



Zaman Serisi Tahmin Probleminin İmmün Plazma Programlama Kullanılarak Çözülmesi

Sibel Arslan*

* Sivas Cumhuriyet Üniversitesi, Teknoloji Fakültesi, Sivas, Türkiye, (ORCID: 0000-0003-3626-553X), sibelarslan@cumhuriyet.edu.tr

(International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT) 2021 – 21-23 October 2021)

(DOI: 10.31590/ejosat.1010078)

ATIF/REFERENCE: Arslan, S., Zaman Serisi Tahmin Probleminin İmmün Plazma Programlama Kullanılarak Çözülmesi. (2021). *Avrupa Bilim ve Teknoloji Dergisi*, (29), 219-224.

Öz

Otomatik programlama, bir sistemde girdi ve çıktı değişkenleri arasındaki ilişkiyi model çıkararak açıklamaya çalışan bir makine öğrenmesi yaklaşımıdır. Zaman içerisinde sembolik regresyon, kümeleme, sınıflandırma, görüntü işleme, devre tasarımı, yol planlama, tahmin gibi birçok mühendislik problemlerini çözmeyi amaçlayan otomatik programlama yöntemleri geliştirilmiştir. Otomatik programlama yöntemlerinden birçoğu doğadan esinlenmektedir. Hızlı yayılan yeni Coronavirüs (COVID-19) salgınıyla mücadelede edilecek için farklı tedavi yöntemleri denenmektedir. İmmün plazma tedavisi, geçmişte birçok farklı salgında ve son olarak COVID-19'da etkili olduğu gösterilmiş tıbbi bir tedavi yöntemidir. İmmün plazma tedavi yaklaşımının uygulama aşamalarına dayanan İmmün Plazma Algoritması (Immune Plasma Algorithm, IPA) kısa süre önce önerilmiş bir meta-sezgisel algoritmadır. IPA, 2020 yılında tanıtılmış yeni bir algoritma olmasına rağmen farklı problemleri çözmek için çeşitli alanlarda uygulanmıştır. Bu çalışmada, IPA algoritmasını temel alan İmmün Plazma Programlama (Immune Plasma Programming, IPP) bir otomatik programlama yöntemi olarak tanıtılmıştır. IPP algoritmasının genel işleyişi IPA'nın aşamalarına benzerdir. Çözümlerin temsili ve iyileştirme mekanizması IPP'nin temel farklarıdır. IPA çözümleri sabit boyutlu diziler şeklinde ifade ederken, IPP çözümleri farklı derinliklere sahip olabilen parçalı ağaçlar olarak ifade eder. Ağaçların en küçük birimi düğümlerle temsil edilir. Düğümler, problemler için özel tanımlanan terminal kümesinden (x, y gibi değişkenler ve sabitler) ve fonksiyon kümesinden (aritmetik operatörler, mantıksal fonksiyonlar, matematiksel fonksiyonlar) seçilirler. Bu düğümlerin birleşimi ile çözümleri temsil eden ağaçlar oluşturulur. Çözümlerin iyileştirme mekanizması olarak Yapay Arı Koloni Programlama'da (Artificial Bee Colony Programming, ABCP) kullanılan bilgi paylaşım mekanizması IPP'ye uyarlanmıştır. Önerilen algoritmanın performansı, literatürde yaygın olarak kullanılan Box-Jenkins zaman serisi kullanılarak incelenmiştir. Çıkarılan modeller, en çok kullanılan otomatik programlama yöntemi ABCP ve Yapay Sinir Ağı modelleri ile kıyaslanmıştır. Sonuçlar, IPP'nin zaman serileri tahmin problemlerinde başarıyla kullanılabileceğini göstermiştir.

Anahtar Kelimeler: Otomatik Programlama Yöntemleri, İmmün Plazma Algoritması, İmmün Plazma Programlama, Zaman Serileri Tahmini, Plazma Tedavisi.

Solving the Problem of Time Series Prediction Using Immune Plasma Programming

Abstract

Automatic programming is a machine learning approach that attempts to explain the relationship between input and output variables in a system by extracting a model. Over time, automatic programming methods have been developed that aim to solve many engineering problems, such as symbolic regression, clustering, classification, image processing, circuit design, path planning, prediction. Most of the automatic programming methods are inspired by nature. To combat the rapidly spreading new coronavirus pandemic (COVID-19), various treatment methods are being tried. Immune plasma treatment is a medical treatment method that has proven to be effective in many different pandemics in the past and most recently in COVID-19. The Immune Plasma Algorithm (IPA) is a recently proposed meta-heuristic algorithm based on the implementation steps of the immune plasma treatment approach. Although IPA is a new algorithm, introduced in 2020, it has already been used in various fields to solve different problems. In this paper, the Immune Plasma Programming (IPP) automatic programming method based on IPA algorithm is presented for the first time. The general procedure of the IPP algorithm is similar to the stages of IPA. The representation of the solutions and the improvement mechanism are the main differences between the two algorithms. IPA expresses the solutions as fixed size arrays, while IPP represents the solutions as fragmented trees that can have different depths. The smallest unit of the trees is represented by nodes. The nodes are selected from a set of terminals (variables and constants such as x, y) and a set of functions (arithmetic operators, logical functions, mathematical functions)

* Sorumlu Yazar: xxxx@xxx.xx.xx

defined for the problems. Solution trees are created by combining these nodes. As the improvement mechanism of the solutions, the information sharing mechanism used in Artificial Bee Colony Programming (ABCP) was adapted to IPP. The performance of the algorithm was evaluated using the Box-Jenkins time series, which is widely used in the literature. The extracted models were compared with the most commonly used automatic programming method ABCP and Artificial Neural Network (ANN) models. The results showed that IPP can be successfully applied to time series prediction problems.

Keywords: Automatic Programming Methods, Immune Plasma Algorithm, Immune Plasma Programming, Time Series Prediction, Plasma Treatment.

1. Giriş

Otomatik programlama, bir görevi yerine getirmek için gerekli bilgileri kullanarak otomatik program kodunu üreten bir bilgisayar programlama tekniğidir (Biermann, 1985). Literatürde Genetik Programlama (Genetic Programming, GP) (Koza, 1992), Karınca Programlama (Ant Programming, AP) (Roux ve Fonlupt, 2000), ABCP (Karaboğa vd., 2012), Yapay Bağışıklık Sistemi Programlama (Artificial Immune System Programming, AISP) (Johnson, 2003), Biyocoğrafya Tabanlı Programlama (Biogeography Based Programming, BBP) (Golofshani, 2015) gibi birçok otomatik programlama yöntemi bulunmaktadır. Bu yöntemlerin sembolik regresyon, öznelik seçimi, sınıflandırma, rota bulma, zaman serileri tahmini gibi birçok karmaşık problemin çözümünde kullanıldığı ve oldukça iyi sonuçlar elde ettiği görülmektedir (Karaboğa vd., 2012; Arslan vd., 2019; Sotto vd. 2016; Cano ve Krawczyk, 2019; Chen vd.,2004). Bu problemlerden zaman serileri periyodik zaman aralıklarında gözlem değerlerinin ifade edildiği diziler olarak tanımlanmaktadır. Bir zaman serisinin belirli bir andaki gözlem değeri daha önceki gözlemlerin değerlerine bağlıdır. Günümüzde iktisat, işletme, mühendislik gibi birçok multidisipliner alanda zaman serileri analizi yapılmaktadır (Akdi, 2003).

Zaman seri modelleri üç farklı sınıfta kategorize edilir: doğrusal modeller, doğrusal olmayan modeller ve her iki modelin kombinasyonunu içeren hibrit modeller (Li ve Tanaka, 2021). Serilerin giriş ve çıkış gözlem değerleri dikkate alınarak bu değerler arasında en uygun ilişkinin fonksiyonel olarak belirlenmesi model çıkarımı için gereklidir. Özellikle doğrusal olmayan ve hibrit modellerde ilgili parametrelerin, değişkenlerin en iyi kombinasyonunun belirlenmesi zaman serileri tahmininde karşılaşılan temel bir problemdir. Bu çalışmada, bu problemle başa çıkabilmek için İmmün Plazma Algoritması (IPA)'nın otomatik programlama versiyonu İmmün Plazma Programlama (Immune Plasma Programming, IPP) tanıtılmıştır. IPP ile çıkarılan modeller ile sıkça kullanılan bir test problemi olan Box-Jenkins zaman serisi tahmin edilmiş ve tahmin sonuçları sıkça kullanılan Yapay Arı Koloni Programlama (Artificial Bee Colony Programming, ABCP) ve Yapay Sinir Ağları (Artificial Neural Network, ANN) yöntemleri ile kıyaslanmıştır.

Makalede IPP'nin tanıtılmasına Bölüm 2'de; problem tanımı, kullanılan parametreler, simülasyon sonuçlarının sunulması ve tartışılması hakkında ayrıntılı bilgiye Bölüm 3'de; sonuçlar ve gelecekteki çalışmalara Bölüm 4'de yer verilmiştir.

2. Materyal ve Metot

Bu bölümde, bu çalışmada önerilen İmmün Plazma Programlama yöntemi anlatılmaktadır. Bu yöntem İmmün Plazma Algoritması'nı (Immune Plasma Algorithm, IPA) temel aldığı için önce bu algoritmadan bahsedilmiş, ardından IPP detaylı olarak anlatılmıştır.

2.1. İmmün Plazma Algoritması (Immune Plasma Algorithm, IPA)

Yarasalardan insanlara bulaşan yeni koronavirüs hastalığı (COVID-19), ilk olarak Aralık 2019'da Çin'in Wuhan Eyaleti'nde görülmüştür (Casella vd., 2020). Virüsün 7 Ocak 2020'de tanımlanmasının ardından Dünya Sağlık Örgütü (World Health Organization, WHO) 11 Mart 2020'de COVID-19'u küresel bir salgın olarak ilan etmiştir (Huang vd., 2020). Bu salgınla mücadele edebilmek için farklı tedavi yöntemleri denenmektedir. İmmün plazma tedavisinde, daha önce iyileşen hastalardan alınan kanın bağışıklık cevabı üreten antikorlardan zengin kısmının hastaya nakledilerek hastanın iyileşmesi amaçlanır. Tedavinin iyileştirme başarısı H1N1 gribi, MERS, SARS, Ebola ve son olarak COVID-19 salgınlarında kanıtlanmıştır. Aslan ve Demirci, COVID-19 salgını ile birlikte uygulama sıklığı artan immün plazma tedavisi yaklaşımının temel özelliklerinden faydalanarak, IPA'yı tanıtmışlardır. Çalışmadaki deneylerde, IPA algoritması ile klasik, yüksek boyutlu test problemleri, karmaşık kıyaslama ve mühendislik problemleri çözülmüştür. Algoritma Genetik Algoritma (Genetic Algorithm, GA), Parçacık Sürü Optimizasyonu (Particle Swarm Optimization, PSO), Diferansiyel Gelişim (Differential Evolution, DE), Yapay Arı Kolonisi (Artificial Bee Colony, ABC) gibi birçok iyi bilinen meta sezgisel algoritma karşısında başarılı sonuçlar üretmiştir (Aslan ve Demirci, 2020). Bu çalışmada IPA'nın otomatik programlama versiyonu olarak IPP yöntemi önerilmiş ve zaman serisi tahmin probleminde bilinen mevcut yöntemlere göre performansı değerlendirilmiştir.

IPA'da, immün plazma tedavisinin uygulama aşamalarını bir meta-sezgisel algoritmaya uyarlanarak algoritmanın araştırma ve çözümleri geliştirme kabiliyetini arttırmak hedeflenmiştir. Algoritma tasarımında, enfeksiyonun bireyler arasında yayılması algoritmanın araştırma; plazma transferi çözümleri geliştirme aşamalarına benzetilmiştir. IPA, popülasyon temelli bir algoritmadır. Algoritmada popülasyondaki her bir birey problemin olası çözümünü, bireylerin ürettiği antikor miktarı çözümün kalitesini temsil eder. Bir bireyin ürettiği antikor miktarı ne kadar yüksekse çözümün kalitesi o kadar fazladır. Başka bir deyişle, amaç fonksiyonu o kadar iyi bir çözüm elde etmiştir (Aslan ve Demirci, 2020). IPA'da problemin olası çözümleri, karar değişkenleri için önerilen değerleri taşıyan ve araştırma boyunca boyutu sabit kalan dizilerle temsil edilir.

IPA'nın temel algoritmik yapısı ve aşamaları aşağıda verilmiştir.

Başlangıç aşaması

REPEAT

Enfeksiyonun yayılma fazı

Plazma transferi fazı

Donörlerin bağışıklık hafızasının kontrolü fazı

UNTIL (durdurma kriteri sağlanıncaya kadar)

Başlangıç aşaması: Bu aşamada her bir birey için başlangıç çözümleri belirlenir. D boyutlu bir problem için x_k çözümünün j 'nci boyut karar değişkeni için ataması Eşitlik (1) ile yapılır.

$$x_{kj} = x_j^{\min} + \text{rand}(0,1)(x_j^{\max} - x_j^{\min}) \quad (1)$$

Burada x_j^{\max} ve x_j^{\min} sırasıyla j 'inci boyutun alabileceği en büyük ve en küçük değerleri ve $\text{rand}(0,1)$ 0 ile 1 arasında normal dağılıma uygun olarak rastgele üretilmiş sayıyı göstermektedir. Oluşturulan bu çözümlerin bağışıklık cevabı/kalitesi amaç fonksiyonu ile değerlendirilmektedir.

Enfeksiyonun yayılma fazı: IPA'da, enfeksiyonun bireyler arasında yayılması algoritmanın araştırma alanını arttırmak için kullanılmıştır. Hasta bireyler, enfeksiyon etkenini doğrudan veya dolaylı yolla bulaştırabilir. Enfekte olmuş bireyin bağışıklık sistemi, o enfeksiyona karşı korumak için özel bir bağışıklık cevabı geliştirir. Popülasyonda enfeksiyonun bireyler arasında yayılması Eşitlik (2) ile gerçekleştirilir.

$$x_{kj}^{\text{inf}} = x_{kj} + \text{rand}(-1,1)(x_{kj} - x_{mj}) \quad (2)$$

x_k bireyinin enfeksiyon öncesinde j 'nci boyutunun değeri x_{kj} ile gösterilirken; x_{kj}^{inf} , enfekte olan x_k bireyinin j 'nci boyutunu ifade etmektedir. x_{mj} , x_k bireyi dışında popülasyondan rastgele seçilen x_m bireyinin j 'nci boyutunun değeridir. $\text{rand}(-1,1)$ ise -1 ile 1 aralığında rastgele üretilen bir sayıdır. Enfeksiyonun yayılmasından sonra, enfekte olmuş birey (x_k^{inf}) ile enfeksiyon öncesindeki birey (x_k) arasında bağışıklık cevapları dikkate alınarak aç gözlü seleksiyon uygulanmaktadır. Böylelikle bağışıklık cevabı yüksek olan birey popülasyonda korunmaktadır. (Aslan ve Demirci, 2021). Bu fazda, popülasyondaki her bir birey enfekte edilerek bağışık cevabının iyileştirilmesi amaçlanır.

Plazma transferi fazı: Enfeksiyonun yayılma fazından sonra IPA, bağışıklık cevabı en yüksek olan bireyleri donör, en kötü bireyleri alıcı olarak tayin eder. Algoritma başlangıcında, donör bireylerin sayısı (NoD , Number of Donors), alıcı bireylerin sayısı (NoR , Number of Receivers) sabit olarak belirlenir. Böylelikle bağışıklık cevabı en iyi toplam NoD bireyi donör, bağışıklık cevabı en kötü toplam NoR bireyi alıcı seçilmiş olunur. Bu fazda her bir alıcı bireye, toplam NoD donör arasından rasgele seçilen bir donör ile plazma transferi gerçekleştirilir. k indisi $\{1, \dots, NoR\}$ kümesinden sırayla seçilen bir alıcı olmak üzere, x_k^{rcv} alıcı bireyine NoD donör arasından rastgele seçilen x_m^{dnr} donörüne ait plazmanın transferi Eşitlik (3) ile ifade edilmiştir.

$$x_{kj}^{\text{rcv-p}} = x_{kj}^{\text{rcv}} + \text{rand}(-1,1)(x_{kj}^{\text{rcv}} - x_{mj}^{\text{dnr}}) \quad (3)$$

Burada, x_k^{rcv} bireyinin plazma transferi öncesi j 'inci boyutunun değeri x_{kj}^{rcv} , x_m^{dnr} donörünün j 'inci boyutunun değeri x_{mj}^{dnr} ile ifade edilmektedir. İlk doz plazma transferinden sonra, plazma transfer edilmiş birey ($x_k^{\text{rcv-p}}$) ile donörü (x_m^{dnr}) arasında açgözlü seçim uygulanır. Eğer $x_k^{\text{rcv-p}}$ 'nin bağışıklık cevabı, x_m^{dnr} 'den daha iyiyse transfer edilmemiş bireyin (x_k^{rcv}) yerine transfer edilmiş birey $x_k^{\text{rcv-p}}$ geçirilir ve bir doz daha plazma transferi gerçekleştirilir (Aslan ve Demirci, 2020). Aksi durumda bireye plazma transferi yapıldığı için x_k^{rcv} yerine donörü x_m^{dnr} geçirilir ve transfer sonlandırılır. İkinci, üçüncü... n'inci doz plazma transferi de Eşitlik (3) ile gerçekleştirilir. Ancak bu transferler sonrası, transfer tedavisinin sonlandırma kararı $x_k^{\text{rcv-p}}$ ile x_k^{rcv} arasında açgözlü seçim ile verilir. Eğer $x_k^{\text{rcv-p}}$ bireyinin bağışıklık cevabı x_k^{rcv} bireyinin bağışıklık cevabında daha kötü olursa plazma tedavisi tamamlanır. Diğer durumda x_k^{rcv} yerine $x_k^{\text{rcv-p}}$ bireyi geçirilir ve bir doz daha plazma transferi uygulanır.

Donörlerin bağışıklık hafızasının kontrolü fazı:

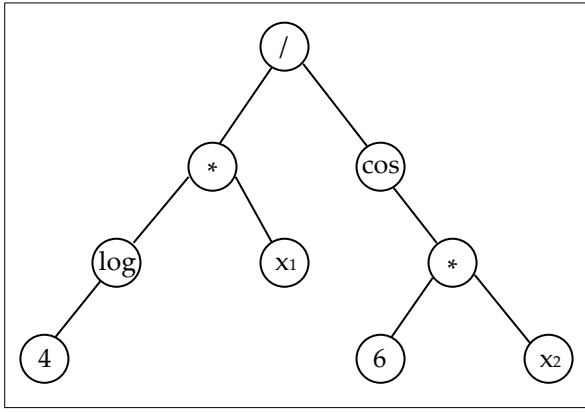
Popülasyonda donör olarak seçilen bireylerin bir enfeksiyona karşı verdiği bağışıklık cevabı zamanla değişebilir. Donörlerin bağışıklık cevabının zamana bağlı değişimi, değerlendirme sayısının (t_c) toplam değerlendirme sayısına (t_{max}) oranından faydalanılarak değerlendirilir. Eğer, toplam NoD donörün her bir x_m^{dnr} için belirlenen (t_c/t_{max}) değeri 0 ile 1 arasında rastgele belirlenen sayıdan küçük ise x_m^{dnr} bireyi tamamen değiştirilir. Diğer durumda ise Eşitlik (4) kullanılarak x_m^{dnr} bireyinin her j boyutu güncellenerek bağışıklık hafızası kontrol edilir (Aslan ve Demirci, 2020).

$$x_{mj}^{\text{dnr}} = x_{mj}^{\text{dnr}} + \text{rand}(-1,1)x_{mj}^{\text{dnr}} \quad (4)$$

2.2. İmmün Plazma Programlama (Immune Plasma Programming, IPP)

IPP, IPA algoritmasına dayanan yüksek seviyeli otomatik programlama metodu olarak önerilmiştir. IPP algoritması için temel aşamalar IPA'nın aşamalarına benzerdir ve aynı anolojiyi kullanmaktadır. İki algoritma arasında en temel farklılık bireylerin temsilinde IPA çözümleri sabit boyutlu diziler şeklinde ifade ederken, IPP çözümleri parçalı ağaçlar olarak ifade eder. IPP'de her parçalı ağaç, sistemin bağımsız değişkenleri ile bağımlı değişkeni arasında en uygun fonksiyonel ilişkiyi belirlemeye çalışır. Ağaçların en küçük birimine düğüm denir. Düğümler, problemler için özel olarak tanımlanmış terminal kümesi (x, y gibi değişkenler ve sabitler) ve fonksiyon kümesi (aritmetik operatörler, mantıksal ve matematiksel fonksiyonlar) arasından seçilir. Şekil 1 IPP için örnek bir çözümü göstermektedir. Çözümün değişkenler arasındaki fonksiyonel ilişkisi ise Eşitlik (5) ile verilmiştir. Bu gösterimlerde bağımsız değişkenler x_1, x_2 , bağımlı değişken $f(x_1, x_2)$ sembolleri ile temsil edilmiştir.

IPP algoritmasında, IPA'dan farklı olarak parçalı ağaçların üretilmesinde kullanılan "ramped half and half" metodu kullanılmıştır. "Ramped half and half" metodu, otomatik programlama yöntemlerinde çözümlerin üretilmesinde kullanılan yöntemlerdendir ve detaylarına (Koza, 1992)'dan ulaşılabilir. IPA'da çözümlerin gösterimi sabit boyutlu dizilerle gerçekleştirildiği için algoritmada enfeksiyonun yayılma, plazma transferi ve donörlerin bağışıklık hafızasının kontrolü fazlarında kullanılan eşitlikler doğrudan kullanılamamaktadır. IPP'de yapılan diğer temel farklılık, iyileştirme mekanizması olarak adlandırılan ve bu fazlardaki eşitlikler yerine kullanılan mekanizma üzerindedir. İyileştirme mekanizması, ABCP'de kullanılan bilgi paylaşım mekanizmasının (Karaboğa vd., 2012) uyarlamasıdır. Bu mekanizma Şekil 2'de gösterilmiştir. Şekil 2a ve 2b'de sırasıyla alıcı (x_k^{rcv}) ve donör (x_m^{dnr}) gösterilmiştir, donörden alınan bilgi (c)'de ve plazma transfer edilmiş birey aday çözüm ($x_k^{\text{rcv-p}}$)'de verilmiştir. Algoritmada, Eşitlik (2), (3) ve (4) yerine eşitliklerdeki ilgili bireyler iyileştirme mekanizmasında kullanılarak fazlar gerçekleştirilmiştir.



$$f(x_1, x_2) = \frac{\log(4)x_1}{\cos(6x_2)} \quad (5)$$

Şekil 1. IPP'de çözümlerin ağaç yapısı ile temsili

Otomatik programlama yöntemlerinde çözümler değerlendirildiğinde, bazı durumlarda ağaçlarda karşılaşılabilecek sifra bölme hatası veya ağaçtaki bir değer veri tipinin alabileceği maksimum değerden yüksek bir değer alma hatası gibi bir durumda Koza (Koza, 1992) ilgili ağaca uygun bir sayısal değer atanarak algoritmanın çalışmasına devam edebileceğini belirtmiştir. Bu çalışmada söz konusu hatalarla karşılaşıldığında alt ağaçlara uygun sayısal değerler atanmıştır.

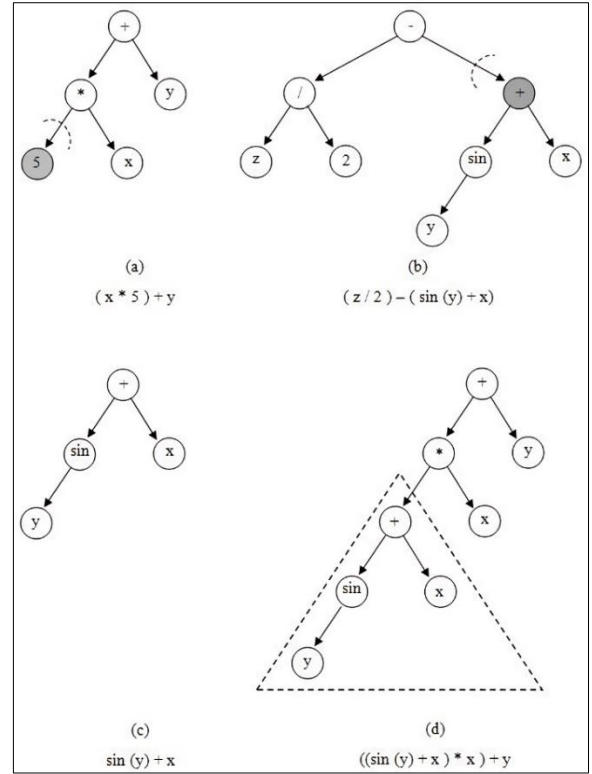
3. Araştırma Sonuçları ve Tartışma

Zaman serileri, bağımlı değişkenlerin kronolojik sırayla ifade edildiği değişkenlerdir. Bu çalışmada gerçek hayat problemlerinden Box-Jenkins zaman serisi tahmin problemi üzerinde IPP yönteminin performansı değerlendirilmiştir.

3.1. Problem Tanımı ve Deneylerde Kullanılan Parametre Değerleri

Box-Jenkins zaman serisi, literatürde test amaçlı kullanılan ve metan-hava karışımının yanma işlemi doğalgaz ile çalışan ısıtma sisteminin verilerinden oluşan 290 örneklilik bir veri setidir (Box ve Jenkins, 1970). Veri setinde girişler $u(t)$ ve $y(t-1)$; çıkış $y(t)$ ile ifade edilmiştir. IPP ile girişler ve çıkış arasında fonksiyonel ilişkiyi belirleyip en az hata ile çıkış tahmin edilmeye çalışılmıştır. Veri setindeki 290 örnekten 200'ü eğitim, 90'ı test için kullanılmıştır. Önerilen yöntemin performansı (Görkemli vd., 2012) ile değerlendirildiğinden, adil bir karşılaştırma olması için (Görkemli vd., 2012)'daki aynı parametre değerleri kullanılmış ve bu değerler Tablo 1'de verilmiştir.

Koşmalarda kullanılan *sqr* fonksiyonu, bir sayının karesini ve bölme fonksiyonu, korumalı fonksiyonunu temsil eder. Korumalı



Şekil 2. IPP iyileştirme mekanizması

fonksiyonda bölen değeri 0'a eşitse, sonuca 1 değeri atanır, aksi takdirde normal bölme gerçekleştirilir. Bu problem için farklı donör sayıları ve alıcı sayıları ile yapılan deneylerde en başarılı donör/alıcı sayısı 1 olarak belirlenmiştir. Çalışma 30 kez bağımsız koşurulmuştur. Koşmalarda (Görkemli vd., 2012)'da kullanılan amaç fonksiyonu Ortalama Hatanın Karesi (Mean Square Error, MSE) Eşitlik (6)'da verilmiştir. Burada y_{actual} tahmin edilen çıkışın gerçek değerini, y_{pred} çözümün ürettiği tahmin değerini temsil etmektedir.

$$MSE = \frac{1}{N} \sum_{j=1}^N (y_{actual} - y_{pred})^2 \quad (6)$$

3.2. Simülasyon Sonuçları ve Tartışmalar

Koşmalarda elde edilen modellerin ortalaması, standart sapması (Görkemli vd., 2012)'daki sonuçlar ile birlikte Tablo 2'de sunulmuştur. Tablo 2'deki sonuçlar değerlendirildiğinde IPP'nin eğitim verisinde ABCP ve YSA modelinden daha iyi olduğu görülmektedir. Bununla birlikte testte ABCP yöntemi en başarılı modeli çıkarmıştır. IPP için en iyi sonucu veren çözümün yakınsama grafikleri Şekil 3 ve Şekil 4'de sunulmuştur.

Şekil 3 ve Şekil 4'deki grafikler incelendiğinde hem eğitim ve hem test verisinde gerçek ve tahmini serinin birbiriyle oldukça uyumlu olduğu görülmektedir. 30 koşma sonucunda elde edilen en iyi modelin denklemi Eşitlik (7)'de verilmiştir. Karmaşık bir zaman serisi eğrisi için Eşitlik (7) analiz edildiğinde, ikinci dereceden nonlineer bir model olduğu ve özellikle eğitim verisi tahmininde oldukça başarılı olduğu değerlendirilmiştir.

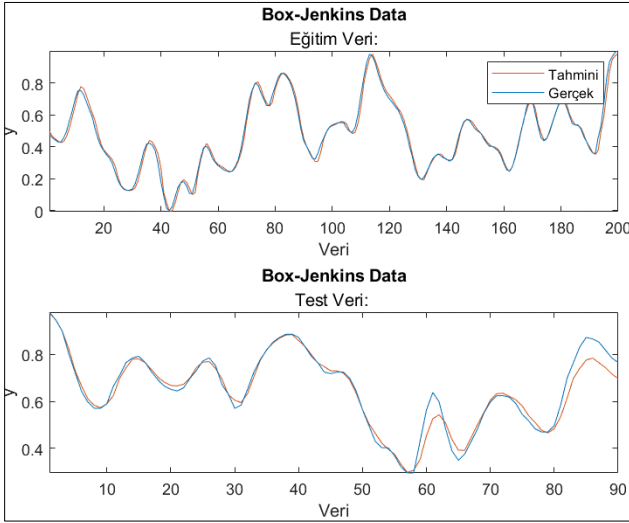
$$(2y(t-1) - u(t))(u(t) - u(t)^2 + 4.857)0.06998 + 0.3229 \quad (7)$$

Tablo 1. Parametre Değerleri

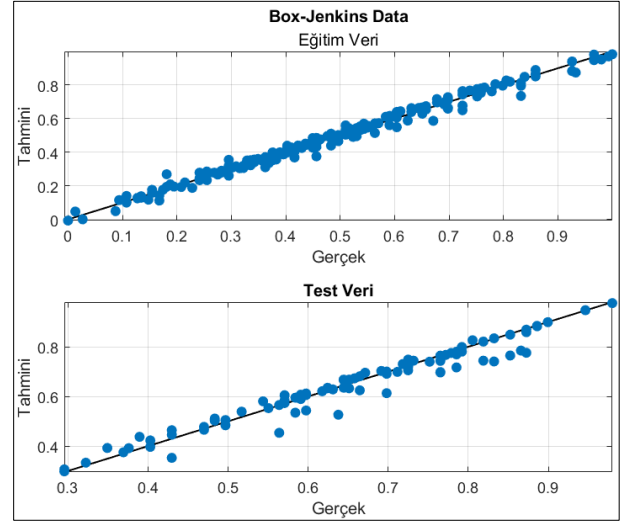
IPP		ABCP	
Parametre	Değer	Parametre	Değer
Popülasyon büyüklüğü	300	Koloni büyüklüğü	300
Değerlendirme sayısı	10,000	Değerlendirme sayısı	10,000
Maksimum ağaç derinliği	4	Maksimum ağaç derinliği	4
Fonksiyon kümesi	+, -, *, /, sqr	Fonksiyon kümesi	+, -, *, /, sqr
Donör Sayısı (NoD)	1	Limit	300
Alıcı Sayısı (NoR)	1		

Tablo 2. IPP'nin ABCP ve Yapay Sinir Ağ Modeli ile kıyaslanması

	IPP		ABCP (Görkemli vd., 2012)		Yapay Sinir Ağ Modeli (Görkemli vd., 2012)	
	Eğitim	Test	Eğitim	Test	Eğitim	Test
Ortalama	0,000391	0,002755	0,000576	0,000022	0,006626	0,010572
Standart Sapma	0,000004	0,009493	0,002658	0,000360	0,000029	0,000711
En İyi	0,000383	0,002667	-		-	



Şekil 3. Box-Jenkins zaman serisinde IPP en iyi modelinin yakınsama grafiği



Şekil 4. Box-Jenkins zaman serisinde IPP en iyi modelinin tahmini ve gerçek veri noktaları arasındaki ilişki

4. Sonuç

Bu çalışmada İmmün Plazma Algoritması'nı temel alan yeni bir otomatik programlama yöntemi İmmün Plazma Programlama tanıtılmıştır. Yöntemin performansı, Box-Jenkins zaman serisi tahmin problemi üzerinde incelenmiştir. Deneyler, IPP ile çıkarılan modellerin tahmini çıkış değerlerinin gerçek değerlere oldukça yakın olduğunu göstermektedir. IPA'nın literatürde yaygın olarak kullanılan meta-sezgisel algoritmalara kıyasla, farklı türlerde optimizasyon problemleri için daha iyi çözümler elde etme yeteneğine sahip olduğu değerlendirilmiştir (Aslan ve Demirci, 2020). Bu yüzden, IPP'nin mevcut yöntemlere göre IPA gibi başarılı sonuçlar elde edeceği düşünülmekte ve IPP'nin literatürde yer alan birçok problemde kullanılabileceği düşünülmektedir. Gelecekteki çalışmalarda, farklı disiplinlerde problemler çözmek için IPP'nin kullanılması ve yeni düzenlemeler getirilerek IPP'nin performansının iyileştirilmesi üzerine çalışma yapılması planlanmaktadır.

5. Teşekkür

Bu çalışmanın yürütülmesinde destek veren kıymetli meslektaşlarım Dr. Öğretim Üyesi Fırat İsmailoğlu'na ve Begüm Yetişkin'e saygılarımı sunar, teşekkür ederim.

Kaynakça

- Akdi, Y. (2003). Zaman Serileri Analizi.
- Arslan, S., & Öztürk, C., (2019). Multi Hive Artificial Bee Colony Programming for high dimensional symbolic regression with feature selection. APPLIED SOFT COMPUTING, vol.78, 515-527.
- Aslan, S., & Demirci, S. (2020). Immune Plasma Algorithm: A Novel Meta-Heuristic for Optimization Problems. IEEE Access, 8, 220227-220245. <https://doi.org/10.1109/access.2020.3043174>
- Aslan, S., & Demirci, S. (2021). Performance Investigation of Parallel Immune Plasma Algorithm. İçinde 2021 International Conference on INnovations in Intelligent SysTems and Applications (INISTA). 2021 International Conference on INnovations in Intelligent SysTems and

- Applications (INISTA). IEEE.
<https://doi.org/10.1109/inista52262.2021.9548547>
- Biermann, A. W. (1985). Automatic programming: A tutorial on formal methodologies. *Journal of Symbolic Computation*, 1(2), 119-142. [https://doi.org/10.1016/s0747-7171\(85\)80010-9](https://doi.org/10.1016/s0747-7171(85)80010-9)
- Box, G., Jenkins, G. (1970). *Time Series Analysis: Forecasting and Control*. Holden Day, San Francisco.
- Cano, A., & Krawczyk, B. (2019). Evolving rule-based classifiers with genetic programming on GPUs for drifting data streams. *Pattern Recognition*, 87, 248-268. <https://doi.org/10.1016/j.patcog.2018.10.024>
- Cascella M., Rajnik M., Cuomo A., Dulebohn, S. C., & Napoli R. D. (2020). Features, evaluation and treatment coronavirus (COVID-19). *StatPearls* [Internet], Stat Pearls Publishing.
- Chen, Y., Yang, B., & Dong, J. (2004). *Evolving Flexible Neural Networks Using Ant Programming and PSO Algorithm*. İçinde *Advances in Neural Networks – ISNN 2004* (ss. 211-216). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-28647-9_36
- Golafshani, E. M. (2015). Introduction of Biogeography-Based Programming as a new algorithm for solving problems. *Applied Mathematics and Computation*, 270, 1-12. <https://doi.org/10.1016/j.amc.2015.08.026>
- Görkemli, B., Öztürk, C., & Karaboğa, D. (2012). Yapay Arı Kolonisi Programlama ile Sistem Modelleme. *Otomatik Kontrol Türk Milli Komitesi 2012 Ulusal Toplantısı (TOK2012)*. 857-860. Niğde, Turkey.
- Huang, C., Wang, Y., Li, X., Ren, L., Zhao, J., Hu, Y., Zhang, L., Fan, G., Xu, J., Gu, X., Cheng, Z., Yu, T., Xia, J., Wei, Y., Wu, W., Xie, X., Yin, W., Li, H., Liu, M., ... Cao, B. (2020). Clinical features of patients infected with 2019 novel coronavirus in Wuhan, China. *The Lancet*, 395(10223), 497-506. [https://doi.org/10.1016/s0140-6736\(20\)30183-5](https://doi.org/10.1016/s0140-6736(20)30183-5)
- Johnson, C. G. (2003). *Artificial Immune System Programming for Symbolic Regression*. *Lecture Notes in Computer Science*, 345-353. Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-36599-0_32
- Karaboğa, D., Öztürk, C., Karaboğa, N., & Görkemli, B. (2012). Artificial bee colony programming for symbolic regression, *Information Sciences*, 209, 1–15.
- Koza, J.R. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, Cambridge: MIT Press.
- Li, Z. & Tanaka, G. (2021). Multi-Reservoir Echo State Networks with Sequence Resampling for Nonlinear Time-Series Prediction. *Neurocomputing*.
- Roux, O., Fonlupt, C. (2000). Ant programming: or how to use ants for automatic programming, in *2nd International Workshop on Ant Algorithms (ANTS'2000)*, 121–129. Brussels, Belgium.
- Sotto, L. F. D. P., de Melo, V. V., & Basgalupp, M. P. (2016). An improved λ -linear genetic programming evaluated in solving the Santa Fe ant trail problem. *Proceedings of the 31st Annual ACM Symposium on Applied Computing. SAC 2016: Symposium on Applied Computing*. <https://doi.org/10.1145/2851613.2851669>