



Kahramanmaraş Sütçü İmam University

Journal of Engineering Sciences



Geliş Tarihi :24.04.2022
Kabul Tarihi :23.07.2022

Received Date :24.04.2022
Accepted Date :23.07.2022

LİNEER OLMAYAN SİSTEMLERİN KİMLİKLENDİRİLMESİ İÇİN KELEBEK OPTİMİZASYON ALGORİTMASI KULLANILARAK İLERİ BESLEMELİ YAPAY SİNİR AĞININ EĞİTİMİ

TRAINING OF FEED FORWARD NEURAL NETWORK BY USING BUTTERFLY OPTIMIZATION ALGORITHM FOR IDENTIFICATION OF NONLINEAR SYSTEMS

Ceren BAŞTEMUR KAYA¹ (ORCID: 0000-0002-0091-3606)

¹Nevşehir Hacı Bektaş Veli Üniversitesi, Bilgisayar Teknolojileri Bölümü, Nevşehir, Türkiye

*Sorumlu Yazar / Corresponding Author: Ceren BAŞTEMUR KAYA , ceren@nevsehir.edu.tr

ÖZET

Gerçek dünyada karşılaştığımız problemlerin pek çoğu lineer olmayan davranış sergilemektedir. Bu yüzden lineer olmayan problemlerin çözümü önemlidir. Bu problemlerin karakteristik yapısının lineer olmaması, onları karmaşık hale getirmektedir. Özellikle bu problemlerin kimliklendirilmesi için etkili yapay zekâ teknikleri gerekmektedir. Gerçek dünya düşünüldüğünde, var olan verilere göre bir model oluşturmak durumunda kalınabilir. Sahip olduğu güçlü yapısından dolayı, yapay sinir ağları (YSA) literatürde kullanılan önemli yapay zekâ tekniklerinden biri olarak karşımıza çıkmaktadır. YSA ile etkili sonuçlar elde edebilmek için başarılı bir eğitim süreci şarttır. Bu çalışmada, lineer olmayan sistemlerin kimliklendirilmesi için ileri beslemeli yapay sinir ağının (İB-YSA) eğitiminde kelebek optimizasyon algoritmasının (KOA) performansı değerlendirilmiştir. Bu kapsamda, YSA'nın ağırlıkları KOA ile belirlenmiştir. Bununla birlikte, İB-YSA'nın eğitiminde popülasyon büyüklüğü ve ağ yapısının etkisi detaylıca incelenmiştir. KOA'nın çözüm kalitesi ve yakınsama hızı açısından performansı değerlendirilmiştir. Uygulamalarda lineer olmayan dört sistem kullanılmıştır. Performans metriği olarak ortalama karesel hata tercih edilmiştir. Tüm sistemler için elde edilen sonuçlar değerlendirildiğinde, lineer olmayan sistemlerin kimliklendirilmesinde KOA tabanlı İB-YSA eğitim sürecinin etkili olduğu gözlemlenmiştir.

Anahtar Kelimeler: Kelebek optimizasyon algoritması, yapay sinir ağı, sistem kimliklendirme.

ABSTRACT

Many of the problems we encounter in the real world exhibit nonlinear behavior. Therefore, it is important to solve nonlinear problems. The nonlinearity of the characteristic nature of these problems complicates them. In particular, effective artificial intelligence techniques are required to identify these problems. Considering the real world, it may be necessary to create a model using known data. The artificial neural networks (ANNs) are one of the important artificial intelligence techniques used in the literature because of its strong structure. A successful training process is essential to obtain effective results with ANNs. In this study, the performance of butterfly optimization algorithm (BOA) was evaluated in training feed forward artificial neural network (FFNN) for the identification of nonlinear systems. In this context, the weights of the ANN were determined by BOA. At the same time, the effects of population size and network structure on the training of FFNN were studied in detail. The performance of the BOA in terms of solution quality and convergence speed was evaluated. In the applications, four nonlinear systems were used. The mean square error was preferred as the performance metric. When the results obtained for all systems were evaluated, it was observed that the FFNN training process based on BOA was effective in the identification of nonlinear systems.

Keywords: Butterfly optimization algorithm, artificial neural network, system identification.

ToCite: BAŞTEMUR KAYA, C., (2022). LİNEER OLMAYAN SİSTEMLERİN KİMLİKLENDİRİLMESİ İÇİN KELEBEK OPTİMİZASYON ALGORİTMASI KULLANILARAK İLERİ BESLEMELİ YAPAY SİNİR AĞININ EĞİTİMİ. *Kahramanmaraş Sütçü İmam Üniversitesi Mühendislik Bilimleri Dergisi*, 25(3), 273-284.

GİRİŞ

Gerçek dünya problemlerinin çözümünde ve karmaşık mühendislik tasarımlarında pek çok parametrenin optimum değerinin bulunması gerekmektedir. Bunun için türev tabanlı optimizasyon yaklaşımları uzun süredir kullanılmasına rağmen, sahip oldukları dezavantajlardan dolayı meta-sezgisel algoritmalar alternatif olarak geliştirilmiştir. Her meta-sezgisel algoritma doğadaki bir zeki davranışı modeller ve ilgili problemleri çözmeye çalışır. Meta-sezgisel yaklaşımlar; geliştirilmesinde kullanılan doğal fenomenler dikkate alınarak evrimsel algoritmalar, sürü zekâsı tabanlı algoritmalar, fizik tabanlı algoritmalar ve insan tabanlı algoritmalar olarak dört gruba ayrılabilir. Meta-sezgisel algoritmalar pek çok gerçek dünya probleminin çözümünde uzun süredir kullanılmaktadır (Aslan, 2022; Aslan & Demirci, 2020).

YSA, sınıflandırma, kümeleme, tahmin, örüntü tanıma ve kimliklendirme problemlerinin çözümünde yoğun bir şekilde kullanılmaktadır. Tarım, finans, fen bilimleri, mühendislik, eğitim, enerji, sosyal bilimler, pazarlama ve yönetim YSA'nın kullanıldığı alanlardan bazılarıdır (Abiodun vd., 2018). YSA ile etkili bir modelleme gerçekleştirmek için etkili bir eğitim süreci gerekmektedir. Sahip olduğu avantajlar nedeniyle, meta-sezgisel optimizasyon algoritmaları YSA eğitiminde başarılı bir şekilde kullanılmaktadır. Yapay arı koloni algoritması (YAK), parçacık sürü optimizasyon algoritması (PSO), genetik algoritma (GA), armoni arama algoritması (AA), diferansiyel gelişim algoritması (DG) ve ateş böceği algoritması (AB) YSA eğitiminde yoğun kullanılan meta-sezgisel algoritmalarından bazılarıdır (Abd Elaziz vd., 2021; Akay vd., 2021; Hemeida vd., 2020).

KOA; YAK, PSO, GA, AA, DG ve AB algoritmaları ile kıyaslandığında daha güncel bir algoritmadır. YSA tabanlı çalışma sayısı diğer algoritmalara göre kısıtlıdır. Literatür genel olarak değerlendirildiğinde; KOA'nın pek çok gerçek dünya probleminin çözümünde kullanıldığı görülmektedir (Alawode vd., 2021; Arora & Singh, 2017; Assiri, 2021; Maheshwari vd., 2021; Zhi vd., 2020). Buna ek olarak, kısıtlı olmakla birlikte YSA ve KOA tabanlı çalışmalarda bulunmaktadır. Irmak ve Gülcü (Irmak & Gülcü), sınıflandırma problemlerinin çözümü için KOA kullanarak İB-YSA'nın eğitimini gerçekleştirmiştir. Önerdikleri yaklaşımın performansını BAT-MLP, SMS-MLP ve BP olarak adlandırdıkları yaklaşımlar ile karşılaştırmışlardır. Önerdikleri yaklaşımın performansının, diğer yaklaşımlardan daha iyi olduğunu rapor etmişlerdir. Li ve ark. (Li vd., 2021) ağ trafiği verilerinin ayırt edici özelliklerini belirlemek için KOA ve YSA tabanlı bir yaklaşım kullanmıştır. Mahboob ve Moghaddam (Mahboob & Moghaddam, 2020) KOA ve İB-YSA tabanlı bir saldırı tespit sistemi önermiştir. Yogananda ve ark. (Yogananda vd., 2021), medikal verilerinin sınıflandırılması için KOA ve YSA tabanlı bir yaklaşım tanıtmıştır.

Gerçek dünyada pek çok problem lineer olmayan davranış sergilemektedir. Bu yüzden lineer olmayan sistemlerin kimliklendirilmesi önemlidir. Lineer olmayan sistemler statik veya dinamik davranışlar sergileyebilirler. Statik sistemler bir fonksiyona göre şekillenirken, dinamik sistemler ise geçmiş eylemlerden etkilenmektedir (Kaya, 2022). Lineer olmayan sistemlerin kimliklendirilmesi, yapısı gereği zor problemlerden biridir. Özellikle bir eğitim algoritmasının performansının değerlendirilmesi için lineer olmayan test sistemleri yoğun olarak kullanılmaktadır. YSA ve ANFIS eğitiminde de lineer olmayan sistemler belirtilen sebeplerden dolayı tercih edilmektedir (Karaboğa & Kaya, 2017; Kaya & Baştemur Kaya, 2021; Kaya, 2022).

YSA ve uyarlamalı ağ tabanlı bulanık çıkarım sistemleri gibi eğitim tabanlı modellerin performansının analiz edilmesinde lineer olmayan sistemler yoğun bir şekilde kullanılmaktadır (Karaboga & Kaya, 2019; Kaya & Baştemur Kaya, 2021). Özellikle lineer olmayan sistemlerin zor ve karmaşık yapıya sahip olması eğitim algoritmalarının başarısının değerlendirilmesi açısından önemlidir. Bununla birlikte, gerçek dünyadaki pek çok problem lineer olmayan davranış sergilemektedir. Algoritmaların sistem kimliklendirme çalışmalarında başarılı olması, farklı gerçek dünya problemlerinin çözümünde kullanılması için referans olabilir. Bu yüzden çalışma kapsamında problem grubu olarak lineer olmayan sistemlerin kimliklendirilmesi seçilmiştir. Bu çalışmada, lineer olmayan sistemlerin kimliklendirilmesi amacıyla KOA kullanılarak İB-YSA'nın eğitimi gerçekleştirilmektedir. Eğitim süreci ve kimliklendirmenin başarılı olması için farklı popülasyon büyüklükleri ve ağ yapılarının çözüme etkisi detaylıca incelenmiştir. Bu çalışma literatüre önemli katkılar sağlayacak olup, yenilik içermektedir.

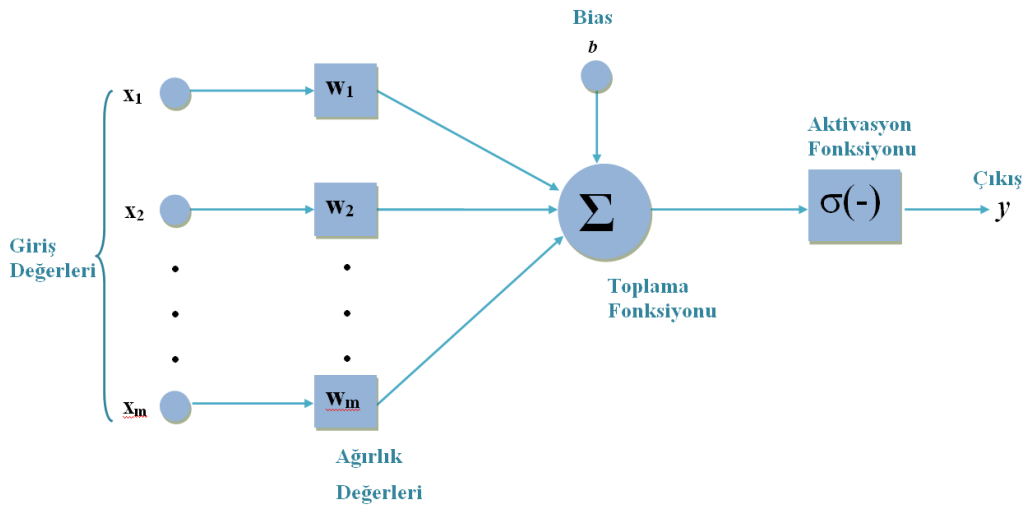
YAPAY SİNİR AĞI

Yapay zekâ tekniklerinden biri olan YSA, yapay nöronların birbirine bağlanmasından oluşur. Yapay bir nöronun genel yapısı Şekil 1'de görülmektedir. Yapay bir nöron; girdiler, bias değeri, ağırlıklar, aktivasyon ve transfer fonksiyonundan oluşmaktadır. Nörondaki girdilerden bir çıktı elde edilir. Çıktı hesaplaması için Denklem (1)

kullanılmaktadır. Denklemdaki x giriş değeri, w girişe karşılık gelen ağırlık değerleri, b bias değeri, f aktivasyon fonksiyonu, y yapay nöronun çıktısına karşılık gelmektedir (Kaya & Baştemur Kaya, 2021; Kaya, 2022).

$$y = f\left(\sum_{i=1}^m w_i x_i + b\right) \quad (1)$$

İB-YSA girdi, gizli ve çıktı olmak üzere üç katmandan oluşmaktadır. İB-YSA'da her nöronda Denklem (1) 'de belirtilen hesaplamalar gerçekleştirilmektedir. Böylelikle her nöron bir sonraki katmandaki nöronları etkiler. Aynı katmanda etkileşim bulunmamaktadır. İB-YSA'da da giriş değerlerine karşılık, çıkış değeri/değerleri elde edilmektedir. Sistem kimliklendirme problemlerinin çözümü için bir modelin oluşturulması gerekmektedir. Bu model ise eğitim süreci sonunda elde edilmektedir. Model oluşturulurken eğitim algoritması kullanılarak ağırlık ve bias değerleri sürekli güncellenerek ağ eğitilir. Eğitim süreci için eğitim veri seti kullanılmaktadır. Ağın öğrenme düzeyi, gerçek çıktı ile tahmin edilen çıktı arasındaki ilişkiyi ifade eden hata değeri ile ilgilidir. Başarılı bir eğitim süreci için hata değerinin düşük olması beklenen bir durumdur. Bunun için etkili bir eğitim algoritmasının kullanılması ve etkili bir eğitim sürecinin gerçekleşmesi gerekmektedir (Kaya & Baştemur Kaya, 2021; Kaya, 2022).



Şekil 1. Yapay Bir Nöronun Genel Yapısı

KELEBEK OPTİMİZASYON ALGORİTMASI

Kelebekler yaşamlarını sürdürebilmek için duyu organlarından yararlanırlar. Duyular arasında en etkin olan ise koku almadır (Blair & Launer, 1997). Kelebeklerin zeki davranışlarının modellenmesiyle KOA geliştirilmiştir (Arora & Singh, 2019). KOA'da arama aracı olarak kelebekler kullanılır. Bir kelebeğin farklı durumlarda yaydığı kokuyu diğer kelebekler hissedebilir. Kelebeğin oluşturduğu bu sosyal ağ ile arama yapılır. KOA'da kelebeğin başka bir kelebek kokusunu alıp o yönde hareket etmesi küresel arama, herhangi bir koku almayı rastgele ilerlemesi ise yerel arama olarak isimlendirilir. KOA'da duysal yöntem (c), güç katsayısı (a) ve uyaran yoğunluğu (I) parametreleri ile algılama yöntemi belirlenir. Yöntem koku, ses, sıcaklık, ışık olabilir ancak KOA'da yöntem koku olarak alınır (Arora & Singh, 2019). Uyaran yoğunluğu uyarının büyüklüğünü simgeler. Güç katsayısı ise uyaran yoğunluğunun yükseltildiği katsayıdır. KOA'da koku (f) formülü Denklem (2)'de verilmiştir.

$$f = cI^a \quad (2)$$

f kokunun diğer kelebek tarafından ne kadar güçlü algılandığını ifade eder. a ve c $[0,1]$ aralığında alınabilir. KOA'da 1:Başlama, 2: Tekrarlama, 3: Son olmak üzere 3 aşama bulunmaktadır. Başlatma aşamasında tarafsız fonksiyon ve fonksiyonun çözüm aralığı, parametre değerleri tanımlanır. Optimizasyon için ilk kelebek popülasyonu oluşturulur. Kelebeklerin konumları, hesaplanan koku ve uygunluk değerleri ile arama aralığında rastgele oluşturulur. Oluşturulan yapay kelebeklerle tekrarlama aşamasına geçilir. Bu aşamada her tekrarlamada tüm kelebekler yeni konumlara hareket eder, uygunluk değerleri hesaplanır. Denklem (2) kullanılarak koku üretilir. Küresel arama aşamasında Denklem (3) kullanılarak kelebek en uygun çözüm g^* 'ye ulaşmaya çalışır.

$$x_i^{t+1} = x_i^t + (r^2 * g^* - x_i^t) * fi \quad (3)$$

t yineleme sayısındaki i 'inci kelebek için çözüm vektörü x_i olarak alınarak denklemdeki x_i^t elde edilir. $r[0,1]$ aralığında rastgele bir sayıdır. Yerel arama aşamasında Denklem (4) kullanılır.

$$x_i^{t+1} = x_i^t + (r^2 * x_j^t - x_k^t) * fi \quad (4)$$

Arama uzayındaki j 'inci kelebek x_j^t , k 'inci kelebek x_k^t 'yi temsil eder. KOA'da küresel arama ile yerel arama arasında geçiş yapmak için p anahtar (değişim) olasılığı kullanılır. Durdurma kriterine ulaşıncaya kadar tekrarlar biter. En iyi uygunluk ile bulunan en iyi çözüm son aşamada sunulur. Algoritmanın yapısı Algoritma 1'de verilmiştir.

Algoritma 1. Kelebek Optimizasyon Algoritması (Arora & Singh, 2019)

1. Uygunluk fonksiyonu $f(\mathbf{x})$, $\mathbf{x}=(x_1, x_2, \dots, x_{\dim})$, \dim =boyut parametrelerini belirle
2. n kelebeğin başlangıç popülasyonu oluştur $\mathbf{x}_i=(i=1, 2, \dots, n)$
3. x_i 'deki uyaran yoğunluğu I_i , $f(\mathbf{x}_i)$ kullanarak belirle
4. Sensör modalitesi c 'yi, güç katsayısı a 'yı ve değiştirme (anahtarlama) olasılığı p 'yi tanımla
5. **while** durdurma kriteri karşılanmadı **do**
6. **foreach** popülasyondaki her bf kelebeği için **do**
7. Denklem (2) kullanarak bf için koku hesapla
8. **endfor**
9. En iyi bf 'yi bul
10. **for** popülasyondaki her bf kelebeği için **do**
11. $[0,1]$ arasında rastgele bir r sayısı üret
12. **if** $r < p$ **then**
13. Denklem (3)'ü kullanarak en iyi kelebeğe/çözüme doğru ilerle
14. **else**
15. Denklem (4)'ü kullanarak rastgele hareket et
16. **endif**
17. **endfor**
18. a 'nın değerini güncelle
19. **endwhile**

SİMÜLASYON SONUÇLARI

Bu çalışmada lineer olmayan sistemlerin kimliklendirilmesi için KOA kullanılarak İB-YSA'nın eğitimi gerçekleştirilmektedir. Uygulama kapsamında dört adet lineer olmayan statik sistem (S1,S2,S3,S4) kullanılmıştır. Her uygulama için pek çok ağ yapısı denenerek etkili sonuçlara ulaşılmaya çalışılmıştır. Buna ek olarak her uygulama en az 30 kere çalıştırılmıştır. Çözüm kalitesinin belirlenmesi için hata metriği olarak ortalama karesel hata (MSE) seçilmiştir. Popülasyon büyüklüğünün (n) 10, 20 ve 50 değerleri için sonuçlar elde edilmiştir. Eğitim sürecine ait durdurma kriteri olarak değerlendirme sayısı kullanılmıştır. Tüm sistemler için değerlendirme sayısı 20000 alınmıştır. Adil bir değerlendirme için, değerlendirme sayısı sabit kalmakla birlikte popülasyon büyüklüğü ve jenerasyon sayısı değişmiştir. $n=10$ olması durumunda, jenerasyon sayısı 2000 olarak alınmıştır. $n=20$ 'de jenerasyon sayısı 1000 olurken, $n=50$ olması durumunda jenerasyon sayısı 400 olmaktadır. YSA tabanlı kimliklendirme çalışmalarında eğitim süreçleri çoğunlukla uzun süreler almaktadır. Ayrıca, sonuçların anlamlılığının analizi için en az 30 kere çalıştırılmaktadır. Bu durum da işlem süresi daha fazla uzatmaktadır. Buna ek olarak, ilk 20000 değerlendirme de yakınsamanın genel olarak daha hızlı olduğu gözlemlenmiştir. Değerlendirme sayısının belirlenmesinde, bu hususlar dikkate alınarak 20000 değeri belirlenmiştir. Anahtar (değişim) olasılığı (p) ise 0.8 alınmıştır. Güç katsayısı (a) ve duyuşal yöntem (c)'in değerleri ise sırasıyla 0.1 ve 0.01'dir.

Tablo 1. Uygulamalarda Kullanılan Statik Sistemler

Sistem	Denklem	Girişler	Çıkış	Veri Seti İçeriği (Eğitim /Test)	Aralık
--------	---------	----------	-------	----------------------------------	--------

S1	$y = 2 \sin(\pi x_1)$	x_1	80/20	[0,1]
S2	$y = \sin(x_1 x_2)$	x_1, x_2	80/20	[-2,2]
S3	$y = 1 + 0.3/(0.9 + x_1) + 0.5 \ln(1 + x_2) + 0.7 \cos(0.4 x_3)$	x_1, x_2, x_3	100/25	[0,1]
S4	$y = e^{2x_1 \sin(\pi x_4)} + \sin(x_2 x_3)$	x_1, x_2, x_3, x_4	100/25	[-0.25, 0.25]

Uygulamalarda kullanılan lineer olmayan statik sistemlere ait bilgiler Tablo 1’de sunulmuştur. S1, S2, S3 ve S4 sisteminin giriş sayıları sırasıyla 1, 2, 3 ve 4 şeklindedir. Tüm sistemler bir çıkışa sahiptir. Tüm sistemlerin denklemleri Tablo 1’de görülmektedir. İlgili sistemlerin kimliklendirilmesinin sağlanabilmesi için belirtilen aralıkta giriş seti oluşturulmuş ve girişlere karşılık bir çıkış seti elde edilmiştir. İlk iki sistem için 100 giriş-çıkış çiftinden oluşan bir veri seti oluşturulurken, son iki sistem için ise 125 giriş-çıkış çiftinden oluşan bir veri seti kullanılmıştır. Sistemlere ait veri setlerindeki veri sayılarının değişkenlik göstermesinde giriş sayıları ve aralık değerleri etkili olmuştur. Lineer olmayan statik sistemlerin YSA tabanlı kimliklendirilmesine yönelik önemli çalışmalardan biri Kaya ve Baştemur Kaya (2021) tarafından yapılmıştır. Bahsi geçen çalışmada çoğunlukla, veri setinin %80’i eğitim süreci için, kalan %20’lik veri ise test sürecinde kullanılmıştır. Bu çalışma dikkate alınarak, veri setinin %80’i eğitim sürecinde, %20’si ise test sürecinde kullanılmıştır. Test verileri belirlenirken, tüm veri setini kapsamaya dikkat edilmiştir. Bu sebeple test veri setinin belirlenmesinde, mod(i, 5) yaklaşımı uygulanmıştır. Burada i, veri setinin indisine karşılık gelmektedir. Yani veri seti baştan sona kadar taranmıştır. Her beş veriden ilk dört tanesi eğitim verisi, son bir tanesi ise test verisi olarak alınmıştır. Yani, eğer mod(i, 5)=0 ise, ilgili veri test veri setine dahil edilmiştir. Oluşturulan YSA modelinde her nöron için Sigmoid fonksiyon kullanılmıştır. Transfer fonksiyonu olarak ise Toplam seçilmiştir. Her nöron için bias değeri kullanılmıştır. Eğitim ve test sürecinde kullanılacak veri seti [0,1] aralığında ölçeklenmiştir.

S1 için gizli katmanda 3, 6, 9, 12 ve 15 nörondan oluşan ağ modellerinin performansları analiz edilmiştir. S1 için elde edilen sonuçlar Tablo 2’de sunulmaktadır. n=10 için en iyi eğitim hata değeri 1-9-1 ağ yapısı ile 0.02340 olarak bulunmuştur. Gizli katmandaki nöron sayısı 9 olana kadar performansın arttığı gözlemlenirken daha sonraki nöron sayısı artışlarında çözüm kalitesi azalmıştır. Eğitim sürecine ait standart sapma değeri de hata değerinde olduğu gibi 1-9-1 de daha etkilidir. n=10 için test hata değerinde en iyi sonuç 1-12-1 ile elde edilmiştir. 1-15-1 ağ yapısı dışında nöron sayısındaki artış çözüm kalitesini iyileştirmiştir. n=20’de eğitim ve test değerleri n=10’dan daha iyi olduğu görülmektedir. En iyi sonuçlar n=10’da olduğu gibi eğitim ve test süreci için sırasıyla 1-9-1 ve 1-12-1 ağ yapılarında elde edilmiştir. En iyi eğitim hata değeri 0.01316 iken, en iyi test hata değeri 0.01522 olmuştur. Genel olarak n=20’de etkili standart sapma değerlerine de ulaşılmıştır. n=50 için en iyi eğitim hata değeri 1-15-1 ile 0.01334 olarak bulunmuştur. Test için ise en etkili ağ yapısı 1-12-1 ile 0.01603 olarak elde edilmiştir. Tüm popülasyon büyüklükleri beraber değerlendirildiğinde en iyi çözüme n=20 ile ulaşıldığı görülmektedir.

Tablo 2. S1 İçin Elde Edilen Sonuçlar

Popülasyon Büyüküğü	Ağ Yapısı	Eğitim		Test	
		Hata	SS	Hata	SS
n=10	1-3-1	0.05351	0.02482	0.05869	0.02715
	1-6-1	0.04058	0.03240	0.04217	0.03406
	1-9-1	0.02340	0.02392	0.02677	0.02564
	1-12-1	0.02360	0.02799	0.02427	0.02793
	1-15-1	0.02764	0.03106	0.03068	0.03187
n=20	1-3-1	0.03608	0.02625	0.03613	0.02542
	1-6-1	0.02189	0.02183	0.02492	0.02330
	1-9-1	0.01316	0.01218	0.01584	0.01335
	1-12-1	0.01323	0.01118	0.01522	0.01249
	1-15-1	0.01887	0.02149	0.02121	0.02279
n=50	1-3-1	0.03205	0.02345	0.03585	0.02546
	1-6-1	0.02436	0.01736	0.02700	0.01720
	1-9-1	0.01881	0.01526	0.02197	0.01621
	1-12-1	0.01341	0.00991	0.01603	0.01033
	1-15-1	0.01334	0.01173	0.01621	0.01336

S2 için elde edilen sonuçlar Tablo 3’de sunulmuştur. En iyi ortalama eğitim ve test hata değerleri n=50 ile bulunmuştur. Elde edilen en iyi eğitim ve test hata değerleri sırasıyla 0.04974 ve 0.06319’dır. n=10 için en başarısız eğitim sonucu 2-3-1 ağ yapısı ile elde edilmiştir. Özellikle, düşük parametre sayısının modeli tanımlamada yeterli olmadığı ortaya çıkmıştır. n=10 için en iyi eğitim hata değeri 2-12-1 ile bulunurken, en iyi test hata değeri ise 2-6-1 ağ yapısına aittir. n=10’da gizli katmandaki nöron sayısının artmasının performansa etkisi değişkenlik göstermiştir. Yani, 2-6-1 ve 2-12-1 ile bulunan eğitim sonuçları 2-9-1’den daha iyidir. Bu yüzden kesin çıkarım yapılamamaktadır. n=20’de nöron sayısının artması eğitim ve test hatalarını zıt yönlü etkilemiştir. Özellikle 2-9-1 ağ yapısına kadar, nöron sayısının artışı eğitim hata değerini iyileştirirken, test hata değerini kötüleştirmiştir. n=20 için en iyi eğitim ve test hata değerleri sırasıyla 2-9-1 ve 2-15-1 ağ yapılarında elde edilmiştir. n=50’de genel olarak nöron sayısının artışı eğitim hata değerini iyileştirirken, test hata değerini kötüleştirmiştir.

Tablo 3. S2 İçin Elde Edilen Sonuçlar

Popülasyon Büyüküğü	Ağ Yapısı	Eğitim		Test	
		Hata	SS	Hata	SS
n=10	2-3-1	0.09039	0.02365	0.06898	0.01917
	2-6-1	0.07494	0.02870	0.06477	0.01214
	2-9-1	0.07826	0.02869	0.07146	0.01924
	2-12-1	0.07361	0.02865	0.07666	0.02987
	2-15-1	0.07677	0.02964	0.07662	0.03143
n=20	2-3-1	0.08498	0.01470	0.06832	0.01781
	2-6-1	0.05949	0.02070	0.07212	0.01683
	2-9-1	0.05553	0.02708	0.07439	0.01671
	2-12-1	0.05808	0.02755	0.07811	0.02168
	2-15-1	0.06054	0.02772	0.06701	0.01312
n=50	2-3-1	0.07979	0.01409	0.06319	0.01191
	2-6-1	0.06117	0.01879	0.06494	0.01600
	2-9-1	0.05698	0.01843	0.06737	0.02407
	2-12-1	0.04974	0.02154	0.07117	0.01580
	2-15-1	0.05395	0.01852	0.06379	0.01831

Tablo 4. S3 İçin Elde Edilen Sonuçlar

Popülasyon Büyüküğü	Ağ Yapısı	Eğitim		Test	
		Hata	SS	Hata	SS
n=10	3-3-1	0.02130	0.01161	0.03766	0.01960
	3-6-1	0.02409	0.01327	0.05013	0.03533
	3-9-1	0.02458	0.01460	0.04071	0.02647
	3-12-1	0.01803	0.01177	0.03578	0.02116
	3-15-1	0.01852	0.01056	0.03734	0.02131
n=20	3-3-1	0.01527	0.00429	0.03219	0.01690
	3-6-1	0.01301	0.00540	0.02753	0.01902
	3-9-1	0.01304	0.00795	0.02523	0.01367
	3-12-1	0.01331	0.00559	0.02454	0.01096
	3-15-1	0.01178	0.00366	0.02325	0.01046
n=50	3-3-1	0.01331	0.00377	0.02329	0.00729
	3-6-1	0.01114	0.00238	0.02268	0.00739
	3-9-1	0.01025	0.00221	0.02081	0.00865
	3-12-1	0.01049	0.00216	0.02125	0.00889
	3-15-1	0.01049	0.00158	0.02030	0.00671

S3 için bulunan sonuçlar Tablo 4’de verilmiştir. n=20 ve n=50 ile elde edilen sonuçlar çoğunlukla n=10’a göre daha başarılıdır. n=10’da nöron sayısındaki artış hem eğitim hem de test sürecindeki sonuçları iyileştirmiştir. Eğitim ve test için en iyi hatalar 3-12-1 ağ yapısı ile sırasıyla 0.01803 ve 0.03578 olarak bulunmuştur. n=20’de nöron

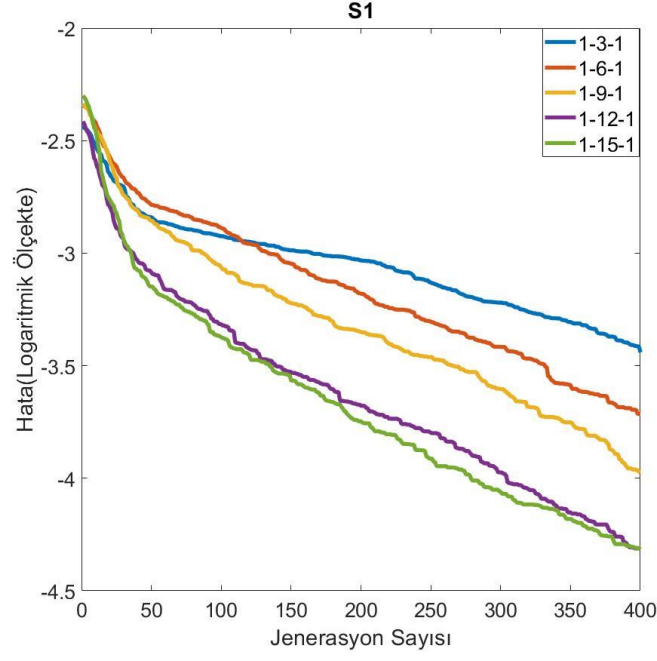
sayısındaki artışın çözüm kalitesini iyileştirdiği net bir şekilde görünmektedir. $n=20$ için en iyi hata değerleri 3-15-1 ağ yapısı ile bulunmuştur. Diğer popülasyon büyüklüklerinde olduğu gibi $n=50$ seçildiğinde, ağ yapısının büyümesi genellikle çözüm kalitesini iyileştirmektedir. $n=50$ 'nin eğitim sonuçlarına bakıldığında 3-9-1, 3-12-1 ve 3-15-1 ile birbirine yakın sonuçlar bulunmuştur. En iyi eğitim hata değeri ise 0.01025 olarak 3-9-1'e aittir. En iyi test hata değeri ise 3-15-1 ile 0.02030 olarak bulunmuştur.

S4 için elde edilen sonuçlar Tablo 5'de sunulmuştur. $n=10$ ve $n=20$ 'de nöron sayısındaki artış çoğunlukla, eğitim ve test sürecine ait çözüm kalitesini iyileştirmiştir. $n=50$ 'de ise değişken davranış sergilenmiştir. En etkili sonuçlar ise $n=20$ ve $n=50$ 'de elde edilmiştir. $n=10$ 'da en iyi eğitim ve test sonucu 4-12-1 ağ yapısı ile bulunmuştur. $n=20$ 'de bulunan en iyi eğitim ve test hata değerleri sırasıyla 0.00657 ve 0.00538'dir. $n=50$ için bulunan en iyi hata değerleri ise sırasıyla 0.00666 ve 0.00537'dir. $n=20$ ve $n=50$ 'nin en iyi sonuçlarının birbirine yakın olduğu görülmektedir.

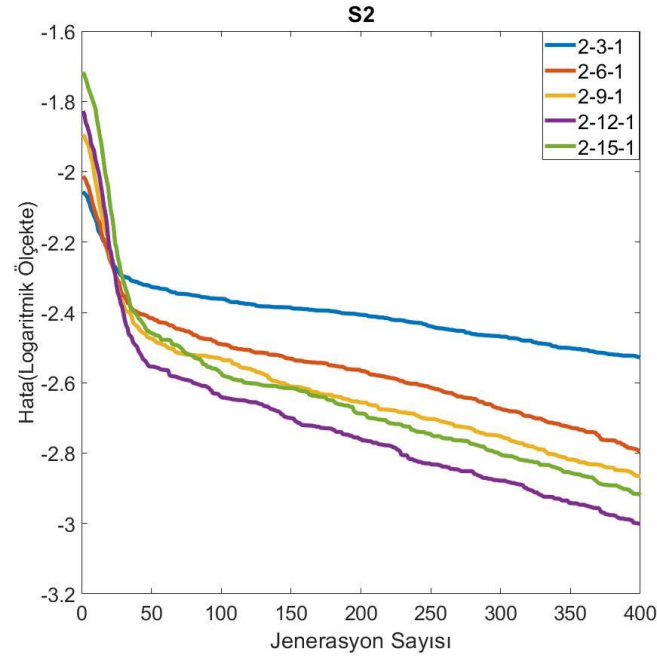
Tablo 5. S4 İçin Elde Edilen Sonuçlar

Popülasyon Büyüküğü	Ağ Yapısı	Eğitim		Test	
		Hata	SS	Hata	SS
n=10	4-3-1	0.00802	0.00138	0.00719	0.00223
	4-6-1	0.00793	0.00158	0.00703	0.00207
	4-9-1	0.00782	0.00125	0.00691	0.00154
	4-12-1	0.00703	0.00083	0.00580	0.00115
	4-15-1	0.00741	0.00090	0.00646	0.00137
n=20	4-3-1	0.00685	0.00118	0.00566	0.00147
	4-6-1	0.00658	0.00079	0.00552	0.00098
	4-9-1	0.00657	0.00081	0.00542	0.00102
	4-12-1	0.00662	0.00057	0.00541	0.00083
	4-15-1	0.00674	0.00065	0.00538	0.00072
n=50	4-3-1	0.00668	0.00060	0.00541	0.00076
	4-6-1	0.00666	0.00046	0.00564	0.00064
	4-9-1	0.00681	0.00072	0.00571	0.00117
	4-12-1	0.00669	0.00052	0.00552	0.00062
	4-15-1	0.00670	0.00038	0.00537	0.00070

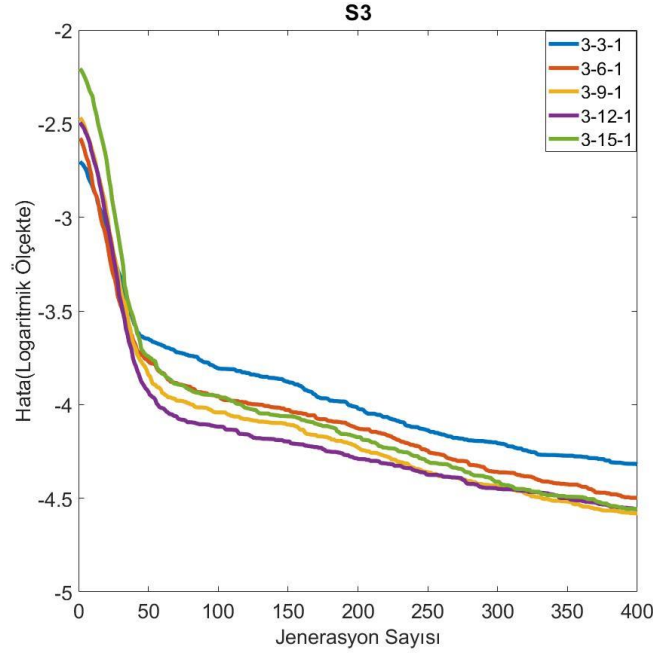
Algoritmanın performansını değerlendirmek için çözüm kalitesinin yanı sıra yakınsama hızı da önemli bir metriktir. Özellikle ağ yapıları performansı önemli ölçüde etkilemektedir. Bu yüzden tüm sistemler için ağ yapılarının yakınsama hızına etkisi incelenmiştir. $n=20$ ve $n=50$ için genel olarak daha etkili sonuçlar elde edilmiştir. Özellikle eğitim ve test hata değerleri birlikte değerlendirildiğinde $n=50$ 'nin daha başarılı olduğu gözlemlenmiştir. Bu nedenle $n=50$ 'ye göre grafikler oluşturulmuştur.



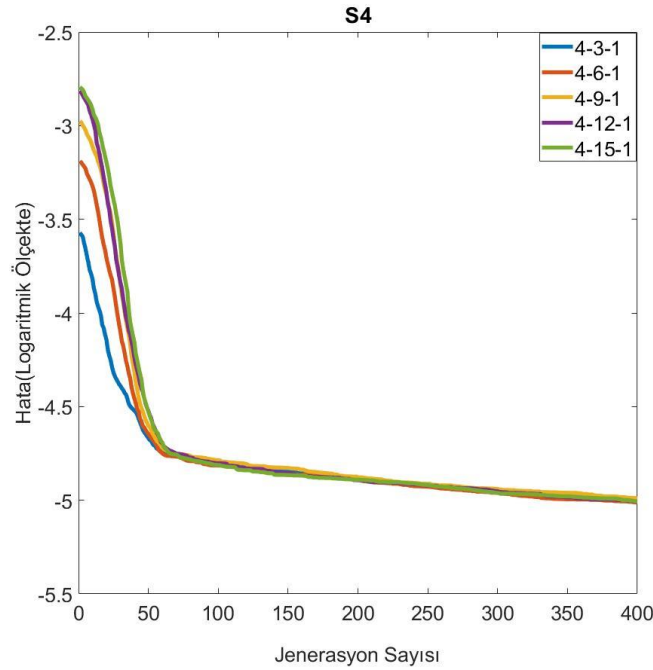
Şekil 2.S1 İçin Ağ Yapılarına Ait Yakınsama Grafiklerinin Karşılaştırması



Şekil 3.S2 İçin Ağ Yapılarına Ait Yakınsama Grafiklerinin Karşılaştırması



Şekil 4.S3 için Ağ Yapılarına Ait Yakınsama Grafiklerinin Karşılaştırması



Şekil 5.S4 için ağ yapılarına ait yakınsama grafiklerinin karşılaştırması.

S1 için ağ yapılarının yakınsamaya etkisi Şekil 2’de karşılaştırılmıştır. Özellikle 1-12-1 ve 1-15-1 ağ yapılarının yakınsama hızlarının birbirine yakın oldukları görülmektedir. Bununla birlikte nöron sayısındaki artış yakınsama hızını olumlu yönde etkilemiştir. Şekil 3’te S2’ye ait yakınsama grafiği sunulmuştur. En iyi yakınsama grafiğinin 2-12-1’e ait olduğu görülmektedir. Buna ek olarak 2-9-1 ve 2-15-1’in yakınsama hızları da birbirine yakındır. Şekil 4’te S3 sistemine ait yakınsama hızları verilmiştir. Jenerasyon sayısı yaklaşık olarak 200 değerine kadar 3-12-1’in yakınsama hızı diğerlerinden daha iyidir. Bu değerden sonra 3-12-1, 3-9-1 ve 3-15-1’in yakınsama hızları birbirine

yakın seyretmiştir. Şekil 5, S4'e ait yakınsama grafiğini vermektedir. Burada görüldüğü üzere tüm ağ yapılarının yakınsama hızı birbirine yakındır. Yani performansla ağ yapısının etkisi sınırlı düzeyde kalmıştır.

Tablo 6. KOA'nın Performansının Farklı Meta-Sezgisel Algoritmalar İle Karşılaştırılması

Sistem	HS		SCA		WOA		KOA (Önerilen)	
	Eğitim	Test	Eğitim	Test	Eğitim	Test	Eğitim	Test
S1	0.13050	0.12586	0.00589	0.00712	0.00806	0.01002	0.01341	0.01603
S2	0.21110	0.20232	0.03990	0.10223	0.04452	0.07880	0.04974	0.07117
S3	0.14054	0.17674	0.01131	0.02528	0.02622	0.05959	0.01049	0.02125
S4	0.07386	0.07413	0.00889	0.00803	0.00745	0.00760	0.00669	0.00552

Tablo 7. Algoritmaların Başarı Sıralamalarının Karşılaştırılması

Sistem	HS		SCA		WOA		KOA (Önerilen)	
	Eğitim	Test	Eğitim	Test	Eğitim	Test	Eğitim	Test
S1	4	4	1	1	2	2	3	3
S2	4	4	1	3	2	2	3	1
S3	4	4	2	2	3	3	1	1
S4	4	4	3	3	2	2	1	1
Toplam	16	16	7	9	9	9	8	6
		32		16		18		14

Tablo 6'da KOA'nın performansı meta-sezgisel algoritmalarla armoni arama (AA-HS), sinüs kosinüs algoritma (SKA-SCA) ve balina optimizasyon algoritması (BOA-WOA) ile karşılaştırılmıştır. Tüm algoritmalar için popülasyon büyüklüğü 50 ve jenerasyon sayısı 400 alınmıştır. Bununla birlikte, KOA'nın analiz çalışmalarında etkili olduğu tespit edilen ağ yapıları kullanılmıştır. Ağ yapılarının belirlenmesinde eğitim ve test süreçlerindeki performansları dikkate alınmıştır. Bu hususlar göz önünde bulundurularak gizli katmanda 12 nöron bulunan ağ yapıları kullanılmıştır. S1 ve S2 için en iyi eğitim hata değeri SCA ile bulunmuştur. S3 ve S4'de ise en iyi eğitim hata değeri KOA'ya aittir. Test sonuçlarında ise üstünlük KOA'ya aittir. S2, S3 ve S4'de en iyi test hata değerleri KOA ile elde edilirken, S1'de en iyi test sonucu SCA'ya aittir. Hem eğitim hem de testte en kötü sonuçlar HS ile bulunmuştur. WOA ise genel olarak 2. veya 3. sıralarda yer almıştır. Algoritmaların problemlere göre başarı sıralaması değişmektedir. En iyi algoritmanın hangisi olduğunu tespit etmek için Tablo 7'de algoritmaların başarı sıralaması karşılaştırılmıştır. Toplam skorlara göre en başarılı algoritma 14 puan ile KOA olmuştur. Sonrasında 16 puan ile SCA gelmektedir. SCA'yı 18 puan ile WOA takip etmektedir. En başarısız algoritma ise 32 puan ile HS olmuştur.

TARTIŞMA VE SONUÇ

Bu çalışmada, lineer olmayan sistemlerin kimliklendirilmesi için gerçekleştirilen IB-YSA'nın eğitiminde KOA'nın performansı değerlendirilmiştir. Farklı popülasyon büyüklükleri ve ağ yapıları için sonuçlar elde edilmiştir. Bu çalışmanın bazı kısıtlılıkları bulunmaktadır. Bunlar popülasyon büyüklüğü, ağ yapısı, değerlendirme sayısı ve kullanılan sistemler üzerinedir. Buradaki analizler sadece popülasyon büyüklükleri $n=\{10, 20, 50\}$ için geçerlidir. Ağ yapısının performansla etkisinin olduğu bir gerçektir. Tüm alternatif ağ yapılarının denemesi, anlamlılığı artırmakla birlikte, uygulamada maliyeti artıran bir durumdur. Bu yüzden ağ yapıları üzerine kısıtlama uygulanmış ve gizli katmandaki nöron sayısının 3, 6, 9, 12 ve 15 olma durumlarına göre analizler gerçekleştirilmiştir. Diğer kısıtlılıklardan biri ise değerlendirme sayısıdır. Kullanılan sistemlerin zorluğuna ve eğitim algoritmasına göre değişmekle birlikte, değerlendirme sayısının artması çözüm kalitesini artırabilir. Bazı durumlarda ise belirli bir değerlendirme sayısından sonra çözüm iyileştirilemeyebilir. Değerlendirme sayısının büyük olması maliyet açısından dezavantajlı bir durumdur. Bu yüzden değerlendirme sayısı çalışmanın kısıtlılıklarından biridir. Çalışmada sadece dört adet statik özellikli lineer olmayan sistem kullanılmıştır. Çalışmadaki analizler bu sistemler özelinde gerçekleştirilmektedir.

Meta-sezgisel algoritmaların karmaşıklığı, birbirlerine avantaj veya dezavantaj sağlayan önemli kriterlerden biridir. Buna karşın, meta-sezgisel algoritmalar ile YSA eğitimi, çok uzun süreler almaktadır. Özellikle lineer olmayan

sistemlerin kimliklendirilmesinde, problem yapısı ve veri yoğunluğu bu süreyi daha da artırabilmektedir. Ayrıca istatistiksel analizlerin yapılabilmesi için her uygulama en az 30 kere çalıştırılmıştır. Bu şekilde çalışma süresi daha da artmaktadır. İlgili problemin çözümü için YSA eğitim sürecinde, genel olarak bütün meta-sezgisel algoritmalarda benzer bir dezavantaj yaşanmaktadır. Çalışma süresinin uzun olması, ilgili problemler için algoritma karmaşıklığını önemsiz kılmaktadır. Bu yüzden ilgili problemin çözümü için literatür değerlendirildiğinde, çoğunlukla algoritma karmaşıklığının karşılaştırma kriterleri arasında yer almadığı görülmektedir.

Eğitim sonuçları dikkate alındığında, popülasyon büyüklüğün çözümü etkilediği gözlemlenmiştir n=20 ve n=50 için elde edilen hata değerlerinin çoğunlukla n=10'dan daha başarılı olduğu değerlendirilmiştir. Benzer durum test süreci içinde geçerlidir. Popülasyon büyüklüğü gibi ağ yapısı da çözüm kalitesini etkileyen hususlardan biridir. Özellikle gizli katmandaki nöron sayısının artması, eğitim sürecinde optimize edilecek parametre sayısını da artırmaktadır. Sonuçlar değerlendirildiğinde, genel olarak gizli katmandaki nöron sayısının 9-15 arasında olması çözüm kalitesini artırmaktadır. Bu durum yakınsama grafiklerinden de görülmektedir. Kullanılan sistem yapısı, elde edilen hata aralığını da doğrudan etkilemektedir. Zor problemlerin modellenmesi daha zor olmakla birlikte, hata değeri de daha yüksek olmaktadır. Daha iyi modelin oluşturulduğu sistemlerde ise hata oranı düşmektedir. Tüm sistemler değerlendirildiğinde e-02 ve e-03 seviyesinde hata değerleri elde edilmiştir. Değerlendirme sayısının artırılması, daha düşük hata değerlerinin elde edilmesini sağlayabilir. Mevcut değerlendirme sayısı dikkate alındığında, sistem kimliklendirme için gerçekleştirilen İB-YSA eğitiminde KOA'nın etkili olduğu görülmektedir.

Bu çalışma kapsamında, KOA'nın sistem kimliklendirme üzerine performansı değerlendirilmiştir. Bilindiği üzere, YSA pekçok alanda farklı problemlerin çözümünde yoğun bir şekilde kullanılmaktadır. Gelecek çalışmalarda ekonomi, tıp, biyomedikal, enerji, sosyal bilimler, eğitim ve mühendislik gibi alanlardaki gerçek dünya problemlerinin çözümü ile KOA tabanlı YSA eğitiminin gerçekleştirilmesi mümkündür.

KAYNAKLAR

Abd Elaziz, M., Dahou, A., Abualigah, L., Yu, L., Alshinwan, M., Khasawneh, A. M., & Lu, S. (2021). Advanced metaheuristic optimization techniques in applications of deep neural networks: a review. *Neural Computing and Applications*, 33(21), 14079-14099.

Abiodun, O. I., Jantan, A., Omolara, A. E., Dada, K. V., Mohamed, N. A., & Arshad, H. (2018). State-of-the-art in artificial neural network applications: A survey. *Heliyon*, 4(11), e00938.

Akay, B., Karaboga, D., & Akay, R. (2021). A comprehensive survey on optimizing deep learning models by metaheuristics. *Artificial Intelligence Review*, 1-66.

Alawode, B. O., Salman, U. T., & Khalid, M. (2021). A Flexible Operation and Sizing of Battery Energy Storage System Based on Butterfly Optimization Algorithm. *Electronics*, 11(1), 109.

Arora, S., & Singh, S. (2017). Node localization in wireless sensor networks using butterfly optimization algorithm. *Arabian Journal for Science and Engineering*, 42(8), 3325-3335.

Arora, S., & Singh, S. (2019). Butterfly optimization algorithm: a novel approach for global optimization. *Soft Computing*, 23(3), 715-734.

Aslan, S. (2022). An immune plasma algorithm with a modified treatment schema for UCAV path planning. *Engineering Applications of Artificial Intelligence*, 112, 104789.

Aslan, S., & Demirci, S. (2020). Immune plasma algorithm: a novel meta-heuristic for optimization problems. *IEEE Access*, 8, 220227-220245.

Assiri, A. S. (2021). On the performance improvement of Butterfly Optimization approaches for global optimization and Feature Selection. *PloS one*, 16(1), e0242612.

Blair, R. B., & Launer, A. E. (1997). Butterfly diversity and human land use: Species assemblages along an urban gradient. *Biological conservation*, 80(1), 113-125.

Hemeida, A. M., Hassan, S. A., Mohamed, A.-A. A., Alkhalaf, S., Mahmoud, M. M., Senjyu, T., & El-Din, A. B. (2020). Nature-inspired algorithms for feed-forward neural network classifiers: a survey of one decade of research. *Ain Shams Engineering Journal*, 11(3), 659-675.

- Irmak, B., & Gülcü, Ş. Training of the Feed-Forward Artificial Neural Networks using Butterfly Optimization Algorithm. *MANAS Journal of Engineering*, 9(2), 160-168.
- Karaboğa, D., & Kaya, E. (2017). Training ANFIS by using the artificial bee colony algorithm. *Turkish Journal of Electrical Engineering & Computer Sciences*, 25(3), 1669-1679.
- Karaboga, D., & Kaya, E. (2019). Training ANFIS by using an adaptive and hybrid artificial bee colony algorithm (aABC) for the identification of nonlinear static systems. *Arabian Journal for Science and Engineering*, 44(4), 3531-3547.
- Kaya, E., & Baştemur Kaya, C. (2021). A novel neural network training algorithm for the identification of nonlinear static systems: Artificial bee colony algorithm based on effective scout bee stage. *Symmetry*, 13(3), 419.
- Kaya, E. (2022). A comprehensive comparison of the performance of metaheuristic algorithms in neural network training for nonlinear system identification. *Mathematics*, 10(9), 1611.
- Li, Y., Ghoreishi, S.-m., & Issakhov, A. (2021). Improving the Accuracy of Network Intrusion Detection System in Medical IoT Systems through Butterfly Optimization Algorithm. *Wireless Personal Communications*, 1-19.
- Mahboob, A. S., & Moghaddam, M. R. O. (2020). An Anomaly-based Intrusion Detection System Using Butterfly Optimization Algorithm. 2020 6th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS).
- Maheshwari, P., Sharma, A. K., & Verma, K. (2021). Energy efficient cluster based routing protocol for WSN using butterfly optimization algorithm and ant colony optimization. *Ad Hoc Networks*, 110, 102317.
- Yogananda, P., Babu, L. A., & Giri, A. A. (2021). Oppositional Butterfly Optimization Algorithm with Multilayer Perceptron for Medical Data Classification. *Turkish Journal of Computer and Mathematics Education*, 12(10), 2721-2731.
- Zhi, Y., Weiqing, W., Haiyun, W., & Khodaei, H. (2020). Improved butterfly optimization algorithm for CCHP driven by PEMFC. *Applied Thermal Engineering*, 173, 114766.