



Dirichlet Sınır Koşullu Diferansiyel Denklemlerin Bir Sınıfının Geometrik Ortalama Optimizasyonu Çözümleri

İclal GÖR^{1,*}, Türkan ÖZYÖRÜK²

¹ Aydın Adnan Menderes Üniversitesi, Fen Fakültesi, Matematik Bölümü, 09010, Aydın, Türkiye

² Aydın Adnan Menderes Üniversitesi, Fen Bilimleri Enstitüsü, Matematik Anabilim Dalı, 09010, Aydın, Türkiye

M A K A L E B İ L G İ S İ

Makale Gönderim 06/06/2024
Revizyon 22/06/2024
Kabul 01/07/2024

Anahtar Sözcükler:

Küresel optimizasyon
Meta-sezgisel algoritma
Geometrik ortalama optimizasyonu
Dirichlet sınır koşullu diferansiyel denklemler

Ö Z E T

Bu çalışma, matematiksel modellemede yaygın olarak karşılaşılan Dirichlet sınır koşullarını sağlayan diferansiyel denklemlerin çözümüne odaklanmaktadır. Bu denklemleri çözmek için Geometrik Ortalama Optimizasyonu (GMO) algoritması kullanılmıştır. Bu amaçla öncelikle ileri beslemeli yapay sinir ağı ile sınır koşullarının sağlanması için ön eğitilmiş ve maliyet fonksiyonu minimize edilerek yapay sinir ağına ait ağırlık güncelleme işlemi gerçekleştirilmiştir. GMO algoritmasının kullanımı, geleneksel çözüm yöntemlerine bir alternatif olarak incelenmiş ve deneysel çalışmalarda benzer performans sağlamıştır. Bu amaçla, Dirichlet koşullarına sahip diferansiyel denklemlerin çözümü araştırılmıştır. Elde edilen sonuçlar, GMO algoritmasının Dirichlet koşullu diferansiyel denklemlerin çözümü için bir alternatif sunduğunu önermektedir.

1. GİRİŞ

Diferansiyel denklemler çeşitli bilimsel problemlerin modellenmesinde kullanılan önemli bir matematik denklemlerdir (Cheng, 2021). Literatürde diferansiyel denklemlerin nümerik çözümleri için önerilen farklı yaklaşımlar yer almaktadır. Yapay sinir ağlarına ek olarak optimizasyon metotları da bu farklı tipteki diferansiyel denklemlerin nümerik çözümlerini hesaplamada tercih edilmektedir. Bu çalışmalardan biri 2006 yılında yüksek mertebeden diferansiyel denklemlerin çözümünde hibrit yapay sinir ağı modeli kullanılarak gerçekleştirilmiştir (Malek ve Beidokhti, 2006). Başka bir çalışmada ise ileri beslemeli yapay sinir ağları kullanılarak başlangıç veya Dirichlet sınır koşullarına sahip birinci ve ikinci mertebeden adi diferansiyel denklemlerin nümerik çözümleri elde edilmiştir (Gör, 2020). Sabir ve arkadaşları Emden-Fowler diferansiyel denklemleri ile temsil edilen üçüncü dereceden doğrusal olmayan çoklu tekil sistemlerin

çözümünde yapay sinir ağları, genetik algoritmalar ve aktif küme algoritmasının etkinliğini kullanarak ileri bir hesaplamalı zeka şeması tasarlamış ve uygulamıştır (Sabir vd., 2020). Adi diferansiyel denklemlerin yaklaşık çözümlerini bulmak için önerilen farklı bir yaklaşımda ise, taban fonksiyonu, ağırlıklı kalıntı yöntemi ve diferansiyel evrim algoritmaları kullanılarak optimize edilen ağsız bir yaklaşım sunulmuş ve bu yöntemin yüksek doğruluk sağladığı gösterilmiştir (Zhang vd., 2020). Fang ve arkadaşları birinci mertebeden tekil gecikmeli diferansiyel denklemleri ve denklem sisteminin yaklaşık çözümünde yapay sinir ağlarını tercih etmiştir (Fang vd. 2020). Gecikmeli diferansiyel denklemlerin farklı bir çeşidi olan pantograf gecikmeli diferansiyel denklemlerinin başlangıç değer problemlerini çözmek için ise Khan ve arkadaşları tarafından yapılan çalışmalarında, doğrusal ve doğrusal olmayan pantograf gecikmeli diferansiyel denklemlerinin başlangıç değer problemlerini çözmek için

*Sorumlu Yazar: iclal@adu.edu.tr



Levenberg-Marquardt Metodu ve Bayes düzenleme metodu tabanlı ileri beslemeli yapay sinir ağları kullanılarak yeni bir hesaplama paradigması sunulmuş ve bu algoritmaların doğruluğu hata histogramları ve regresyon ölçütleri ile doğrulanmıştır (Khan vd., 2020). Gecikmeli diferansiyel denklemlerin nümerik çözümü için farklı bir yaklaşım ise Panghal ve Kumar tarafından önerilmiştir. Yaptıkları çalışmada gecikmeli diferansiyel denklemler ileri beslemeli yapay sinir ağları kullanılarak çözülmüş, aşırı makine öğrenmesi algoritması ile eğitilen farklı derinliklerdeki ağ mimarilerinin performansı incelenmiş ve derin ağların çözümler üzerindeki etkisi analiz edilmiştir (Panghal ve Kumar, 2021). Kesirli diferansiyel denklemlerin başlangıç değer problemini çözmek için ise 2021 yılında Wu ve arkadaşları tarafından dalgacık fonksiyonu tabanlı dalgacık sinir ağı önerilmiş ve algoritmanın yakınsama koşulları verilerek orijinal denklemin yaklaşık çözümüne dönüştürülmesi sağlanmıştır (Wu vd. 2021). Diğer bir çalışmada ise, yapay sinir ağlarının farklı bir çeşidi olan tek katmanlı radyal tabanlı fonksiyon ağı ile aşırı makine öğrenmesi algoritması adı diferansiyel denklemleri çözmek için yeni bir sayısal yöntem geliştirilmiş ve deneysel karşılaştırmalar, bu yöntemin mevcut yöntemlerden daha iyi performans gösterdiğini kanıtlamıştır (Liu vd., 2023).

Özel olarak Dirichlet koşullarını sağlayan diferansiyel denklemler, çeşitli bilim ve mühendislik disiplinlerinde geniş bir uygulama yelpazesine sahiptir. Bu denklemler, birçok fiziksel sistem ve mühendislik problemlerinde karşılaşılan önemli matematiksel problemlerdir. Bu denklemlerin farklı çeşitleri için literatürde önerilen çözüm yöntemleri yer almaktadır. Kowalski tarafından yapılan çalışmada Dirichlet sınır değer koşullarına sahip bir Duffing tipi denklem için çözümün parametreye bağıllığını incelemek amacıyla doğrudan değişkenler yöntemi kullanılmıştır (Kowalski, 2013). Dirichlet sınır koşullarına sahip ikinci dereceden lineer Fredholm integro-diferansiyel denklemler için ise hızlı çok ölçekli Galerkin yöntemi geliştirilen çalışmada, önerilen yöntemin kararlı olduğu, optimal yakınsama düzenine sahip olduğu ve neredeyse lineer hesaplama karmaşıklığına sahip olduğu logaritmik bir faktörle kanıtlanmıştır (Chen vd., 2020). Kesirli Dirichlet sınır değer probleminin zayıf çözümlerinin varlığının incelendiği çalışmada ise Sousa ve arkadaşları Nehari manifold ve değişkenler yöntemleri kullanarak farklı tipteki Dirichlet sınır değer problemi için çözüm önermiştir (Sousa vd., 2022).

Bununla birlikte Dirichlet koşulu içeren denklemlerin çözümünün varlığı birçok çalışmada incelenmiştir. 2018 yılında yapılan çalışmada ani etkiler içeren Dirichlet sınır değer problemi için aşık olmaya klasik çözümlerin varlığı, varyasyonel yöntemler ve kritik nokta teorisi kullanılarak gösterilmiştir (Graef vd., 2018). Ancak, Dirichlet koşulu içeren denklemleri çözmek genellikle analitik olarak zordur ve sayısal çözümleri karmaşıktır. Bu nedenle Dirichlet koşullarını sağlayan diferansiyel denklemlerin nümerik çözümlerini elde etme amacıyla literatürde farklı yaklaşımlar yer almaktadır. Nümerik çözümleri elde etmede önerilen bu yaklaşımlara alternatif olarak yapay sinir ağlarının çözümü örnek olarak verilebilir. Günel ve arkadaşları 2018 yılında yaptıkları çalışmada, adi diferansiyel denklemler için Dirichlet sınır değer problemlerinin sayısal çözümlerini elde etmek amacıyla devirli sinirsel ağ kullanılmış, ağın eğitiminde Parçacık Sürü Optimizasyonu yöntemi kullanılmış ve farklı aktivasyon fonksiyonlarıyla elde edilen çözümler karşılaştırılarak devirli sinirsel ağların etkinliği değerlendirilmiştir (Günel vd. 2018). Aynı yılda yapılan başka bir çalışmada, Dirichlet sınır koşulları ve zamanla değişen gecikmelere sahip stokastik reaksiyon-difüzyon sinir ağlarının ϕ -tipi kararlılığı ve sağlam kararlılığı incelenmiş; bu kararlılığı sağlamak için gerekli kriterler belirlenmiş ve teorik sonuçlar sayısal simülasyonlarla doğrulanmıştır (Sheng vd., 2018). Üçüncü dereceden doğrusal olmayan ve çoklu tekil sistemleri çözmek için ise başka bir çalışmada yapay sinir ağları, genetik algoritmalar ve aktif küme algoritması kullanılarak ileri bir hesaplamalı zeka yöntemi tasarlanmış ve uygulanmış, bu yöntemin etkinliği beş farklı tekil model varyantı üzerinde test edilmiştir. (Sabir vd., 2020). Cheng ve Zhang tarafından yapılan çalışmada akışkan dinamiği problemlerini çözmek için Navier-Stokes denklemlerini ve başlangıç ile sınır koşullarını maliyet fonksiyonuna dahil eden Resnet blokları ile birleşik fiziksel sinir ağı geliştirilmiş ve bu yöntemin, geleneksel derin öğrenme yöntemlerinden daha güçlü öngörü yeteneği sağladığı gösterilmiştir (Cheng ve Zhang, 2021). Dirichlet sınır koşullarına sahip adi diferansiyel denklemlerin sayısal çözümlerini elde etmek için 2022 yılında önerilen çalışmada, sürü zekası tekniklerinin ileri beslemeli sinir ağlarıyla kullanımı incelenmiş ve optimizasyon algoritmalarının sayısal çözümünü iyileştirmek için yapay arı kolonisi, karınca kolonisi optimizasyonu, kütle çekimi arama algoritması ve parçacık sürü optimizasyonu gibi yöntemlerin yanı sıra bu yöntemlerin hibrit kombinasyonları uygulanmıştır (Günel ve Gör, 2022).

Bu çalışmada ise literatürde önerilen Geometrik Ortalama Optimizasyonu (GMO) ile Dirichlet koşulu içeren diferansiyel denklemlerin nümerik çözümlerinin incelenmesi amaçlanmıştır. GMO bir optimizasyon algoritmasıdır ve genellikle küresel optimizasyon problemlerini çözmek için kullanılır. Temel amacı, bir hedef fonksiyonun belirli bir değerini minimize etmek veya maksimize etmek için en iyi çözümü bulmaktır. Bu amaçla matematikte bilinen geometrik ortalamaya benzer bir yaklaşım kullanılarak sözde geometrik ortalama kavramı ile optimizasyon algoritması önerilmiştir (Rezaei vd., 2021).

GMO algoritmasının adımlarında, bir arama ajanının uygunluk değeri, karşıt ajanlarının uygunluklarına göre belirlenir. Her iterasyonda, ajanların o ana kadar ulaştıkları en uygun konum, yani ajanın mevcut iterasyona kadar elde ettiği en iyi konum, kesin olarak belirlenir. Bu adımdan sonra, verilen bir ajanın o ana kadar ki en iyi konumuna ilişkin her karşıt ajan için, üyelik değerlerinin çarpılması ile ikili uygunluk indeksleri hesaplanır. Bu ifade, algoritmanın minimizasyon süreci içinde hem uygunluğu hem de çeşitliliği eşzamanlı olarak artırarak potansiyel olarak daha elverişli bir senaryo oluşturma kapasitesini vurgular. Literatürde, mühendislik alanında kafes yapı tasarımı gibi çok amaçlı yaklaşım ve uygulama alanını içeren Geometrik Ortalama Optimizasyon algoritması üzerine bazı araştırmalar bulunmaktadır (Rezaei vd., 2023).

Çalışmanın takip eden bölümünde GMO algoritması ve Dirichlet sınır GMO ile çözümü detaylı bir şekilde açıklanacaktır. Bir sonraki bölümde Dirichlet koşulu içeren lineer ve lineer olmayan diferansiyel denklemlerin nümerik çözümlerini incelemek amacıyla diferansiyel denklemlerin GMO ve PSO ile çözümleri incelenecektir. Çalışmanın son bölümünde ise elde edilen çözümler analiz edilerek yöntemlerin etkinliği tartışılacaktır.

2. MATERYAL VE METOT

Bu bölümde Dirichlet koşulu sağlayan diferansiyel denklemlerin çözümü için bilinen optimizasyon algoritmalarından GMO algoritması anlatılacaktır.

2.1. Geometrik Ortalama Optimizasyonu

Geometrik Ortalama Optimizasyon Algoritmasının başlangıç aşamasında arama alanında rastgele ajanlar oluşturularak bu ajanların problem

üzerindeki sonuçları hesaplanır (Rezaei vd., 2023). Üyelik değerleri (*MFI*) her ajan için belirlenir. Zadeh'in öne sürdüğü gibi bulanık mantığın temel ilkelerine dayanan yöntem ile hesaplanan üyelik değerleri, birden fazla değişken arasındaki benzerliğin ayırt edilmesine olanak tanır (Zadeh, 1965). Çarpım bazlı Larsen uygulama fonksiyonu olarak bilinen bu özel çarpımsal prosedür, bulanık sistem çalışmaları alanında yaygın olarak kullanılmaktadır. Ardından sözde geometrik ortalama adı verilen işlem uygulanarak ikili uygunluk indeksleri hesaplanır. Her bir ajanın ikili uygunluk indeksi hesaplanırken, karşıt ajanların üyelik değerleri kullanılır. Bu durumda ajanların birbirlerinden etkilendiğini söylenebilir. Elde edilen verilerle oluşturulan kılavuz ajan ve hız vektörüne bağlı olarak ajanların konum güncellemeleri yapılır. Ayrıca bu algoritmada Gauss mutasyonu ve elitizm işlemleri uygulanmaktadır. Gauss mutasyonu ile algoritmaya rassalık katılmıştır. Elitizm işlemi ile de elit bireyler üzerinden algoritma adımları tamamlanarak algoritmanın daha hızlı bir şekilde sonuca ulaşması amaçlanmıştır.

Geometrik Ortalama Optimizasyonu (GMO), çeşitli problemlerin çözümünde diğer algoritmalara göre avantajlar sağlayan bir optimizasyon algoritmasıdır. Her optimizasyon algoritmasında olduğu gibi GMO'da da sorunun çözümü için iki görevin yerine getirilmesi gerekmektedir: keşif ve kullanım. Diğer meta-sezgisel algoritmalarda keşif ve kullanım için iki farklı değişkenin bulunduğunu ve bu durumda algoritmanın uygunluğu yüksek ancak çeşitliliğin düşük olabileceği veya tam tersi bilinir. Bu durumda hangi ajanın rehber olarak seçileceğini belirlemek zor olabilir. GMO algoritması, ikili uygunluk indeksi (*DFI*) adı verilen bir değişkenle ajanların hem uygunluğu hem de çeşitliliği hakkında bilgi sağlanabilir. Bu indeks, popülasyondaki tüm ajanlar için hesaplanır ve arama süreci sırasında her arama ajanı için yerel bir rehber tanımlamak için kullanılır.

İlk olarak Zadeh tarafından ortaya atılan bulanık mantık kavramlarına bakıldığında, çeşitli değişkenler arasındaki benzerlik, bu değişkenlere atfedilen bulanık üyelik fonksiyonu (*MF*) değerlerinin çarpılmasıyla ifade edilebilir (Zadeh, 1965). Bu çarpma, bulanık sistemlerde geniş çapta uygulanan, çarpım tabanlı Larsen uygulama fonksiyonu olarak bilinir. (*MF*) değerlerini tanımlamak için Eşitlik 1 kullanılır.

$$f(x) = \frac{1}{1+e^{a(x-c)}}; a < 0 \quad (1)$$

Denklem 1 de bulunan, a ve c parametreleri sigmoidal MF 'nin belirli özelliklerini belirtir. Ayrıca e Napier sabitidir. Her bir ajan için MF değeri hesaplaması $j = 1, 2, \dots, N$ için Eşitlik 2 ile verdiği gibi yapılır.

$$MF_j^t = \frac{1}{1 + e^{-\frac{4}{\sigma^t \sqrt{e}} \times (Z_{best_j}^t - \mu^t)}} \quad (2)$$

Burada $Z_{best_j}^t$, j 'inci ajanın şimdiye kadarki kişisel en amaç fonksiyonu değeridir; μ^t ve σ^t , t 'inci iterasyondaki ortalama ve standart sapmalardır.

Daha sonra bu MF değerlerine bağlı olarak ikili uygunluk indeksi dediğimiz değer Eşitlik 3 ile hesaplanır:

$$DFI_i^t = \prod_{j=1, j \neq i}^N MF_j^t \quad (3)$$

Eşitlik 3 ile her bir ajan için, ikili eşleşme indeksi hesaplamalarında karşıt ajanların (kendisi dışındaki diğer ajanların) MF değerlerinin kullanıldığı görülmektedir. Yani algoritmamızdaki etmenlerin birbirleriyle etkileşim halinde oldukları söylenebilir.

Rehber ajanların belirlenmesinde Eşitlik 4 kullanılmaktadır.

$$Y_i^t = \frac{\sum_{j=1, j \neq i}^N DFI_j^t \times X^t}{\sum_{j \in N_{best}} DFI_j^t + \varepsilon_i} \quad (4)$$

Rehber ajanların çeşitliliğini artırmak ve rastgelelik eklemek için Gauss mutasyon işlemi uygulanır. Mutasyonun uygulanmasını Eşitlik 5. ile verilen denklem ile yapılır.

$$Y_{i,mut}^t = Y_i^t + \omega \times randn \times (STD_{max}^t - STD^t) \quad (5)$$

Yeni konum vektörünü belirlerken tek yapılması gereken, elde edilen hız vektörü kadar ileri gitmektir. Bunun için gerekli olan yeni hız vektörü Eşitlik 6 ile verilmiştir.

$$\begin{aligned} V_i^{t+1} &= \omega \times V_i^t + \varphi \times (Y_{i,mut}^t - X_i^t) \varphi \\ &= 1 + (2 \times rand - 1) \times \omega \end{aligned} \quad (6)$$

Elde edilen hız vektörüne bağlı olarak konum güncellenmesi Eşitlik 7. ile verilmiştir.

$$X_i^{t+1} = X_i^t + V_i^{t+1} \quad (7)$$

Bir sonraki bölümde GMO algoritması kullanılarak Dirichlet Sınır Değer Probleminin çözümü anlatılacaktır.

2.2. Dirichlet Sınır Koşulunun Geometrik Ortalama Optimizasyon Algoritması ile Çözümü

İlk olarak Dirichlet sınır değeri probleminin çözümü için yapay sinir ağı oluşturulmuştur. Kullanılan yapay sinir ağı, gizli katmanında 10 nöron bulunan tek katmanlı ileri beslemeli sinir ağıdır. GMO algoritması ile yapay sinir ağı probleminin sınır koşullarını sağlaması amacıyla eğitilmiştir. Eğitim aşamasında Levenberg-Marquardt yöntemi kullanılmaktadır. Eğitim sonrasında sinir ağı ile elde edilen hatayı GMO kullanılarak en aza indirmek amaçlanmaktadır. Böylece yapay sinir ağının ağırlıkları ve eşik değerleri güncellenmektedir.

İkinci dereceden bir diferansiyel denklem için Dirichlet sınır değeri problemi Eşitlik 8 ile verildiği gibi tanımlanır.

$$\begin{cases} y''(t) = f(t, y(t), y'(t)), t \in [a, b] \\ y(a) = A, \\ y(b) = B \end{cases} \quad (8)$$

Tanımda verilen Dirichlet sınır değeri problemi için sınır değer koşullarını sağlayan deneme fonksiyonu Eşitlik 9. ile tanımlanır.

$$y_T(t, \vec{p}) = \frac{(t-b)A - (t-a)B}{(a-b)} + (t-a)(t-b)Net(t, \vec{p}) \quad (9)$$

Bilinmeyen parametre değerlerinin belirlenmesi, Eşitlik 11 ile verilen maliyet fonksiyonunun en aza indirilmesini gerektirir. Eşitlik 10 ile tanımlanan deneme fonksiyonunun giriş değerine göre kısmi türevleri Eşitlik 11 ve Eşitlik 12 tarafından verilmiştir.

$$E = \frac{1}{2} \sum_{j=1}^n \left\{ \frac{\partial^2 y_T(t_j)}{\partial t_j^2} - f \left(t_j, y_T(t_j), \frac{\partial y_T(t_j)}{\partial t_j} \right) \right\}^2 \quad (10)$$

$$\begin{aligned} \frac{\partial y_T(t_j)}{\partial t_j} &= \frac{A-B}{a-b} + (2t_j - a - b)Net(t_j, \vec{p}) \\ &\quad + (t_j - a)(t_j - b) \frac{\partial Net(t_j, \vec{p})}{\partial t_j} \end{aligned} \quad (11)$$

$$\begin{aligned} \frac{\partial^2 y_T(t_j)}{\partial t_j^2} &= 2Net(t_j, \vec{p}) + 2(2t_j - a - b) \frac{\partial Net(t_j, \vec{p})}{\partial t_j} \\ &\quad + (t_j - a)(t_j - b) \frac{\partial^2 Net(t_j, \vec{p})}{\partial t_j^2} \end{aligned} \quad (12)$$

Yapay sinir ağlarının birinci ve ikinci dereceden kısmi türevi sırasıyla Eşitlik 13 ve Eşitlik 14 ile verilmiştir.

$$\frac{\partial^2 Net(t_j, \vec{p})}{\partial t_j^2} = 2Net(t_j, \vec{p}) + 2(2t_j - a - b) \frac{\partial Net(t_j, \vec{p})}{\partial t_j}$$

$$+(t_j - a)(t_j - b) \frac{\partial^2 \text{Net}(t_j, \vec{p})}{\partial t_j^2} \quad (13)$$

$$\frac{\partial \text{Net}(t_j, \vec{p})}{\partial t_j} = \frac{\text{Net}(t_j + h, \vec{\omega}) - 2 * \text{Net}(t_j, \vec{\omega}) + \text{Net}(t_j - h, \vec{\omega})}{h^2} \quad (14)$$

Dirichlet sınır problemleri için Gradyan Düşüm Algoritmasını kullanırken, E maliyet fonksiyonunun bilinmeyen parametrelere göre türevlerinin

Çizelge 1: Örnek 1 ve Örnek 2 için PSO'da ve GMO'da kullanılan parametre değerleri.

Parametre	Parametre değerleri
YSA'daki nöron sayısı, m	10
Popülasyon sayısı, N	50
Arama uzayının alt sınırı	-1
Arama uzayının üst sınırı	1
Arama uzayının boyutu, d	3m
Eğitim kümesi için adım uzunluğu, h	0.01
İterasyon sayısı	100

hesaplanması gerekir. Ancak yüksek boyutlu problemlerde yerel minimuma takılma olasılığı yüksektir. Bu nedenle, optimum parametreleri belirlemek için bir sonraki bölümde Geometrik Ortalama Optimizasyon Algoritması kullanılarak Dirichlet koşulu içeren farklı tipte denklemlerin çözümü incelenmiştir.

3. BULGULAR VE TARTIŞMA

GMO algoritmasının Dirichlet koşulu içeren diferansiyel denklemlerin nümerik çözümlerinde alternatif bir optimizasyon algoritması olduğunu incelemek amacıyla lineer ve lineer olmayan diferansiyel denklemlerin nümerik çözümleri bu bölümde incelenecektir. Eşitlik 8 ile verilen diferansiyel denklem lineer, Eşitlik 9 ile verilen denklem ise lineer değildir (Gor, 2020). Bu çalışmada kullanılan parametreler Örnek 1 ve Örnek 2 için sırasıyla PSO ve GMO algoritmaları için Çizelge 1 ile verilmiştir.

Örnek 1.

$$\begin{cases} y'' = \frac{-2t}{1+t^2} y' + y + \frac{2}{1+t^2} - \log(1+t^2), t \in [0,1] \\ y(0) = 0, \\ y(1) = \log(2) \end{cases} \quad (15)$$

Eşitlik 15 ile verilen denklemin gerçek çözümü $\log(1+t^2)$ 'dir. Verilen denklemin çözümlerinde PSO ve GMO ile elde edilen mutlak hata değerlerinin sonuçları eğitim ve test kümeleri için sırasıyla Çizelge 2 ve Çizelge 3 ile verilmiştir. Verilen çizelgelerde görüldüğü üzere elde edilen çözümler eğitim ve test kümeleri için bazı noktalarda verilmiştir.

Maliyet fonksiyonunun her iterasyonda değişimi ise Şekil 3 ve Şekil 4 ile sırasıyla PSO ve GMO için verilmiştir.

Ayrıca eğitim ve test kümesi için elde edilen ortalama karesel hata değerleri (MSE) ile saniye cinsinden geçen süreler PSO ve GMO için Çizelge 4 ile verilmiştir.

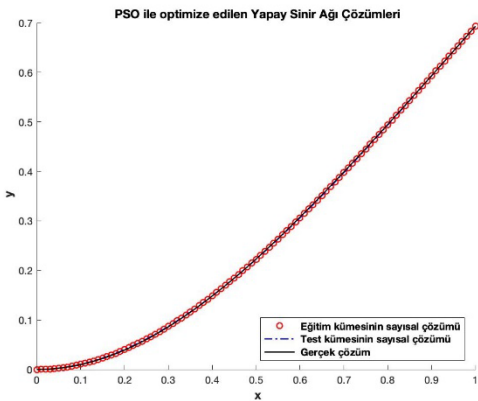
Çizelge 2: Örnek 1'in eğitim kümesindeki bazı noktalar için nümerik çözümünden elde edilen mutlak hata değerler.

x	PSO Çözümü	GMO Çözümü
0.000	0.000	0.000
0.010	1.396×10^{-5}	3.400×10^{-3}
0.020	1.241×10^{-5}	6.670×10^{-3}
0.030	2.634×10^{-6}	9.810×10^{-3}
0.040	2.927×10^{-5}	1.282×10^{-2}
0.050	6.570×10^{-5}	1.570×10^{-2}
0.060	1.102×10^{-4}	1.845×10^{-2}
0.070	2.171×10^{-4}	2.107×10^{-2}
0.080	8.257×10^{-4}	2.356×10^{-2}
0.090	2.767×10^{-4}	2.593×10^{-2}
0.100	3.386×10^{-4}	2.817×10^{-2}
0.150	6.438×10^{-4}	3.753×10^{-2}
0.200	8.668×10^{-4}	4.396×10^{-2}
0.250	9.459×10^{-4}	4.768×10^{-2}
0.300	8.634×10^{-4}	4.899×10^{-2}

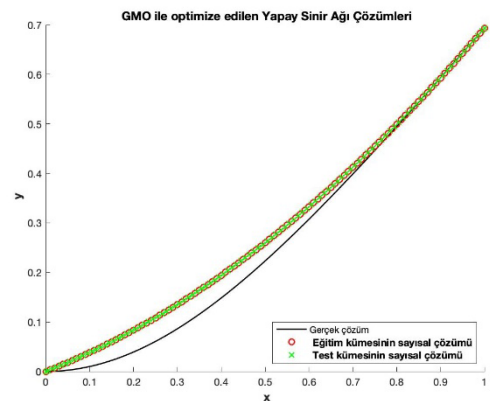
0.350	6.353×10^{-4}	4.820×10^{-2}
0.400	3.005×10^{-4}	4.566×10^{-2}
0.450	8.951×10^{-4}	4.174×10^{-2}
0.500	4.797×10^{-4}	3.679×10^{-2}
0.550	8.190×10^{-4}	3.117×10^{-2}
0.600	1.066×10^{-3}	2.523×10^{-2}
0.650	1.194×10^{-3}	1.928×10^{-2}
0.700	1.1919×10^{-3}	1.365×10^{-2}
0.750	1.066×10^{-3}	8.591×10^{-2}
0.800	8.435×10^{-4}	4.370×10^{-3}
0.850	5.629×10^{-4}	1.210×10^{-3}
0.900	2.811×10^{-4}	6.860×10^{-4}
0.950	6.714×10^{-4}	1.141×10^{-3}
1.000	0.000	0.000

Çizelge 3: Örnek 1'in test kümesindeki bazı noktalar için nümerik çözümünden elde edilen mutlak hata değerleri.

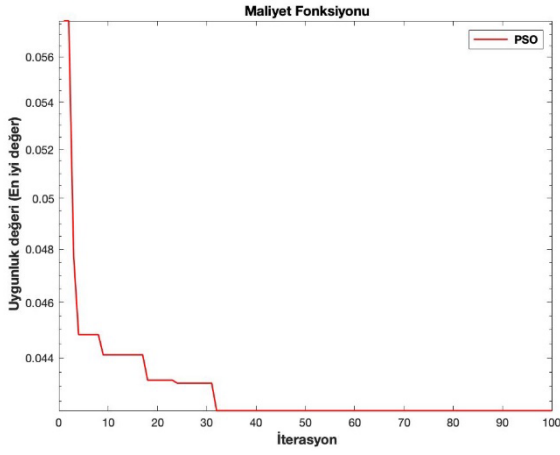
x	PSO Çözümü	GMO Çözümü
0.000	0.000	0.000
0.005	9.055×10^{-6}	1.717×10^{-3}
0.015	1.500×10^{-5}	5.052×10^{-3}
0.025	6.458×10^{-6}	8.256×10^{-3}
0.035	1.462×10^{-5}	1.133×10^{-2}
0.045	4.637×10^{-5}	1.427×10^{-2}
0.055	8.705×10^{-5}	1.709×10^{-2}
0.065	1.349×10^{-4}	1.978×10^{-2}
0.075	1.886×10^{-4}	2.233×10^{-2}
0.085	2.465×10^{-4}	2.476×10^{-2}
0.095	3.074×10^{-4}	2.707×10^{-2}
0.105	3.700×10^{-4}	2.925×10^{-2}
0.205	8.818×10^{-4}	4.445×10^{-2}
0.255	9.448×10^{-4}	4.791×10^{-2}
0.305	8.466×10^{-4}	4.900×10^{-2}
0.355	6.059×10^{-4}	4.802×10^{-2}
0.405	2.631×10^{-4}	4.533×10^{-2}
0.455	1.294×10^{-4}	4.128×10^{-2}
0.505	5.167×10^{-4}	3.625×10^{-2}
0.555	8.484×10^{-4}	3.058×10^{-2}
0.605	1.084×10^{-3}	2.463×10^{-2}
0.655	1.199×10^{-3}	1.870×10^{-2}
0.705	1.184×10^{-3}	1.311×10^{-2}
0.755	1.048×10^{-3}	8.127×10^{-3}
0.805	8.172×10^{-4}	4.002×10^{-3}
0.855	5.337×10^{-4}	9.607×10^{-4}
0.905	2.555×10^{-4}	7.991×10^{-4}
0.955	5.243×10^{-5}	1.900×10^{-4}
1.000	0.000	0.000



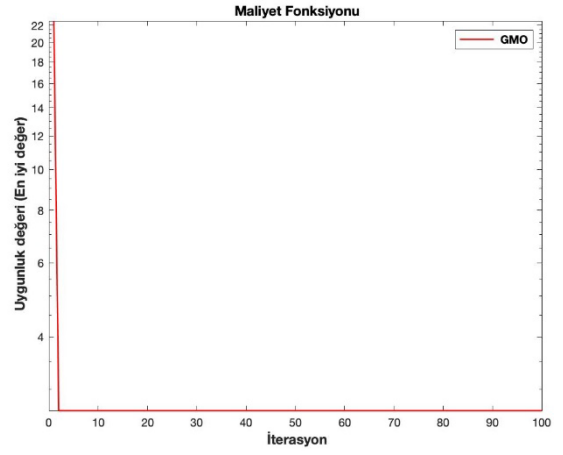
Şekil 1: Örnek 1 için PSO kullanılarak elde edilen yaklaşık çözüm grafiği



Şekil 2: Örnek 1 için GMO kullanılarak elde edilen yaklaşık çözüm grafiği



Şekil 3: Örnek 1 için PSO kullanılarak elde edilen iterasyona bağlı maliyet fonksiyonu grafiği



Şekil 4: Örnek 1 için GMO kullanılarak elde edilen iterasyona bağlı maliyet fonksiyonu grafiği

Çizelge 4: Örnek 1'in eğitim ve test kümesi için elde edilen ortalama karesel hata değerleri ve saniye cinsinden eğitim ve test süreleri için geçen süreler.

	PSO	GMO
Eğitim Kümesi için MSE	$7.877 \times 10^{-7} \pm 2.673 \times 10^{-7}$	$9.192 \times 10^{-3} \pm 8.473 \times 10^{-4}$
Test Kümesi için MSE	$7.799 \times 10^{-7} \pm 2.647 \times 10^{-7}$	$9.264 \times 10^{-3} \pm 8.539 \times 10^{-4}$
Modelin Eğitim Süresi	$5.509 \pm 2.733 \times 10^{-1}$	$9.538 \times 10^1 \pm 1.523$
Modelin Test Süresi	$2.776 \times 10^{-4} \pm 1.018 \times 10^{-4}$	$9.050 \times 10^{-3} \pm 8.472 \times 10^{-4}$

Örnek 2.

$$\begin{cases} y'' + y^2 = 0, t \in [0,1] \\ y(0) = 1, \\ y(1) = 0 \end{cases} \quad (16)$$

Eşitlik 16 ile verilen Dirichlet koşulu içeren diferansiyel denklemin gerçek çözümü $\tan\left(\frac{\pi}{4}(1-t)\right)$ ' dir. Denklem belirlenmiş noktalarda hem eğitim hem test kümeleri için elde edilen nümerik çözümler sırasıyla Çizelge 5 ve Çizelge 6 ile verilmiştir.

Örnek 2 için yaklaşık çözüm grafikleri olan Şekil 5 ve Şekil 6, sırasıyla PSO ve GMO

algoritmaları kullanılarak elde edilen yaklaşık çözüm grafikleri olarak verilmiştir.

Maliyet fonksiyonunun her iterasyonda değişimi ise Şekil 3 ve Şekil 4 ile sırasıyla PSO ve GMO için verilmiştir.

Bununla birlikte elde edilen ortalama karesel hata değerleri (MSE) PSO ve GMO için elde edilmiştir. Buna göre eğitim ve test kümesi için elde edilen MSE değerleri Çizelge 7 ile vermiştir. Çizelge de ayrıca çalışmada yapılan zaman analizine göre, sırasıyla eğitim ve test kümeleri için saniye cinsinden geçen süreler sunulmuştur.

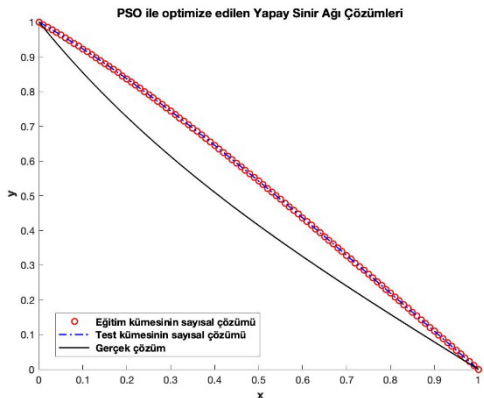
Çizelge 5: Örnek 2'nin eğitim kümesindeki bazı noktalar için nümerik çözümünden elde edilen mutlak hata değerleri.

x	PSO Çözümü	GMO Çözümü
0.000	1.110×10^{-16}	1.110×10^{-16}
0.010	8.245×10^{-3}	6.359×10^{-4}
0.020	1.615×10^{-2}	1.133×10^{-3}
0.030	2.373×10^{-2}	1.497×10^{-3}
0.040	3.099×10^{-2}	1.737×10^{-3}
0.050	3.794×10^{-2}	1.860×10^{-3}
0.060	4.458×10^{-2}	1.870×10^{-3}
0.070	5.0933×10^{-2}	1.775×10^{-3}
0.080	5.699×10^{-2}	1.581×10^{-3}
0.090	6.277×10^{-2}	1.294×10^{-3}
0.100	6.827×10^{-2}	9.193×10^{-4}
0.150	9.194×10^{-2}	2.086×10^{-3}
0.200	1.097×10^{-1}	6.543×10^{-3}
0.250	1.223×10^{-1}	1.193×10^{-2}
0.300	1.304×10^{-1}	1.780×10^{-2}

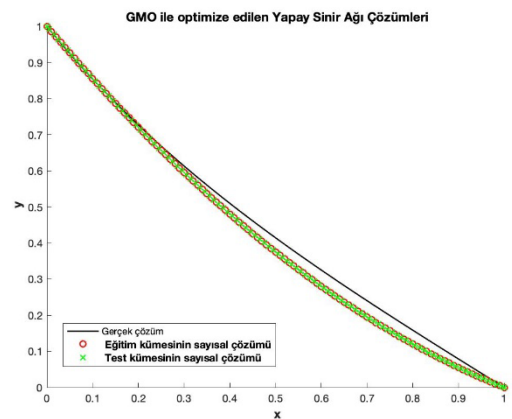
0.350	1.345×10^{-1}	2.378×10^{-2}
0.400	1.352×10^{-1}	2.953×10^{-2}
0.450	1.327×10^{-1}	3.476×10^{-2}
0.500	1.277×10^{-1}	3.921×10^{-2}
0.550	1.204×10^{-1}	4.267×10^{-2}
0.600	1.112×10^{-1}	4.492×10^{-2}
0.650	1.003×10^{-1}	4.578×10^{-2}
0.700	8.821×10^{-2}	4.508×10^{-2}
0.750	7.495×10^{-2}	4.266×10^{-2}
0.800	6.084×10^{-2}	3.838×10^{-2}
0.850	4.610×10^{-2}	3.211×10^{-2}
0.900	3.093×10^{-2}	2.370×10^{-2}
0.950	1.550×10^{-2}	1.304×10^{-2}
1.000	0.000	0.000

Çizelge 6: Örnek 2'nin test kümesindeki bazı noktalar için nümerik çözümünden elde edilen mutlak hata değerleri.

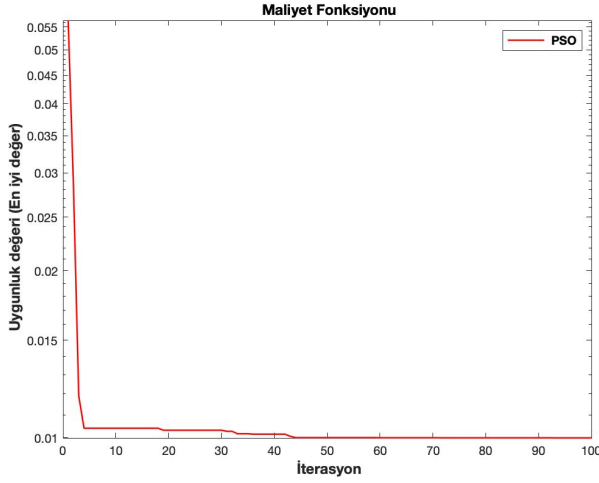
x	PSO Çözümü	GMO Çözümü
0.000	1.110×10^{-16}	1.110×10^{-16}
0.005	4.165×10^{-3}	3.358×10^{-4}
0.015	1.224×10^{-2}	9.012×10^{-4}
0.025	1.998×10^{-2}	1.331×10^{-3}
0.035	2.740×10^{-2}	1.633×10^{-3}
0.045	3.450×10^{-2}	1.813×10^{-3}
0.055	4.130×10^{-2}	1.878×10^{-3}
0.065	4.779×10^{-2}	1.835×10^{-3}
0.075	5.399×10^{-2}	1.690×10^{-3}
0.085	5.991×10^{-2}	1.449×10^{-3}
0.095	6.556×10^{-2}	1.117×10^{-4}
0.105	7.093×10^{-2}	7.007×10^{-3}
0.205	1.112×10^{-1}	7.047×10^{-3}
0.255	1.233×10^{-1}	1.250×10^{-2}
0.305	1.310×10^{-1}	1.840×10^{-2}
0.355	1.347×10^{-1}	2.437×10^{-2}
0.405	1.350×10^{-1}	3.008×10^{-2}
0.455	1.324×10^{-1}	3.524×10^{-2}
0.505	1.271×10^{-1}	3.961×10^{-2}
0.555	1.196×10^{-1}	4.295×10^{-2}
0.605	1.102×10^{-1}	4.507×10^{-2}
0.655	9.922×10^{-2}	4.578×10^{-2}
0.705	8.693×10^{-2}	4.492×10^{-2}
0.755	7.357×10^{-2}	4.232×10^{-2}
0.805	5.939×10^{-2}	3.785×10^{-2}
0.855	4.460×10^{-2}	3.137×10^{-2}
0.905	2.939×10^{-2}	2.274×10^{-2}
0.955	1.395×10^{-2}	1.185×10^{-2}
1.000	0.000	0.000



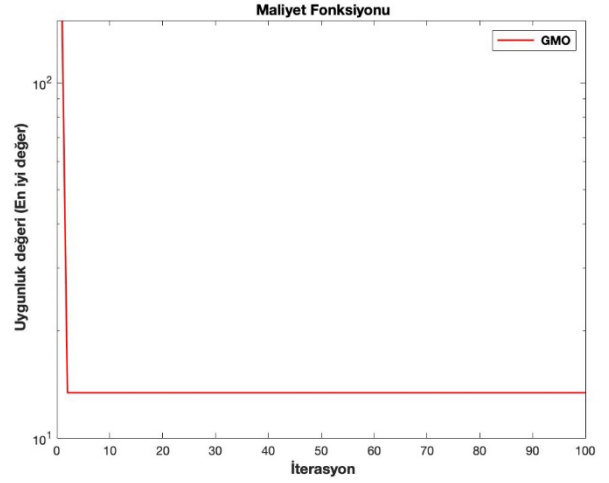
Şekil 5: Örnek 2 için PSO kullanılarak elde edilen yaklaşık çözüm grafiği



Şekil 6: Örnek 2 için GMO kullanılarak elde edilen yaklaşık çözüm grafiği



Şekil 7: Örnek 2 için PSO kullanılarak elde edilen iterasyona bağlı maliyet fonksiyonu grafiği



Şekil 8: Örnek 2 için GMO kullanılarak elde edilen iterasyona bağlı maliyet fonksiyonu grafiği

Çizelge 7: Örnek 1'in eğitim ve test kümesi için elde edilen ortalama karesel hata değerleri ve saniye cinsinden eğitim ve test süreleri için geçen süreler.

	PSO	GMO
Eğitim Kümesi için MSE	$9.321 \times 10^{-3} \pm 8.542 \times 10^{-5}$	$3.897 \times 10^{-3} \pm 0.000$
Test Kümesi için MSE	$9.230 \times 10^{-3} \pm 8.459 \times 10^{-5}$	$3.927 \times 10^{-3} \pm 0.000$
Modelin Eğitim Süresi	4.633± 2.221	$8.647 \times 10^1 \pm 2.846$
Modelin Test Süresi	$2.818 \times 10^{-4} \pm 2.385 \times 10^{-4}$	$7.941 \times 10^{-3} \pm 6.616 \times 10^{-4}$

4. SONUÇ

Bu çalışmada optimizasyon algoritmalarından GMO algoritması ile Dirichlet koşulu içeren diferansiyel denklemlerin nümerik çözümleri elde edilmiştir. Bu amaçla tek ara katmanlı ileri beslemeli yapay sinir ağı modeli kullanılarak ara katmanda 10 nöron öncelikle sınır koşullarını sağlayabilmesi için ön eğitimden geçirilmiştir. Eğitim aşamasında Levenberg Marquat metodu kullanılmıştır. Daha sonra GMO algoritması kullanılarak maliyet fonksiyonu minimize edilerek ağırlık ve eşik değerleri güncellenmiştir. GMO algoritmasının performansını incelemek amacıyla literatürde sıklıkla kullanılan PSO algoritması ile karşılaştırılmıştır.

Elde edilen nümerik çözümlere göre, GMO algoritmasının Dirichlet koşullu diferansiyel denklemlerin nümerik çözümü için alternatif bir optimizasyon algoritması olduğu sonucuna varılmıştır. Lineer ve lineer olmayan tipteki Dirichlet koşullu diferansiyel denklem için PSO algoritmasının zaman analizine göre GMO algoritmasından daha hızlı sonuç verdiği gözlemlenmiştir. Buna ek olarak GMO algoritmasının lineer olmayan tipteki denklem için analitik çözüme daha yakın bir çözüm elde edildiği görülmektedir.

GMO algoritmasının PSO'ya yakın sonuçlar vermesine ek olarak, literatürde var olan optimizasyon algoritmalarındaki stratejiler kullanılarak GMO algoritmasının performansının

daha da iyileştirilmesi sağlanabilir. Bu stratejilere örnek olarak Sobol, Halton, Hammersley ve Latin hiperküp gibi farklı dağıtım algoritmaları kullanarak ilk popülasyonu oluşturma işlemi ile rastgele fakat dengeli popülasyon oluşturulması verilebilir (Sirsant ve Reddy, 2022; Ahmad ve Rauf, 2021; Navarro vd., 2022). Popülasyon oluşturma kısmında yapılabilecek değişikliğe ek olarak anlık en iyi seçim (greedy selection) kullanılarak yerel olarak en iyi olan seçeneği tercih edilebilir ve bu seçimler sonucunda global olarak en iyi çözüme ulaşmak hedeflenebilir (Liu ve Qin, 2020). Ayrıca algoritmanın en iyi çözümlerini kaybetmeden ilerlemesi ve daha hızlı bir şekilde yakınsamaya ulaşması hedeflenen elitizm prensibi de GMO algoritması için uygulanabilir (Huang ve Li, 2020). Dinamik popülasyon yaklaşımı da diğer bir alternatif olarak düşünülebilir (Li vd., 2021). Kendinden öğrenme stratejisi (Self-learning strategy) ile algoritmanın kendi performansını iyileştirmek için önceki deneyimlerinden öğrenmesini ve bu deneyimlerden faydalanarak daha iyi sonuçlar elde etmesini sağlayan yöntem GMO için tercih edilebilir (Deng vd., 2024). Buna ek olarak, GMO algoritması için karşıtlık temelli öğrenme (Opposition-Based Learning) tekniği ile bir çözümün karşıt değerini hesaplayarak arama alanını daha etkin bir şekilde taramayı amaçlayan bir yaklaşım denenebilir (Huang ve Li, 2020; Chen vd., 2022; Alomoush vd., 2019). GMO algoritmasının keşif yeteneğini arttırmak için ise, bölgesel hakimiyet

politikası (Regional domination policy) uygulanabilir (Günel vd., 2018; Günel vd., 2020). Ayrıca Lévy uçuşu (Lévy flight) ile yerel optimal çözümlerden kaçabilmesini ve popülasyon arama alanını genişletebilmesi amacıyla GMO algoritması için iyileştirme düşünülebilir. Bahsedilen stratejiler ile sınırlı olmamakla beraber farklı yaklaşımlar tercih edilebilir.

Çalışmada kullanılan kodlara şu adresten erişilebilir: <https://github.com/iclalgor/GMO-Solutions-of-a-Class-of-DEs-with-DBCs/tree/main>

KAYNAKÇA

- Ahmad, J. ve Rauf, H. T. (2021). Comparison of Different Bat Initialization Techniques for Global Optimization Problems. *International Journal of Applied Metaheuristic Computing*, 12-1. doi: [10.4018/IJAMC.2021010109](https://doi.org/10.4018/IJAMC.2021010109)
- Alomoush A. A., Alsewari. A. A., Alamri, H.S., Zamli, K. Z., Alomoush, W. ve Younis. M. I. (2019). Modified Opposition Based Learning to Improve Harmony Search Variants Exploration. In: Saeed, F., Mohammed, F., Gazem, N. (eds) *Emerging Trends in Intelligent Computing and Informatics. IRICT 2019. Advances in Intelligent Systems and Computing*, 1073. Springer, Cham. doi: [10.1007/978-3-030-33582-3_27](https://doi.org/10.1007/978-3-030-33582-3_27)
- Chen, J., He, M. ve Huang, Y. (2020). A fast multiscale Galerkin method for solving second order linear Fredholm integro-differential equation with Dirichlet boundary conditions. *Journal of Computational and Applied Mathematics*, 364, 112352. doi: [10.1016/j.cam.2019.112352](https://doi.org/10.1016/j.cam.2019.112352)
- Chen, D., Liu, J., Yao, C., Zhang, Z. ve Du, X. (2022). Multi-strategy improved salp swarm algorithm and its application in reliability optimization, *Mathematical Biosciences and Engineering*. 19(5): 5269–5292. doi: [10.3934/mbe.2022247](https://doi.org/10.3934/mbe.2022247)
- Cheng, C. ve Zhang, G.T. (2021). Deep Learning Method Based on Physics Informed Neural Network with Resnet Block for Solving Fluid Flow Problems. *Water*, 13, 423. doi: [10.3390/w13040423](https://doi.org/10.3390/w13040423)
- Deng, X., He, D. ve Qu, L.. (2024). A Multi-strategy Enhanced Arithmetic Optimization Algorithm and Its Application in Path Planning of Mobile Robots. *Neural Processing Letters*, 56:18. doi: [10.1007/s11063-024-11467-6](https://doi.org/10.1007/s11063-024-11467-6)
- Fang, J., Liu, C., Simos, T. E., Famelis, I. T. (2020). Neural Network Solution of Single-Delay Differential Equations. *Mediterranean Journal of Mathematics*, 5-15. doi: [10.1007/s00009-019-1452-5](https://doi.org/10.1007/s00009-019-1452-5)
- Gör, I. (2020). *Diferansiyel Denklemlerin Yapay Sinir Ağları ile Nümerik Çözümleri*. Aydın Adnan Menderes Üniversitesi, Aydın.
- Graef, J. R., Heidarkhani, S., Kong, L. (2018), Existence of Solutions to An Impulsive Dirichlet Boundary Value Problem, *Fixed Point Theory*, 19, 1: 225-234. doi: [10.24193/fpt-ro.2018.1.18](https://doi.org/10.24193/fpt-ro.2018.1.18)
- Günel, K., Gör, İ. ve Tekeli, K. (2020). ICA-RD: The Regional Domination Policy for Imperialist Competitive Algorithm from Imperialism to Internationalism. *Arab Journal for Science and Engineering*, 45, 10529–10589. doi: [10.1007/s13369-020-04787-x](https://doi.org/10.1007/s13369-020-04787-x)
- Günel, K., Isman, G. ve Kocakula, M. (2018). Simple recurrent neural networks for the numerical solutions of ODEs with Dirichlet boundary conditions, *J. BAUN Inst. Sci. Technol.*, 20(3) Special Issue, 143-153. doi: [10.25092/baunfbcd.483922](https://doi.org/10.25092/baunfbcd.483922)
- Günel, K. ve Gör, I. (2022). Solving Dirichlet boundary problems for ODEs via swarm intelligence. *Mathematical Sciences*. 16:325–341. doi: [10.1007/s40096-021-00424-2](https://doi.org/10.1007/s40096-021-00424-2)
- Huang, Z. S. ve Li, W.-L. (2020). Novel Multi-Strategy Enhanced Whale Optimization Algorithm. 2nd IEEE Eurasia Conference on IOT, Communication and Engineering 2020. ISBN: 978-1-7281-8060-1. doi: [10.1109/ECICE50847.2020.9301990](https://doi.org/10.1109/ECICE50847.2020.9301990)
- Khan, I., Raja, M. A. Z., Shoaib, M., Kumam, P., Alrabaiah, H., Shah, Z. ve Islam, S. (2020). Design of Neural Network with Levenberg-Marquardt and Bayesian Regularization Backpropagation for Solving Pantograph Delay Differential Equations. *IEEE Access*. 8: 137918-137933 doi: [10.1109/ACCESS.2020.3011820](https://doi.org/10.1109/ACCESS.2020.3011820)
- Kowalski, P. (2013). Dirichlet boundary value problem for Duffing's equation. *Electronic Journal of Qualitative Theory of Differential Equations*. 37, 1-10. doi: [10.14232/ejqtde.2013.1.37](https://doi.org/10.14232/ejqtde.2013.1.37)
- Li, K., Fu, X., Wang, F. ve Jalil, H. (2021). Survey of Lévy Flight-Based Metaheuristics for Optimization. *Mathematics*. 10(15), 2785. doi: [10.3390/math10152785](https://doi.org/10.3390/math10152785)
- Li, K., An, Q., Deng, Q. ve Wang, G-G. (2022). A dynamic population reduction differential evolution algorithm combining linear and nonlinear strategy piecewise functions. *Concurrency Computat Pract Exper*. 34, 6773. doi: [10.1002/cpe.6773](https://doi.org/10.1002/cpe.6773)
- Liu, M. Peng, W., Hou, M. ve Tian, Z. (2023). Radial basis function neural network with extreme learning machine algorithm for solving ordinary differential equations. *Soft Computing*, 27:3955–3964. doi: [10.1007/s00500-022-07529-3](https://doi.org/10.1007/s00500-022-07529-3)
- Liu, X. ve Qin, X. (2020). A probability-based core dandelion guided dandelion algorithm and application to traffic flow prediction. *Engineering*

- Applications of Artificial Intelligence*, 96: 103922. doi: [10.1016/j.engappai.2020.103922](https://doi.org/10.1016/j.engappai.2020.103922)
- Malek, A. ve Beidokhti, R. S. (2006). Numerical solution for high order differential equations using a hybrid neural network Optimization method. *Applied Mathematics and Computation*. 183, 260-271. doi: [10.1016/j.amc.2006.05.068](https://doi.org/10.1016/j.amc.2006.05.068)
- Navarro, M. A., Oliva, D. Ramos-Michel, A., Morales-Castañeda, B., Zaldívar, D. ve Luque-Chang, A. (2022). A Review of the Use of Quasi-random Number Generators to Initialize the Population in Meta-heuristic Algorithms. *Archives of Computational Methods in Engineering*. 29:5149–5184. doi: [10.1007/s11831-022-09759-y](https://doi.org/10.1007/s11831-022-09759-y)
- Panghal, S. ve Kumar, M. (2022). Neural network method: delay and system of delay differential equations. *Engineering with Computers* 38 (Suppl 3):S2423–S2432. doi: [10.1007/s00366-021-01373-z](https://doi.org/10.1007/s00366-021-01373-z)
- Rezaei, F., Safavi, H. R., Elaziz, M. A. ve Mirjalili, S. (2023). GMO: geometric mean optimizer for solving engineering problems. *Soft Computing*. 27(15):10571–10606. doi: [10.1007/s00500-023-08202-z](https://doi.org/10.1007/s00500-023-08202-z)
- Sabir, Z., Wahab, H. A., Umar, M., Sakar, M. G. ve Raja, M. A. Z. (2020). Novel design of Morlet wavelet neural network for solving second order Lane–Emden equation. *Mathematics and Computers in Simulation*, 172, 1–14. doi: [10.1016/j.matcom.2020.01.005](https://doi.org/10.1016/j.matcom.2020.01.005)
- Sabir, Z., Umar, M., Guirao, J. L. G., Shoaib, M. ve Raja, M. A. Z (2021). Integrated intelligent computing paradigm for nonlinear multi- singular third-order Emden–Fowler equation. *Neural Computing and Applications*, 33:3417–3436. doi: [10.1007/s00521-020-05187-w](https://doi.org/10.1007/s00521-020-05187-w)
- Sheng, Y. Zhang, H. ve Zeng, Z. (2018). Stability and Robust Stability of Stochastic Reaction–Diffusion Neural Networks With Infinite Discrete and Distributed Delays. *IEEE Transactions On Systems, Man, and Cybernetics: Systems*, 50(5): 1721-1732. doi: [10.1109/TSMC.2017.2783905](https://doi.org/10.1109/TSMC.2017.2783905)
- Sirsant, S.ve Reddy, M. J. (2022). Improved MOSADE algorithm incorporating Sobol sequences for multi-objective design of Water Distribution Networks. *Applied Soft Computing*. 120. 108682. doi: [10.1016/j.asoc.2022.108682](https://doi.org/10.1016/j.asoc.2022.108682)
- Sousa, J.V.d.C., Nyamoradi, N. ve Lamine, M. (2022). Nehari manifold and fractional Dirichlet boundary value problem. *Anal.Math.Phys.* 12, 143. doi: [10.1007/s13324-022-00754-x](https://doi.org/10.1007/s13324-022-00754-x)
- Wu, M., Zhang, J., Huang, Z., Li, X. ve Dong, Y. (2021). Numerical solutions of wavelet neural networks for fractional differential equations. *Math Meth Appl Sci.* 2023;46:3031–3044. doi: [10.1002/mma.7449](https://doi.org/10.1002/mma.7449)
- Zhang, Z., Cai, Y. ve Zhang, D. (2020). Solving Ordinary Differential Equations with Adaptive Differential Evolution. *IEEE Access*, 8: 128908 - 128922. doi: [10.1109/ACCESS.2020.3008823](https://doi.org/10.1109/ACCESS.2020.3008823).
- Zadeh, L. A. (1965). Fuzzy sets. *Information and control*. 8(3):338–353. doi: [10.1016/S0019-9958\(65\)90241-X](https://doi.org/10.1016/S0019-9958(65)90241-X)