

Geliştirilmiş Atom Arama Optimizasyon Algoritması ile Çok Katmanlı Algılayıcı Eğitimi

A Novel Improved Atom Search Optimization Algorithm for Training Multilayer Perceptron

Davut İzci



Batman Üniversitesi, Batman Türkiye
davut.izci@batman.edu.tr

Öz

Bu makalede atom arama optimizasyonu (ASO) ve benzetilmiş tavlama (SA) algoritmalarının hibritleştirilmesiyle geliştirilen ve iASO olarak isimlendirilen yeni bir hibrit algoritma ele alınmaktadır. SA tekniğinin kullanımı ile ASO algoritmasının arama yeteneği güçlendirilmiştir. Önerilen hibrit algoritmanın doğrusal olmayan sistemleri optimize etmedeki yeteneğini gözlemek üzere çok katmanlı algılayıcıyı (MLP) eğiticisi olarak kullanılmıştır. Iris, Balloon, XOR, Breast Cancer ve Heart olmak üzere çeşitli veri kümeleri kullanılmış ve elde edilen sonuçlar orijinal ASO, sine cosine algoritması (SCA), parçacık sürüsü optimizasyonu (PSO), karınca kolonisi optimizasyonu (ACO), gri kurt optimizasyonu (GWO) ve genetik algoritma (GA) gibi rekabetçi algoritmalar kullanılarak oluşturulmuş diğer MLP eğiticileri ile karşılaştırılmıştır. Balloon veri seti için önerilen iASO algoritması sadece GA algoritmasının gerisinde bir performans sergilemiş olup, karşılaştırma yapılan diğer tüm algoritmalara göre daha iyi performans sergilediği gözlemlenmiştir. Sonuçlar, Balloon veri seti dışında önerilen yaklaşımla daha düşük ortalama kare hatasının (MSE) ortalama ve standart sapmasının elde edildiğini göstermiş ve dolayısıyla daha iyi performansının olduğunu açıkça göstermiştir.

Anahtar Kelimeler: Çok katmanlı algılayıcı, sezgisel-üstü algoritmalar, atom arama optimizasyonu, benzetilmiş tavlama algoritması

Abstract

A novel hybrid algorithm developed by merging atom search optimization (ASO) and simulated annealing (SA) algorithms is presented. The search capability of ASO was improved by using simulated annealing (SA) algorithm. The proposed hybrid algorithm was named as iASO and used for training multilayer perceptron (MLP) to observe its ability for optimizing non-linear systems. Several datasets (Iris, Balloon, XOR, Breast cancer and Heart) were used, and the obtained results were compared with respective recent competitive algorithms such as original ASO, sine cosine algorithm (SCA), particle swarm optimization (PSO), ant colony optimization (ACO), grey wolf optimization (GWO), and genetic algorithm (GA). The performance of the GA was found to be better than the proposed iASO algorithm only for Balloon dataset. However, the proposed iASO was found to have better performance than all other compared algorithms for the respective dataset. Apart from Balloon dataset, the results clearly indicated the performance of the proposed algorithm to be better as the lower average and

standard deviation of mean square error were achieved via the proposed approach.

Keywords: Multilayer perceptron, metaheuristic algorithms, atom search optimization, simulated annealing

1. Giriş

İnsan beyni, yapay sinir ağları (ANN) tarafından hesaplama modelleri aracılığıyla taklit edilebilmekte ve bu sayede geleneksel matematiksel denklemlerin çözmekte zorlandığı karmaşık ve doğrusal olmayan problemler için kullanılabilir [1]. İleri beslemeli sinir ağlarının bir parçası olan çok katmanlı algılayıcı (MLP) [2], oldukça benimsenen bir sinir ağı olup belirli uygulamalarda kullanılması durumunda eğitilmesi gereken bir yapıya sahiptir [3]. Literatür incelendiğinde, söz konusu bu sinir ağının eğitimi için deterministik yaklaşımlara denk gelmek mümkündür [4]. Ancak, yavaş yakınsama ve yerel optimumda sıkışma gibi durumlar ilgili eğitim sürecinde sıklıkla karşılaşılan sorunlardandır. Bu nedenle, bu tür bir yapının eğitimi için daha yetenekli algoritmalara ihtiyaç duyulmaktadır. Bu noktada sezgisel-üstü algoritmalar daha yetenekli yapılar olarak kullanılabilir.

Sezgisel-üstü algoritmalar MLP eğitiminde arzu edilen sonuçlara ulaşmayı sağlayan alternatif bir yöntem olarak başarısını göstermiştir [5]. Bu algoritmalarından bazıları gri kurt optimizasyonu (GWO) [6], karınca aslanı optimizasyonu (ALO) [7], şempanze optimizasyon algoritması (ChOA) [8], çekirge optimizasyon algoritması (GOA) [9], salp sürüsü algoritması (SSA) [10] ve bu algoritmanın geliştirilmiş versiyonları [11], [12], yer-çekimi arama algoritması (GSA) [13], manyetik optimizasyon algoritması (MOA) [14], yapay alg algoritması (AAA) [15], ağaç tohum algoritması (TSA) [16] ve Harris şahinleri optimizasyonu (HHO) [16] olarak sıralanabilir.

Atom arama optimizasyonu (ASO) [17], son zamanlarda geliştirilen ve moleküler dinamikten esinlenen popülasyon-temelli sezgisel-üstü algoritmalarındandır. Söz konusu bu algoritma çeşitli optimizasyon problemlerinin üstesinden gelmek üzere başarıyla kullanılmıştır [17]–[21]. Ancak diğer birçok sezgisel-üstü algoritmada olduğu gibi, orijinal ASO algoritması da erken yakınsama ve yerel optimuma takılma gibi sorunlar barındırmaktadır [18] ve bu nedenle keşif ve sömürü aşamalarının iyileştirilmesine ihtiyaç duymaktadır.

Sezgisel-üstü algoritmaların sahip olduğu sınırlamaların üstesinden gelmek üzere çeşitli stratejiler önerilmiştir [19]. Hibritleştirme yaklaşımı daha etkili sonuçlar sağladığından,

önerilen stratejiler arasında öne çıkan ve tercih edilen bir yöntem olmuştur. Optimizasyon problemlerinin çözülmesi için sezgisel-üstü yapılar ile hibritleştirilen algoritmalarından biri de benzetilmiş tavlama (SA) algoritmasıdır [22].

SA algoritması yerel arama açısından güçlü olan ve daha az hesaplama süresi gerektiren, bununla beraber uygulanması kolay olan sezgisel-üstü bir tekniktir [23]. Bu özelliğinden dolayı SA algoritmasının hibrit bir model oluşturulmasına olanak tanıyacak şekilde kullanılması mümkündür. Bu durum da söz konusu bu çalışmanın temel motivasyonunu oluşturmakta olup, ASO algoritmasının yukarıda sözü edilen dezavantajlı taraflarının üstesinden gelinmesi amaçlanmaktadır. Bu yolla ASO algoritmasının çözüm kalitesi iyileştirilebilir ve dolayısıyla optimizasyon problemlerinin çözümü için daha iyi bir yapı elde edilebilir.

Bu makale, ASO algoritmasının keşif ve sömürü aşamaları arasındaki denge eksikliğini gidermek üzere SA algoritmasının mükemmel yerel arama yeteneğinin kullanılmasını esas alarak geliştirilmiş atom arama optimizasyonu (iASO) olarak isimlendirilen yeni bir hibrit algoritmayı önermektedir. Geliştirilen bu hibrit algoritma, daha kötü çözümler için SA tekniğini kullanıp, potansiyeli olan komşu çözümlerin de ihmal edilmemesini sağlamaktadır. SA ile ASO algoritmalarının hibritleştirilmesi yoluyla elde edilen iASO algoritmasının, doğrusal olmayan sistem optimizasyonundaki performansını gözlemek üzere MLP eğitiminde kullanılmıştır. Böylelikle, söz konusu bu amaç için daha önce önerilen diğer algoritmalara kıyasla daha iyi performans gösteren bir yapının oluşturulması amaçlanmıştır.

MLP eğitiminde önerilen algoritmanın performansını gözlemek için farklı zorluk seviyelerine sahip Iris, Balloon, XOR, Breast Cancer ve Heart [24] veri kümeleri kullanılmıştır. Elde edilen sonuçlar adil bir karşılaştırma için tüm algoritmaların benzer koşullar altında çalıştırıldığı orijinal ASO, sinüs kosinüs algoritması (SCA), parçacık sürüsü optimizasyonu (PSO), karınca kolonisi optimizasyonu (ACO), gri kurt optimizasyonu (GWO) ve genetik algoritma (GA) gibi yapılar ile oluşturulan MLP eğitimcilerinin sonuçlarıyla karşılaştırılmıştır. Önerilen hibrit algoritma için yapılan testler GA algoritmasının sadece Balloon veri seti için önerilen algoritmadan daha iyi performans sergilediğini göstermiş olup geri kalan diğer algoritmalara göre önerilen iASO algoritmasının yine de daha iyi olduğu gözlemlenmiştir. Diğer tüm veri setleri için ise iASO yapısının daha düşük bir ortalama kare hatasına (MSE) ve standart sapmaya sahip olduğunu ve bu sebeple MLP eğitimi için verimli bir optimizasyon algoritması olduğunu gösterirken, karşılaştırmalar bu algoritmanın rakip diğer algoritmalarından daha iyi performansa sahip olduğunu belirlemiştir.

2. ASO Algoritması

Bir önceki bölümde de bahsedildiği üzere, ASO algoritması moleküler dinamiklerden esinlenen popülasyon tabanlı bir küresel optimizasyon tekniğidir [25]. Diğer bir ifade ile, söz konusu bu algoritma klasik mekanik göre davranan atomik hareketin matematiksel bir temsildir. Newton'un ikinci yasasına göre, bir atomik sistemdeki ilişki eşitlik (1)'de olduğu gibi yazılabilir.

$$a_i = \frac{F_i + G_i}{m_i} \quad (1)$$

Burada, F_i ve G_i , i . atom üzerinde birlikte etkiyen etkileşim ve kısıtlama kuvvetlerini temsil eder. Atomun ivme ve kütlesi ise sırasıyla a_i ve m_i ile gösterilmektedir. d boyutunda ve t zamanında, j . atomdan kaynaklı i . atoma etki eden etkileşim kuvveti bağıntı (2) ile ifade edilebilir.

$$F'_{ij}(t) = -\eta(t) \left[2(h_{ij}(t))^{13} - (h_{ij}(t))^7 \right] \quad (2)$$

Bu bağıntıda, $\eta(t)$ derinlik fonksiyonu olarak adlandırılır ve bağıntı (3)'te olduğu gibi tanımlanmaktadır. α derinlik ağırlığını, T ise maksimum iterasyon sayısını göstermektedir. Söz konusu bu fonksiyon, itme veya çekme bölgelerinin düzenlenmesi için kullanılır.

$$\eta(t) = \alpha \left(1 - \frac{t-1}{T} \right)^3 e^{-\frac{20t}{T}} \quad (3)$$

Eşitlik (2)'de verilen $h_{ij}(t)$ terimi, (4) numaralı bağıntı ile ifade edilmekte olup; ilgili eşitlikteki r , iki atom arasındaki mesafeyi, h_{min} alt sınırı, h_{max} ise üst sınırı temsil etmektedir.

$$h_{ij}(t) = \begin{cases} h_{min}, & \frac{r_{ij}(t)}{\sigma(t)} < h_{min} \\ \frac{r_{ij}(t)}{\sigma(t)}, & h_{min} \leq \frac{r_{ij}(t)}{\sigma(t)} \leq h_{max} \\ h_{max}, & \frac{r_{ij}(t)}{\sigma(t)} > h_{max} \end{cases} \quad (4)$$

Son eşitlikte verilen bu fonksiyon, itme, çekme veya dengenin oluşmasına yardımcı olur. Keşif özelliği alt itme ($h = 1.1$) ve üst çekme sınırlarına ($h = 1.24$) sahip olunca artırılmış olur. Eşitlik (5)'te verilen g_0 ve u terimlerinin sınırları sırasıyla 1.1 ve 1.24 değerlerine eşittir.

$$h_{min} = g_0 + g(t), \quad h_{max} = u \quad (5)$$

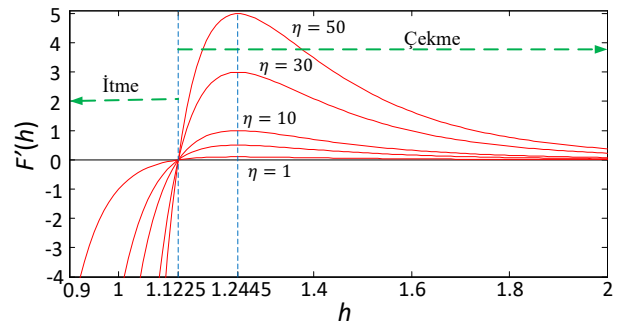
Algoritmanın keşiften sömürüye kaymasını sağlayan sürüklenme faktörü, g ile ifade edilmekte ve bağıntı (6) ile verilmektedir.

$$g(t) = 0.1 \times \sin\left(\frac{\pi}{2} \times \frac{t}{T}\right) \quad (6)$$

Eşitlik (4)'te verilen $\sigma(t)$, uzunluk ölçeğini temsil etmekte ve bağıntı (7)'de verildiği şekliyle tanımlanmaktadır.

$$\sigma(t) = \left\| x_{ij}(t), \frac{\sum_{j \in K_{best}} x_{ij}(t)}{K(t)} \right\|_2 \quad (7)$$

Burada K_{best} , en iyi uygunluk fonksiyonu değerlerini içeren ilk K atomlarının olduğu bir popülasyonunu belirtir. F' fonksiyonunun $\eta(t)$ değerlerine göre davranışı (h değerlerine karşılık gelen) Şekil 1'de gösterilmektedir.



Şekil 1. Farklı η değerlerine karşılık gelen F' fonksiyonu

Diğer atomlardan dolayı d . boyutta i . atom üzerinde etkili olan rasgele ağırlıklara sahip bileşenlerin toplamı, toplam kuvvet olarak ifade edilmekte ve (8) numaralı eşitlik aracılığıyla verilmektedir.

$$F_i^d(t) = \sum_{j \in K_{best}} rand_j F_{ij}^d(t) \quad (8)$$

Eşitlik (8)'deki $rand_j$, terimi $[0,1]$ aralığında rastgele bir sayıyı temsil etmektedir. Moleküler dinamikte atomik hareket, geometrik kısıtlamadan büyük ölçüde etkilenir. ASO algoritmasında bu durum her atom ile en iyi atom arasında kovalent bir bağ varsayımı ile basitleştirilmektedir. Böylelikle, i . Atomun kısıtlanması (9) bağıntısı ile verilen halde yazılabilir.

$$\theta_i(t) = [|x_i(t) - x_{best}(t)|^2 - (b_{i,best})^2] \quad (9)$$

İlgili eşitlikte $x_{best}(t)$, t iterasyonundaki en iyi atom pozisyonunu temsil ederken, $b_{i,best}$ en iyi atom ile i . atom arasındaki sabit bağ uzunluğunu göstermektedir. Böylece, kısıtlama kuvveti eşitlik (10)'da verildiği haliyle elde edilebilir.

$$G_i^d(t) = \lambda(t) (x_{best}^d(t) - x_i^d(t)) \quad (10)$$

Burada, $\lambda(t)$ Langrajyan çarpanıdır ve eşitlik (11)'de olduğu gibi tanımlanır.

$$\lambda(t) = \beta e^{-\frac{20t}{T}} \quad (11)$$

Bağıntı (11)'deki verilen β , çarpan ağırlığını ifade etmektedir. i atomunun t anındaki ivmesi eşitlik (12)'deki gibi yazılabilir.

$$a_i^d(t) = \frac{F_i^d(t)}{m_i^d(t)} + \frac{G_i^d(t)}{m_i^d(t)} \quad (12)$$

Burada $m_i(t)$, t anındaki i atomunun kütesidir. Söz konusu son denklemden de görüleceği üzere, daha büyük kütleli bir atom daha iyi bir fonksiyon uygunluk değeri sağlamakta ve böylece daha az ivmeye neden olmaktadır. i atomunun kütesi eşitlik (13)'te verilen ifade ile hesaplanabilmektedir.

$$m_i(t) = \frac{M_i(t)}{\sum_{j=1}^N M_j(t)} \quad (13)$$

$$M_i(t) = e^{-\frac{Fit_i(t) - Fit_{best}(t)}{Fit_{worst}(t) - Fit_{best}(t)}} \quad (14)$$

$Fit_{best}(t)$ ve $Fit_{worst}(t)$ sırası ile t iterasyonundaki minimum ve maksimum uygunluk değerlerine sahip atomları temsil etmektedirler. Söz konusu bu uygunluk değerleri de eşitlik (15) ve (16) ile verilmektedir.

$$Fit_{best}(t) = \min_{i \in \{1,2,\dots,N\}} Fit_i(t) \quad (15)$$

$$Fit_{worst}(t) = \max_{i \in \{1,2,\dots,N\}} Fit_i(t) \quad (16)$$

Algoritmayı basitleştirmek adına $t + 1$ iterasyonundaki i atomunun hızı ve konumu aşağıdaki gibi ifade edilebilir.

$$v_i^d(t+1) = rand^d v_i^d(t) + a_i^d(t) \quad (17)$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \quad (18)$$

Daha iyi bir keşfi yeteneği için her atom daha iyi uygunluk değerine sahip K komşu atomları ile mümkün olduğunca fazlaca etkileşim içinde olmalıdır. Bahsedilen durumun aksine, sömürü yeteneğinin iyileştirilmesi için her atom daha iyi uygunluk

değerine sahip K komşu atomları ile mümkün olduğunca daha az etkileşim içinde olmalıdır. Burada bahsi geçen K , zamana bağlı bir fonksiyonu temsil eder ve iterasyon sayısına göre kademeli düşüşü göstermek için eşitlik (19)'da olduğu gibi hesaplanır.

$$K(t) = N - (N - 2) \times \sqrt{\frac{t}{T}} \quad (19)$$

3. SA Tekniği

Bu algoritma, metalürjideki tavlama sürecini taklit ederek çözüm üreten bir yöntemdir [26]. Algoritma süreci, temel olarak ısıtma ve soğutma aşamaları aracılığı ile daha az kusurlu tek tip kristal üretilme işleminin simüle edilmiş halinden ibarettir [27]. SA, X_i çözümü için rastgele bir başlangıç değer kümesiyle başlar ve bunun aracılığıyla X'_i komşu çözümünü belirler. Ardından, X_i ve X'_i için uygunluk değerini hesaplar. X'_i çözümüne ait olan $F(X'_i)$ uygunluk değerinin, X_i çözümünün $F(X_i)$ uygunluk değerinden küçük olması durumunda SA tarafından $X_i = X'_i$ şeklinde bir ayarlama yapılmaktadır. Bununla beraber, bahsedilen bu ilişki olmasa bile, SA algoritması X_i çözümünü X'_i çözümü ile değiştirme yoluna yine de gidebilir. Böyle bir durumun gerçekleştirilmesi, eşitlik (20)'de tanımlanan p olasılığına bağlıdır.

$$p = e^{-\frac{\Delta F}{T_k}}; \Delta F = F(X'_i) - F(X_i) \quad (20)$$

Burada F uygunluk fonksiyonunu, T ise sıcaklık kontrol parametresini ifade etmektedir. SA algoritması X_i çözümünü, $p < rand(0,1)$ durumunda X'_i çözümü ile değiştirmeyecektir. Bir sonraki aşamada aşağıdaki denklemleri kullanarak sıcaklık değerini azaltmaktadır.

$$T_{k+1} = \mu T_k \quad (21)$$

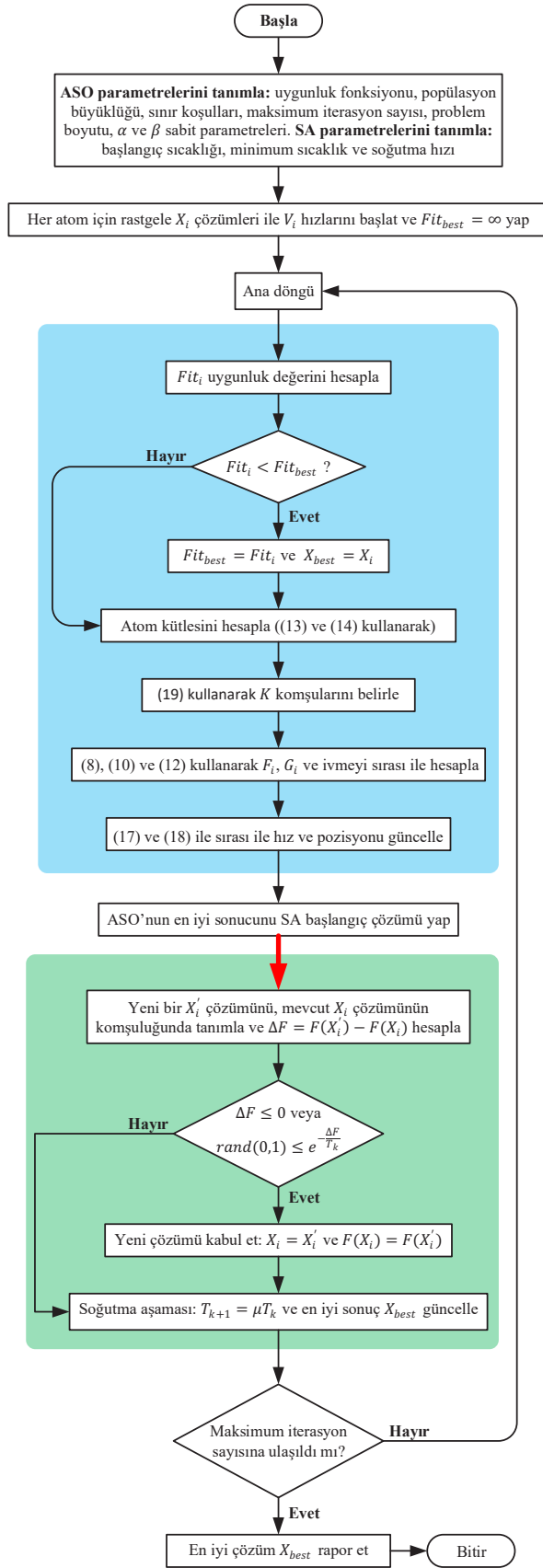
Burada μ , 0 ile 1 arasında rastgele değişen soğutma katsayısını ifade etmektedir.

4. Önerilen Geliştirilmiş Algoritma

Giriş bölümünde bahsedildiği üzere, ASO algoritmasının orijinal versiyonu, çeşitli mühendislik problemlerini çözmeye başarı sergilemiştir. Ancak, diğer küresel optimizasyon tekniklerinin birçoğunda olduğu gibi, bu algoritma da büyük boyutlu sorunlara çözümler sağlamak için yeterince verimli değildir. Bu sorunun üstesinden gelip ASO algoritmasının etkinliğini artırmak için bu çalışmada SA tekniği ASO algoritması ile hibritleştirilerek yeni bir yaklaşım elde edilmiştir.

SA algoritmasının temel faydalarından biri, tepe tırmanma hareketleri ile yerel minimumdan kaçma yeteneğidir. Söz konusu bu durum global bir çözüm arama açısından yararlıdır. Bu algoritma, mevcut çözümü daha iyi olan komşu çözüme doğru kaydırarak arama alanının keşfedilmesi sağlar. Bu yapısı ile de ASO algoritmasının yerel minimuma takılmamasına yardımcı olmaktadır.

Bu çalışma ile MLP eğitimi için önerilen yeni hibrit iASO algoritması, ASO algoritmasının hızlı keşif ve SA algoritmasının da tepe tırmanma özelliğini kullanmaktadır. Önerilen hibrit algoritmanın akış şeması Şekil 2'de gösterilmektedir. İlgili diyagramdan da görülebileceği gibi önerilen hibrit algoritma, ASO ve SA algoritmalarının parametrelerinin başlatılması ve uygunluk değerlerinin sonsuza ayarlanması ile rastgele bir atom kümesi üzerinden başlatılmaktadır.



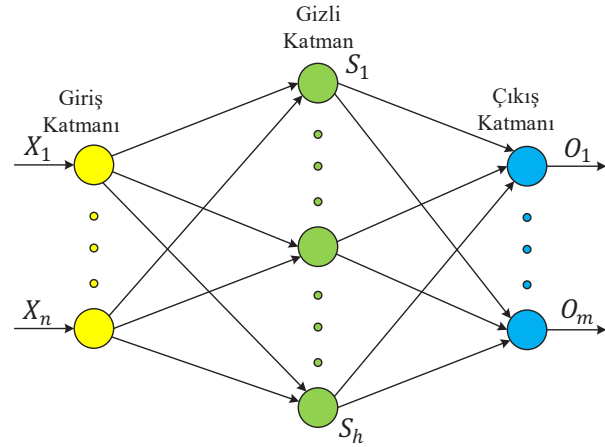
Şekil 2. Önerilen algoritmanın akış şeması

Söz konusu parametreler elde edildikten sonra, iterasyonlar boyunca her bir atom için uygunluk değeri hesaplanır ve ardından elde edilen uygunluk değeri en iyi uygunluk değeri ile karşılaştırılır. Daha iyi değerler olması durumunda, algoritma en iyi çözümü ve uygunluk değerini günceller ve akış şemasındaki adımların geri kalanı yürütülür. Her bir yinelemede, ASO algoritmasının ilgili iterasyon için elde ettiği en iyi çözüm, SA algoritmasının başlangıç çözümü olarak alınmaktadır. Bu yolla ASO algoritmasının arama alanını keşfetme yeteneği artırılmaktadır.

5. Önerilen Algoritma ile MLP Eğitimi

5.1. Çok Katmanlı Algılayıcı

MLP, veri geçişinin gerçekleştiği katmanlı bir yapıya sahip olan ve nöronlar tek yönlü olarak düzenlendiği ileri beslemeli bir sinir ağı olarak tanımlanabilir [28]. Giriş katmanı, gizli katman ve çıkış katmanı olarak adlandırılan paralel katmanlar MLP'lerin yapısını meydana getirmektedir. Şekil 3, bu üç katmana sahip bir MLP'yi göstermektedir. İlgili şekilde n giriş düğümleri sayısını, h gizli katmanı ve m 'de çıkış düğümlerini ifade etmektedir.



Şekil 3. MLP sinir ağının yapısı

MLP çıkışı birkaç adımda hesaplanmaktadır. İlk olarak, ağırlıklı toplamlar eşitlik (22) kullanılarak hesaplanır.

$$s_j = \sum_{i=1}^n (W_{ij}X_i) - \theta_j, j = 1, 2, \dots, h \quad (22)$$

Bu eşitlikte, W_{ij} , giriş katmanının i . düğümünden gizli katmanın j . düğümüne olan bağlantı ağırlığı; X_i , i . girişi; θ_j ise j . gizli katmana ait bias değerini temsil etmektedir. İkinci olarak, her gizli düğümün çıktısı eşitlik (23)'te olduğu gibi hesaplanır.

$$S_j = \text{sigmoid}(s_j) = \frac{1}{1 + e^{-s_j}}, j = 1, 2, \dots, h \quad (23)$$

Gizli düğümlerin çıktıları hesaplandıktan sonra, son çıkışlar eşitlik (24) ve (25)'te olduğu gibi tanımlanır.

$$o_k = \sum_{j=1}^h (\omega_{jk}S_j) - \theta'_k, \quad (24)$$

$$O_k = \text{sigmoid}(o_k) = \frac{1}{1 + e^{-o_k}}, \quad (25)$$

Son iki denklemde $k = 1, 2, \dots, m$ olarak ifade edilmektedir. ω_{jk} , gizli düğüm j 'den çıkış düğümü k 'ya olan bağlantı ağırlığını temsil etmektedir.

MLP'nin nihai çıkışının kalitesi bias değeri ve bağlantı ağırlıklarına bağlı olduğundan bu iki parametre MLP eğitiminde kritik bir rol oynamaktadır. Bu nedenle, bir MLP eğitimi bu parametreler için optimum değerleri bulmak anlamına gelmektedir.

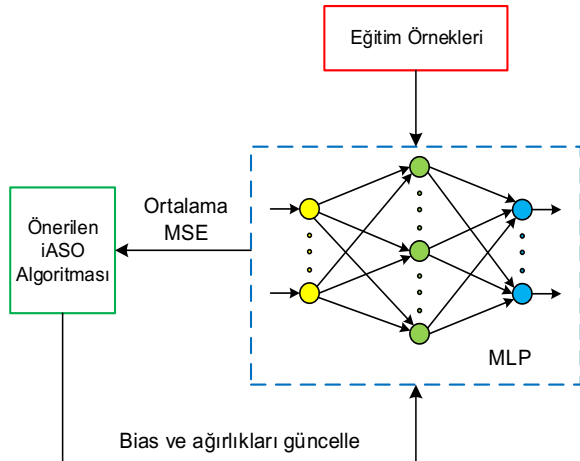
5.2. Önerilen Algoritma Tabanlı MLP Eğiticisi

Sezgisel-üstü algoritmalar kullanarak üç farklı yöntem ile MLP'leri eğitmek mümkündür. İlk yöntem, optimum bağlantı ağırlıklarını ve bias değerlerini bulmak ile ilgilidir. Bu yolla MLP için sezgisel-üstü yöntemler aracılığı ile minimum hatanın elde edilmesi sağlanır. Bahsi geçen bu yöntemde MLP mimarisi öğrenme sürecinde olduğu gibi kalır. İkinci yöntem ise spesifik bir problem durumunda sezgisel-üstü algoritmaları kullanarak MLP için uygun bir mimari yapı bulmakla ilgilidir. Sezgisel-üstü algoritmaların kullanılabileceği sonucu yaklaşım ise gradyan tabanlı öğrenme algoritmasının öğrenme hızı ve momentum gibi parametrelerinin ayarlanması olayıdır.

Önerilen iASO algoritması için yukarıda açıklanan ilk yöntem uygulanmıştır. Çünkü öğrenme algoritması için uygun bias ve ağırlıkların elde edilerek MLP hatasının en aza indirilmesi gerekmektedir. MLP eğitiminde söz konusu bu iki parametrenin temsili önemli bir özelliktir. Bunları temsil etmek için ise binary, matris ve vektör gibi üç yöntem mevcuttur [29]. Bu makalede, bias ve ağırlıkların temsili için vektör yöntemi kullanılmıştır. Bias ve ağırlıkların vektör biçiminde temsil edilmesinden sonra algoritmanın her bir aday çözümünü değerlendirmek üzere amaç fonksiyonu tanımlanmalıdır. Bu çalışmada, eşitlik (26) ile verilen MSE amaç fonksiyonu olarak belirlenmiştir.

$$E = \sum_{k=1}^q \frac{\sum_{i=1}^m (o_i^k - d_i^k)^2}{q} \quad (26)$$

Burada, m çıkışların sayısını, q ise eğitim örneklerinin sayısını ifade etmektedir. d_i^k , k . eğitim örneği kullanıldığında i . giriş biriminin istenen çıkışı ifade ediyor iken o_i^k , söz konusu bu eğitim örneği girişte kullanılıyor iken elde edilen gerçek çıkıştır.



Şekil 4. iASO tabanlı MLP eğiticisi

Şekil 4, önerilen hibrit iASO algoritmasını kullanan MLP eğitiminin genel sürecini göstermektedir. Şekilden de görüldüğü gibi, iASO algoritması MLP'ye bias ve ağırlıkları sağlamakta ve tüm eğitim örnekleri için ortalama MSE değerini almaktadır. iASO algoritması, tüm eğitim örneklerinin ortalama kare hatasını en aza indirmek için bias ve ağırlıkları iteratif olarak değiştirir.

6. İzlenen Yöntem ve Sonuçlar

XOR, Balloon, Iris, Heart ve Breast Cancer olmak üzere beş adet sınıflandırma veri kümesi önerilen iASO algoritmasını kıyaslamak için kullanılmıştır. Bu veri kümeleri kaynak [24]'den elde edilmiştir. Her bir aday çözüm, eğitim algoritmasında $[-10, 10]^D$ aralığından rastgele seçilmiştir. Sınıflandırma problemleri için maksimum iterasyon sayısı 250 ayarlanıp aday çözümlerin popülasyon büyüklüğü Iris, Heart ve Breast Cancer veri kümeleri için 200 alınırken, XOR ve Balloon veri kümeleri için ise 50 olarak seçilmiştir. Veri kümeleri Tablo 1'de sunulduğu gibi sınıflandırılmıştır. Tablo 1'de gösterilen veri kümelerinin farklı zorluk seviyeleri bulunmaktadır. Örneğin, Heart veri kümesi zor olarak değerlendirilirken, XOR veri kümesi basittir.

Tablo 1. Sınıflandırma veri kümeleri

Veri Kümeleri	Özelliklerin Sayısı	Eğitim Örneklerinin Sayısı	Test Örneklerinin Sayısı	Sınıf Sayısı
XOR	3	8	8	2
Balloon	4	16	16	2
Iris	4	150	150	3
Breast cancer	9	599	100	2
Heart	22	80	187	2

Önerilen algoritma, veri kümelerine 30 kez uygulanmış olup elde edilen sonuçlar Tablo 2'den Tablo 6'ya kadar verilmiştir. Bu sonuçlar, algoritmanın son iterasyonunda elde edilen en iyi MSE değerinin ortalamasını ve standart sapmasından oluşmaktadır. Son yinelemede MSE'nin daha düşük ortalama ve standart sapması daha iyi bir performansın göstergesidir. Önerilen iASO algoritmasının performansı, söz konusu verilen sınıflandırma problemlerini çözmek için kullanılan orijinal ASO, SCA [3], GWO [30], PSO [30], ACO [30] ve GA [30] algoritmaları ile karşılaştırılarak değerlendirilmiştir. Çok sayıda eğitim örneği problemin zorluğunu azaltırken, çok sayıda özellik sınır ağırlığı daha büyük boyuta sahip olmasına neden olmakta, dolayısıyla, daha fazla ağırlık belirlenmesini gerektirmektedir ve bu sebepten problemin zorluğunu arttırmaktadır. Dikkate alınan veri kümeleri ve elde edilen sonuçları devam eden paragraflarda açıklanmaktadır.

Kullanılan veri kümelerinden olan XOR, iyi bilinen ve doğrusal olmayan bir sınıflandırma problemidir. Giriş vektöründeki 1'lerin sayısını tanımlamak bu problemin amacıdır. Giriş vektörünü oluşturan 1'lerin tek sayıda olması durumunda çıkış 1 iken; çift sayıda olması durumunda 0'dır. Bu sorunu çözmek için 3-7-1 yapısına sahip MLP kullanıldı. Bu veri kümesi için elde edilen sonuçlar Tablo 2'de sunulmuştur. Önerilen algoritma performansının söz konusu bu problemi çözmeye daha iyi olduğu açıkça gösterilmektedir.

Diğer bir veri kümesi olan Balloon veri seti, string formatında olan renk, yaş, hareket ve boyut gibi 4 özelliğe sahip 16 örnek

içerir. Bu veri kümesinin sınıflandırılması için 4-9-1'lik bir MLP yapısı kullanılmıştır. Elde edilen sonuçlar Tablo 3'te sunulmuş olup, GA algoritmasının önerilen algoritmadan daha iyi bir performans sergileyerek minimum hatayı sağladığı görülmüştür ancak önerilen algoritmanın bu noktada performansı diğer tüm algoritmalarından daha iyi olmuştur. Karşılaştırmada kullanılan tüm algoritmaların sınıflandırma oranları aynı ve %100 olmuştur. Önerilen algoritma sadece Ballon veri seti için GA algoritmasının gerisinde bir performans sergilemiştir ancak ilerleyen paragraflarda da bahsedildiği üzere geri kalan diğer tüm veri setlerinde en iyi sonuçların elde edilmesini sağlamıştır.

Tablo 2. XOR sınıflandırma problemi için deneysel sonuçlar

Eğitim Algoritması	Ortalama	Standart Sapma	Sınıflandırma (%)
iASO-MLP (önerilen)	3.78E-06	7.03E-08	100.00
ASO-MLP	6.38E-03	1.92E-02	100.00
SCA-MLP [3]	3.52E-02	2.69E-02	87.50
GWO-MLP [30]	9.41E-03	2.95E-02	100.00
PSO-MLP [30]	8.40E-02	3.59E-02	37.50
GA-MLP [30]	1.81E-04	4.13E-04	100
ACO-MLP [30]	1.80E-01	2.53E-02	62.50

Üçüncü veri kümesi olarak kullanılan Iris veri seti, Virginica, Versicolor ve Setosa olmak üzere üç sınıf altında değerlendirilebilecek 150 örnek içermektedir. Taç yaprak uzunluğu ve genişliği ile çanak yaprak genişliği ve uzunluğu bu örneklerde bulunan dört özelliklerdir. Bu sınıflandırma probleminin çözümü için 4-9-3 yapısına sahip bir MLP kullanılmıştır. Elde edilen sonuçlar Tablo 4'te verilmiştir. Bu tablodan da görüleceği üzere yapılan karşılaştırmalar sonucu önerilen algoritmanın, MSE amaç fonksiyonu ve sınıflandırma oranı değerlerine dayalı olarak MLP'yi eğitmek üzere daha iyi performans göstererek diğer algoritmalarından üstün olduğu anlaşılmaktadır.

Tablo 3. Balloon sınıflandırma problemi için deneysel sonuçlar

Eğitim Algoritması	Ortalama	Standart Sapma	Sınıflandırma (%)
iASO-MLP (önerilen)	8.18E-16	2.71E-15	100.00
ASO-MLP	3.47E-08	1.52E-07	100.00
SCA-MLP [3]	7.35E-06	1.04E-05	100.00
GWO-MLP [30]	9.38E-15	2.81E-14	100.00
PSO-MLP [30]	5.85E-04	7.49E-04	100.00
GA-MLP [30]	5.08E-24	1.06E-23	100.00
ACO-MLP [30]	4.85E-03	7.76E-03	100.00

Bir diğer veri kümesi olan Breast Cancer veri seti 9 özellik ve 699 örnekten oluşur. Marjinal yapıma ve yığın kalınlığı ile hücre şekli ve boyutu bu veri setinin özelliklerindedir. İyi huylu kanser için çıkış 2 iken kötü huylu kanser için 4'tür. Bu sınıflandırma probleminin çözümü için 9-19-1 yapısına sahip MLP kullanılmıştır. Tablo 5 ile bu problemin sonuçlarını verilmektedir. Tablodan da görüldüğü üzere, önerilen algoritma

vasıtasıyla MSE amaç fonksiyonunun ortalama ve standart sapması en düşük değerini almıştır ve en yüksek sınıflandırma oranı sağlanmıştır. Bu da önerilen algoritmanın daha iyi performansa sahip olduğunun açık bir göstergesidir.

Bu çalışma için kullanılan sonuncu veri kümesini Heart veri seti oluşturmaktadır. Bu veri seti 267 kardiyak tomografi görüntüsünü içermektedir. Bu görüntüleri özetlemek üzere 22 özellik çıkarılmıştır. 22-45-1 yapısına sahip MLP, 80 örnek kullanılarak eğitilmiştir. Bir hastanın durumu, veri setinin binary formu kullanılarak normal veya normal değil olarak ifade edilmektedir. Tablo 6 sonuçları listelemekte ve önerilen algoritmanın diğerlerinden daha iyi sonuçlar ve sınıflandırma oranı sağlayabileceğini açıkça göstermektedir.

Tablo 4. Iris sınıflandırma problemi için deneysel sonuçlar

Eğitim Algoritması	Ortalama	Standart Sapma	Sınıflandırma (%)
iASO-MLP (önerilen)	1.51E-02	2.35E-03	91.33
ASO-MLP	1.64E-02	3.17E-03	89.33
SCA-MLP [3]	1.35E-01	3.68E-02	54.67
GWO-MLP [30]	2.29E-02	3.20E-03	91.33
PSO-MLP [30]	2.29E-01	5.72E-02	37.33
GA-MLP [30]	8.99E-02	1.24E-01	89.33
ACO-MLP [30]	4.06E-01	5.38E-02	32.66

Tablo 5. Breast Cancer sınıflandırma problemi için deneysel sonuçlar

Eğitim Algoritması	Ortalama	Standart Sapma	Sınıflandırma (%)
iASO-MLP (önerilen)	1.08E-03	9.21E-05	100.00
ASO-MLP	3.62E-03	1.88E-03	99.00
SCA-MLP [3]	1.15E-02	5.36E-03	97.00
GWO-MLP [30]	1.20E-03	7.45E-05	99.00
PSO-MLP [30]	3.49E-02	2.47E-03	11.00
GA-MLP [30]	3.03E-03	1.50E-03	98.00
ACO-MLP [30]	1.35E-02	2.14E-03	40.00

Tablo 6. Heart sınıflandırma problemi için deneysel sonuçlar

Eğitim Algoritması	Ortalama	Standart Sapma	Sınıflandırma (%)
iASO-MLP (önerilen)	7.48E-02	4.27E-03	76.25
ASO-MLP	9.64E-02	1.52E-02	73.75
SCA-MLP [3]	1.79E-01	1.28E-02	64.17
GWO-MLP [30]	1.23E-01	7.70E-03	75.00
PSO-MLP [30]	1.89E-01	8.94E-03	68.75
GA-MLP [30]	9.30E-02	2.25E-02	58.75
ACO-MLP [30]	2.28E-01	4.98E-03	00.00

7. Sonuç

Bu çalışmada, orijinal ASO algoritması ile SA tekniği hibritleştirilerek iASO adı verilen yeni bir hibrit algoritma

geliştirilmiştir. Önerilen iASO algoritması MLP eğitiminde kullanılarak performansı değerlendirilmiştir. MLP eğitimi ise XOR, Balloon, Breast Cancer, Iris ve Heart veri kümeleri kullanılarak gerçekleştirilmiştir. Elde edilen sonuçlar, klasik ASO, SCA, GWO, PSO, ACO ve GA gibi algoritmaların kullanımıyla gerçekleştirilen eğitimcilerin performansları önerilen iASO tabanlı MLP eğitimcisi ile karşılaştırılarak değerlendirilmiştir. Önerilen algoritma sadece Balloon veri seti için GA algoritmasının gerisinde bir performans sergilemiş olup söz konusu bu veri seti için GA dışındaki diğer karşılaştırılan algoritmalara göre daha iyi olduğu gözlemlenmiştir. Diğer tüm veri setleri için önerilen iASO'nun, diğer tekniklerle kıyaslandığında MSE ortalaması ve standart sapması ile yüzdelik sınıflandırma oranı açısından etkin olduğu gözlemlenmiştir.

Kaynaklar

- [1] P. Bansal, S. Kumar, S. Pasrija, and S. Singh, "A hybrid grasshopper and new cat swarm optimization algorithm for feature selection and optimization of multi-layer perceptron," *Soft Comput.*, pp. 1–27, 2020, doi: 10.1007/s00500-020-04877-w.
- [2] S. Haykin, *Neural networks: A comprehensive foundation*, 2nd editio. Prentice Hall PTR, 1999.
- [3] S. Gupta and K. Deep, "A novel hybrid sine cosine algorithm for global optimization and its application to train multilayer perceptrons," *Appl. Intell.*, vol. 50, no. 4, pp. 993–1026, 2020, doi: 10.1007/s10489-019-01570-w.
- [4] A. A. Suratgar, M. B. Tavakoli, and A. Hoseinabadi, "Modified Levenberg-Marquardt method for neural networks training," *World Acad Sci Eng Technol*, vol. 6, no. 1, pp. 46–48, 2005.
- [5] E. Eker, M. Kayri, S. Ekinici, and D. Izci, "Training Multi-Layer Perceptron Using Harris Hawks Optimization," in *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, 2020, pp. 1–5, doi: 10.1109/HORA49412.2020.9152874.
- [6] H. Faris, S. Mirjalili, and I. Aljarah, "Automatic selection of hidden neurons and weights in neural networks using grey wolf optimizer based on a hybrid encoding scheme," *Int. J. Mach. Learn. Cybern.*, vol. 10, no. 10, pp. 2901–2920, 2019, doi: 10.1007/s13042-018-00913-2.
- [7] A. A. Heidari, H. Faris, S. Mirjalili, I. Aljarah, and M. Mafarja, "Ant lion optimizer: Theory, literature review, and application in multi-layer perceptron neural networks," in *Studies in Computational Intelligence*, vol. 811, S. Mirjalili, J. Song Dong, and A. Lewis, Eds. Cham: Springer International Publishing, 2020, pp. 23–46.
- [8] M. Khishe and M. R. Mosavi, "Classification of underwater acoustical dataset using neural network trained by Chimp Optimization Algorithm," *Appl. Acoust.*, vol. 157, p. 107005, 2020, doi: 10.1016/j.apacoust.2019.107005.
- [9] A. A. Heidari, H. Faris, I. Aljarah, and S. Mirjalili, "An efficient hybrid multilayer perceptron neural network with grasshopper optimization," *Soft Comput.*, vol. 23, no. 17, pp. 7941–7958, 2019, doi: 10.1007/s00500-018-3424-2.
- [10] M. Khishe and H. Mohammadi, "Passive sonar target classification using multi-layer perceptron trained by salp swarm algorithm," *Ocean Eng.*, vol. 181, pp. 98–108, 2019, doi: 10.1016/j.oceaneng.2019.04.013.
- [11] D. Bairathi and D. Gopalani, "Numerical optimization and feed-forward neural networks training using an improved optimization algorithm: multiple leader salp swarm algorithm," *Evol. Intell.*, pp. 1–17, 2019, doi: 10.1007/s12065-019-00269-8.
- [12] Y. Yin, Q. Tu, and X. Chen, "Enhanced Salp Swarm Algorithm based on random walk and its application to training feedforward neural networks," *Soft Comput.*, vol. 24, no. 19, pp. 14791–14807, 2020, doi: 10.1007/s00500-020-04832-9.
- [13] R. García-Ródenas, L. J. Linares, and J. A. López-Gómez, "Memetic algorithms for training feedforward neural networks: an approach based on gravitational search algorithm," *Neural Comput. Appl.*, 2020, doi: 10.1007/s00521-020-05131-y.
- [14] S. Mirjalili and A. S. Sadiq, "Magnetic Optimization Algorithm for training Multi Layer Perceptron," in *2011 IEEE 3rd International Conference on Communication Software and Networks*, 2011, pp. 42–46, doi: 10.1109/ICCSN.2011.6014845.
- [15] B. Turkoglu and E. Kaya, "Training multi-layer perceptron with artificial algae algorithm," *Eng. Sci. Technol. an Int. J.*, 2020, doi: 10.1016/j.jestch.2020.07.001.
- [16] A. C. Cinar, "Training Feed-Forward Multi-Layer Perceptron Artificial Neural Networks with a Tree-Seed Algorithm," *Arab. J. Sci. Eng.*, 2020, doi: 10.1007/s13369-020-04872-1.
- [17] W. Zhao, L. Wang, and Z. Zhang, "Atom search optimization and its application to solve a hydrogeologic parameter estimation problem," *Knowledge-Based Syst.*, vol. 163, pp. 283–304, 2019, doi: 10.1016/j.knosys.2018.08.030.
- [18] P. Sun, Y. Zhang, J. Liu, and J. Bi, "An Improved Atom Search Optimization with Cellular Automata, a Lévy Flight and an Adaptive Weight Strategy," *IEEE Access*, vol. 8, pp. 49137–49159, 2020, doi: 10.1109/ACCESS.2020.2979921.
- [19] P. Sun, H. Liu, Y. Zhang, L. Tu, and Q. Meng, "An intensify atom search optimization for engineering design problems," *Appl. Math. Model.*, vol. 89, pp. 837–859, 2021, doi: 10.1016/j.apm.2020.07.052.
- [20] B. Hekimoğlu, "Optimal Tuning of Fractional Order PID Controller for DC Motor Speed Control via Chaotic Atom Search Optimization Algorithm," *IEEE Access*, vol. 7, pp. 38100–38114, 2019, doi: 10.1109/ACCESS.2019.2905961.
- [21] S. Ekinici, A. Demiroren, H. Zeynelgil, and B. Hekimoğlu, "An opposition-based atom search optimization algorithm for automatic voltage regulator system," *J. Fac. Eng. Archit. Gazi Univ.*, vol. 35, pp. 1141–1158, Apr. 2020, doi: 10.17341/gazimmmfd.598576.
- [22] X. Pan, L. Xue, Y. Lu, and N. Sun, "Hybrid particle swarm optimization with simulated annealing," *Multimed. Tools Appl.*, vol. 78, no. 21, pp. 29921–29936, 2019, doi: 10.1007/s11042-018-6602-4.
- [23] F. Javidrad and M. Nazari, "A new hybrid particle swarm and simulated annealing stochastic optimization method," *Appl. Soft Comput. J.*, vol. 60, pp. 634–654, 2017, doi: 10.1016/j.asoc.2017.07.023.
- [24] C. L. Blake and C. J. Merz, "UCI Repository of machine learning databases," *University of California*, 1998. <http://archive.ics.uci.edu/ml/>.
- [25] W. Zhao, L. Wang, and Z. Zhang, "A novel atom search optimization for dispersion coefficient estimation in groundwater," *Futur. Gener. Comput. Syst.*, vol. 91, pp. 601–610, 2019, doi: 10.1016/j.future.2018.05.037.
- [26] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi,

- “Optimization by simulated annealing,” *Science (80-.)*, vol. 220, no. 4598, pp. 671–680, 1983, doi: 10.1126/science.220.4598.671.
- [27] B. Hekimoğlu and S. Ekinçi, “Optimally designed PID controller for a DC-DC buck converter via a hybrid whale optimization algorithm with simulated annealing,” *Electrica*, vol. 20, no. 1, pp. 19–27, 2020, doi: 10.5152/ELECTRICA.2020.19034.
- [28] T. Şengüler, E. Karatoprak, and S. Şeker, “A new MLP approach for the detection of the incipient bearing damage,” *Adv. Electr. Comput. Eng.*, vol. 10, no. 3, pp. 34–39, 2010, doi: 10.4316/aecce.2010.03006.
- [29] J. R. Zhang, J. Zhang, T. M. Lok, and M. R. Lyu, “A hybrid particle swarm optimization-back-propagation algorithm for feedforward neural network training,” *Appl. Math. Comput.*, vol. 185, no. 2, pp. 1026–1037, 2007, doi: 10.1016/j.amc.2006.07.025.
- [30] S. Mirjalili, “How effective is the Grey Wolf optimizer in training multi-layer perceptrons,” *Appl. Intell.*, vol. 43, no. 1, pp. 150–161, 2015, doi: 10.1007/s10489-014-0645-7.

Davut İZCİ



Davut İZCİ, lisans derecesini Dicle Üniversitesi Elektrik - Elektronik Mühendisliği Bölümü'nden, yüksek lisans ve doktora derecelerini ise İngiltere'de bulunan Newcastle Üniversitesi'nden (University of Newcastle upon Tyne) sırası ile mekatronik ve mikrosistemler alanlarından almıştır. Şuan da Batman Üniversitesi Elektronik ve Otomasyon Bölümü'nde Dr. Öğretim Üyesi olarak çalışmaktadır. Dr. İZCİ'nin araştırma alanları mikrosistemler, sensörler ve algılama uygulamaları, robotik, kontrol ve uygulamaları ile optimizasyon algoritmalarıdır.