Araştırma Makalesi
Zeki Sistemler Teori ve Uygulamaları Dergisi 4(2) (2021) 76-85
DOI: 10.38016/jista.857881

Harris Şahinleri ve Balina Optimizasyon Algoritmalarının Kısıt İşleme Teknikleriyle Uygulaması: Karşılaştırmalı bir çalışma

Zeynep Garip[1]*, Murat Erhan Çimen[2], Ali Fuat Boz[2]

[1]Sakarya Uygulamalı Bilimler Üniversitesi, Bilgisayar Mühendisliği Bölümü, Sakarya, Türkiye

[2]Sakarya Uygulamalı Bilimler Üniversitesi, Elektrik ve Elektronik Mühendisliği Bölümü, Sakarya, Türkiye

zbatik@subu.edu.tr, muratcimen@subu.edu.tr, afboz@subu.edu.tr

Öz
Bu çalışmada kısıtlı optimizasyon problemlerin çözümünde, doğadan ilham alan meta-sezgisel algoritmaların etkisine odaklanılmıştır. Kısıtlı ihlal tekniklerinden olan ölüm ceza, statik ceza, dinamik ceza, bariyer fonksiyon ve Deb uygulanabilirlik kuralı Balina Optimizasyon (BOA) ve Harris Şahini Optimizasyon (HŞO) algoritmaları üzerinde test edilmiştir. Algoritmaların performansını test etmede kısıtsız ve kısıtlı benchmark fonksiyonları ve optimal güç akışı minimizasyon problemi kullanılmıştır. Ayrıca BOA ve HŞO algoritmaların optimal güç akışında gösterdikleri performanslarını karşılaştırmak amacıyla literatürde bulunan algoritmalarla kullanılmıştır. Sonuç olarak kısıt ihlal yöntemlerine entegre edilmiş algoritmaların kısıtlı optimizasyon problemlerin çözümünde etkili olduğu görülmüştür.

Anahtar kelimeler: Meta-sezgisel Algoritma; Kısıt İhlal Yöntemleri; Optimal Güç Akışı; Optimizasyon.

Application of Harris Hawks and Whale Optimization Algorithm with Constraint Handling Techniques: A comparative study

Abstract
This study focuses on the effect of meta-heuristic algorithms inspired by nature in solving constrained optimization problems. Death penalty, static penalty, dynamic penalty, barrier function and Deb feasibility rule, which are the constraint handling techniques, were tested on Whale Optimization (WOA) and Harris Hawk Optimization (HHO) algorithms. Unconstrained and constrained benchmark functions and optimal power flow minimization problem were used to test the performance of algorithms. Furthermore, in order to compare the performance of WOA and HHO algorithms in optimal power flow, it was used with algorithms found in the literature.. As a result, it has been observed that algorithms integrated into constraint handling methods are effective in solving constrained optimization problems.

Keywords: Meta-heuristic Algorithm; Constraint Handling Methods; Optimal Power Flow; Optimization.

1. Giriş (Introduction)

Kısıtlı optimizasyon problemlerini çözmede meta-sezgisel algoritmaların kısıtlı optimizasyona uyarlanması önemli bir görevdir. Kısıtlama yöntemleri ceza fonksiyonlarının varlığına bağlı olarak iki gruba ayrılır. İlki ceza temeli kısıtlı yöntemlerinde amaç fonksiyonuna uygulanabilir bölge dışında bulunan fonksiyona değeri ceza olarak eklenir. Bu yöntemlere ilave olarak Deb uygulanabilirlik yönteminin önerilmesi algoritmaların kısıtlı problemlerin çözümüne farklı bir bakış açısı kazandırmıştır. Literatürde ölüm cezası (Fan vd., 2019), statik ceza (Tsipianitis vd., 2020), dinamik ceza (Paszkowicz 2009), bariyer fonksiyonu (Matias vd., 2015), epsilon kısıtlama (Fang vd., 2020), stokastik sıralama (Bansal vd., 2009), Deb uygulanabilirlik kuralları (Babalik vd., 2018) gibi çeşitli kısıtlama teknikleri bulunmaktadır. Ayrıca kısıtlama teknikleri üzerinde yürütülen çalışmalarda sezgisel algoritmaların performanslarının arttırılmasına yönelik çalışmalar yapılmaya devam etmektedir.

Son zamanlarda popülasyon tabanlı meta-sezgisel algoritmaların optimizasyon performansları, basit ve etkili yapıları, adaptasyon ve uygulama kolaylığı nedeniyle

* Sorumlu yazar.
E-posta adresi: zbatik@subu.edu.tr

Alındı : 11 Ocak 2021
Revizyon : 15 Mart 2021
Kabul : 20 Mart 2021

araştırmacılar tarafından sıklıkla tercih edilmektedir (Babalik vd., 2018). Meta sezgisel algoritmalar keşif ve sömürü olmak üzere iki arama yöntemini içermektedir. Bu iki yöntemin dengelenmesi algoritmaların farklılaşmasını sağlar. Bununla birlikte arama sürecindeki rastgelelikten dolayı keşif ve sömürü arasında denge sağlanması kolay değildir. Ayrıca tüm sorunları çözecek evrensel bir algoritma yoktur. Başka bir deyişle bir algoritma bazı problemler için başarılı sonuçlar verirken diğerlerinde başarılı sonuçlar vermeyebilir.

Meta sezgisel algoritmalar, ayrık veya sürekli, tek veya çok amaçlı, kısıtlı veya kısıtsız optimizasyon problemlerinin çözümünde etkili bir şekilde kullanılmaktadır (Haklı 2019). Burada çözülmesi gereken önemli, sorun arama süresi boyunca en uygun değerin seçilmesidir (Chen vd., 2019). Kısıtlı optimizasyon problemlerine yapısal optimizasyon, mühendislik tasarımı, VLSI tasarımı, ekonomi gibi birçok alanda karşılaşılmaktadır (Akay vd., 2011). Son zamanlarda, doğadaki birey rekabetinden ve sürü işbirliğinden esinlenerek kısıtlı optimizasyon problemini çözmek için genetik algoritma (Jelovic vd., 2020), diferansiyel evrimi (Amaratunga vd., 2018), yapay arı koloni (Akay vd., 2011), parçacık sürüsü optimizasyonu (Kohler vd., 2019), çiçek tozlaşma algoritması (Fan vd., 2019), ağaç tohum algoritması (Babalik vd., 2019), sinüs kosinüs algoritması(Chen vd., 2020) ve ateş böceği algoritma (Tuba vd., 2014), Harris Şahini (Aljarah vd., 2019) ve Balina Optimizasyon algoritmaları (Lewis vd., 2016) gibi birçok meta sezgisel algoritma kullanılmıştır.

Literatürde kısıtlı optimizasyon problemlerini çözmek için çeşitli kısıtlama yöntemleri ve meta-sezgisel algoritmalar birlikte kullanılmaktadır. Babalık ve ark. tarafından kısıtlı optimizasyon tekniklerini çözmek amacıyla kısıtlama tekniklerinden biri olan Deb'in kuralları kullanılarak ağaç tohumu algoritması modifiye edilmiştir (Babalik vd., 2018). Karaboğa ve ark. kısıtlı problemleri çözmek için ABC algoritmasının seçim mekanizmasını Deb'in seçim mekanizmasıyla değiştirmiştir. Geliştirilen algoritmada kısıtlı optimizasyon problemlerin çözümünde uygunluk değerlerine göre uygulanabilir ve ihlallere göre uygun olmayan çözümlere olasılık değeri atanarak kısıtlamalar çözülmeye çalışılmıştır (Akay vd., 2011). Miranda-Varela ve ark. hedeflenen fonksiyonun değerine ve kısıt ihlallerinin toplamına yaklaşmak amacıyla kısıt yaklaşımları ve en yakın komşu regresyonu tabanlı diferansiyel algoritmaları arasındaki ilişki üzerinde durulmuştur (Miranda-Varela vd., 2018). Samanipour ve ark. dominat olmayan genetik algoritmayla kısıtlama tekniklerinin etkinliğini artırmak amacıyla onarım yaklaşımı önermişlerdir (Jelovic vd., 2020). Rodrigues ve ark. optimizasyon problemlerindeki kısıtları göz önünde bulundurarak, uyarlanabilir genişletilmiş dengeli bir sıralama yöntemi geliştirmişlerdir (Guimarães vd., 2018). Gandomi ve ark. tarafından önerilen yaklaşımda; kısıtlamaları ihlal etmeyecek şekilde uyarlanabilen ve değişkenin sınırlarını onararak uygulanabilir çözümlerin değerlendirilmesini sağlamışlardır (Deb vd., 2020).

Tablo 1. Literatür Karşılaştırması (Literature Comparison)

Yöntem	Kısıtlama Tekniği	Referans
BOA, HŞO	Ölüm cezası, Statik ceza, Dinamik ceza, Bariyer ceza ve Deb uygulanabilirlik kuralı	Bu çalışma
Ağaç Tohum Algoritması	Deb Kuralları	(Babalik vd., 2018)
Yapay Arı Koloni Algoritması	Deb seçim mekanizması	(Akay vd., 2011)
Diferansiyel Evrim Algoritması	Uygunluk kuralları, ε-kısıt methot, Stokastik sıralama, Çeşitlilik sağlama	(Miranda-Varela vd., 2018)
Genetik Algoritma	Uyarlanabilir eşik yaklaşımı, Deterministik Uygun Olmama Sıralaması	(Jelovic vd., 2020)
Genişletilmiş dengeli sıralama yöntemi	Deb Uygulanabilirlik kuralı, Stokastik Sıralama, Ceza Yöntemleri	(Guimarães vd., 2018)
İç nokta, Sıralı Karesel Programlama, Doğrudan Arama, Benzetim Tavlama, Parçacık Sürü Optimizasyonu, Genetik Algoritma, Diferansiyel Evrim Algoritması	Sınır Belirleme	(Deb vd., 2020)

Bu çalışmada ölüm cezası, statik ceza, dinamik ceza, bariyer ceza ve Deb uygulanabilirlik kısıtlama teknikleri ve kambur balinaların avcılık davranışından esinlenilen Balina Optimizasyon algoritması (BOA) ve Harris şahinlerinin yiyecek arama davranışından ilham alınan Harris Şahini Optimizasyon Algoritması (HŞO), kısıtlı optimizasyon problemlerini çözmek için uyarlanmıştır.

Bu çalışmanın başlıca katkıları aşağıdaki gibi özetlenebilir:

1. Ölüm cezası, statik ceza, dinamik ceza, bariyer ceza ve Deb uygulanabilirlik kısıtlama teknikleri incelenmiştir.
2. Kısıt işlemeye yöntemlerine HŞO ve BOA meta-sezgisel algoritmaları adapte edilmiştir.
3. Algoritmaların performansları kısıtsız ve kısıtlı fonksiyonlar ve optimal yük akışı mühendislik problemi üzerinde incelenmiştir.
4. BOA ve HŞO algoritmaların optimal güç akışında gösterdikleri performanslarını karşılaştırmak amacıyla literatürde bulunan HŞO (Islam vd., 2020), Gri Kurt Optimizasyon (GKO) (Islam vd., 2020), BOA (Islam vd., 2020), GA (Bouktir vd., 2008), Harris Şahini-

Diferansiyel Evrim (HŞODE) (Birogul, 2019) ve Modifiyeli diferansiyel evrim (Sayah vd., 2008) algoritmaları kullanılmıştır.

2. Optimizasyon (Optimization)

Optimizasyon problemlerinde belirli sınırlar arasındaki parametreler ile en uygun çözümü bulmaktır. Buna ek olarak optimizasyon işleminde tüm kısıtlamaları sağlayan birçok çözüm içerisinden en uygun çözüm ya da yaklaşık çözümün bulunması amaçlanmaktadır (Mert vd., 2021). Bu bölümde kısıtlı optimizasyon probleminin tanımı yapıldıktan sonra Harris şahin optimizasyon ve balina optimizasyon algoritmaları açıklanmıştır.

2.1. Kısıtlı problemlerin matematiksel modellenmesi (Mathematical modeling of constrained problems)

Optimize edilmek istenen amaç fonksiyonuna $f(X)$ eşitlik ve/veya eşitsizlik kısıdı (sırasıyla $h_i(X) = 0$, $g_i(X) \leq 0$) eklendiğinde problem kısıtlı optimizasyon problemi haline dönüşmektedir (Garcia vd., 2017). Eşitlik 1'de verilen çok değişkenli bir kısıtlı optimizasyon probleminde $f(X)$ fonksiyonun minimum olması istenmektedir. Bu problemin çözümünde N adet eşitlik $h(X)$ kısıtı ve M adet eşitsizlik $g(X)$kısıtlarında sağlaması gerekmektedir.

$$\begin{aligned} &minimum\ f(X) && X = [x_1, x_2, \ldots, x_D] \in R^D \\ &h_i(X) = 0 && (i = 1, 2, \ldots, N) \\ &g_i(X) \leq 0 && (j = 1, 2, \ldots, M) \end{aligned} \quad (1)$$

Kısıtlı optimizasyon problemlerinde kısıt fonksiyonları amaç fonksiyonla aynı derece de etkilidir. Çünkü çözümün uygunluğu amaç fonksiyon kullanılarak hesaplanırken, çözümün uygulanabilirliği kısıtların ihlaline bağlıdır. Bu nedenle, bir çözümün uygulanabilirliği, uygunluk değerinden daha önemlidir (Babalik vd., 2018).

2.2. Balina Optimizasyon Algoritma (Whale Optimization Algorithm)

Balina optimizasyonu algoritması (BOA), optimizasyon problemlerinde kullanılmak amacıyla kambur balinaların avcılık davranışından ilham alınarak Marjalili ve Lewis tarafından önerilmiştir (Lewis vd., 2016). Yalnızca kambur balinalarda gözlenen yiyecek arama davranışı, kabarcık-ağ besleme yöntemidir. Balinalar, avlanma sırasında avı çevrelerken dairesel bir yol boyunca kabarcıklar oluşturur. Optimizasyonun gerçekleştirilmesi için, spiral kabarcık-ağ besleme davranışının işlem adımlarını gösteren sözde kodu Şekil 1'de verilmiştir.

```
Balina popülasyonunu oluştur Xi (i =1, 2, …, n)
Her bir ajan uygunluk popülasyonu hesapla
X* =en iyi arama ajanı
while (t < maksimum iterasyon sayısı)
for her arama ajanı
a, A, C, l, ve p değerlerini güncelle
if (p < 0.5)
  if (|A| < 1)
Eş.(2) ile mevcut arama ajanın konumunu güncelle
  else if (|A| ≥1)
Rasgele arama ajanı seç (X_rand)
Eş.(3) ile mevcut arama ajanın konumunu güncelle
  end if
else if (p≥0.5)
Eş.(8) ile mevcut arama ajanın konumunu güncelle
end if
end for
Herhangi bir arama ajanının arama alanının ötesine
geçip geçmediğini kontrol edin ve değiştirin
Her bir arama ajanın uygunluğunu hesapla
Daha iyi bir çözüm varsa X* güncelle
t=t+1
end while
return X*
```

Şekil 1. BOA'nın sözde kodu (Pseudo Code of WOA)

2.1.1. Avı Kuşatma (Encircling prey)

Kambur balinalar avlanırken avının yerini bulabilir ve avı çevreleyebilir. BOA'da arama alanındaki en uygun tasarımın yeri önceden bilinmediğinden, mevcut en iyi aday çözümün hedef av olduğunu veya optimum duruma yakın olduğunu varsayar. En iyi arama ajanı tanımlandıktan sonra, diğer arama ajanları konumlarını en iyi arama ajanına doğru güncellemeye çalışacaktır. Kambur balinaların avını kuşatma davranışının matematiksel modeli Denklem 2 ve 3'de gösterilmiştir. Denklem 2-3'de bulunan $\overrightarrow{X}(t)$ ajanın konumunu, t iterasyon, $\overrightarrow{X^*}$ en iyi çözümü ifade ederken Denklem 4 ve 5'da $\vec{A}, \vec{C}$ yakınsama değerlerini temsil etmektedir. $\vec{r}$ [0,1] rastgele sayı ve $\vec{a}$ iterasyon boyunca 2'den sıfıra doğrusal olarak azalan vektörü göstermektedir.

$$\vec{D} = \left|\vec{C}\, \vec{X}^*(t) - \overrightarrow{X}(t)\right| \quad (2)$$

$$\vec{X}^*(t+1) = \left|\vec{X}^*(t) - \vec{A}.\vec{D}\right| \quad (3)$$

$$\vec{A} = 2\vec{a}\,.\vec{r} - \vec{a} \quad (4)$$

$$\vec{C} = 2.\vec{r} \quad (5)$$

2.1.2. Kabarcık-net saldırı yöntemi (Bubble-net attacking method)

Kambur balinaların kabarcık-net saldırma yönteminde avına doğru ilerken arama çevresini küçültme ve ava doğru spiral şeklinde yol alma bulunmaktadır. Balinaların Denklem 8'de bulunan $\vec{a}$ değerinin düşürülmesiyle arama çevrelerini küçülterek avı yakalama davranışlarını sergilerler. $\vec{A}$ değeri de $\vec{a}$ değerine bağlı olduğundan dolayı 2'den sıfıra doğrusal olarak azalır. Kambur balinaların avını yakalarken oluşturdukları spiral şeklin matematiksel modeli Denklem 6 ve 7'de verilmiştir.

$$\overrightarrow{D'} = \left|\vec{X}^{*}(\mathrm{t}) - \overrightarrow{X}(\mathrm{t})\right| \tag{6}$$

$$\vec{X}^{*}(\mathrm{t}+1) = \overrightarrow{D'}.e^{bl}.\cos(2\pi l) + \vec{X}^{*}(\mathrm{t}) \tag{7}$$

Denklem 6 ve 7'de verilen D' balina ve en iyi av arasındaki mesafe, b logaritmik spiral sabiti, l ise [-1,1] arasında rastgele sayıdır. Kambur balinalar avına doğru hareket ederken yüzde 50 olasılıkla ya daralan hareket modelini ya da spiral hareket modelinden birini seçmektedir. Denklem 8'de bulunan p parametresi [0,1] aralığında rasgele sayıdır.

$$\vec{X}(t+1) = \begin{cases} \vec{X}^{*}(\mathrm{t}) - \vec{A}.\overrightarrow{D} & p < 0.5 \\ \overrightarrow{D'}.e^{bl}.cos(2\pi l) + \overrightarrow{X^{*}}(t) & p \geq 0.5 \end{cases} \tag{8}$$

2.2. Harris Şahini Optimizasyon Algoritması (Harris Hawks Optimization Algorithm)

Harris Şahini optimizasyon algoritması(HHO), optimizasyon problemlerinde kullanılmak amacıyla Harris şahinlerinin yiyecek arama davranışından ilham alınarak Ali Asghar Heidari ve arkadaşları tarafından önerilmiştir. Şahinler izleme, kuşatma ve saldırı özelliklerini kullanarak çeşitli aşamalarla işbirliğine dayalı yiyecek arama işlemini verimli bir şekilde gerçekleştirirler (He vd., 2020). Algoritmanın işlem adımlarını gösteren sözde kodu Şekil 2'de verilmiştir.

N boyutunda rasgele popülasyonu oluştur
while (Durdurma kriteri sağlanana kadar) **do**
Şahinlerin uygunluk değerlerini hesapla
Xtavsan konumu en iyi konum olarak ata
for (herbir şahin (X_i)) **do**
BAşlangıç enerjisi E_0 ve atlama kuvveti *J güncelle*
E_0=2rand()-1, J=2(1-rand())
Eş.(11) ile E'yi güncelle
if (|E|≥ 1) **then** ▷ Keşif aşaması
Eş.(9) ile konumu güncelle
if (|E|< 1) **then** ▷ Sömürü aşaması
if (r ≥0.5 and |E|≥ 0.5) **then** ▷ Yumuşak kuşatma stratejisi-
Eş.(12) ile konumu güncelle
else if (r ≥0.5 and |E|< 0.5) **then** ▷ Sert kuşatma stratejisi-
Eş.(13) ile konumu güncelle
else if (r < 0.5 and |E|≥ 0.5) **then** ▷ Hızlı saldırılarla yumuşak kuşatma stratejisi
Eş.(14) ile konumu güncelle
else if (r < 0.5 and |E|< 0.5) **then** ▷ Aşamalı hızlı saldırılarla sert kuşatma stratejisi- Eş.(16) ile konumu güncelle
Return *Xtavsan*

Şekil 2. HŞO algoritmasının sözde kodu(Pseudo Code of HHO)

2.2.1. Keşif aşaması (Exploration phase)

Harris şahinleri avını dev ağaçlarda veya telgraf direklerinde keskin bakışlarıyla arayarak keşif aşamasını gerçekleştirmektedir. HŞO algoritmasında, arama davranışı global keşif aşaması olarak kabul edilir. Global keşif stratejileri Denklem 9 ile matematiksel olarak ifade edilmektedir. Denklem 9'daki stratejilerden hangisinin seçileceğini olasılık değeri olan q değeri belirler.

$$x_i^{t+1} = \begin{cases} x_{rand}^{t} - r_1 * |x_{rand}^{t} - 2 * r_2 * x_i^{t}|, & q \geq 0.5 \\ \left(x_{tavşan} - x_{mean}^{t}\right) - r_3 * \left(lb + r_4 * (ub - lb)\right), & q < 0.5 \end{cases} \tag{9}$$

x_i^t değişkeni Harris Şahinin mevcut konumunu verirken x_i^{t+1} her bir iterasyondaki konum vektörüyken $x_{tavsan}(t)$ avın konum vektörüdür. r_1,r_2,r_3,r_4 ve q ise 0 ve 1 aralığındaki rassal sayılardır. ub ve lb, sırasıyla popülasyonun üst ve alt sınır değerleridir. x_{rand}^t mevcut popülasyondan rastgele seçilen bir şahini gösterirken, x_{mean}^t mevcut şahin popülasyonunun ortalama konum değerlerini vermektedir. Ortalama konum değeri t iterasyonda N şahin sayısı kullanılarak denklem 10 kullanılarak elde edilir (Aljarah vd., 2019).

$$x_{mean}^{t} = \frac{1}{n}\sum_{i=1}^{n} x_i^t \tag{10}$$

2.2.2. Keşiften sömürüye geçiş aşaması (Transition from exploration to exploitation)

HŞO algoritmasında global aramadan yerel sömürmeye geçiş kaçan avın enerjisine bağlı olarak oluşan enerji faktörü (E) ile kontrol edilmektedir. Enerji faktörü denklem 11 ile modellenmektedir. Bu modelde E kaçan avın enerjisini, T maksimum yineleme sayısını ve E_0 (-1, 1) avın enerjisinin başlangıç halidir.

$$E = 2E_0(1 - \frac{t}{T}) \tag{11}$$

Kaçan avın enerjisi $|E| \geq 1$ olduğunda, şahinler bir tavşanın yerini keşfetmek için farklı bölgelerde arama yaparlar böylece keşif aşaması gerçekleşir. $|E| < 1$ olduğu sömürü aşamasında ise çözümlerin komşuluğundan yararlanılmaktadır.

2.2.3. Sömürü Aşaması (Exploitation stage)

Harris şahinleri hedef avı bulduktan sonra avın etrafında bir çember oluşturur. Şahinler avın davranışa göre saldırı türünü belirlemektedir. Avın kaçma davranışlarına ve Harris'in şahinlerinin kovalamaca stratejilerine göre, saldırı aşamasını modellemek amacıyla dört olası strateji önerilmiştir. Stratejiler rastgele sayı(r) ve avın kaçan enerjisine(E) bağlıdır. r (0, 1) avın kuşatma halkasından kaçıp kaçamayacağına karar vermek için kullanılır.

Yumuşak kuşatma stratejisi r ≥ 0.5 ve | E | ≥ 0.5 olduğu durumdur. Bu durumda avın kaçma şansı yoktur, ancak kuşatma halkasından kaçacak kadar enerjisi olduğu için şahinler yumuşak kuşatma ile avlanırlar. Matematiksel modellemesi denklem 12'de verilmiştir. Δx^t

popülasyondaki uygun avdan mevcut arasındaki vektör mesafesidir. r_5 (0,1) 'de eşit olarak dağıtılmış rasgele bir sayı iken J ise avın kaçış sırasındaki atlama uzunluğunu gösterir.

$$x_i^{t+1} = \Delta x_i^t - E * \left|J * x_{tavşan} - x_i^t\right|$$
$$\Delta x_i^t = x_{tavşan} - x_i^t \qquad (12)$$
$$J = 2 * (1 - r_5)$$

Sert kuşatma stratejisi r ≥ 0.5 and |E| < 0.5 olduğu durumdur. Avın kaçma şansı olmadığı için enerjisi de yetersizdir. Bu durumda şahinler sert kuşatma ile avlanırlar.

$$x_i^{t+1} = x_{tavşan} - E * \left|\Delta x_i^t\right| \qquad (13)$$

Aşamalı hızlı saldırılarla yumuşak kuşatma stratejisinde r <0.5 ve | E | ≥ 0.5 olduğu durumdur. Bu durumda kaçmak için gerekli enerjiye sahip avın kuşatma halkasından kaçma şansı vardır. Bundan dolayı şahinler avı yakalamak amacıyla daha akıllı ve yumuşak bir kuşatma halkası oluşturacaktır. Bu strateji iki adımdan oluşmaktadır. Şahinler ava doğru konumu ilk adımla iyileşmezse ikinci adımla konumu güncellenmektedir. İlk adımda yumuşak kuşatma stratejisinde bulunan konum denklemi kullanılmaktadır. İkinci adım olan güncelleme modu Denklem 14 ile modellenmiştir. $s \in \mathbb{R}^{dim}$, 1xdim boyutunda rastgele bir vektördür. Levy fonksiyonu denklem 15 ile tanımlanmıştır. Burada u, v (0,1) arası rastgele sayı, β ise 1.5'tir.

$$z = \Delta x_i^t - E * \left|J * x_{tavşan} - x_i^t\right| + s * levy(dim) \qquad (14)$$

$$levy(x) = 0.01x \frac{u-\sigma}{|\mu|^{\frac{1}{\beta}}}, \sigma = \left(\frac{\Gamma(1+\beta)x\sin(\frac{\pi\beta}{2})}{\Gamma(1+\beta)x2^{\frac{\beta-1}{2}}}\right)^{\frac{1}{\beta}} \qquad (15)$$

Aşamalı hızlı saldırılarla sert kuşatma stratejisinde |E| <0.5 ve r < 0.5 olduğu durumdur. Bu durumda kaçmak için gerekli enerjisi olmayan avın kuşatma halkasından kaçma şansı yoktur. Bundan dolayı şahinler ava saldırmadan önce sert bir kuşatma halkasıyla yakalayarak öldürmektedir. Matematiksel modellemesi 16 denklemi ile ifade edilmektedir.

$$x_i^{t+1} = \begin{cases} y, & if\ f(y) < f(x_i^t) \\ z, & if\ f(z) < f(x_i^t) \end{cases}$$

$$y = x_{tavşan} - E * \left|J * x_{tavşan} - x_{mean}^t\right|$$

$$z = y + s * levy(dim) \qquad (16)$$

3. Kısıt İşleme Yöntemleri (Constraint Handling Methods)

Meta-sezgisel algoritmalar kısıtlı optimizasyon problemlerini çözmek için tasarlanmamıştır fakat bu sorunları çözmede çok etkilidirler. Kısıt işleme yöntemlerinin amaç fonksiyona eklenmesi arama uzayının uygun bölgelerinde arama yapmasını sağlar.

3.1. Ölüm Ceza Yöntemi (Death Penalty Methods)

Ölüm ceza yöntemi literatürdeki en kolay ve uygulanabilir kısıt işleme yöntemlerinden biridir. Bu yöntemde elde edilen olası çözümler uygun bölgede ise ceza değeri eklenmez. Aksi durumda yani olası çözümler içinde uygun bölgede bulunmazsa kısıtları ihlal eden çözümlere çok yüksek bir hata ataması yapılmaktadır. Yöntemin uygunluk fonksiyonu Denklem 17 ile modellenmektedir. Denklem 17'de verilen s değeri uygun bölge içerisindeki kısıt sayısını ifade etmektedir. K değeri ise çözümlerin uygun bölgede olması durumlarda eklenen yüksek bir değerdir.

$$\varphi(x) = \begin{cases} f(x) & kısıt\ ihlali\ yok \\ K - \sum_{i=1}^{s} \frac{K}{M} & kısıt\ ihlali\ var \end{cases} \qquad (17)$$

Bu yöntemde kısıtlar ihlal edildiğinde, uygunluk fonksiyonunun değerine bakılmaksızın ihlaller aynı hata değeri ile değerlendirilir. Diğer bir deyişle, uygunluk fonksiyon değeri ve ihlallerin büyüklükleri önemli değildir.

3.2. Statik Ceza Yöntemi (Static Penalty Methods)

Statik ceza kısıt işleme yönteminde cezalandırma işleminde her bir ihlal değeri sabit bir sayı ile çarpılarak hesaplanmaktadır. Çözümlerin uygun bölmeye alınması için uygunluk fonksiyonu düzenlenerek Eşitlik 18'deki matematiksel model elde edilir. Bu eşitlikte λ ve μ sabit parametreler 1 ile ∞ arasında değeri değişmektedir.

$$\varphi(X) = f(X) + \lambda \sum_{i=1}^{N} \left|h_i(X)\right| + \mu \sum_{j=1}^{M} \max\left\{0, h_j(X)\right\} \qquad (18)$$

3.3. Dinamik Ceza Yöntemi (Dynamic Penalty Methods)

Dinamik ceza yönteminde kısıtlama ihlal edildiğinde dinamik veya artan bir oranda ceza işlemi gerçekleşmektedir. Çözümlerin uygun bölmeye taşımak amacıyla ceza katsayısının güncellenmesi gerekmektedir. Ceza katsayısının güncellenmesi Eşitlik 19'da matematiksel modeldeki λ(t) iterasyon sayısına bağlı olarak dinamik olarak değişmektedir. λ(t) = $(\alpha t)^{\beta}$ matematiksel modelinde belirtildiği gibi $\lambda(t)$ değerinin dışında β değişkenin iterasyona bağlı olarak değişmesinin dinamik ceza yönteminin etkinliğini arttırmaktadır (Batık vd., 2019, Hatamlou vd., 2016).

$$\varphi(X) = f(X) + \lambda(t)\left(\sum_{i=1}^{N} \left(h_i(X)\right)^2 + \sum_{j=1}^{M} \left(\max\left\{0, h_j(X)\right\}\right)^2\right) \qquad (19)$$

Bu çalışmada kullanılan işlem parametreleri α = 0.5 ve β = 1.2 olarak alınmıştır. Fakat simülasyon çalışmalarında amaç fonksiyonun çok yüksek değerlerde olduğu görüldüğünden dolayı hata fonksiyonun yetersiz olduğu belirlenmiştir. Bu sebepten dolayı $\lambda(t)$ değeri $\lambda(t) = 10^3(1+(\alpha t)^\beta)$ matematiksel modeli ile hesaplanmaktadır.

3.4. Bariyer Fonksiyon (Barrier Function)

Bariyer fonksiyon kısıt işleme yönteminde eşitlik kısıtları lagrange çarpanlarıyla sağlanmaktadır. Eşitsizlik kısıtları ise çözümler uygun olmayan bölgeye yaklaştıkça ceza fonksiyonu büyük veya sonsuz değerler almaktadır. Sonraki aşama işe elde edilen ceza fonksiyon değeri amaç fonksiyonun değerine eklenmektedir. Bariyer fonksiyonun matematiksel modellemesi Eşitlik 20 veya 21 ile sağlanmaktadır. Buna ek olarak bu fonksiyon değerleri hesaplanırken iterasyonlar sayısı artması durumunda $\mu(t) > 0$, $\mu \rightarrow 0$ ve $\mu(t) = 1/t$ veya $\mu(t) = 1/\sqrt{t}$ seçilmesi gerekmektedir.

$$\varphi(X) = f(X) + \mu(t)\sum_{j=1}^{M} -\log\left(g_j(X)\right) \quad (20)$$

$$\varphi(X) = f(X) + \mu(t)\sum_{j=1}^{M} \frac{1}{g_j(X)} \quad (21)$$

3.5. Deb'in Uygulanabilirlik Kuralı (Deb Feasibility Rule)

Deb'in uygulanabilirlik kuralında kısıtlı problemlerin çözümünde uygun veya uygun olmayan bölgeler için bir takım kurallar bulunmaktadır. Problemin çözümündeki her bir bireyin ihlal derecesi Eşitlik 22 kullanılarak hesaplanmaktadır. Hesaplanan değerlerin göre seçim yapılmaktadır. Debin önermiş olduğu kurallar göz önünde bulundurularak uygun olmayan bölgede bulunan tüm çözümler uygun bölgeye doğru hareket ettirilmekte veya yönlendirilmeye çalışılmaktadır.

$$\varphi(X) = \sum_{i=1}^{N}(h_i(X))^2 + \sum_{j=1}^{M}\left(\max\{0, h_j(X)\}\right)^2 \quad (22)$$

Debin önermiş olduğu bu uygulanabilirlik kuralları aşağıdaki gibidir.

1. İki çözümde uygun bölgede olduğu durumda uygunluk fonksiyon değeri küçük olan tercih edilir.
2. Çözümlerden biri uygun ve diğeri uygun olmayan bölgede olduğu durumda uygun bölgedeki çözüm tercih edilir.
3. Her iki çözümde uygun olmayan bölgede olduğu durumda kısıt ihlali az olan çözüm tercih edilir.

4. Deneysel çalışma ve tartışma (Experimental study and discussion)

Bu çalışmada kısıt işlemeye yöntemlerine HŞO ve BOA meta-sezgisel algoritmaları adapte edilmiştir. HŞO ve BOA meta-sezgisel algoritmaların performansını değerlendirmek amacıyla 10 kısıtsız (Zhang vd., 2017) ve 5 kısıtlı (Wu vd., 2017) benchmark fonksiyonu ve mühendislik problemi olan optimal güç akışı (Amaratunga vd., 2018) kullanılmıştır. Bu problemler çok boyutlu, doğrusal, doğrusal olmayan ve ikinci dereceden çeşitli amaç fonksiyonlarını içermektedir. Buna ek olarak kısıtlı fonksiyonların sınırları da doğrusal veya doğrusal olmayan problemlerdir. HŞO ve BOA algoritmaların sürü boyutları 300 ve iterasyon sayıları maksimum 2500 olarak belirlenmiştir. Algoritmaların istatiksel sonuçlarını değerlendirmek amacıyla her bir algoritma ve kısıt işleme teknikleri 20 bağımsız çalışma gerçekleştirilmiştir.

4.1. Deney 1: Benchmark Problemleri (Test 1: Benchmark Problems)

Bu çalışmada ilk olarak sezgisel algoritmalar kısıtsız optimizasyon yöntemlerinde kullanılmış ardından kısıtlı problemlerde kullanılan kısıtlı işleme teknikleri ilave edilerek kısıtlı optimizasyon problemlerine uygulanmıştır. Kullanılan sezgisel HŞO ve BOA algoritmalarının performanslarının istatistiksel olarak karşılaştırılabilmek amacıyla kullanılan kısıtsız benchmark problemlerinin değişkenlerinin aralıkları ve değişkenlerinin boyut değerleri Tablo 2'de verilmiştir (Dzubera et.al. 1996).

BOA ve HŞO algoritmaların kısıtsız benchmark fonksiyonlarına uygulandığında elde edilen ortalama(ort) ve standart sapma(std) sonuçları Tablo 3'de verilmiştir. Tablo 3'de görüldüğü gibi HŞO algoritması BOA'ya göre on problem içerisinde $(F_1, F_2, F_3, F_4, F_5, F_6, F_7, F_8, F_9, F_{10})$ global minimuma daha yakındır. BOA sadece F_7 fonksiyonunda global optimuma yaklaşmıştır.

Tablo 2. Kısıtsız benchmark problemleri (Unconstrained benchmark problems)

Fonksiyon	**Boyut**	**Aralık**
$F_1(x) = -20exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) - exp\left(\frac{1}{n}\sum_{i=1}^{n} cos(2\pi x_i)\right) + 20 + e$	30	[-35,35]
$F_2(x) = \sum_{i=1}^{D} \lvert x_i\, sin(x_i) + 0.1x_i \rvert$	30	[-10,10]
$F_3(x) = \lvert x_1^2 + x_2^2 + x_1x_2 \rvert + \lvert sin(x_1) \rvert + \lvert cos(x_2) \rvert$	2	[-500,500]
$F_4(x) = (1.5 - x_1 + x_1x_2)^2 + (2.25 - x_1 + x_1x_2^2)^2 + (2.625 - x_1 + x_1x_2^3)^2$	2	[-4.5,4.4]
$F_5(x) = 1 - cos\left(2\pi\sqrt{\sum_{i=1}^{D} \lvert x_i^2 \rvert}\right) + 0.1\sqrt{\sum_{i=1}^{D} x_i^2}$	30	[-100,100]
$F_6(x) = (x_1 + 1.7x_2)\, sin(x_1) - 1.5x_3 - 0.1x_4\, cos(x_4 + x_5 - x_1) + 0.2x_5^2 - x_2 - 1$	5	[-100,100]
$F_7(x) = -e^{-0.5}\sum_{i=1}^{D} x_i^2$	30	[-1,1]
$F_8(x) = \left(10000\left\lvert\sum_{i=1}^{D} x_i\right\rvert\right)^{0.5}$	30	[-10,10]
$F_9(x) = \sum_{i=1}^{n} ix_i^4 + random[0,1]$	30	[-1,28, 1,28]
$F_{10}(x) = \sum_{i=1}^{D-1} \lvert 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \rvert$	30	[-30,30]

Tablo 3. Kısıtsız benchmark fonksiyonları için HŞO ve BOA algoritmalarından elde edilen istatiksel sonuçlar (Statistical results obtained from HŞO and BOA algorithms for unconstrained benchmark functions)

		F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}
Optimal		***0***	***0***	***1***	***0***	***0***	***-529,871***	***0***	***0***	***0***	***0***
HŞO	Ort	**8,88E-16**	**1,04E-276**	**1**	**0**	**0**	**-528,351**	**3,06E-07**	**0**	**4,17E-06**	**8,13E-06**
	Std	**0,00E+00**	**0**	**0**	**0**	**0**	2,8238	**1,63E-22**	**0**	**4,18E-06**	**1,33E-05**
BOA	Ort	2,84E-15	0,2969	**1**	2,99E-18	0,1148	-529,83	**3,06E-07**	**0**	0,000137	23,7434
	Std	1,81E-15	1,3279	**0**	3,25E-18	0,0586	**0,02605**	**1,63E-22**	**0**	0,000168	0,2557

Kısıtlı optimizasyonda HŞO ve BOA algoritmalarının performanslarının karşılaştırmak amacıyla kullanılan eşitlik ve eşitsizlik kısıtları içeren benchmark problemlerinin değişkenlerinin aralıkları, değişkenlerinin boyutları ve fonksiyonun minimum değerleri Tablo 3'de verilmiştir. Ölüm ceza, statik ceza, dinamik ceza, bariyer ceza ve Deb kuralları kısıtlama tekniklerine uyarlanmış HŞO ve BOA algoritmalarından elde edilen ortalama, standart sapma deneysel sonuçları ve optimal değerleri Tablo 5'te verilmiştir. Tablo 5'e göre kısıt işleme tekniklerinden ölüm ceza yönteminde beş kısıtlı fonksiyonda da $(F_{11}, F_{12}, F_{13}, F_{14}, F_{15})$ HŞO global optimal sonuca yaklaşmıştır. Statik ceza yöntemindeF F_{11}, F_{12} ve F_{13} kısıtlı fonksiyonlarda en iyi performansa HŞO algoritması sahipken F_{14} ve F_{15} fonksiyonlarında BOA en iyi sonucu vermiştir. Dinamik ceza yönteminde $F_{11}, F_{12}, F_{13}, F_{14}$ ve F_{15} kısıtlı fonksiyonlarda HŞO global optimal yaklaşmıştır. Dinamik ceza yönteminde $F_{11}, F_{12}, F_{13}, F_{14}$ kısıtlı fonksiyonların HŞO algoritması en iyi performans gösterirken F_{15} kısıtlı fonksiyonunda ise her iki algoritmada benzer sonuç vermiştir. Son olarak Deb uygulanabilirlik yönteminde ise sonuç dinamik ceza yöntemiyle aynıdır. Bu sonuçlara ek olarak kısıt işleme yöntemleri içerisinde algoritmaların ortalama ve standart sapma değerleri bakımından performansları karşılaştırıldığında HŞO algoritması daha iyi sonuç verdiği görülmektedir.

4.2. Deney 2: Optimal Güç Akış Problemi (Optimal Power Flow Problem)

Kısıt işleme tekniklerinin performanslarını ölçmek ve karşılaştırmak amacıyla bir çok araştırmacı tarafından bilinen ve optimizasyon literatüründe karşılaşılan yaygın bir problem olan Optimal Güç Akışı (OGA) ele alınmıştır. OGA problemindeki karmaşık güç akışı eşitlikleri, eşitsizlik kısıtlamaları, enerji sistemindeki fiziksel kısıtlamaları gibi kısıtların fazla olmasından kaynaklanmaktadır. Bu karmaşıklık eşitlik veya eşitsizlik olabileceğinden bu değerler enerji üretim maliyetini etkilemektedir. OGA probleminde kısıtlar sağlanarak yakıt maliyetini minimum olacak şekilde minimizasyon gerçekleştirilecektir. Yakıt maliyetini minimum olarak hesaplanmasında ölüm ceza, statik ceza, dinamik ceza, bariyer ceza ve Deb uygulanabilirlik kısıtlama tekniklerine uyarlanmış HŞO ve BOA algoritmaları uyarlanmıştır.

Bu problemde literatürde çokça karşılaşılan IEEE30 baralı optimal güç akışı problemi ele alınmıştır. Bu problem amaç fonksiyonu, kısıtlar ve ağ yapısına bağlı olacak şekilde farklı şekilde modellenerek kullanılabilir. Bu çalışmada toplam yakıt maliyetini en aza indirecek şekilde Denklem 23'te görüldüğü şekilde matematiksel modeli çıkartılmıştır. Denklem 23'teki n sistemdeki toplam jeneratör sayısını, P_i i. bara da üretilen aktif güçleri, a_i, b_i ve c_i jeneratör yakıt maliyeti katsayılarını göstermektedir.

$$\begin{gathered}\text{minimize} \quad \sum_{i=1}^{N} F_i(P_i) \\ F_i(P_i) = a_i P_i^2 + b_i P_i + c_i \\ P_i^{\min} \le P_i \le P_i^{\max}\end{gathered} \tag{23}$$

Üretilecek olan güç (P_i), tüketilecek güç (D) ve iletim kayıplarının (P_l) toplamı olmalıdır. Dolayısıyla iletim kayıplarının hesaplanması Denklem 24 ile hesaplanmaktadır.

$$\begin{gathered}\sum_{i=1}^{N} P_i = D + P_l \\ P_i = real\left(\sum_{j}^{N} V_i Y_{ij} \overline{V}_j\right) \quad i = 1,2,...,N \\ Q_i = imag\left(\sum_{j}^{N} V_i Y_{ij} \overline{V}_j\right) \quad i = 1,2,...,N\end{gathered} \tag{24}$$

Tablo 4. Kısıtlı benchmark problemleri (Constrained bencmark problems)

Kısıtlı Fonksiyon Eşitlikleri	Boyut	F_{min}
$F_{11}(x) = 5\sum_{i=1}^{4} x_i - 5\sum_{i=1}^{4} x_i^2 - \sum_{i=5}^{13} x_i$ $g_1(x) = 2x_1 + 2x_2 + x_{10} + x_{11} - 10 \le 0$ $\quad g_2(x) = 2x_1 + 2x_3 + x_{10} + x_{11} - 10 \le 0$ $g_3(x) = 2x_2 + 2x_3 + x_{11} + x_{12} - 10 \le 0$ $\quad g_4(x) = -8x_1 + x_{10} \le 0$ $g_5(x) = -8x_2 + x_{11} \le 0$ $\quad g_6(x) = -8x_3 + x_{12} \le 0$ $g_7(x) = -2x_4 - x_5 + x_{10} \le 0$ $\quad g_8(x) = -2x_6 - x_7 + x_{11} \le 0$ $g_9(x) = -2x_8 - x_9 + x_{12} \le 0$ $\quad 0 \le x_i \le 1 \quad i = 1,2,\ldots,9$ $0 \le x_i \le 100 \quad i = 10,11,12$ $\quad 0 \le x_{13} \le 1$	13	-15
$F_{12}(x) = -(\sqrt{n})^n \prod_{i=1}^{n} x_i$ $h_1(x) = \sum_{i=1}^{n} x_i^2 - 1 = 0 \qquad 0 \le x_i \le 1 \quad i = 1,2,\ldots,10$	10	-1.0005000
$F_{13}(x) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1$ $g_1(x) = 85.334407 + 0.0056858x_1x_5 + 0.0006262x_1x_4 - 0.0022053x_3x_5 - 92 \le 0$ $g_2(x) = -85.334407 - 0.0056858x_1x_5 - 0.0006262x_1x_4 + 0.0022053x_3x_5 \le 0$ $g_3(x) = 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2 - 110 \le 0$ $g_4(x) = -80.51249 - 0.0071317x_2x_5 - 0.0029955x_1x_2 - 0.0021813x_3^2 + 90 \le 0$ $g_5(x) = 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.0019085x_3x_5 - 25 \le 0$ $g_6(x) = -9.300961 - 0.0047026x_3x_5 - 0.0012547x_1x_3 - 0.0019085x_3x_5 + 20 \le 0$ $78 \le x_1 \le 102$ $33 \le x_2 \le 45$ $27 \le x_i \le 45 \quad i = 2,3,5$	5	-3.06655×10^4
$F_{14}(x) = (x_1 - 10)^3 + (x_2 - 20)^3$ $g_1(x) = -(x_1 - 5)^2 + (x_2 - 5)^2 + 100 \le 0$ $g_2(x) = -(x_1 - 6)^2 + (x_2 - 5)^2 - 82.81 \le 0$ $13 \le x_1 \le 100$ $0 \le x_2 \le 100$	2	-6961.813875
$F_{15}(x) = (x_1)^2 + (x_2 - 1)^2$ Subject to $h_1(x) = x_2 - x_1^2 \quad -1 \le x_1, x_2 \le 1$	2	0.7499

Tablo 5 . Kısıtlı Benchmark Problemlerinin Optimizasyon Sonuçları (Optimization Results of Constrained Benchmark Problems)

Algoritma	Kısıt	Kriter	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}
			-15	***-1.0005***	***$-3.06655x10^{4}$***	***-6961.813875***	***0.7499***
BOA	Ölüm Ceza	Ort	-9,92257	-0,00329	-30375,4	-6963,21	0,749001
		Std	3,051379	0,014556	182,6252	1,407344	6,48E-07
HŞO		Ort	***-14,5482***	***-0,10744***	***-30580***	***-6964,14***	***0,749015***
		Std	0,878496	0,090828	144,3007	1,14E-06	4,39E-05
BOA	Statik	Ort	-9,24107	-1,86E-68	-30413,4	***-6963,47***	***7,49E-01***
		Std	2,054122	0,057546	168,4201	0,46082	7,01E-04
HŞO		Ort	***-14,7276***	***-0,01824***	***-30561,8***	-6961,81	1,00E+00
		Std	0,444701	4,55E-68	173,8887	4,13E-06	8,82E-17
BOA	Dinamik	Ort	-9,44737	-7,67E-69	-30429,3	-6961,24	9,73E-01
		Std	2,916546	1,71E-68	156,9226	0,364233	6,92E-02
HŞO		Ort	***-14,8243***	***-8,72E-03***	***-30640,3***	***-6961,81***	***9,11E-01***
		Std	0,218548	2,33E-02	53,86185	7,27E-07	9,99E-02
BOA	Bariyer	Ort	-7,43006	-0,00114	-30448,8	-6961,33	***1,00E+00***
		Std	4,990804	0,004272	150,9557	0,284288	0,00E+00
HŞO		Ort	**-14,8398**	***-0,00226***	***-30560,6***	***-6961,81***	***1,00E+00***
		Std	0,102318	0,010125	161,9542	2,03E-06	5,09E-17
BOA	Deb	Ort	-9,51114	-0,01077	-29725,3	-5607,19	***1,00E+00***
		Std	2,122866	0,026659	254,2305	2411,676	0
HŞO		Ort	***-11,6678***	-***0,01841***	***-30526,9***	***-6961,8***	***1,00E+00***
		Std	1,962415	0,05898	106,9022	0,042547	0

OGA yakıt maliyeti kısıt problemi, kısıt işleme tekniklerine uyarlanmış HŞO ve BOA algoritmaları ile çözümünden elde edilen minimum, ortalama, standart sapma deneysel sonuçları Tablo 6'da verilmiştir.

Tablo 6. OGA'da yakıt maliyet minimizasyonunun istatiksel sonuçlarının karşılaştırılması ($/sa) (Comparison of statistical results of fuel cost minimization in OPF)

Kısıt Yöntemi	HŞO			BOA		
	Min. ($/sa)	Ort. ($/sa)	Std	Min. ($/sa)	Ort. ($/sa)	Std
Ölüm Ceza	800.40	**805.0**	6.600	805.39	813.00	7.14
Statik Ceza	799.92	**804.0**	5.750	801.72	810.00	6.77
Dinamik Ceza	799.96	**802.0**	2.120	805.26	813.00	6.35
Bariyer Ceza	799.94	**802.0**	2.120	801.78	810.00	4.77
Deb Kuralı	800.37	**803.0**	1.908	801.82	805.38	2.24

Tablo 6'a göre OGA kısıt probleminde ölüm ceza, statik ceza, dinamik ceza, bariyer ceza ve Deb uygulanabilirlik kuralı kısıt işleme tekniklerinde HŞO global optimale yaklaşmıştır. Buna ek olarak kısıt işleme tekniklerinden HŞO algoritmasında dinamik ceza ve bariyer ceza, BOA algoritmasında ise Deb uygulanabilirlik kuralı daha etkili olduğu görülmüştür.

BOA ve HŞO algoritmaların optimal güç akışında gösterdikleri performanslarını karşılaştırmak amacıyla literatürde bulunan HŞO (Islam vd., 2020), Gri Kurt Optimizasyon (GKO) (Islam vd., 2020), BOA (Islam vd., 2020), GA (Bouktir vd., 2008), Harris Şahini-Diferansiyel Evrim (HŞODE) (Birogul, 2019) ve Modifiyeli diferansiyel evrim (Sayah vd., 2008) algoritmaları kullanılmıştır. Çalışmada kullanılan BOA ve HŞO algoritmaları literatürdeki algoritmalarla karşılaştırıldığında HŞO algoritması en iyi değer bakımından daha iyi sonuç vermiştir. Bunun nedeni bu algoritmaların rastlantısal olarak çalışması ve belirlenen iterasyon ve sürü sayıları farklı olması ve genellikle kısıt işleme tekniklerinin ve parametrelerinin değişebilmesinden kaynaklanmaktadır. Bu doğrultuda bağıl bir değerlendirme yapıldığında literatürde genellikle kısıtlı problemler ölüm, statik ve dinamik ceza fonksiyonları çözümlenmektedir.

Tablo 7. Optimal güç akışı için HŞO, GKO, BOA, GA, HŞODE ve MDA algoritmalarıyla elde edilen optimal çözümler (The minimum solutions obtained by the HŞO, GKO, BOA, GA, HŞODE, and MDA algorithms for optimal power flow)

Algoritma	Minimum Değer ($/sa)
HŞO (Islam vd., 2020)	801.8290
GKO (Islam vd., 2020)	801.8440
WOA (Islam vd., 2020)	801.8400
GA (Bouktir vd., 2008)	804.1000
HHODE (Birogul 2019)	800.9959
MDE (Sayah vd., 2008)	802.3700

Fakat Tablo 6'da görüleceği üzere bariyer ceza fonksiyonu ortalama olarak daha iyi sonuç vermiştir. Lakin bu çalışmada yapılan denemelerde sadece minimum değerler baz alınarak kıyas yapıldığında ise Tablo 6 ve Tablo 7 incelendiğinde Tablo 6'da bulunan HŞO ile kullanılan Statik Ceza fonksiyonu literatürdeki çalışmalardan daha iyi sonuç verdiği görülmektedir.

5. Sonuçlar (Results)

Bu çalışmada doğadan ilham alan meta-sezgisel optimizasyon algoritmaları için etkili bir kısıt işleme teknikleri sunmaktadır. Ölüm cezası, statik ceza, dinamik ceza, bariyer ceza ve Deb kuralları kısıtlama teknikleriyle BOA ve HŞO algoritmaları kısıtsız ve kısıtlı optimizasyon problemlerinin optimize edilmesinde kullanılmıştır ve performansları karşılaştırılmıştır. Ayrıca bu algoritmaların performansları literatürde var olan IEEE30 baralı optimal güç akışı minimizasyon problemine başarıyla uygulanmıştır. Deneysel sonuçlara göre kısıtlı ve kısıtsız benchmark problemlerinde ve optimal güç akışı mühendislik probleminde HŞO algoritmasının performansı daha iyi olduğu istatistiksel olarak görülmüştür. Ayrıca kısıt işleme yöntemlerinin BOA ve HŞO algoritmaları üzerinde efektif olduğu görülmüştür.

Kaynaklar (References)

Akay B., Karaboga D., 2011. A modified Artificial Bee Colony (ABC) algorithm for constrained optimization problems. Applied Soft Computing, 11, 3021–3031.

Aljarah, I., Chen, H., Faris, H., Heidari, A. A., Mafarja, M., Mirjalili, S. 2019. Harris hawks optimization: Algorithm and applications. Future Generation Computer Systems, 97, 849–872.

Amaratunga, G.A.J., Biswas, P.P., Mallipeddi, R., Suganthan P.N., 2018.Optimal power flow solutions using differential evolution algorithm integrated with effective constraint handling techniques. Engineering Applications of Artificial Intelligence, 68, 81–100.

Babalik, A., Cinar, A.C., Kiran M.S., 2018, A modification of tree-seed algorithm using Deb's rules for constrained optimization. Applied Soft Computing, 63, 289–305.

Batık, Z. G., Boz, A. F., Çimen, M.E., Karayel D.2019. The Chaos-Based Whale Optimization Algorithms Global Optimization, Chaos Theory and Applications, 1, 51-63.

Birogul, S. 2019. Hybrid harris hawk optimization based on differential evolution (HHODE) algorithm for optimal power flow problem. IEEE Access, 7, 184468-184488.

Bouktir, T.,Slimani, L., Mahdad, B. 2008. Optimal power dispatch for large scale power system using stochastic search algorithms. International Journal of Power and Energy Systems, 28(2), 118.

Chen S., Gu Y., Jiang, S., Nouioua, M., Li Z., Zhang, S., 2019. FSB-EA: Fuzzy search bias guided constraint handling technique for evolutionary algorithm. Expert Systems with Applications, 119, 20–35.

Chen, H., Wang, M., Zha X., 2020. A multi-strategy enhanced sine cosine algorithm for global optimization and constrained practical engineering problems. Applied Mathematics and Computation, 369, 124872.

Deb, K., Gandomi, A.H., 2020 .Implicit constraints handling for efficient search of feasible solutions. Comput. Methods Appl. Mech. Engrg. 363, 112917.

Dzubera, J., Mathias, K., Rana S., D. Whitley, 1996. Evaluating Evolutionary Algorithms. Artif Intell, 85, 245-276.

Fan, Q.-W., He, X.-S., Karamanoglu M. and Yang, X.-S., 2019. Comparison of Constraint-Handling Techniques for Metaheuristic Optimization. ICCS 2019: Computational Science, 357-366.

Garcia R., Jacob, B. P., Lemonge, A., Lima, B., 2017. A rank-based constraint handling technique for engineering design optimization problems solved by genetic algorithms. Computers and Structures, 187, 77–87.

Guimarães, S., Lima, B., Rodrigues, M., 2018. E-BRM: A constraint handling technique to solve optimization problems with evolutionary algorithms. Applied Soft Computing, 72, 14–29.

Haklı H., 2019. A novel approach based on elephant herdıng optımızatıon for constraıned optımızatıon problems. Selcuk Univ. J. Eng. Sci. Tech., 7,405-419.

Hatamlou, A., Mirjalili, S., Mirjalili, S. M., 2016. Multi-verse optimizer: a nature-inspired algorithm for global optimization. Neural Computing and Applications, 27, 495-513.

He W., Peng, X., Peng, X., Qu, C. 2020. Harris Hawks optimization with information Exchange, Applied Mathematical Modelling, 84, 52–75.

Islam, M.Z., Abdul Wahab, N.I., Veerasamy, V., Hizam, H., Mailah, N.F., Guerrero, J.M., Mohd Nasir M.N., 2020. A Harris Hawks Optimization Based Single-and Multi-Objective Optimal Power Flow Considering Environmental Emission. Sustainability, 12(13), 5248.

Jelovic, J., Samanipour, F., 2020. Adaptive repair method for constraint handling in multi-objective genetic algorithm based on relationship between constraints and variables. Applied Soft Computing Journal, 90, 106-143.

Kohler, M., Tanscheit, R., Vellasco, M.M.B.R., 2019. PSO+: A new particle swarm optimization algorithm for constrained problems. Applied Soft Computing, 85, 105865.

Lewis, A., Mirjalili, S., 2016. The whale optimization algorithm. Advances in Engineering Software, 95, 5167.

Mert, A., Tezel, B.T., 2021. A cooperative system for metaheuristic algorithms. Expert Systems with Applications, 165, 113976.

Miranda-Varela, M.,, Mezura-Montes, E. 2018. Constraint-handling techniques in surrogate-assisted evolutionary optimization. An empirical study. Applied Soft Computing Journal 73, 215–229.

Sayah, S., Zehar, K. 2008. Modified differential evolution algorithm for optimal power flow with non-smooth cost functions. Energy conversion and Management, 49(11),3036-3042.

Tuba, M. and Bacanin, N., 2014. Improved seeker optimization algorithm hybridized with firefly algorithm for constrained optimization problems. Neuro computing, 143, 197-207.