



Düzce Üniversitesi Bilim ve Teknoloji Dergisi

Araştırma Makalesi

Türkiye Özelinde Ekonomik Yük Dağıtım Problemi için Sürüngen Arama Algoritmasının Geliştirilmiş Versiyonu

 Ayşe ERDOĞAN YILDIRIM^{a,*}

^a Bilgisayar Mühendisliği Bölümü, Mühendislik Fakültesi, Fırat Üniversitesi, Elazığ, TÜRKİYE

* Sorumlu yazarın e-posta adresi: ayseerdogan@firat.edu.tr

DOI: 10.29130/dubited.1185476

Öz

Sürüngen Arama Algoritması, sürüngenlerin avlanma kalıplarının modellenmesiyle geliştirilmiş yeni nesil sürü tabanlı bir optimizasyon yöntemidir. Algoritma, özellikle timsah türleri için tasarlanmıştır. Algoritmada timsahların yüksekte yürüme, göbekte yürüme, avlanma koordinasyonu, avlanma işbirliği gibi davranışları farklı operatörler olarak modellenmiştir. Sürüngen Arama Algoritması daha önce kıyaslama fonksiyonlarına ve gerçek dünya mühendislik tasarım problemlerine uygulanmıştır. Bu alanlarda, bu yeni nesil buluşsal yöntem başarılı sonuçlar elde etmiştir. Termik santrallerde enerji üretim maliyetlerinin düşürülmesi çok önemli bir konudur. Enerji verimliliğini sağlamak için bu alanda birçok optimizasyon yöntemi kullanılmıştır. Bu makalede, Türkiye'deki güç sistemleri için ekonomik yük dağıtım problemine Sürüngen Arama Algoritması ve algoritmanın geliştirilmiş versiyonu ayrı ayrı uygulanarak bu yaklaşımların başarısı değerlendirilmektedir. Bu çalışma, Sürüngen Arama Algoritmasının performansının yapılan güncellemelerle geliştirildiğini ve algoritmanın yeni versiyonunun termik santrallerde yakıt maliyet optimizasyonu alanında daha etkili bir yöntem olduğunu ortaya koymaktadır.

Anahtar Kelimeler: Sürüngen arama algoritması, Ekonomik yük dağıtım problemi, Güç sistemleri, Optimizasyon, Meta-sezgisel optimizasyon algoritmaları

Improved Version of Reptile Search Algorithm for Economic Load Dispatch Problem in Turkey

ABSTRACT

Reptile Search Algorithm is a new generation swarm-based optimization method developed by modeling the hunting patterns of reptiles. The algorithm is designed specifically for the crocodile species. Behaviors of crocodiles such as high walking, belly walking, hunting coordination, hunting cooperation are modeled as different operators in the algorithm. Reptile Search Algorithm has been previously applied to benchmark functions and real world engineering design problems. In these fields, this new generation heuristic method achieved successful results. Reducing energy production costs in thermal power plants is a very important issue. Many optimization methods have been used in this area to provide energy efficiency. In this article, the success of the approaches are evaluated by applying the Reptile Search Algorithm and the improved version of the algorithm separately to the economic load dispatch problem for power systems in Turkey. This study reveals that the performance of the Reptile Search Algorithm has been improved with the updates and the new version of the algorithm is a more effective method in the field of fuel cost optimization in thermal power plants.

Keywords: Reptile search algorithm, Economic load dispatch problem, Power systems, Optimization, Meta-heuristic optimization algorithms

I. GİRİŞ

Elektrik, her alanda yaygın kullanımıyla günümüzün vazgeçilmez ihtiyaçlarından biridir. Bu nedenle, çağımız için elektrik enerjisi üretimi büyük öneme sahip bir konudur. Elektrik enerjisi üretimini en ekonomik şekilde gerçekleştirmek, hem üreticisi hem de tüketicisi açısından faydalıdır. Ekonomik Yük Dağıtım (EYD) problemi tam bu noktada karşımıza çıkmaktadır. EYD, elektrik enerjisi üretimi yapılan termik güç santrallerinde, üretim sürecinde kullanılan biyo-yakıt miktarında tasarruf sağlayarak elektrik üretim maliyetini düşürmek amacıyla ortaya konmuş bir optimizasyon problemidir. Literatürdeki birçok çalışma, matematiksel ya da meta-sezgisel yöntemler kullanarak bu probleme çözüm aramıştır. Son yıllarda, sağladığı avantajlar nedeniyle meta-sezgisel yöntemlerin kullanımı daha çok tercih edilmektedir.

Yalçınöz vd. [1] yapmış oldukları çalışmada, 6 generatörlü sistem için EYD problemini Tabu Araştırması yöntemiyle çözümlenerek, FLCGA (Bulanık mantık tabanlı genetik algoritma), AECGA (Gelişmiş mühendislik koşullandırılmış genetik algoritma), IHNN (Geliştirilmiş Hopfield Sinir Ağı) ve AHNN (Aiyer'in Hopfield Sinir Ağı) yöntemleriyle karşılaştırmıştır. Kurban ve Başaran [2] çalışmalarıyla Türkiye'de bulunan 14 baralı 6 termik santralli güç sisteminin gerekliliklerini belirleyerek EYD problemini Lagrange yöntemini kullanarak çözmüşlerdir. Duman ve Öztürk [3] gerçekleştirdikleri çalışmayla, Türkiye'deki 6-termik santral için elektrik enerjisi üretiminin minimum yakıt tüketimi ile gerçekleştirilmesi amacıyla, Tavlama Benzetimi, Tabu Arama Algoritması ve Genetik Algoritmayı kullanmışlardır. Rahmani vd. [4] çalışmalarında, evrimsel olarak değiştirilmiş Parçacık Sürü Optimizasyonuna dayalı olarak geliştirdikleri yöntemi kullanarak EYD problemini 3-, 6-, 15- ve 40-üniteli test sistemleri için çözümlenmişlerdir. Bindu ve Reddy [5] 3- ve 6- üniteli test sistemleri için yakıt maliyet optimizasyonu problemine Guguk Kuşu Arama Algoritmasını uygulamışlardır. Banerjee vd. [6] EYD probleminin çözümünde Öğretme-Öğrenme Tabanlı Optimizasyon yöntemini valf noktası yükleme etkisini dikkate aldıkları 3-, 13- ve 40-üniteli sistemler için kullanmışlardır. Turgut ve Demir [7], Yapay İşbirlikçi Algoritmasını kullanarak 13 ve 40 üniteli test sistemleri için EYD problemine çözüm aramışlardır. Akkaş vd. [8] yaptıkları çalışmayla VIKOR yöntemini kullanarak 6 generatörlü sistem için EYD problemine çözüm getirmeye çalışmışlardır. Ali ve Abd Elazim [9] çalışmalarında, EYD problemini 6 generatörlü test sistemi için ve çift amaçlı kombine ekonomik emisyon dağıtım (KEED) problemini 10 generatörlü test sistemi için Mayın Patlatma Algoritması kullanarak çözmeye çalışmışlardır. Reddy ve Reddy [10] yaptıkları çalışmada, 6-üniteli test sisteminde EYD probleminin çözümü için, valf noktası yükleme etkisini dikkate alan, dikkate almayan, emisyon minimizasyonu yapan ve yasak işletim bölgeleri ile rampa hız sınırını dikkate alan üzere 4 farklı sistem için Karınca Aslanı Optimizasyon Algoritmasını uygulamışlar ve önerdikleri yöntemi literatürdeki diğer meta-sezgisel algoritmalar ile karşılaştırarak, üstünlüğünü ortaya koymuşlardır. Malik vd. [11] çalışmalarında, 3-, 6- ve 15-generatörlü test sistemleri için EYD probleminin çözümünü Modifiye edilmiş Balina Optimizasyon Algoritmasını kullanarak gerçekleştirmişlerdir. Fu vd. [12] yaptıkları çalışmayla, 6- ve 15- generatörlü test sistemleri için Geliştirilmiş Kuş Sürü Algoritması ile yakıt maliyeti optimizasyon problemini çözmüşlerdir. Andıç vd. [13] ise çalışmalarıyla, Türkiye'deki güç sistemleri için EYD problemini meta-sezgisel bir algoritma olan Karga Arama Algoritmasıyla çözmeyi önermişlerdir. Önceki çalışmalara kıyasla daha iyi bir performans elde etmişlerdir. Deb vd. [14] çalışmalarında, 6-üniteli test sistemi için hem EYD hem de KEED problemlerine meta-sezgisel bir algoritma olan Türbülanslı Su Akışı Optimizasyonu ile çözüm aramışlardır. Elde ettiği sonuçları, Guguk Kuşu Arama Algoritması, Gri Kurt Algoritması, Sinüs Kosinüs Algoritması, Toprak Solucanı Optimizasyon Algoritması, Tulumlu Sürü Algoritması, Güve Arama Algoritması ve Öğretme-Öğrenme Tabanlı Optimizasyon yöntemleri ile karşılaştırarak üstünlüğünü ortaya koymuşlardır. Al-Betar vd. [15] çalışmalarında, 3-, 13- ve 40-üniteli test sistemleri için rampa hızı limitleri ve yasaklanmış çalışma bölgesi kısıtlamaları olmaksızın geliştirdikleri Hibrit Sinüs-Kosinüs yöntemini uygularken, 6- ve 15-üniteli sistemler için bu kısıtlamaları dikkate almışlardır. Aribowo [16] çalışmasında, 3- ve 6-üniteli sistemler için Martı Optimizasyon Algoritması, Deniz Yırtıcı Algoritması, Sinüs Ağaç-Tohum Algoritması, Şempanze Optimizasyon Algoritması, Denge Optimize Edici ve Giza Piramitleri İnşaatı olmak üzere 6 farklı meta-sezgisel yöntemi EYD probleminin çözümü için uygulayarak karşılaştırmalı sonuçları sunmuşlardır. Said vd. [17], EYD ve

KEED problemlerinin çözümü için Arama ve Kurtarma Optimizasyon Algoritmasını kullanarak 6- üniteli ve 10- üniteli test sistemlerinin maliyet optimizasyonunu gerçekleştirmişlerdir. Solucan Optimizasyon Algoritması, Gri Kurt Optimizasyonu, Sinüs-Cosinüs Optimizasyonu gibi çeşitli meta-sezgisel yöntemlerde karşılaştırma yaparak önerilen yöntemin üstünlüğünü göstermişlerdir.

Önerilen çalışmada, daha önce Lagrange yöntemi [2], Genetik Algoritma [18], Tavlama Benzetimi, Tabu Arama Algoritması [3] ve Karga Arama Algoritması [13] ile çözüm aranan Türkiye’de bulunan 14-baralı ve 6- termik santral içeren güç sisteminin gereksinimlerine uygun olarak tasarlanmış EYD problemine, Sürüngen Arama Algoritmasının (SAA) ve geliştirilmiş versiyonu ile çözüm aranmıştır. Önerilen yöntem, önceki çalışmalarda elde edilen sonuçlarla karşılaştırılmıştır. Deneysel sonuçlar Geliştirilmiş Sürüngen Arama Algoritmasının (GSAA) EYD probleminin çözümü için uygun bir yöntem olduğunu ortaya koymuştur.

Çalışmanın anlatımı aşağıdaki şekilde planlanmıştır: 2.Bölüm EYD probleminin anlatımı ve formüllerle tanıtımını içermektedir. Ayrıca SAA ve önerilen geliştirilmiş versiyonu bu bölümde açıklanmıştır. 3. Bölümde GSAA kullanılarak 14 baralı 6 termik santrali bulunan Türkiye’deki güç sistemleri özelinde gerçekleştirilen uygulamanın sonuçları verilmiş ve karşılaştırmalı olarak değerlendirilmiştir. Sonuç bölümünde, geliştirilen yöntemin ilgili problemdeki başarısı tartışılmıştır.

II. MATERİYAL VE METOD

A. EKONOMİK YÜK DAĞITIM PROBLEMİ

Ekonomik Yük Dağıtım (EYD) problemi, güç sistemlerinde elektrik üretim sürecinde, farklı termik santraller arasında yükün optimum şekilde dağıtımını ile kullanılan biyo-yakıt miktarını düşürmeyi ve böylece üretim maliyetini azaltmayı amaçlayan bir çeşit optimizasyon problemidir. Termik santraller arasındaki yük dağıtım maliyeti, generatörlerin çıkış gücüne bağlı denklemlerle hesaplanmaktadır. Yük dağıtım maliyetini düşürmek için termik santrallerin generatör çıkış güçlerini optimize etmek gerekmektedir. Bu doğrusal olmayan bir optimizasyon problemi olarak karşımıza çıkmaktadır. Termik santrallerin yakıt maliyetleri Denklem 1’deki formülasyon ile hesaplanmaktadır.

$$F_i = c_i + b_i P_i + a_i P_i^2 \quad (1)$$

Burada, P_i her bir generatörün taşıdığı yükü gösterirken a_i , b_i ve c_i değişkenleri yük dağıtım sabitlerini göstermektedir [19-21].

Güç sistemi içerisinde çalışan tüm termik santrallerdeki toplam yakıt maliyeti ise, Denklem 2 kullanılarak hesaplanmaktadır.

$$F_T = F_1 + F_2 + \dots + F_K \quad (2)$$

Toplam yakıt maliyetinin minimizasyonu amacıyla kullanılan yük sabitlerini içeren EYD probleminin amaç fonksiyonu Denklem 3’de verilmektedir.

$$C = \text{Min} \sum_{i=1}^N F_i(P_{Gi}) = \text{Min} \sum_{i=1}^N c_i + b_i P_{Gi} + a_i P_{Gi}^2 \quad (3)$$

Denklem 1 ve Denklem 3, valf noktası yükleme etkisini dikkate almayan sistemler için geçerlidir. Ayrıca, bu çalışmada, sistem içerisinde meydana gelen güç kayıpları da göz ardı edilmektedir. Sistem içerisindeki her termik santralin yük kapasitesi birbirinden farklıdır. Bu sınırlamalar aşağıdaki eşitlik ile belirlenir (Denklem 4).

$$P_i^{\min} \leq P_i \leq P_i^{\max} \quad (4)$$

Burada P_i^{min} ve P_i^{max} sırasıyla generatörlerin yüklenebileceği alt sınır ve üst sınırları göstermektedir [1].

B. SÜRÜNGEN ARAMA ALGORİTMASI

Sürüngen Arama Algoritması (SAA) [22], sürüngenler arasında özellikle timsah türü için modellenmiş bir algoritmadır. Timsahların avlarını yakalamak için gerçekleştirdikleri hareket ve davranışları modelleyerek geliştirilmiştir. Timsahların avlanma stratejileri iki temel adımdan oluşmaktadır. Bunlar; Keşif (Çevreleme) ve Sömürü (Avlanma) aşamalarıdır. Çevreleme aşaması da kendi içerisinde Yüksekte Yürüyüş ve Göbekte Yürüyüş olmak üzere iki bölüme ayrılmaktadır. Bunlar SAA'nın operatörlerini oluşturmaktadır. Benzer şekilde; Avlanma aşaması da iki bölüme ayrılmaktadır. Bunlar; Avlanma Koordinasyonu ve Avlanma İşbirliğidir. Bu dört operatör algoritmanın iterasyon sayısına göre Tablo 1.'deki şekilde işletilirler [22-23].

Tablo 1. SAA'nın Operatörleri

Şart	Operatör	Denklemleri
$t \leq \frac{T}{4}$	Yüksekte Yürüyüş	$Best_j(t) \times -\eta_{ij}(t) \times \beta - R_{ij}(t) \times rand$
$t > \frac{T}{4}$ ve $t \leq 2\frac{T}{4}$	Göbekte Yürüyüş	$Best_j(t) \times x_{r_1j} \times ES(t) \times rand$
$t > 2\frac{T}{4}$ ve $t \leq 3\frac{T}{4}$	Avlanma Koordinasyonu	$Best_j(t) \times P_{ij}(t) \times rand$
$t > 3\frac{T}{4}$ ve $t \leq T$	Avlanma İşbirliği	$Best_j(t) - \eta_{ij}(t) \times \varepsilon - R_{ij}(t) \times rand$

Tablo 1.'deki t , şu anki iterasyon değerini gösterirken, T , algoritma için maksimum iterasyon sayısını belirtmektedir. Burada $Best_j$, popülasyonun j . sütunundaki en iyi değeri belirtmektedir. η_{ij} algoritmanın arama operatörüdür. β , keşif aşamasında doğruluğu kontrol eden bir parametredir. R_{ij} , arama alanını küçültmek amacıyla kullanılan bir fonksiyondur. r_1 , 1 ile N arasında rastgele üretilen bir tamsayıdır. ES , [-2,2] aralığında değerler alan Evrimsel Algı parametresidir. ε , küçük bir değerdir. ε (eps) değeri harici olarak tanımlanmadığı zaman, MATLAB tarafından 2^{-52} olarak atanmaktadır. Ancak GSAA yöntemi için 10^{-2} olarak belirlenmiştir. P_{ij} ise, mevcut konum ile en iyi konum arasındaki yüzdesel farkı göstermektedir. P_{ij} değeri Denklem 5'deki matematiksel ifade ile hesaplanmaktadır.

$$P_{ij}(t) = \alpha_1 + (x_{i,j}(t) - mean(x_i(t))) / (Best_j(t) * (UB - LB) + \varepsilon) \quad (5)$$

Burada, $x_{i,j}$ mevcut konumu gösterirken, $mean()$ fonksiyonu ile tüm konumların ortalaması hesaplanmaktadır. $Best_j$, popülasyonun j . sütunundaki en iyi değeri belirtmektedir. α_1 ise sabit bir değerdir. Genellikle $\alpha_1=0.1$ olarak alınmaktadır. Ayrıca, UB ve LB sırasıyla, problemin parametre değerlerinin üst limit ve alt limitlerini belirtmektedir. ε ise küçük bir değer olarak belirlenir [22].

C. GELİŞTİRİLMİŞ SÜRÜNGEN ARAMA ALGORİTMASI

SAA'nın optimizasyon problemlerindeki etkililiğini arttırmak amacıyla operatörleri üzerinde bazı güncellemeler yapılmıştır. Öncelikle performans üzerindeki etkisinin düşük olduğu tespit edilen Yüksekte Yürüyüş Operatörü, algoritmanın başlangıç adımındakine benzer şekilde, sınır şartları gözetilerek rastgele değerler üreten bir operatörle değiştirilmiştir.

İkinci olarak, yine etkisi düşük olan Avlanma İşbirliği operatörünün yerine Bal Porsuğu Algoritmasının bir operatörü eklenerek geliştirilmiştir [24]. Bal Porsuğu Algoritması, bal porsuğu adı verilen memeli hayvan türünün avlarına ve bala ulaşma davranışlarını modelleyen, doğadan

esinlenerek geliştirilmiş bir meta-sezgisel yöntemdir. Bu algoritma, bal porsuğunun yiyecek arama şekli olarak tanımlanan kazma ve bal modu adında iki operatöre sahiptir. Kazma operatörü, bal porsuğunun böcek, kurbağa, kaplumbağa, kertenkele gibi avlarını yakalamak için kullandığı çevreleme, kazma ve yakalama aşamalarından esinlenerek tasarlanmıştır. Bal modu ise, bal porsuğunun, kovana götüren kuşu takip etmesinin modellendiği operatördür [24-25]. Bal Porsuğu Algoritmasının Bal modu operatörü, SAA' nın yeni versiyonu için performansı artırıcı olarak eklenmiştir. Problem için algoritmanın tüm operatörleri tek tek uygulanarak performansları test edilmiştir. Buna göre, performans üzerinde etkisinin düşük olduğu görülen GSAA' nın Avlanma İşbirliği operatörü, Bal Porsuğu Algoritmasının Bal Modu operatörü ile değiştirilmiştir. Bal Modu operatörünün matematiksel ifadesi Denklem 6'de verilmektedir.

$$x_{i,j}(t+1) = Best_j(t) + F * rand() * \alpha_2 * d_i \quad (6)$$

Burada, $Best_j(t)$, j . pozisyon için şu ana kadar elde edilen en iyi sonucu belirtmektedir. F , arama yönünü değiştirmek amacıyla kullanılan bayrağı ifade etmektedir. F bayrağını hesaplamak için Denklem 7 kullanılmaktadır.

$$F = \begin{cases} 1 & \text{if } r_2 \leq 0.5 \\ -1 & \text{else} \end{cases} \quad (7)$$

Burada r_2 , $rand()$ fonksiyonu ile rastgele hesaplanan $[0,1]$ arasında bir sayıdır. Denklem 5'de α_2 , yoğunluk faktörünü belirtmektedir. α_2 'nın hesaplanması için Denklem 8 kullanılmaktadır. Burada, t , mevcut iterasyon değerini gösterirken, T , algoritma için maksimum iterasyon sayısını belirtmektedir. C ise bir sabittir. $C \geq 1$ olmalıdır.

$$\alpha_2 = C \times \exp\left(\frac{-t}{T}\right) \quad (8)$$

d_i , i . konum için en iyi değerle mevcut değer arasındaki farkı göstermektedir [24].

Son olarak, algoritmanın mevcut iterasyon değeri (t) ile maksimum iterasyon değerine (T) bağlı olarak hesaplanan operatörlerin yürütülme şartları, $mod(t)$ 'ye göre hesaplanmıştır. Böylece operatörleri işletirken, iterasyon sayısını %25'lik dilimlere ayrılmak yerine; ilk iterasyondan itibaren her operatör sırayla işletilmiştir. İterasyon adımları arasında, uygulanacak işlem çeşitliliği artırılmıştır. Bu da performansı olumlu yönde etkilemiştir. GSAA' nın operatörleri Tablo 2.'de gösterilmektedir.

Tablo 2. GSAA' nın Operatörleri

Şart	Operatör	Denklemleri
$mod(t, 4) == 0$	Geliştirilmiş Yüksekte Yürüyüş	$rand(1,1) * (UB - LB) + LB$
$mod(t, 4) == 1$	Göbekte Yürüyüş	$Best_j(t) \times x_{r_1j} \times ES(t) \times rand$
$mod(t, 4) == 2$	Avlanma Koordinasyonu	$Best_j(t) \times P_{ij}(t) \times rand$
$mod(t, 4) == 3$	Geliştirilmiş Avlanma İşbirliği	$Best_j(t) + F * rand() * \alpha_2 * d_i$

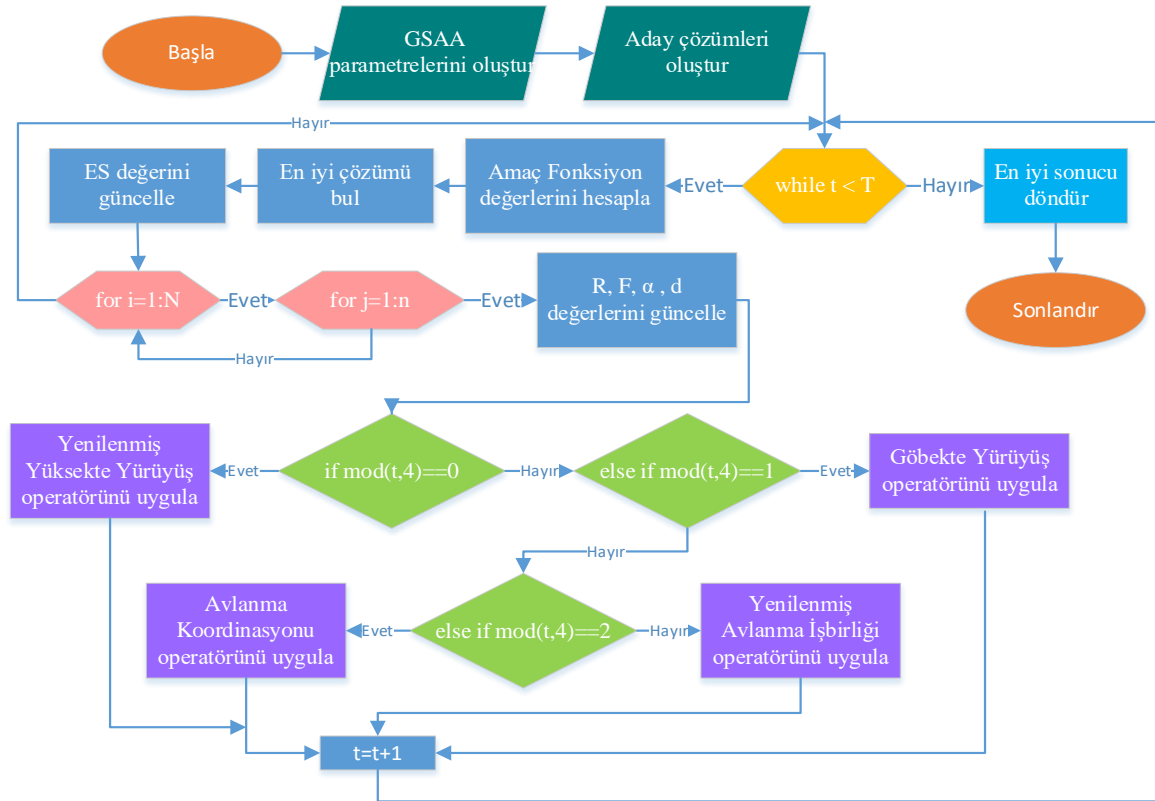
Burada, UB ve LB sırasıyla EYD probleminin parametre değerlerinin üst limit ve alt limitlerini belirtmektedir. F , arama yönünü değiştirmek amacıyla kullanılan bir bayraktır. α_2 , yoğunluk faktörüdür. Bu yolla, iterasyon sayısı arttıkça randomizasyon azalır [24].

GSAA'nın sözde (pseudo) kodu aşağıda verilmektedir.

Algoritma 1 GSAA'nın Sözde Kodu

```
1: Başlangıç
2: GSAA parametrelerini oluştur ( $\beta, \varepsilon, T, N$ )
3: Aday çözümleri rastgele(random) değerlerle oluştur ( $X: i = 1, \dots, N$ )
4: while ( $t < T$ )
5:   Aday çözümler (X) için Amaç Fonksiyon değerlerini hesapla
6:   En iyi çözümü bul
7:   ES değerini güncelle
8:   for ( $i=1$ 'den  $N$ 'e kadar) (N: Aday çözüm sayısı (Sürüngen sayısı))
9:     for ( $j=1$ 'den  $n$ 'e kadar) (n: Problemin parametre sayısı)
10:       $R_{ij}, F, \alpha_2, d_i$  değerlerini güncelle
11:      if ( $\text{mod}(t, 4) == 0$ ) ise,
12:         $\text{rand}(1,1) * (UB - LB) + LB$ 
13:      else if ( $\text{mod}(t, 4) == 1$ ) ise,
14:         $Best_j(t) \times x_{r_1j} \times ES(t) \times \text{rand}$ 
15:      else if ( $\text{mod}(t, 4) == 2$ ) ise,
16:         $Best_j(t) \times P_{ij}(t) \times \text{rand}$ 
17:      else
18:         $Best_j(t) + F * \text{rand}() * \alpha_2 * d_i$ 
19:      end if
20:    end for
21:  end for
22:   $t=t+1$ 
23: end while
24: En iyi sonucu geri döndür (Best(X))
```

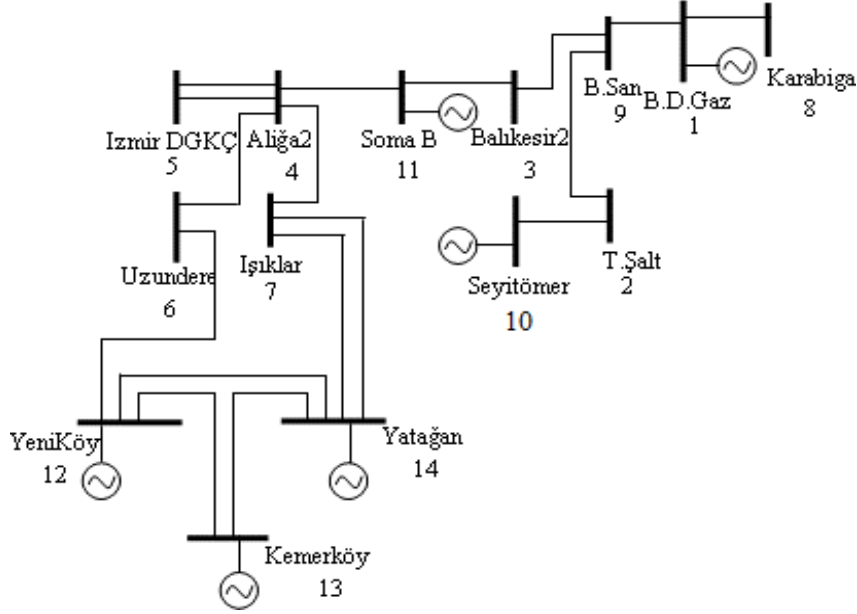
Şekil 1.'de GSAA için akış diyagramı verilmektedir.



Şekil 1. GSAA'nın akış diyagramı

III. BULGULAR

Türkiye’de bulunan 14 baralı, 380kV’luk 6 generatörlü bir güç sistemi Şekil 2.’de gösterilmiştir. Yapılan çalışmada, bu sistemin generatörleri arasında optimum yük dağılımı sağlamak ve mümkün olan en ekonomik şekilde elektrik üretim sürecini tamamlamak amacıyla GSAA kullanılmıştır.



Şekil 2. Türkiye’de bulunan 14 baralı 6 generatörlü güç sistemi örneği [13]

Türkiye özelinde tasarlanan bu sistemin yük dağıtım sabitlerini ve yük talebini de içeren sistem gereklilikleri Tablo 3.’de verilmektedir. Sistemin yük talebi 2734.9 MW olarak belirlenmiştir. Ayrıca, bu problemde valf noktası yükleme etkisi ve sistem içerisinde meydana gelen güç kayıpları dikkate alınmamıştır.

Tablo 3. Türkiye’deki generatörlerin yük dağıtım sabitleri ve sistemin yük talebi [13]

Generatörler	P_{min} (MW)	P_{max} (MW)	a	b	c
Bursa D. Gaz	318	1432	6780.5	5.682	0.0106
Seyitömer	150	600	1564.4	3.1288	0.0139
Soma B.	210	990	5134.1	6.2232	0.0168
YeniKöy	110	420	1159.5	3.3128	0.0210
Kemerköy	140	630	1697	3.2324	0.0137
Yatağan	140	630	1822.8	3.472	0.0147
Sistemin Yük Talebi (MW)					2734.9

Sistem gereksinimleri alındıktan sonra, SAA ve GSAA için parametre değerleri ayarlanmıştır. Bu problem için sürüngen (timsah) sayısı 100 ve iterasyon sayısı 10.000 olarak belirlenmiştir. SAA ve GSAA için parametre değerleri Tablo 4.’de ayrıntılı olarak verilmiştir.

Tablo 4. EYD problemi için SAA ve GSAA parametreleri

Açıklamaları	Parametreler	Değerleri
Parametre sayısı	n (Ortak)	6
Aday çözüm sayısı (Sürüngen sayısı)	N (Ortak)	100
Maksimum iterasyon sayısı	T (Ortak)	10000
Keşif doğruluk sabiti	α_1 (Ortak)	0.1
Keşif doğruluk sabiti	β (SAA)	0.005
Sabit bir değer	C (GSAA)	2
Küçük bir değer	ε (Ortak)	10^{-2}

SAA ve GSAA ile Türkiye'deki güç sistemleri için EYD probleminin çözümü, MATLAB 2018b kullanılarak Intel Core-i5 işlemcili 2.5 GHz ve Turbo 3.1 GHz PC üzerinde gerçekleştirilmiştir.

Tablo 5. SAA ve GSAA ile Türkiye'deki 14 baralı 6 generatörlü güç sistemi için EYD probleminin çözümü

Algoritma			SAA	GSAA
Generatörler	P_{min} (MW)	P_{max} (MW)	P_i (MW)	P_i (MW)
Bursa D. Gaz	318	1432	413.08843	553.71409
Seyitömer	150	600	600	517.15158
Soma B.	210	990	210	334.31148
Yeniköy	110	420	252.06404	335.69595
Kemerköy	140	630	630	520.79677
Yatağan	140	630	630	473.23011
Sistemin Yük Talebi (MW)			2734.9	2734.9
Toplam Üretim Maliyeti (\$/h)			49160.82201	47661.10055

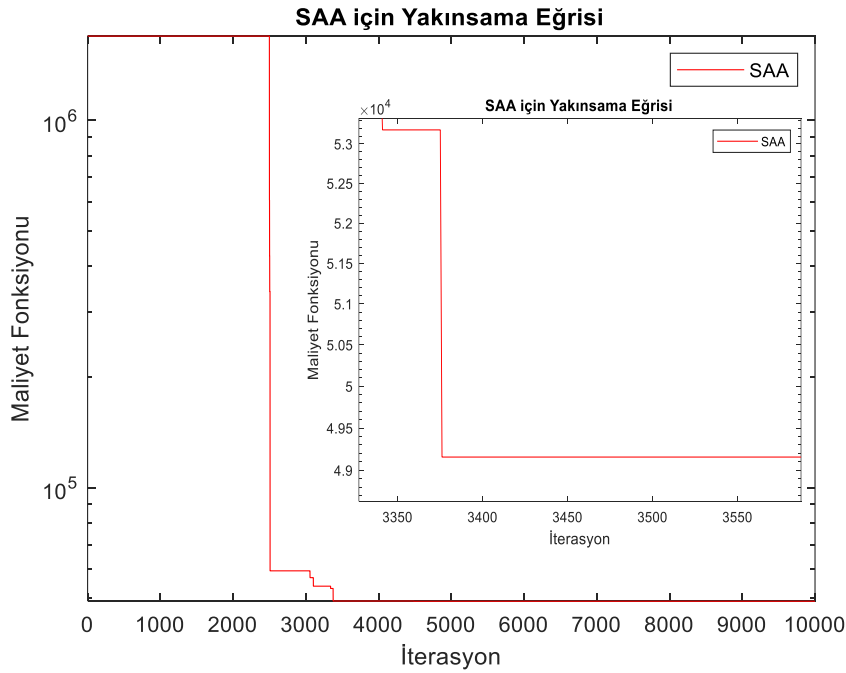
SAA ve GSAA ile Türkiye'deki 14 baralı 6 generatörlü 380kV'luk güç sisteminin EYD probleminin çözümü neticesinde Tablo 5.'de gösterilen generatör güç değerleri elde edilmiştir. Buna göre, SAA ile toplam üretim maliyeti 49160.822\$/h olarak elde edilirken, GSAA ile toplam üretim maliyeti 47661.100 \$/h olarak elde edilmiştir. Buna göre, GSAA, SAA'ya nazaran toplam üretim maliyetinde % 3.15'lik bir iyileştirme sağlamıştır. Tablo 5.'deki sonuçlara göre, SAA'nın geliştirilmiş yeni versiyonunun EYD probleminin çözümünde, ilk versiyonuna nazaran oldukça başarılı olduğu görülmektedir. Elde edilen bu sonuçlar, daha önce Türkiye'deki güç sistemleri için EYD probleminin çözümünde kullanılan Benzetilmiş Tolama yöntemi, Genetik Algoritmalar, Tabu Arama Algoritması ve Karga Arama Algoritması ile karşılaştırılmıştır. Karşılaştırmalı sonuçlar Tablo 6.'da verilmektedir.

Tablo 6. Türkiye'deki 14 baralı 6 generatörlü güç sistemi için EYD probleminin çözümünün karşılaştırmalı sonuçları

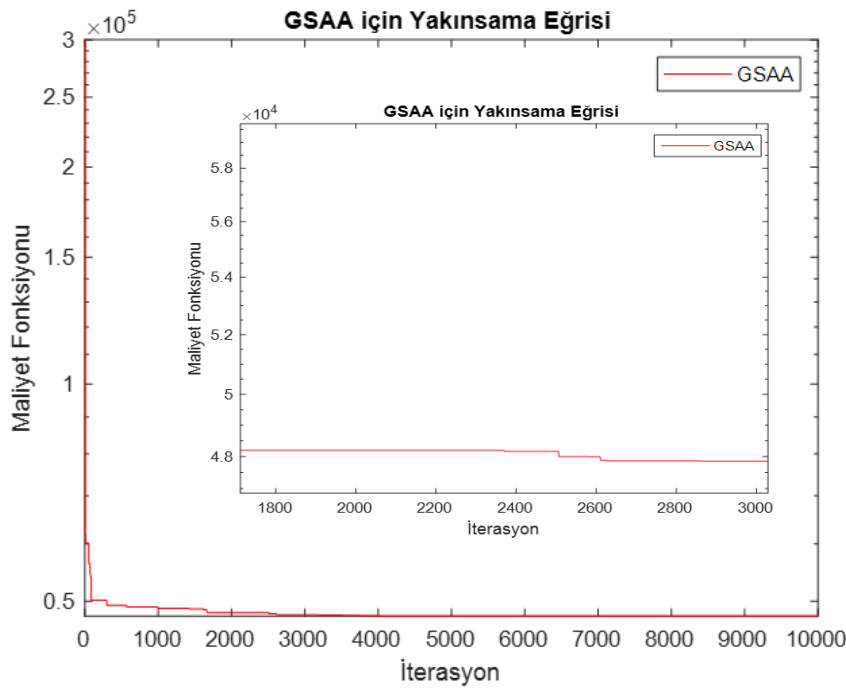
Generatörler	Benzetilmiş Tavlama Yöntemi [3]	Genetik Algoritma [18]	Tabu Arama Algoritması [3]	Karga Arama Algoritması [13]	GSAA
Bursa D. Gaz	483.5981	554.0455	553.6206	557.3107	553.7140
Seyitömer	478.4897	496.9588	503.5243	519.3833	517.1515
Soma B.	294.8322	320.8097	358.3317	330.1128	334.3114
Yeniköy	374.3331	357.1712	321.1134	336.7874	335.6959
Kemerköy	582.2331	519.2502	519.5332	519.3440	520.7967
Yatağan	521.4198	486.6204	478.7977	471.9617	473.2301
Toplam Üretim Maliyeti(\$/h)	47876	47679.2861	47678	47661.6375	47661.1005

Buna göre, Türkiye özelindeki bu güç sisteminin optimum yük dağılımını sağlamak amacıyla ortaya konmuş bu problem için, GSAA' nın Benzetilmiş Tavlama [3], Genetik Algoritma [18], Tabu Arama [3] ve Karga Arama Algoritmasından [13] daha iyi bir sonuç elde ettiği görülmektedir. GSAA, toplam üretim maliyeti bakımından Benzetilmiş Tavlama Yöntemine göre %0.45'lik bir iyileştirme sağlarken, Karga Arama Algoritmasına göre %0.001 iyileştirme sağlamıştır.

Şekil 3. ve Şekil 4. sırasıyla, SAA ve GSAA ile Türkiye özelindeki EYD probleminin çözümü için maliyet fonksiyonunun iterasyon sayısına bağlı değişimini gösteren yakınsama grafiğini vermektedir.



Şekil 3. Türkiye özelindeki EYD probleminin SAA ile çözümüne ait yakınsama grafiği



Şekil 4. Türkiye özelindeki EYD probleminin GSAA ile çözümüne ait yakınsama grafiği

Toplam üretim maliyeti bakımından en iyi performansı gösteren GSAA' nın algoritma yürütme zamanı 17.14 sn. olarak hesaplanmıştır. SAA' nın yürütme zamanı ise 21.54 sn. olarak hesaplanmıştır. Ayrıca, Tablo 6.'daki sonuçlar elde edilirken, SAA ve GSAA yöntemleri 20 kez koşturularak ulaşılan en iyi sonuçlar alınmıştır.

IV. SONUC

Bu makalede, timsah türünün avlanma davranışlarından esinlenerek geliştirilmiş bir meta-sezgisel yöntem olan SAA için yeni bir versiyon önerilmiştir. Buna göre; SAA' nın performans üzerindeki etkisi düşük olan Yüksekte Yürüyüş operatörünün içeriği, parametrelerin sınır şartlarına uygun olarak rastgele değerler üreten bir operatör ile güncellenmiştir. Ayrıca, performans üzerinde olumlu katkısının az olduğu belirlenen Avlanma İşbirliği operatörü de, başka bir meta-sezgisel algoritma olan Bal Porsuğu algoritmasının bir operatör ile değiştirilmiştir. Son olarak, algoritmanın hangi operatörünün ne zaman uygulanacağına karar verilen şart güncellenmiştir. Eski versiyonunda, mevcut iterasyon sayısı ve maksimum iterasyon sayısına bağlı olarak hangi operatörün uygulanacağı belirlenirken [22], yeni versiyonda sadece mevcut iterasyon değerine göre mod alma yöntemiyle belirlenmektedir. Yapılan bu değişiklik ve güncellemelerle, SAA' nın yeni versiyonu optimizasyon problemleri için daha etkin bir yöntem haline gelmiştir.

SAA'nın ilk versiyonu ve önerilen yeni versiyonu kullanılarak Türkiye'de bulunan 14 baralı 6 generatörlü güç sistemi için EYD probleminin çözümü gerçekleştirilmiştir. Deneysel sonuçlar, GSAA' nın Türkiye'deki güç sistemi için EYD probleminin çözümünde SAA'ya nazaran oldukça iyi bir performans sergilediğini ortaya koymuştur. Elde edilen sonuçlar, daha önce aynı problemin çözümünde kullanılan Benzetilmiş Tavlama [3], Genetik Algoritma [18], Tabu Arama [3] ve Karga Arama Algoritması [13] ile karşılaştırılmıştır. Toplam üretim maliyeti bakımından en iyi sonucun GSAA ile elde edildiği görülmüştür. Bu sonuçlar, SAA'da yapılan iyileştirmelerin etkili sonuçlar verdiğini ve önerilen yeni versiyonun EYD alanındaki optimizasyon problemleri için daha etkili bir yöntem olarak literatüre katkı sağlayabileceğini ortaya koymuştur.

Gelecekte yapılması öngörülen çalışmalarda, SAA ve GSAA' nın kıyaslama problemleri için performans analizinin yapılması planlanmaktadır. Ayrıca, daha büyük güç sistemleri için GSAA yönteminin kullanılarak başarısının değerlendirilmesi hedeflenmektedir. Buna ek olarak, Bal Porsuğu Algoritması ile GSAA' nın hibrit bir yaklaşımının tasarlanması ve farklı optimizasyon problemleri için denemesi mümkündür.

V. KAYNAKLAR

- [1] T. Yalçınöz, T. Yavuzer ve H. Altun, "Tabu araştırma algoritması kullanılarak ekonomik yük dağıtım probleminin çözümü," *Elektrik-Elektronik-Bilgisayar Mühendisliği Sempozyumu*, Bursa, Türkiye, 2002.
- [2] M. Kurban ve Ü. Başaran, "Türkiye'deki 380 kV'luk 14 baralı güç sisteminde ekonomik yüklenme analizi," *Elektrik-Elektronik, Bilgisayar Mühendisliği 11. Ulusal Kongresi*, İstanbul, Türkiye, 2005.
- [3] S. Duman ve A. Öztürk, "Türkiye'deki güç sisteminde tavlama benzetimi, genetik algoritma ve tabu araştırma algoritmaları kullanılarak ekonomik dağıtım," *Eng. Sci.*, c. 5, s.1, ss. 64-78, 2010.
- [4] R. Rahmani, M.F. Othman, R. Yusof, and M. Khalid, "Solving economic dispatch problem using particle swarm optimization by an evolutionary technique for initializing particles," *J. Theor. Appl. Inf. Technol.*, vol. 46, no. 2, pp. 526-536, 2012.

- [5] A.H. Bindu, and M.D. Reddy, "Economic load dispatch using cuckoo search algorithm," *Int. J. Eng. Res. Appl. (IJERA)*, vol. 3, no. 4, pp. 498-502, 2013.
- [6] S. Banerjee, D. Maity, and C.K. Chanda, "Teaching learning based optimization for economic load dispatch problem considering valve point loading effect," *Int. J. Electr. Power Energy Syst.*, vol. 73, pp. 456-464, 2015.
- [7] M.S. Turgut ve G.K. Demir, "Ekonomik yük dağıtım problemi için yapay işbirlikçi algoritması ile çözümü," *Dokuz Eylül Üniversitesi-Mühendislik Fakültesi Fen ve Mühendislik Dergisi*, c. 19, s. 55, ss. 16-27, 2017.
- [8] Ö.P. Akkaş, Y. Arıkan, and E. Çam, "Load dispatch for a power system in terms of economy and environment by using VIKOR method," *Dokuz Eylül University-Faculty of Engineering Journal of Science and Engineering*, vol. 19, no. 57, pp. 733-741, 2017.
- [9] E.S. Ali, and S.M. Abd Elazim, "Mine blast algorithm for environmental economic load dispatch with valve loading effect," *Neural Comput & Applic*, vol. 30, pp. 261-270, 2018.
- [10] Y.V.K. Reddy and M.D. Reddy, "Economic load dispatch problem with ant lion optimization using practical constraints," *Gazi Univ. J. Sci.*, vol. 32, no. 2, pp. 524-542, 2019.
- [11] M.U. Malik, M.M. Zaid and A. Ahmad, "Solution of economic load dispatch problem by modified whales optimization algorithm," *2019 International Symposium on Recent Advances in Electrical Engineering (RAEE)*, Islamabad, Pakistan, 2019.
- [12] C. Fu, S. Zhang, and K-H. Chao, "Energy management of a power system for economic load dispatch using the artificial intelligent algorithm," *Electronics*, vol. 9, no. 108, pp. 1-19, 2020.
- [13] C. Andiç, A. Öztürk ve S. Tosun, "Türkiye'deki güç sisteminde karga arama algoritması kullanılarak ekonomik yük dağıtım," *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, c. 8, s. 2020, ss. 428-436, 2020.
- [14] S. Deb, E.H. Houssein, M. Said and D.S. Abdelminaam, "Performance of turbulent flow of water optimization on economic load dispatch problem," *IEEE Access*, vol. 9, pp. 77882-77893, 2021.
- [15] M.A. Al-Betar, M.A. Awadallah, R.A. Zitar, and K. Assaleh, "Economic load dispatch using memetic sine cosine algorithm," *J. Ambient Intell. Hum. Comput.*, vol. 14, pp. 11685-11713, 2023.
- [16] W. Aribowo, "Comparison study on economic load dispatch using metaheuristic algorithm," *Gazi Univ. J. Sci.*, vol. 35, no. 1, pp. 26-40, 2022.
- [17] M. Said, E. H. Houssein, S. Deb, R. M. Ghoniem and A. G. Elsayed, "Economic load dispatch problem based on search and rescue optimization algorithm," *IEEE Access*, vol. 10, pp. 47109-47123, 2022.
- [18] M.K. Döşoğlu, S. Duman ve A. Öztürk, "Genetik algoritma kullanılarak ekonomik dağıtım analizi: Türkiye uygulaması," *J. Polytech.*, c. 12, s. 3, ss. 167-172, 2009.
- [19] I. Ahmed, A.R. Rao, A. Shah, E. Alamzeb, and J.A. Khan, "Performance of various metaheuristic techniques for economic dispatch problem with valve point loading effects and multiple fueling options," *Adv. Electr. Eng.*, vol. 2014, no. 765053, pp. 1-14, 2014. <https://doi.org/10.1155/2014/765053>.
- [20] A.E. Yıldırım, "Optimization of fuel cost in electric power systems using harmony search algorithm," *Int. J. Eng. Res. Dev.*, vol. 13, no. 2, pp. 531-544, 2021.

- [21] I. Ahmed, U-E-H. Alvi, A. Basit, T. Khursheed, A. Alvi, K-S. Hong, and M. Rehan, "A novel hybrid soft computing optimization framework for dynamic economic dispatch problem of complex non-convex contiguous constrained machines," *PLoS ONE*, vol. 17, no. 1, pp. 1-32, 2022.
- [22] L. Abualigah, M. Abd Elaziz, P. Sumari, Z.W. Geem, and A.H. Gandomi, "Reptile search algorithm (RSA): A nature-inspired meta-heuristic optimizer," *Expert Syst. Appl.*, vol. 191, no. 116158, 2022.
- [23] A.E. Yıldırım, "Solving optimum load dispatch problem with valve-point loading effect using reptile search algorithm," *The Hodja Akhmet Yassawi International Conference of Scientific Research*, Lankaran, Azerbaijan, 2022, pp. 510-516.
- [24] F.A. Hashim, E.H. Houssein, K. Hussain, M.S. Mabrouk, and W. Al-Atabany, "Honey badger algorithm: New metaheuristic algorithm for solving optimization problems," *Math. Comput. Simul.*, vol. 192, pp. 84-110, 2022.
- [25] Y. Xiao, X. Sun, Y. Guo, H. Cui, Y. Wang, J. Li, and S. Li, "An enhanced honey badger algorithm based on Lévy flight and refraction opposition-based learning for engineering design problems," *J. Intell. Fuzzy Syst.*, vol. 43, no. 4, pp. 4517-4540, 2022.