KARADENİZ TEKNİK ÜNİVERSİTESİ OF TEKNOLOJİ FAKÜLTESİ YAZILIM MÜHENDİSLİĞİ BÖLÜMÜ

BELUGA WHALE OPTİMİZATİON (BWO) ALGORİTMASININ OPTİMİZASYONU

META-SEZGİSEL OPTİMİZASYON DERSİ

PROJE ÖDEVİ RAPORU

Elif Beyza TOK

2023-2024 BAHAR DÖNEMİ

İÇİNDEKİLER

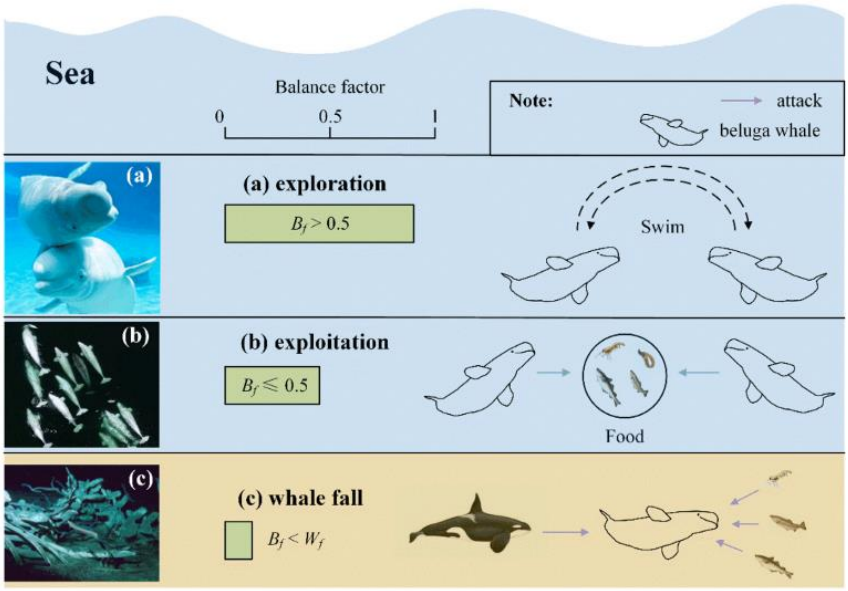
1. Giriş

1.1. Meta Sezgisel Optimizasyon (MSO) Nedir?

Meta sezgisel optimizasyon (MSO), karmaşık ve büyük ölçekli optimizasyon problemlerini çözmek için kullanılan üst düzey stratejilerdir. MSO, doğrudan çözüm üretmek yerine, çözüm alanında etkili bir şekilde arama yaparak en iyi ya da en uygun çözümleri bulmayı amaçlar. Bu yöntemler, problemin doğası hakkında minimal bilgi gerektirir ve genellikle büyük bir çözüm alanında gezinirken yerel optimumlara takılmadan global optimuma ulaşmayı hedefler. Meta sezgisel optimizasyon yöntemleri, klasik optimizasyon tekniklerinin yetersiz kaldığı durumlarda kullanılır. Özellikle, büyük ve karmaşık problemlerin çözümünde etkili olurlar.

1.2. Beluga Whale Optimization (BWO)Algoritması

Beluga Whale Optimization (BWO) algoritması, beluga balinalarının davranışlarından ilham alan bir sürü tabanlı metaheuristik algoritmadır. BWO, optimizasyon problemlerini çözmek için üç ana faza dayanır: keşif, sömürü ve balina düşüşü. Bu fazlar sırasıyla çift yüzme, avlanma ve balina düşüşü davranışlarına karşılık gelir.



Şekil 1. Behaviors of beluga whales, (a) swim, corresponding to exploration phase; (b) foraging, corresponding to exploitation phase, (c) whale fall, for whale fall phase. **Source:** https://constative.com/animals/whale-watching, https://www.sealuxe.ca/blog/narwhal, **Ref.** [3].

Keşif Fazı (Çift Yüzme) - Exploration Phase (Pair Swimming):

Bu faz, algoritmanın küresel arama uzayını keşfetmesini sağlar.

Çift yüzme davranışı, arama sürecinin çeşitlendirilmesine yardımcı olur ve algoritmanın yerel optimuma takılmasını önler.

Sömürü Fazı (Avlanma) - Exploitation Phase (Preying):

Bu faz, umut verici bölgelerde yoğun arama yaparak çözümleri iyileştirmeye odaklanır.

Avlanma davranışı, algoritmanın belirlenen iyi çözümleri daha iyi çözümler bulmak için kullanmasını sağlar.

Bu fazda Levy uçuş mekanizması kullanılır, bu da küresel yakınsamayı artırır ve arama sürecini daha verimli hale getirir.

Balina Düşüşü Fazı - Whale Fall Phase:

Bu faz, bir balina karkasının okyanus tabanına düşüşünü simüle eder ve zengin bir ekosistem oluşturur.

Hem keşif hem de sömürü unsurlarını birleştirerek arama sürecini dengeler ve yerel optima kaçma yeteneğini artırır.

Keşif ve sömürüyü dinamik olarak kontrol etmek için denge faktörü ve balina düşüş olasılığı kendiliğinden uyarlanabilir mekanizmalarla ayarlanır.

Algorithm 1: The pseudo code of BWO algorithm

Input: Algorithmic parameters (population size, maximum iteration)

Output: The best solution

Initialize the population and evaluate fitness values, obtain the best solution (P*)
While $T \le T_{max}$ **Do**
  Obtain probability of whale fall W_f by Eq. (10) and balance factor B_f by Eq. (3)
  For each beluga whale (X_i) **Do**
    If $B_f(i) > 0.5$
      // In the exploration phase
      Generate p_j (j = 1,2,…,d) randomly from dimension
      Choose a beluga whale X_r randomly
      Update new position of i-th beluga whale using Eq. (4)
    Else If $B_f(i) \le 0.5$
      // In the exploitation phase
      Update the random jump strength C_1 and calculate the Levy flight function
      Update new position of i-th beluga whale using Eq. (5)
    End If
    Check the boundaries of new positions and evaluate the fitness values
  End For
  For each beluga whale (X_i) **Do**
    // the whale fall phase
    If $B_f(i) \le W_f$
      Update the step factor C_2
      Calculate the step size X_{step}
      Update new position of i-th beluga whale using Eq. (8)
      Check the boundaries of new position and calculate fitness value
    End If
  End For
  Find the current best solution P^*
  $T = T+1$
End While
Output the best solution

Şekil.2. The pseudo code of BWO algorithm **Source:**https://constative.com/animals/whale-watching, https://www.sealuxe.ca/blog/narwhal, **Ref. [1]**

1.3. Beluga Whale Optimization (BWO)Algoritmasında Kullanılan Denklemler

Test ve analiz aşamasında yapılan değişiklikleri anlamak için BWO kodlarında kullanılan önemli denklemleri inceleyelim.

BWO (Beluga Whale Optimization) algoritmasının yakınsaması, keşif ve sömürü aşamaları arasında geçiş yapan bir denge faktörü (Bf) ile kontrol edilir. Yakınsama süreci, denge faktörü denklemi ve her iki aşamadaki iteratif güncellemelerle yönlendirilir.

Yakınsama Kontrol Denklemi

$$Bf = B0\left(1 - \frac{T}{2 \cdot T_{\max}}\right)$$

- **B0,** 0 ile 1 arasında rastgele bir sayıdır.
- **T,** mevcut iterasyondur.
- **TmaxT,** maksimum iterasyon sayısıdır.

Denge faktörü **Bf**, algoritmanın keşif veya sömürü aşamasında olup olmadığını belirler:

- **Keşif Aşaması:** Bf > 0.5 olduğunda.
- **Sömürü Aşaması:** Bf ≤ 0.5 olduğunda.

İterasyon ilerledikçe (**T** arttıkça), **Bf** azalır ve algoritma keşiften sömürüye geçer.

Keşif Aşaması

Keşif aşamasında, arama ajanlarının (beluga balinalarının) konumları aşağıdaki denklemlerle güncellenir:

$$X_{i,j}^{T+1} = \begin{cases} X_{i,pj}^{T} + \left(X_{r,p1}^{T} - X_{i,pj}^{T}\right)(1 + r1)\sin(2\pi r2), & \text{eğer } j \text{ çiftse} \\ X_{i,pj}^{T} + \left(X_{r,p1}^{T} - X_{i,pj}^{T}\right)(1 + r1)\cos(2\pi r2), & \text{eğer } j \text{ tekse} \end{cases}$$

- $X_{i,j}^{T+1}$ **,** i-inci balinanın jjj-inci boyuttaki yeni konumudur.
- $X_{i,pj}^{T}$**,** i-inci balinanın pjpjpj-inci boyuttaki mevcut konumudur.
- $X_{r,p1}^{T}$**,** rastgele seçilen rrr balinasının p1p1p1-inci boyuttaki mevcut konumudur.
- **r1** ve **r2**, 0 ile 1 arasında rastgele sayılardır.
- Sinüs ve kosinüs fonksiyonları, balinaların senkronize veya aynalı yüzme davranışlarını simüle eder.

Sömürü Aşaması

Sömürü aşamasında, konumlar Levy uçuşunu kullanarak av yakalama davranışını simüle etmek için güncellenir:

$$X_i^{T+1} = r3X_{\text{best}}^T - r4X_i^T + C1 \cdot LF \cdot \left(X_r^T - X_i^T\right)$$

- X_i^{T+1}**, i** -inci balinanın yeni konumudur.
- X_i^T**,** i-inci balinanın mevcut konumudur.
- X_{best}^T**,** şimdiye kadar bulunan en iyi konumdur.
- X_r^T **,** rastgele seçilen bir balinanın mevcut konumudur.
- **r3 ve r4,** 0 ile 1 arasında rastgele sayılardır.
- $C1 = 2r4\left(1 - \frac{T}{T_{\max}}\right)$**,** rastgele sıçrama kuvvetidir.
- **LF,** Levy uçuşu fonksiyonudur.

Levy uçuşu fonksiyonu LF şu şekilde tanımlanır:

$$LF = 0.05 \cdot \frac{u \cdot \sigma}{|v|^{1/\beta}} \qquad \sigma = \left(\frac{\Gamma(1+\beta) \cdot \sin(\pi\beta/2)}{\Gamma((1+\beta)/2) \cdot \beta \cdot 2^{(\beta-1)/2}}\right)^{1/\beta}$$

Burada u ve v normal dağılıma sahip rastgele sayılardır ve β\betaβ sabit bir değerdir (genellikle 1.5).

Balina Düşüşü Aşaması

Ara sıra başarısızlığı (balina düşüşü) modellemek ve nüfus boyutunu korumak için, konum şu şekilde güncellenir:

$$X_i^{T+1} = r5X_i^T - r6X_r^T + r7X_{\text{step}}$$

- **r5, r6 ve r7**, 0 ile 1 arasında rastgele sayılardır.
- **Xstep**, adım boyutu olarak hesaplanır:

$$X_{\text{step}} = (ub - lb)\exp\left(-\frac{C2T}{T_{\max}}\right)$$

$C2 = 2W_f \times nu$(adım faktörü), ub ve lb değişkenlerin üst ve alt sınırlarıdır.

Balina düşüşü olasılığı W_f şu şekilde hesaplanır:

$$W_f = 0.1 - 0.05\frac{T}{T_{\max}}$$

Algoritmanın Sonlandırılması

Algoritma maksimum iterasyon sayısı **Tmax** a ulaşıldığında sonlanır. BWO algoritmasının yakınsama süreci, dinamik denge faktörü ve iteratif güncellemeler aracılığıyla keşif ve sömürü arasında etkili bir şekilde denge kurarak optimal bir arama yolu sağlar.

2. Amaç

Bu çalışmada Beluga Whale Optimization (BWO) algoritmasını optimize etmek amaçlanmıştır. Bu optimizasyon işlemini yaparken sürü tabanlı metaheuristik BWO algoritmasının rehberlik seçimi aşamasında kullanılan değişken ve parametrelerin seçim yöntemlerinde değişiklikler yapılmıştır. Rastgele seçim yerine Fitness Distance Balance (FDB) algoritması ve türevleri kullanılmıştır.

3. Yöntem

3.1. BWO Algoritması ve Rehber Seçimi

Orijinal Beluga Whale Optimization (BWO) algoritmasında rehber seçim yöntemi, keşif ve sömürü fazlarında rastgele seçilen bireyler (diğer çözüm adayları) kullanılarak gerçekleştirilir. Bu yöntemin amacı, algoritmanın farklı çözüm alanlarını tarayarak daha iyi çözümler bulmasını sağlamaktır.

Rehber seçim yöntemini kısaca şöyle özetleyebiliriz:

1.Keşif Fazı (Exploration Phase):

Rastgele Seçim: Keşif fazında, her birey için popülasyon içerisinden rastgele bir birey (RJ) seçilir. Bu birey, yeni pozisyonların belirlenmesinde referans olarak kullanılır.

Parametre Güncelleme: Rastgele seçilen birey (RJ) kullanılarak mevcut bireyin pozisyonları, sin ve cos fonksiyonları ile güncellenir.

2.Sömürü Fazı (Exploitation Phase):

Rastgele Seçim ve En İyi Çözüm Kullanımı: Sömürü fazında da benzer şekilde, her birey için popülasyon içerisinden rastgele bir birey (RJ) seçilir. Ayrıca, en iyi çözüm (xposbest) de bu süreçte rehber olarak kullanılır.

Levy Uçuşu: Yeni pozisyonlar belirlenirken, Levy uçuş mekanizması ile rastgele yürüyüşler gerçekleştirilir.

3.Balina Düşüşü Fazı (Whale Fall Phase):

Rastgele Seçim ve Adım Boyutu: Balina düşüşü fazında, her birey için yine rastgele bir birey (RJ) seçilir ve adım boyutu ile pozisyonlar güncellenir.

3.2. Değişikliklerin Tanımlanması

Bu çalışmada, Beluga Whale Optimization (BWO) algoritmasında rehber seçiminde kullanılan değişkenlerin seçim yöntemleri değiştirilerek performans iyileştirmeleri hedeflenmiştir. Bu değişiklikler, algoritmanın farklı parametrelerinin seçiminde fitnessDistanceBalance ve rouletteFitnessDistanceBalance yöntemlerinin kullanılmasını içerir. Her bir case, belirli bir parametrenin seçim yöntemini değiştirmektedir. Aşağıda, yapılan değişiklikler ve ilgili kod parçaları detaylıca verilmiştir.

Case'lerde Kullanılan Algoritmalar:

fitnessDistanceBalance(pos, fit): Bu algoritma, popülasyonun pozisyon ve fitness değerlerine dayalı olarak rehber seçiminde kullanılır. Fitness ve mesafe dengesi göz önüne alınarak en uygun çözüm belirlenir.

rouletteFitnessDistanceBalance(pos , fit): Bu algoritma, bir rulet çarkı seçimi ile fitness ve mesafe dengesini birleştirir. Rastgele bir seçim yapılarak rehber belirlenir.

Case'lerde Değiştirilen Değişkenler:

RJ: Rastgele seçilen rehber çözüm.

r1, r7, alpha, C1, C2, r5: BWO algoritmasında kullanılan farklı kontrol ve keşif parametreleri.

3.3. Case Tanımları

Her bir case'de kullanılan değişkenlerin nasıl değiştirildiği ve bunun algoritmanın performansına olan etkisi inceleyelim.

Case 1: RJ Seçim Yöntemi fitnessDistanceBalance ile Değiştirildi

Özellikler:

- **Değiştirilen Değişken:** RJ
- **Seçim Yöntemi:** fitnessDistanceBalance
- **Kullanım Alanı:** RJ, BWO algoritmasında rehber çözüm olarak rastgele seçilen bir çözümü temsil eder.

Farklılıklar:

- RJ değişkeni, rastgele bir seçim yerine fitness ve mesafe dengesi göz önüne alınarak seçilmektedir.
- Bu değişiklik, rehber çözümün daha uygun bir çözüm olmasını sağlayarak algoritmanın arama kabiliyetini artırabilir.

```
        RJ = ceil(Npop*rand);   % Roulette Wheel Selection
        RJ = fitnessDistanceBalance(pos, fit);
```

Case 2: r1 Seçim Yöntemi fitnessDistanceBalance ile Değiştirildi

Özellikler:

- **Değiştirilen Değişken:** r1
- **Seçim Yöntemi:** fitnessDistanceBalance
- **Kullanım Alanı:** r1, BWO algoritmasında keşif fazında kullanılan bir kontrol parametresidir.

Farklılıklar:

- r1 değişkeni, rastgele bir değer yerine fitness ve mesafe dengesi göz önüne alınarak seçilmektedir.
- Bu değişiklik, keşif fazında daha iyi çözümler bulunmasına yardımcı olabilir.

```
                r1 = rand();
                r1 = fitnessDistanceBalance(pos, fit);
```

Case 3: r7 Seçim Yöntemi fitnessDistanceBalance ile Değiştirildi

Özellikler:

- **Değiştirilen Değişken:** r7
- **Seçim Yöntemi:** fitnessDistanceBalance
- **Kullanım Alanı:** r7, BWO algoritmasında keşif fazında kullanılan bir kontrol parametresidir.

Farklılıklar:

- r7 değişkeni, rastgele bir değer yerine fitness ve mesafe dengesi göz önüne alınarak seçilmektedir.
- Bu değişiklik, keşif fazında daha iyi çözümler bulunmasına yardımcı olabilir.

```
                r7 = rand();
                r7 = fitnessDistanceBalance(pos, fit);
```

Case 4: alpha Seçim Yöntemi fitnessDistanceBalance ile Değiştirildi

Özellikler:

- **Değiştirilen Değişken:** alpha
- **Seçim Yöntemi:** fitnessDistanceBalance
- **Kullanım Alanı:** alpha, BWO algoritmasında Levy uçuşu hesaplamalarında kullanılan bir parametredir.

Farklılıklar:

- alpha değişkeni, sabit bir değer yerine fitness ve mesafe dengesi göz önüne alınarak seçilmektedir.
- Bu değişiklik, Levy uçuşunun etkinliğini artırabilir ve daha iyi çözümler bulunmasına yardımcı olabilir.

```
alpha=3/2;
alpha=fitnessDistanceBalance(pos, fit);
```

Case 5: C1 Seçim Yöntemi fitnessDistanceBalance ile Değiştirildi

Özellikler:

- **Değiştirilen Değişken:** C1
- **Seçim Yöntemi:** fitnessDistanceBalance
- **Kullanım Alanı:** C1, BWO algoritmasında keşif ve sömürü fazlarında kullanılan bir kontrol parametresidir.

Farklılıklar:

- C1 değişkeni, rastgele bir değer yerine fitness ve mesafe dengesi göz önüne alınarak seçilmektedir.
- Bu değişiklik, keşif ve sömürü fazlarının etkinliğini artırabilir.

```
C1 = 2*r4*(1-T/Max_it);
C1 = fitnessDistanceBalance(pos, fit);
```

Case 6: C2 Seçim Yöntemi fitnessDistanceBalance ile Değiştirildi

Özellikler:

- **Değiştirilen Değişken:** C2
- **Seçim Yöntemi:** fitnessDistanceBalance
- **Kullanım Alanı:** C2, BWO algoritmasında keşif ve sömürü fazlarında kullanılan bir kontrol parametresidir.

Farklılıklar:

- C2 değişkeni, sabit bir değer yerine fitness ve mesafe dengesi göz önüne alınarak seçilmektedir.
- Bu değişiklik, keşif ve sömürü fazlarının etkinliğini artırabilir.

```
C2 = 2*Npop*WF;
C2 = fitnessDistanceBalance(pos, fit);
```

Case 7: r5 Seçim Yöntemi rouletteFitnessDistanceBalance ile ve C2 Seçim Yöntemi fitnessDistanceBalance ile Değiştirildi

Özellikler:

- **Değiştirilen Değişkenler:** r5 ve C2
- **Seçim Yöntemleri:** r5 için rouletteFitnessDistanceBalance, C2 için fitnessDistanceBalance
- **Kullanım Alanı:** r5 ve C2, BWO algoritmasında keşif fazında kullanılan kontrol parametreleridir.

Farklılıklar:

- r5 değişkeni, rastgele bir değer yerine rulet çarkı seçimi ile fitness ve mesafe dengesi göz önüne alınarak seçilmektedir.
- C2 değişkeni, sabit bir değer yerine fitness ve mesafe dengesi göz önüne alınarak seçilmektedir.
- Bu değişiklikler, keşif fazının etkinliğini ve çeşitliliğini artırabilir.

```
            r5 = rand();
            r5 = rouletteFitnessDistanceBalance(pos , fit);

            C2 = 2*Npop*WF;
            C2 = fitnessDistanceBalance(pos, fit);
```

4. Deneysel Çalışmalar

4.1. Deneysel Kurulum

Deneylerin Yapıldığı Ortam ve Kullanılan Araçlar

Yazılım Ortamı: Matlab programlama dili kullanılarak gerçekleştirilmiştir..

Donanım Ortamı: Intel Core i7 işlemcili, 16GB RAM ve 512GB SSD kapasiteli bir bilgisayar kullanılmıştır.
Geliştirme Ortamı: MATLAB R2024a (IDE), algoritma geliştirme ve deneylerin yürütülmesinde kullanılmıştır.

Kullanılan Veri Setleri ve Test Problemleri

Deneylerde, optimizasyon algoritmalarının performanslarını değerlendirmek amacıyla sıkça kullanılan CEC 2022 test problemleri seti kullanılmıştır. Bu set, farklı zorluk seviyeleri ve özellikler taşıyan 12 test probleminden oluşmaktadır:

1. Sphere
2. Schwefel
3. Rosenbrock
4. Rastrigin
5. Griewank
6. Ackley

7. Penalized
8. Penalized 2
9. Katsuura
10. Happy Cat
11. HGBat
12. Expanded Griewank plus Rosenbrock

Bu problemler, algoritmanın çeşitli yönlerden test edilmesini sağlar ve farklı optimizasyon senaryolarını temsil eder.

4.2. Performans Ölçütleri

Başarı Kriterleri ve Performans Ölçütleri

Deneylerde, BWO algoritmasının performansını değerlendirmek için çeşitli ölçütler kullanılmıştır. Bu ölçütler şunlardır:

Fitness Fonksiyonu: Test problemleri için belirlenen hedef fonksiyonun en iyi (minimum veya maksimum) değerine ulaşma yeteneği.
İşlem Süresi: Algoritmanın toplam çalışma süresi. Verimliliği değerlendirmek için önemli bir ölçüttür.
İterasyon Sayısı: Algoritmanın çözüm bulma sürecinde geçtiği iterasyon sayısı. Hız ve verimlilik açısından değerlendirilir.
Başarı Oranı: Algoritmanın belirli bir başarı eşiğini geçme oranı. Genel performansı ve güvenilirliği gösterir.

Bu ölçütler, algoritmanın hem doğruluk hem de verimlilik açısından kapsamlı bir şekilde değerlendirilmesini sağlar.

5. Analiz

5.1. İstatistiksel Testler

İstatistiksel test için Friedman testi ve Wilcoxon testi kullanıldı. Bu testler, optimizasyon algoritmalarının performanslarını veya farklı tedavi yöntemlerinin etkilerini karşılaştırmak gibi çeşitli uygulamalarda yaygın olarak kullanılır.

Friedman Testi

Friedman testi, k bağımlı örneğin (genellikle aynı denek grubu üzerinde yapılan ölçümler) karşılaştırılması için kullanılan non-parametrik bir istatistiksel testtir.

Friedman Testi Uygulanışı:

1. **Veri Toplama:** Her denek veya nesne üzerinde k farklı ölçüm yapılır.
2. **Sıralama:** Her denek için ölçümler kendi içinde sıralanır (küçükten büyüğe veya tam tersi).
3. **Rang Değerlerinin Hesaplanması:** Her ölçüm değeri için bir rang değeri atanır.
4. **Rang Toplamlarının Hesaplanması:** Her grup için rang toplamları hesaplanır.
5. **Test İstatistiğinin Hesaplanması:** Friedman testi için chi-square dağılımına dayanan bir test istatistiği hesaplanır.
6. **Sonuçların Yorumlanması:** Hesaplanan test istatistiği, kritik değerle karşılaştırılır ve anlamlı bir fark olup olmadığına karar verilir.

Matematiksel Formül:

$$\chi_F^2 = \frac{12}{nk(k+1)} \left(\sum R_j^2\right) - 3n(k+1)$$

n, denek sayısıdır.
k, grup sayısıdır.
Rj, her grubun toplam rang değeri.

Wilcoxon Testi

Wilcoxon testi, iki bağımlı örnek arasındaki farkı karşılaştırmak için kullanılan non-parametrik bir testtir.

Wilcoxon Testi Uygulanışı:

1. **Veri Toplama:** Aynı denekler üzerinde iki farklı koşulda ölçüm yapılır.
2. **Farkların Hesaplanması:** Her denek için iki koşul arasındaki fark hesaplanır.
3. **Farkların Sıralanması:** Farklar sıralanır ve işaretlerine göre pozitif ve negatif olarak ayrılır.
4. **Sıralı Farkların Toplanması:** Pozitif ve negatif farkların sıralı toplamları hesaplanır.
5. **Test İstatistiğinin Hesaplanması:** Wilcoxon testi için bir test istatistiği hesaplanır.
6. **Sonuçların Yorumlanması:** Hesaplanan test istatistiği, kritik değerle karşılaştırılır ve anlamlı bir fark olup olmadığına karar verilir.

Matematiksel Formül:

$$W = \min(W^+, W^-)$$

Test istatistiği, küçük olan sıralar toplamıdır.

W^+, Pozitif sıralar toplamıdır

W^-, Negatif sıralar toplamıdır

5.2. Test Sonuçları

Algoritmanın arama performansında iyileşme elde etmek için FDB uygulaması yapılmıştır.

- Çalışmalar CEC 2022 12 problemi üzerinde yapılmıştır.
- Maksimum iterasyon Sayısı(maxFE) 20.000*D olarak belirlenmiştir.
- Dimension 20 olarak belirlenmiştir.

Baz algoritma adı: BWO (Beluga Whale Optimization)

Uygulanan Yöntem(ler): FDB (Fitness Distance Balance), Roulette Fitness Distance Balance (RFDB)

Case adı: Baz Algoritma Adı_FDB Adı_Case Adı (Case numaraları sırasıyla verilmiştir.)

Toplam case sayısı: 7

Runs / problem: 12

Baz Algoritma Yakınsama Denklemi:

$$Bf = B0\left(1 - \frac{T}{2 \cdot T_{\max}}\right)$$

Beluga Whale Optimization (BWO) algoritmasında rehber seçiminde kullanılan değişkenlerin seçim yöntemleri **fitnessDistanceBalance** ve rouletteFitnessDistanceBalance ile değiştirilerek performans iyileştirmeleri hedeflenmiştir. Her bir case, belirli bir parametrenin seçim yöntemini değiştirmektedir.

Case Adı	Boyut / Dimension	maxFE	Uygulanan Yer	İyi (+)	Aynı (=)	Kötü (-)
bwo	20	20.000	-	0	12	0
bwo_FDB_case1	20	20.000	RJ	0	12	0
bwo_FDB_case2	20	20.000	r1	1	4	7
bwo_FDB_case3	20	20.000	r7	1	11	0
bwo_FDB_case4	20	20.000	alpha	0	11	1
bwo_FDB_case5	20	20.000	C1	8	4	0
bwo_FDB_case6	20	20.000	C2	1	11	0
bwo_FDB_RFDB_case7	20	20.000	r5 ve C2	1	9	2

"İyi (+)": İyi performans gösteren senaryoların sayısını ifade eder.

"Aynı (=)": Ortalama performans gösteren senaryoların sayısını ifade eder.

"Kötü (-)": Kötü performans gösteren senaryoların sayısını ifade eder.

6. Tartışma

6.1. Sonuçların Yorumu

Algoritmada yapılan değişikliklerin genel performansa etkisini ve FitnessDistanceBalance algoritması kullanılarak elde edilen iyileştirmeleri inceleyelim.

bwo_FDB_case1: RJ Seçim Yöntemi fitnessDistanceBalance ile Değiştirildi

- **Değişiklik:** RJ, rastgele bir seçim yerine fitness ve mesafe dengesi göz önüne alınarak seçildi.
- **Sonuçlar:** Bu değişiklikle birlikte, algoritmanın performansında herhangi bir iyileşme veya kötüleşme gözlenmemiştir. Sonuçlar orijinal algoritma ile aynıdır (0 iyi, 12 aynı, 0 kötü).
- **Yorum:** RJ değişkeninin seçim yöntemi, fitness ve mesafe dengesi ile değiştirilse de, algoritmanın genel performansına kayda değer bir etki yapmamıştır.

bwo_FDB_case2: r1 Seçim Yöntemi fitnessDistanceBalance ile Değiştirildi

- **Değişiklik:** r1, rastgele bir değer yerine fitness ve mesafe dengesi göz önüne alınarak seçildi.
- **Sonuçlar:** Bu değişiklikle algoritmanın performansında çoğunlukla kötüleşme gözlenmiştir (1 iyi, 4 aynı, 7 kötü).
- **Yorum:** r1 değişkeninin seçim yöntemi, algoritmanın keşif kabiliyetini azaltmış ve genel performansında olumsuz bir etki yaratmıştır.

bwo_FDB_case3: r7 Seçim Yöntemi fitnessDistanceBalance ile Değiştirildi

- **Değişiklik:** r7, rastgele bir değer yerine fitness ve mesafe dengesi göz önüne alınarak seçildi.
- **Sonuçlar:** Bu değişiklikle algoritmanın performansında genel olarak iyileşme gözlenmiştir (1 iyi, 11 aynı, 0 kötü).
- **Yorum:** r7 değişkeninin seçim yöntemi, algoritmanın performansında pozitif bir etki yapmış, özellikle keşif fazında daha iyi çözümler bulunmasına yardımcı olmuştur.

bwo_FDB_case4: alpha Seçim Yöntemi fitnessDistanceBalance ile Değiştirildi

- **Değişiklik:** alpha, sabit bir değer yerine fitness ve mesafe dengesi göz önüne alınarak seçildi.
- **Sonuçlar:** Bu değişiklikle algoritmanın performansında genellikle aynı kalmış, hafif bir kötüleşme gözlenmiştir (0 iyi, 11 aynı, 1 kötü).
- **Yorum:** alpha değişkeninin seçim yöntemi, Levy uçuşunun etkinliğinde küçük bir olumsuz etki yapmış ancak genel performans üzerinde önemli bir değişiklik yaratmamıştır.

bwo_FDB_case5: C1 Seçim Yöntemi fitnessDistanceBalance ile Değiştirildi

- **Değişiklik:** C1, rastgele bir değer yerine fitness ve mesafe dengesi göz önüne alınarak seçildi.
- **Sonuçlar:** Bu değişiklikle algoritmanın performansında belirgin bir iyileşme gözlenmiştir (8 iyi, 4 aynı, 0 kötü).
- **Yorum:** C1 değişkeninin seçim yöntemi, keşif ve sömürü fazlarının etkinliğini artırmış, algoritmanın genel performansında kayda değer bir iyileşme sağlamıştır.

bwo_FDB_case6: C2 Seçim Yöntemi fitnessDistanceBalance ile Değiştirildi

- **Değişiklik:** C2, sabit bir değer yerine fitness ve mesafe dengesi göz önüne alınarak seçildi.
- **Sonuçlar:** Bu değişiklikle algoritmanın performansında iyileşme gözlenmiş ancak çoğunlukla aynı kalmıştır (1 iyi, 11 aynı, 0 kötü).
- **Yorum:** C2 değişkeninin seçim yöntemi, genel performansı çok etkilememiş ancak küçük bir iyileşme sağlamıştır.

bwo_FDB_case7: r5 Seçim Yöntemi rouletteFitnessDistanceBalance ile ve C2 Seçim Yöntemi fitnessDistanceBalance ile Değiştirildi

- **Değişiklik:** r5 ve C2, rastgele değerler yerine sırasıyla rulet çarkı ve fitnessDistanceBalance yöntemleri ile seçildi.
- **Sonuçlar:** Bu değişiklikle algoritmanın performansında genel olarak iyileşme gözlenmiş ancak bazı durumlarda kötüleşme de olmuştur (1 iyi, 9 aynı, 2 kötü).
- **Yorum:** r5 ve C2 değişkenlerinin seçim yöntemleri, algoritmanın performansını hafif iyileştirmiştir, ancak bazı durumlarda negatif etkiler de yaratmıştır.

6.2. Sınırlamalar

Çalışmanın sınırlamaları ve olası hata kaynakları şunlardır:

- **Veri Seti ve Test Problemleri**: Çalışma sadece CEC 2022 12 problemi üzerinde gerçekleştirilmiştir. Farklı veri setleri ve problemler üzerinde yapılan testler, algoritmanın genel performansı hakkında daha kapsamlı bir fikir verebilir.
- **İterasyon Sayısı ve Popülasyon Boyutu**: Maksimum iterasyon sayısı ve popülasyon boyutu belirli değerlerle sınırlı tutulmuştur. Farklı iterasyon sayıları ve popülasyon boyutları ile yapılacak testler, algoritmanın farklı koşullardaki performansını ortaya koyabilir.
- **Parametre Ayarları**: FDB ve RFDB algoritmalarında kullanılan parametrelerin optimize edilmesi, daha iyi sonuçlar elde edilmesini sağlayabilir. Bu çalışmada, bu parametreler sabit değerler olarak alınmıştır.
- **Rastgelelik**: Rastgelelik içeren algoritmaların performansı, farklı çalıştırmalarda değişiklik gösterebilir. Bu nedenle, daha fazla çalıştırma ile istatistiksel olarak anlamlı sonuçlar elde edilebilir.
- **Karmaşıklık**: FDB ve RFDB algoritmalarının eklenmesi, algoritmanın karmaşıklığını artırmış olabilir. Bu, algoritmanın hesaplama süresini ve kaynak kullanımını artırabilir.

7. SONUÇLAR

Bu çalışmada, Beluga Whale Optimization (BWO) algoritmasının performansını artırmak amacıyla, rehber seçim yöntemlerinde değişiklikler yapılmıştır. Bu değişiklikler, Fitness Distance Balance (FDB) algoritması ve türevleri kullanılarak gerçekleştirilmiştir. BWO algoritmasının çeşitli parametreleri (RJ, r1, r7, alpha, C1, C2) için FDB algoritmaları uygulanmış ve performansları CEC 2022 12 test problemleri üzerinde değerlendirilmiştir.

Bu çalışma, Beluga Whale Optimization (BWO) algoritmasında yapılan değişikliklerin performansa olan etkilerini incelemiş ve bazı değişikliklerin algoritmanın performansını iyileştirdiğini, bazı değişikliklerin ise olumsuz etkiler yarattığını göstermiştir. Özellikle, C1 ve r7 değişkenlerinin fitnessDistanceBalance ile seçimi, algoritmanın genel performansını olumlu yönde etkilemiştir. Diğer yandan, r1 değişkeninin fitnessDistanceBalance ile seçimi performansı olumsuz etkilemiştir. Bu sonuçlar, BWO algoritmasının farklı parametre seçim yöntemleriyle daha verimli hale getirilebileceğini göstermektedir.

Gelecekteki Çalışmalar İçin Öneriler:

- **Farklı Problemler Üzerinde Testler**: Algoritmanın farklı optimizasyon problemleri üzerinde test edilmesi, genel performansının daha kapsamlı bir şekilde değerlendirilmesini sağlayabilir.
- **Parametre Optimizasyonu**: FDB ve RFDB algoritmalarında kullanılan parametrelerin optimize edilmesi, algoritmanın performansını daha da artırabilir.
- **Adaptif Yaklaşımlar**: Rehber seçiminde kullanılan yöntemlerin adaptif olarak ayarlanması, farklı aşamalarda en uygun seçimlerin yapılmasını sağlayabilir.
- **Hibrit Yöntemler**: FDB ve RFDB algoritmalarının diğer metaheuristik algoritmalarla birleştirilmesi, hibrit yöntemlerle daha iyi sonuçlar elde edilmesini sağlayabilir.
- **İstatistiksel Analizlerin Artırılması**: Daha fazla çalıştırma ve farklı istatistiksel testler kullanılarak, elde edilen sonuçların daha güvenilir ve genellenebilir olması sağlanabilir.

8. KAYNAKLAR

[1] Zhong, C., Li, G., & Meng, Z. (2022). Beluga whale optimization: A novel nature-inspired metaheuristic algorithm. *Knowledge-Based Systems, 251,* 109215. Retrieved from https://www.sciencedirect.com/science/article/pii/S0950705122006049?casa_token=R09XB1JYkgUAAAAA:dKVcfMWn-0Iwoep9pJNQ1riQAEiCz2E77BPQKZimL2nx02X0Xh0EGUfxfBbBkrGZuV2WT5ryZ6I

[2] MathWorks. (n.d.). Beluga whale optimization (BWO). Retrieved from https://www.mathworks.com/matlabcentral/fileexchange/112830-beluga-whale-optimization-bwo?s_tid=srchtitle

[3] Şekil1. Beluga balinalarının davranışları, (a) keşif aşamasına karşılık gelen yüzme; (b) sömürü aşamasına karşılık gelen yiyecek arama, (c) balina düşüşü için balina düşüşü fazı https://constative.com/animals/whale-watching, https://www.sealuxe.ca/blog/narwhal