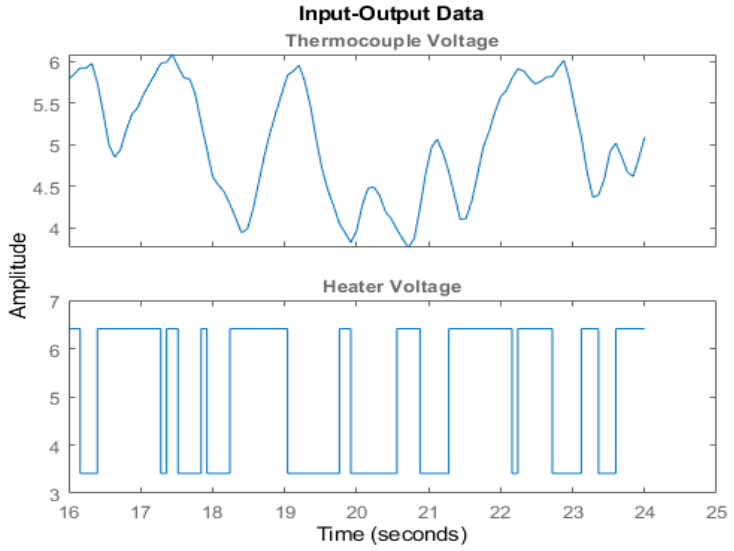
Siyah kutu sistem tanımlama

Siyah kutu sistem tanımlamada, sistemin önceden tanımlanmış bir matematiksel modeli yoktur ve model denklemler kullanılarak değil de sistem için tanımlanmış giriş ve çıkış verileri kullanılarak elde edilir. Tanımlanması gereken model şu şekildedir:

$$y(t) = f[y_1(t-1), \dots, y_1(t-n_{y_1}), \dots, y_r(t-1), \dots, y_r(t-n_{y_r}), u_1(t-1), \dots, u_1(t-n_{u_1}), \dots, u_m(t-1), \dots, u_m(t-n_{u_m}), e_1(t-1), \dots, e_1(t-n_{e_1}), \dots, e_r(t-1), \dots, e_r(t-n_{e_r})] \tag{20}$$

G. Ön Analiz

Algoritmaların kullanılmasında PC olarak 2.5 GHz hızında 6 çekirdekli Intel Core i5-3210M CPU, 6 Gb RAM ve 1 TB SSD kullanılmıştır. Yazılım olarak Python 3.8 sürümü ile birlikte mealpy 2.5.0, pandas 1.4.2 ve control 0.9.2 kütüphaneleri kullanılmıştır. IDE olarak Spyder 5.1.5 ve Jupyter Notebook 6.4.8 tercih edilmiştir. Deney seti için Matlabda bulunan bir veri seti kullanılmıştır ve bu veriler laboratuar ölçeğinde "hairdryer" deney setinden (Feedback's Process Trainer PT326) elde edilmiştir. Deney düzeneği hava sıcaklığı ölçümüne dayanmaktadır ve giriş olarak ısıtıcıya uygulanan gerilim, çıkış olarak da termokupl aracılığıyla ölçülen hava sıcaklığı kullanılmaktadır. Şekil 5'te ısıtıcı girişine uygulanan gerilime karşılık çıkışta ölçülen termokupl gerilimini göstermektedir.



Şekil-5. Isıtıcı girişine uygulanan gerilime karşılık çıkışta ölçülen termokupl gerilimi

III. BULGULAR

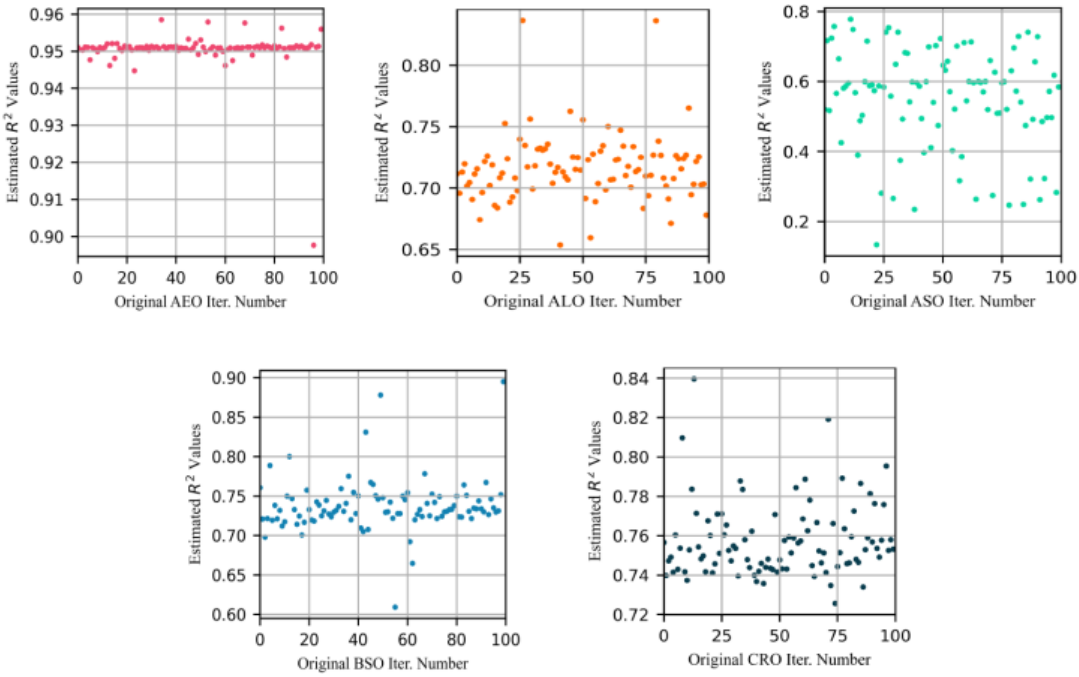
Bu çalışmada meta-heuristik algoritmalardan Yapay Ekosistem Optimizasyonu (AEO), Çiçek Tozlaşması Algoritması (FPA), Karınca Aslanı Algoritması (ALO), Güve-Alev Algoritması (MFO) ve Halat Çekme Optimizasyonu (TWO) ile ilgili durdurma kriterlerinin sınır değerlerine bağlı olarak performans değerleri karşılaştırılmış olup aşağıdaki sonuçlara varılmıştır.

A. GÜVENILIRLIK ANALIZI VE SINIRLILIKLAR

Performans odaklı birçok çalışmada durdurma kriterlerine yeterince önem verilmediğinden, doğru durdurma kriterleri seçiminin sistem modelinin tahmininde kritik bir öneme sahip olmaktadır. Özellikle sınırlı kaynaklara sahip gömülü donanımlar için minimum sürede çözüme ulaşılması önem arz etmektedir. AEO tabanlı algoritmaların performansının incelenmesinin yanı sıra, farklı durdurma kriterlerine bağlı olarak elde edilen sonuçları da karşılaştırmaktadır [22].

1. Zaman Sınırlılığı Performansı

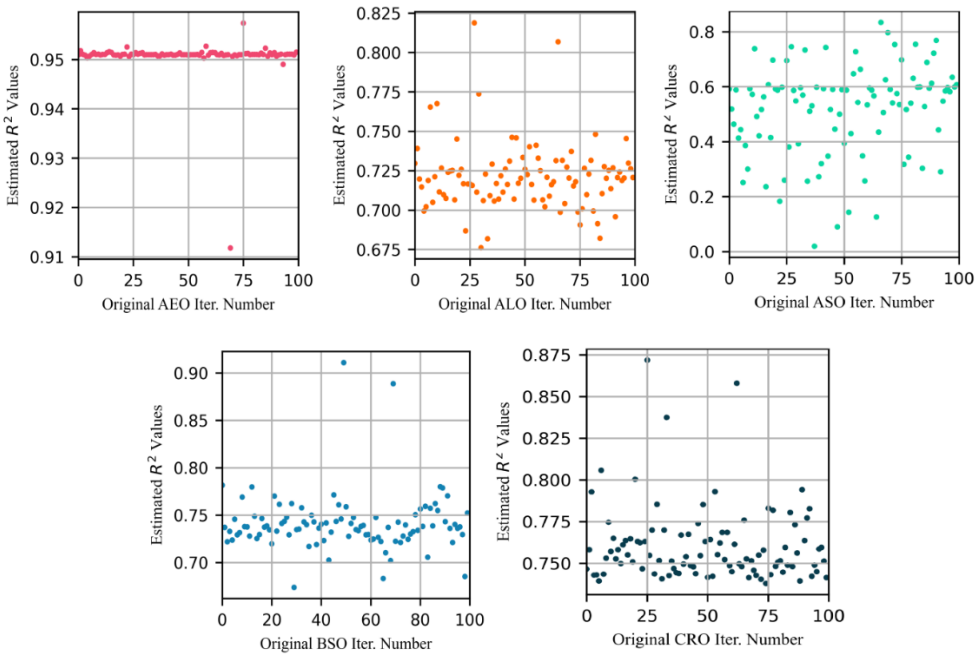
Ele alınan AEO ve diğer algoritmalar maksimum zaman sınırlılığı belirlenerek (45 saniye) test edilmiştir. 100 defa çalıştırıldıktan sonra çıkan karakteristik değerler Şekil 6' da sunulmuştur. Bu şekle göre Original AEO algoritması bu karşılaştırma sonucunda R^2 değerleri 0,94 – 0,96 arasında en yüksek ve kararlı R^2 değerlerine ulaştığı görülmüştür. Original CRO algoritmasında çoğunluk R^2 değerleri 0,72 – 0,84 etrafında toplandığından kararlı bir sonuç çıkmamıştır. Original ASO algoritması ise bu karşılaştırma sonucunda 0,2 - 0,8 değerleri arasında en düşük R^2 değerlerine ulaştığı görülmüştür.



Şekil-6. Zaman sınırlılıkları altında (TB-45) Algoritmalarının R^2 performansı

2. Maksimum Jenerasyon Sınırlılığı Performansı

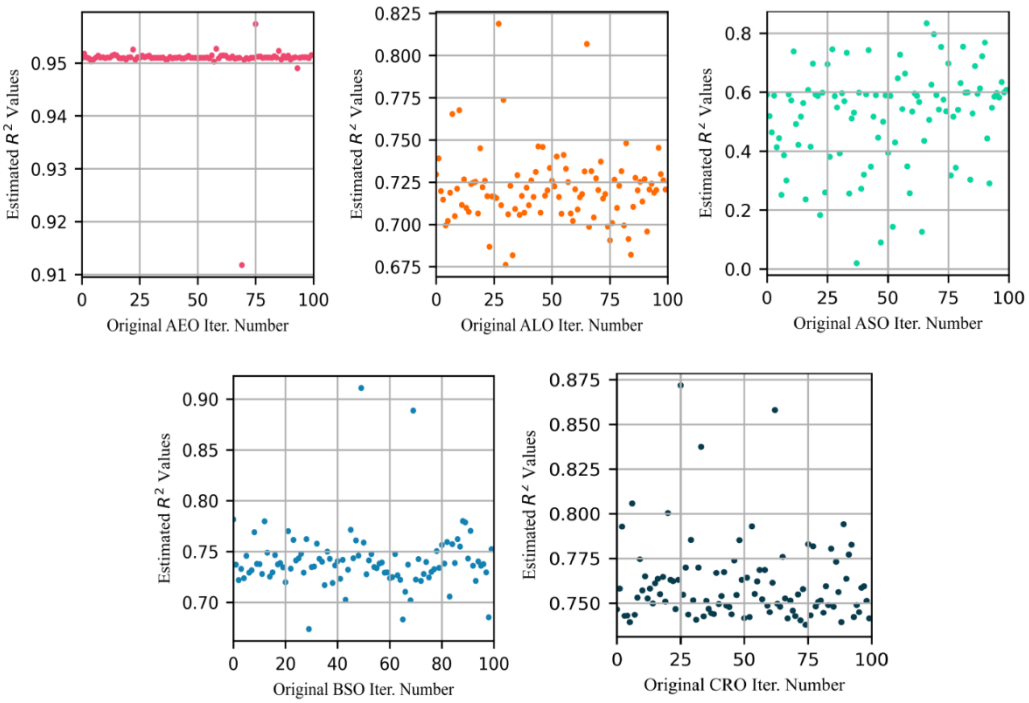
Maksimum jenerasyon sayısı (30) dikkate alınarak hesaplanan R^2 değerleri Şekil 7' de sunulmuştur. Bütün algoritmalar 100 defa çalıştırılmıştır. Original AEO algoritması bu karşılaştırma sonucunda 0,91 – 0,95 değerleri arasında en yüksek ve kararlı R^2 değerlerine ulaştığı görülmüştür. Original CRO algoritmasında 0,75 – 0,87 aralığında R2 değerlerine ulaştığı görülmüştür. Original ASO algoritması ise bu karşılaştırma sonucunda 0,0 - 0,8 değerleri arasında en düşük R^2 değerlerine ulaştığı görülmüştür.



Şekil-7. Maksimum jenerasyon (MG-30) sınırlılıkları altında algoritmalarının R^2 performansı

3. Fonksiyon Hesaplama Sınırlılığı

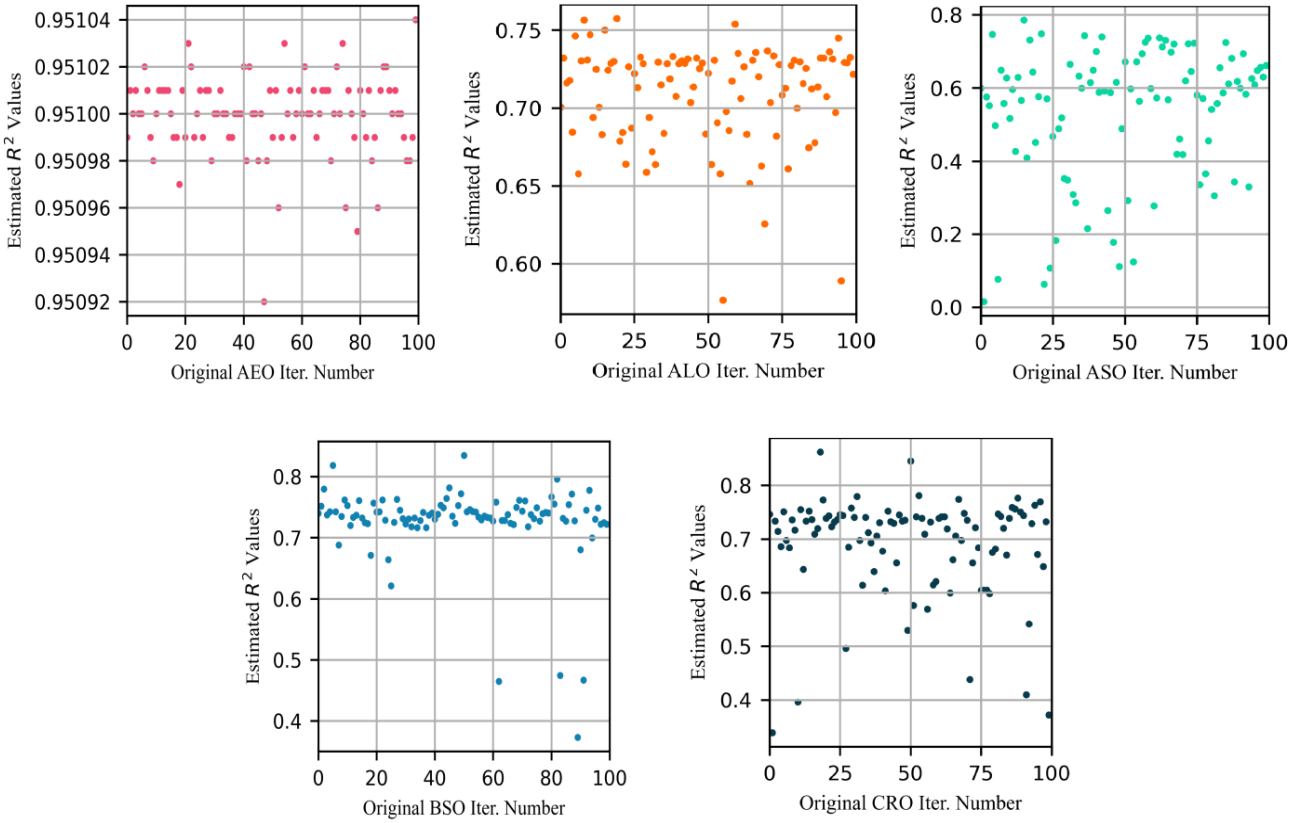
Fonksiyon hesaplama sınırlılığı dikkate alınarak yapılan hesaplamalarda algoritmalar 4000 defa çalıştırılarak performans sonuçları incelenmiştir. Bu değerin seçiminde algoritmaların en azından 100 çalıştırma boyunca 0.9 R^2 değerini geçmesi dikkate alınmıştır. Original AEO algotirması 100 çalışıtırmada 0.91 üstü R^2 performansına sahip olmuştur. Original ALO algoritması ise 100 çalıştırma boyunca 0,67 – 0,82 değerleri arasında olmuştur. Original ASO algoritması halen en düşük performanslara sahip algoritmadır. Maksimum fonksiyon hesaplama kriteri dikkate alınarak elde edilen R^2 değerleri Şekil 8' de sunulduğu üzere performansı en yüksek algoritma olarak karşımıza çıkmaktadır.



Şekil-8. Maksimum Fonksiyon Hesaplama sınırlılığı altında algoritmalarının R^2 performansı

4. Erken Durdurma Sınırlılığı

Erken durdurma (ES=3) kriteri, AEO algoritmalarının hesaplanan minimum hata değerinin 3 döngü boyunca hiç değişmemesi sonunda araştırma sürecinin bitirilmesi olarak tanımlanabilir. Bu kriter algoritmalara çözümü araştırmak için oldukça uzun sürmektedir. Dolayısıyla AEO algoritması lokal optimum noktalarına takılmadan sonucu en iyi hesaplayan sistem tanımlama problemleri için R^2 değerleri 0,95 etrafında değerler alarak en uygun niteliğe sahiptir. Şekil 9'da Original AEO algoritması 100 iterasyon sonunda kararlı bir şekilde global çözüm noktasını her seferinde belirlemeyi başarmıştır. Bu da AEO algoritmasını performans ve çözüm konusunda ön plana çıkarmaktadır. Original AEO algoritması bu süreçte daha başarılı gözükürken her bir iterasyon süresi ortalama olarak 138 saniye sürmüştür. Bu süre Original ALO için 13 sn, Original ASO için 8 sn, Original BSO için 41 sn, Origial CRO için 7 sn olarak hesaplanmıştır. Karşılaştırmalara sonucunda Original AEO algoritması yüksek çözüm süreleri ile dikkate alınırken diğer algoritmalar daha düşük süreler ile ortaya çıkmaktadır.



Şekil-9. Maksimum erken durdurma sınırlılığı altında algoritmaların R^2 performansı

5. Farklı Sınırlamalar Altında Performans İndekslerinin Karşılaştırılması

Zaman sınırlılığı temel olarak performans sonucuna bakılmaksızın verilen süre sonunda algoritmanın durdurulmasını temel almaktadır. Bu çalışmada verinin doğası, algoritmaların yapısı ve donanımsal kaynaklar dikkate alınarak süre sınırlaması 45 sn olarak belirlenmiştir. Tablo-1'de 5 farklı algoritma 100 iterasyon çalışması sonunda 5. döngüde çıkan sonuçları sunulmuştur.

Tablo 1- Zaman sınırlılıkları altında çözüm yeteneklerinin karşılaştırılması

Algoritmalar	MAPE	MAE	MSE	R^2	Time	Transfer Fonksiyonları
Original AEO	0,02843	0,13701	0,02924	0,94768	45,82	$T_f(s) = \frac{-1.50s + 12.65}{s^2 + 6.14s + 12.95}$
Original ALO	0,06798	0,33375	0,16516	0,7045	46,14	$T_f(s) = \frac{-10.81s + 97.34}{s^2 + 60.67s + 100}$
Original ASO	0,08282	0,39716	0,24365	0,56406	45,26	$T_f(s) = \frac{-5.03s + 74.96}{s^2 + 84.30s + 83.42}$
Original BSO	0,06721	0,32456	0,15681	0,71945	45,62	$T_f(s) = \frac{-7.21s + 57.30}{s^2 + 33.17s + 58.99}$
Original CRO	0,06597	0,31821	0,15216	0,72777	45,32	$T_f(s) = \frac{-9.97s + 73.91}{s^2 + 43.19s + 75.97}$

Maksimum jenerasyon sınırlandırma kriterinde, çalışmakta olan algoritma, verilen maksimum nesil sayısına ulaştığında durdurulur. Tablo 2'de maksimum jenerasyon sınırlılığı 30 olarak seçildikten sonra hesaplanan parametreler görülmektedir. Bu tabloda 5 farklı algoritma 100 iterasyon çalışması sonunda 5. döngüde çıkan sonuçları sunulmuştur. AEO algoritması en uzun çözüm süresine sahip olmasına rağmen en uygun değerler çıkmıştır.

Tablo 2- Maksimum jenerasyon sınırlılıkları altında çözüm yeteneklerinin karşılaştırılması

Algoritmalar	MAPE	MAE	MSE	R2	Time	Transfer Fonksiyonları
Original AEO	0,02744	0,13198	0,02737	0,95103	83.92	$T_f(s) = \frac{-1.49s + 12.59}{s^2 + 6.11s + 12.90}$
Original ALO	0,06947	0,33863	0,17168	0,69284	51,15	$T_f(s) = \frac{-11.27s + 97.26}{s^2 + 60.41s + 100}$
Original ASO	0,08194	0,39324	0,23909	0,57223	54,78	$T_f(s) = \frac{-10.03s + 73.01}{s^2 + 43.02s + 72.29}$
Original BSO	0,06299	0,30406	0,13791	0,75325	42,59	$T_f(s) = \frac{-8.23s + 65.16}{s^2 + 38.07s + 66.84}$
Original CRO	0,06374	0,30694	0,14056	0,74851	24,10	$T_f(s) = \frac{-8.12s + 48.88}{s^2 + 26.57s + 50.73}$

Fonksiyon hesaplama sınırlandırma kriterinde, çalışmakta olan algoritma, verilen maksimum fonksiyon diğerine ulaştığında durdurulmaktadır. Bu durdurma kriteri maksimum jenerasyon durdurma kriterine oldukça benzemektedir. Yapılan çalışma boyunca veri setinin doğası gereği maksimum fonksiyon sınırlandırma değeri 4000 olarak belirlenmiştir. Bu şartlar altında 5 farklı algoritma için hesaplanan transfer fonksiyonları ve performans göstergeleri Tablo 3'te sunulmuştur. Diğer kısıtlamalar altında olduğu gibi algoritmalar farklı algoritma değerlerine sahip olmuştur.

Tablo 3- Fonksiyon hesaplama sınırlılıkları altında çözüm yeteneklerinin karşılaştırılması

Algoritmalar	MAPE	MAE	MSE	R2	Time	Transfer Fonksiyonları
Original AEO	0,02755	0,13246	0,02755	0,95071	55,59	$T_f(s) = \frac{-1.49s + 12.61}{s^2 + 6.12s + 12.91}$
Original ALO	0,06928	0,33421	0,16652	0,70206	67,29	$T_f(s) = \frac{-13.31s + 97.59}{s^2 + 60.55s + 100}$
Original ASO	0,1001	0,47568	0,33634	0,39823	71,14	$T_f(s) = \frac{-19.08s + 100}{s^2 + 100s + 100}$
Original BSO	0,06411	0,30883	0,14203	0,74588	56,55	$T_f(s) = \frac{-8.48s + 67.20}{s^2 + 39.33s + 68.92}$
Original CRO	0,06847	0,33061	0,16409	0,70641	57,08	$T_f(s) = \frac{-9.33s + 80.51}{s^2 + 46.26s + 82.73}$

Erken durdurma sınırlılığı temel olarak global en iyi çözümünün artık hiç azalmadığı 3. nesil için algoritmanın durdurulmasını sağlar. Erken durdurma kriter değeri 3 seçildiği zaman tüm AEO algoritmalarının performansı oldukça iyi olmakla birlikte çözüm süreleri oldukça artmaktadır. Tablo 4'te 5 farklı algoritma için hesaplanan transfer fonksiyonları MAPE, MAE, MSE, R^2 değerleri sunulmuştur.

Tablo 4-Erken durdurma sınırlılıkları altında çözüm yeteneklerinin karşılaştırılması

Algoritmalar	MAPE	MAE	MSE	R^2	Time	Transfer Fonksiyonları
Original AEO	0,02745	0,13203	0,02739	0,951	131,08	$T_f(s) = \frac{-1.49s + 12.60}{s^2 + 6.12s + 12.90}$
Original ALO	0,06397	0,30631	0,14181	0,74627	14,76	$T_f(s) = \frac{-4.72s + 51.72}{s^2 + 26.47s + 52.84}$
Original ASO	0,08961	0,42803	0,28265	0,49428	7,07	$T_f(s) = \frac{-17.04s + 42.09}{s^2 + 29.95s + 42.79}$
Original BSO	0,05357	0,25709	0,10153	0,81835	7,00	$T_f(s) = \frac{-8.52s + 88.20}{s^2 + 47.01s + 90.66}$
Original CRO	0,07147	0,34896	0,1819	0,67456	6.72	$T_f(s) = \frac{-7.14s + 50.75}{s^2 + 37.18s + 54.08}$

IV. TARTIŞMA VE SONUÇLAR

Bu çalışmada, beş farklı optimizasyon algoritması sistem tanımlama problemlerine uygulanmıştır. Orta düzeyde bir bilgisayar kullanılmış meta heuristik algoritmalarının sistem tanımlama problemlerinde transfer fonksiyonu bulmak için kolaylıkla kullanılabileceği gösterilmiştir. Böylece karmaşık ve elde etmesi zahmetli olan sistem denklemleri yerine günümüzde popülerleşen makine öğrenme algoritmaları kullanılmıştır. Kullanılan algoritmalar 100 defa çalıştırılarak performansları bir problemi rastgele çözüp çözemedikleri ele alınmıştır. Sunulan performans sonuçları meta heuristik algoritmalarının parametre tahmininde ne kadar güvenli olduğunu göstermek amacıyla yapılmıştır. Sonuç olarak, AEO algoritmasının 100 iterasyon sonunda yüksek bir performans değerine ulaştığı

tespit edilmiştir. Makalede; zaman, maksimum jenerasyon, fonksiyon hesaplama ve erken durdurma sınırlılıkları gibi faktörler dikkate alınarak algoritmalar çalıştırılmıştır.

V. KAYNAKLAR

[1] Zhao, W., Wang, L., & Zhang, Z. (2020). Artificial ecosystem-based optimization: a novel nature-inspired meta-heuristic algorithm. *Neural Computing and Applications*, *32*, 9383-9425.

[2] Izci, D., Hekimoğlu, B., & Ekinci, S. (2021). A new artificial ecosystem-based optimization integrated with Nelder-Mead method for PID controller design of buck converter. *Alexandria Engineering Journal*, 61(3), 2030-2044.

[3] Üstüner, M., & Taşkın, S. (2015). Çok Girişli Çok Çıkışlı Sistemlerde Etkileşimin Yok Edilmesi: Proses Kontrol Sistemi Uygulaması. *CBÜ Fen Bil. Dergi*, *11*(2).

[4] Fissore, D. (2021). 2.3 System Identification: Linear methods

[5] El-Dabah, M. A., El-Sehiemy, R. A., Becherif, M., & Ebrahim, M. A. (2021). Parameter estimation of triple diode photovoltaic model using an artificial ecosystem-based optimizer. *International Transactions on Electrical Energy Systems*, *31*(11), e13043.

[6] Omotoso, H. O., Al-Shaalan, A. M., Farh, H. M., & Al-Shamma'a, A. A. (2022). Techno-economic evaluation of hybrid energy systems using artificial ecosystem-based optimization with demand side management. *Electronics*, *11*(2), 204.

[7] Abdel-Basset, M., & Shawky, L. A. (2019). Flower pollination algorithm: a comprehensive review. *Artificial Intelligence Review*, *52*, 2533-2557.

[8] Al-Betar, M. A., Awadallah, M. A., Abu Doush, I., Hammouri, A. I., Mafarja, M., & Alyasseri, Z. A. A. (2019). Island flower pollination algorithm for global optimization. *The Journal of Supercomputing*, *75*, 5280-5323.

[9] Kopciewicz, P., & Łukasik, S. (2020). Exploiting flower constancy in flower pollination algorithm: improved biotic flower pollination algorithm and its experimental evaluation. *Neural Computing and Applications*, *32*(16), 11999-12010.

[10] Liu, Y., Qin, W., Zhang, J., Li, M., Zheng, Q., & Wang, J. (2021). Multi-Objective Ant Lion Optimizer Based on Time Weight. *IEICE TRANSACTIONS on Information and Systems*, *104*(6), 901-904.

[11] Guo, M. W., Wang, J. S., Zhu, L. F., Guo, S. S., & Xie, W. (2020). Improved ant lion optimizer based on spiral complex path searching patterns. *IEEE Access*, *8*, 22094-22126.

[12] Abualigah, L., Shehab, M., Alshinwan, M., Mirjalili, S., & Elaziz, M. A. (2021). Ant lion optimizer: a comprehensive survey of its variants and applications. *Archives of Computational Methods in Engineering*, *28*, 1397-1416.

[13] Jiang, L., Hao, K., Tang, X. S., Wang, T., & Liu, X. (2021, October). An improved moth-flame optimization algorithm based on fusion mechanism. In *IECON 2021–47th Annual Conference of the IEEE Industrial Electronics Society* (pp. 1-6). IEEE.

[14] Kaur, K., Singh, U., & Salgotra, R. (2020). An enhanced moth flame optimization. *Neural Computing and Applications*, *32*, 2315-2349.

[15] Worch, E., Samiappan, S., Zhou, M., & Ball, J. E. (2020, September). Hyperspectral band selection using moth-flame metaheuristic optimization. In *IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium* (pp. 1271-1274). IEEE.

[16] Ghelichi, M., Goltabar, A. M., Tavakoli, H. R., & Karamodin, A. (2021). Neuro-fuzzy active control optimized by Tug of war optimization method for seismically excited benchmark highway bridge. Numerical Algebra, Control and Optimization, 11(3), 333-351.

[17] Kaveh, A., & Zolghadr, A. (2016). A novel meta-heuristic algorithm: tug of war optimization.

[18] YÜZGEÇ, U., & KILIÇ, H. Kutulama Problemi için Geliştirilmiş Karınca Aslanı Optimizasyonu Algoritması. *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi*, *11*(2), 13-19.

[19] Ozkaya, U., & Seyfi, L. (2017). Optimal Rectangular Microstrip Antenna with and without Air Gaps Design by Means of Particle Swarm Optimization and Vortex Search Algorithm. International Journal of Computer and Communication Engineering, 6(1), 75.

[20] Kaur, K., Singh, U., & Salgotra, R. (2020). An enhanced moth flame optimization. *Neural Computing and Applications*, *32*, 2315-2349.

[21] Kaveh, A., & Zolghadr, A. (2016). A novel meta-heuristic algorithm: tug of war optimization.

[22] Ravber, M., Liu, S. H., Mernik, M., & Črepinšek, M. (2022). Maximum number of generations as a stopping criterion considered harmful. *Applied Soft Computing*, 128, 109478.