**Arctic Puffin Optimization (APO – Deniz Papağanı Optimizasyonu)**

**Özet**

Bu çalışmada, yenilikçi bir yaklaşımla Deniz Papağanı Optimizasyonu tanıtılmaktadır. Bu meta-sezgisel optimizasyon algoritması, deniz papağanının hayatta kalma ve avlanma davranışlarından ilham almıştır. APO, keşif (hava uçuşu) ve sömürü (su altı yiyecek arama) olmak üzere iki aşamadan oluşmaktadır. Keşif aşamasında, Levy uçuşu ve hız faktörü mekanizmaları kullanılarak algoritmanın yerel optima tuzaklarından kaçınma ve yakınsama hızını artırma kapasitesi geliştirilmiştir. Sömürü aşamasında ise sinerji ve uyarlanabilir değişim faktörleri gibi stratejiler kullanılarak algoritmanın mevcut en iyi çözümü etkili bir şekilde kullanması ve arama yönünü belirlemesi sağlanmaktadır. Ayrıca, keşif ve sömürü aşamaları arasındaki dinamik geçiş, davranış dönüşüm faktörü ile gerçekleştirilmekte olup, bu durum küresel arama ile yerel geliştirme arasında etkili bir denge kurulmasını sağlamaktadır. APO algoritmasının ileri düzeydeki ve uygulanabilirliğini doğrulamak amacıyla dokuz gelişmiş optimizasyon algoritması ile karşılaştırılmıştır. CEC2017, CEC2019 ve CEC2022 test setlerinde, APO algoritması sırasıyla %72, %70 ve %75 oranında diğer algoritmalardan üstün performans sergilemiştir. Wilcoxon işaretli-rütbe testi sonuçları ve Friedman ortalama sıralaması, APO algoritmasının üstünlüğünü istatistiksel olarak kanıtlamaktadır. Ayrıca, on üç gerçek dünya mühendislik problemi üzerinde yapılan testlerde, APO vakaların %85'inde diğer algoritmalardan daha iyi performans göstererek, karmaşık gerçek dünya optimizasyon problemlerini çözme potansiyelini ortaya koymaktadır. Sonuç olarak, APO, çeşitli karmaşık optimizasyon problemlerini çözmedeki üstün performansı ile pratik değerini ve avantajlarını kanıtlamaktadır.



Şekil 1 - Deniz Papağanı / Puffin Kuşu Görseli

Şekil 1’de görseli gördükten sonra göz aşinalığımız olduğu anlaşılabilmektedir. En çok kullanılan tarayıcılardan biri olan Puffin Browser adını ve logosunu bu kuştan almaktadır.   
Puffin tarayıcısında puffin kuşunun yani deniz papağanının kullanılmasının sebepleri aşağıda yer almaktadır.

**Hız:** Puffin kuşları, havada en hızlı uçan kuşlardan biridir. Bu da tarayıcının hızını ve çevikliğini temsil eder. Puffin Browser, bulut tabanlı sunucular kullanarak web sayfalarını daha hızlı yüklemeyi ve işlemlerini hızlandırmayı vaat ediyor.

**Güvenlik:** Puffin kuşları, kayalıklarda yuva yaparlar ve bu da onları yırtıcılara karşı korur. Puffin Browser ise, kullanıcıların çevrimiçi gizliliğini ve güvenliğini korumak için çeşitli güvenlik özellikleri sunar. Bu özelliklerden bazıları şunlardır:

**Etkileyicilik:** Puffin kuşları, sevimli ve ilgi çekici hayvanlardır. Bu da Puffin Browser'ı daha ilgi çekici ve kullanımı keyifli bir tarayıcı haline getirir.

**Benzersizlik:** Puffin kuşları, nispeten az bilinen bir kuş türüdür. Bu da Puffin Browser'ı diğer tarayıcılardan farklı ve özel kılar.

**Anlam:** Puffin kuşları, özgürlük ve bağımsızlıkla ilişkilendirilir. Bu da Puffin Browser'ın kullanıcıların çevrimiçi ortamda özgürce ve bağımsız bir şekilde gezinmelerine olanak tanıma arzusunu temsil eder.

**1. Giriş**

Optimizasyon, belirli kısıtlar altında hedef fonksiyonun çıktısını maksimize veya minimize ederek en uygun çözümü bulmayı amaçlamaktadır. Gerçek dünya mühendislik problemlerinin karmaşıklığı arttıkça, optimizasyon tekniklerine olan talep de giderek artmaktadır. Gradyan inişi ve simpleks yöntemi [1,2] gibi geleneksel optimizasyon yöntemleri, belirli koşullar altında kesin çözümler sağlasa da, problem karmaşıklığı arttıkça, çeşitli kısıtlar ve çoklu zirvelerle karşılaşıldığında genellikle yetersiz kalmaktadır. Bu yöntemler, etkili çözümler için problemin özelliklerinin önceden anlaşılmasını gerektirir ve büyük ölçekli problemlerle başa çıkarken önemli hesaplama kaynakları tüketirler. Ayrıca, bu yaklaşımlar yerel optima sorununa duyarlıdır ve bu da küresel optimumu bulmayı zorlaştırır. Bu sınırlamalar, bu tür yöntemlerin günümüzün karmaşık mühendislik problemlerinin gereksinimlerini karşılamada yetersiz kalmasına neden olmaktadır.

Son yıllarda, biyolojik veya fiziksel fenomenlerden ilham alan meta-sezgisel algoritmalar sürekli olarak ortaya çıkmaktadır [[3], [4], [5]]. Geleneksel yöntemlere kıyasla, meta-sezgisel algoritmaların basit kavramları ve çerçeveleri vardır, gradyan güncelleme bilgisine ihtiyaç duymazlar. Karmaşık kısıtlar altında optimal çözümleri makul bir zaman diliminde etkili bir şekilde bulabilirler [6]. Bu nedenle, bu yöntemler mühendislik optimizasyon problemlerindeki karmaşık zorlukları ele almak için daha uygundur ve endüstriyel mühendislik problemleri, özellik seçimi, çok seviyeli görüntü segmentasyonu, topluluk tespiti, görüntü sınıflandırma ve diğer alanlar dahil olmak üzere birçok pratik mühendislik sorununa geniş çapta uygulanmıştır [[7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18]].

İyi tasarlanmış bir meta-sezgisel algoritma, küresel keşif ve yerel sömürü olmak üzere iki temel özellik arasında bir denge kurmalıdır. Bu denge, algoritmanın arama alanında kapsamlı bir keşif yaparken küresel optimal çözümler arama ihtiyacını temsil eder. Değerli çözümler keşfedildikçe, algoritma bu bölgelerde yerel optimal çözümler elde etmek için derinlemesine araştırma yapmalıdır. Bu durum, algoritmanın yerel optima tuzaklarından kaçınmasını, erken yakınsamayı önlemesini ve küresel optimuma hızla yakınsamasını sağlamak için keşif ve sömürüyü nasıl dengeleyebileceğimiz sorusunu ortaya çıkarır. Bu özellikler, çoklu zirveler ve kısıtlar içeren optimizasyon görevlerinde özellikle önemlidir. Meta-sezgisel algoritmaların, olası çözümleri keşfetmek için arama alanını esnek bir şekilde küresel olarak keşfetmesi gerekir. Değerli çözüm bölgeleri belirlendikten sonra, algoritma yerel sömürüye odaklanmalı ve optimal yerel çözümleri bulmak için aramayı yoğunlaştırmalıdır. Böyle bir denge, problem karmaşıklığıyla başa çıkmak ve algoritmanın çeşitli problem senaryolarında uyarlanabilirliğini artırmak için önemlidir.

Mevcut çeşitli optimizasyon algoritmaları olmasına rağmen, "no free lunch" (NFL) teoremi [19] nedeniyle yeni meta-sezgisel algoritmaların sürekli olarak geliştirilmesi gerekmektedir. Bu teorem, farklı algoritmaların performansının, farklı özelliklere sahip optimizasyon problemlerini çözerken değişiklik gösterdiğini öne sürer. Tek bir algoritma, tüm problemler için optimal performans sergileyemez. Dahası, bir algoritma bir tür problemde iyileşme gösterse bile, bu iyileşme başka bir problem türünde dengelenebilir. Bu durum, araştırmacıların, farklı problem türlerinin optimizasyon gereksinimlerini karşılamak ve mevcut algoritmaların eksikliklerini gidermek için sürekli olarak yeni meta-sezgisel algoritmalar arayıp geliştirmelerine yol açmaktadır.

Bu motivasyon doğrultusunda, bu makalede, yeni bir meta-sezgisel optimizasyon algoritması olan APO algoritmasını öneriyoruz. Bu algoritma, kutup martılarının davranışlarını iki aşamada simüle eder: hava uçuşu ve su altı yiyecek arama, keşif ve sömürü arasında daha iyi bir denge sağlamayı amaçlar. Algoritma, hava uçuşu aşamasında hava araması ve dalış avlanması, su altı yiyecek arama aşamasında ise yiyecek toplama, yoğunlaştırılmış arama ve yırtıcıdan kaçınma gibi birkaç önemli işlemden oluşur. Bu adımlar aracılığıyla, algoritma arama alanını amaçlı bir şekilde keşfetmek, keşif ve sömürü arasında daha iyi bir denge kurmak ve yerel optima düşmekten kaçınmak üzere tasarlanmıştır. Bu makalenin özgün katkıları ve yenilikleri aşağıda özetlenmiştir.

1. Küresel optimizasyon problemi için, bu makale çeşitli optimizasyon senaryoları için yeni bir APO algoritması önermektedir.
2. Hava uçuşu aşamasında (keşif), Levy uçuşları ve hız faktörlerinin tanıtılması, algoritmanın yerel optimumdan çıkma yeteneğini artırır ve yüksek boyutlu karmaşık optimizasyon problemlerinde daha iyi bir çözüm bulma olasılığını artırır.
3. Su altı yiyecek arama aşamasında (sömürü), Levy uçuşları, sinerji faktörleri ve uyarlanabilir değişim faktörleri kullanılarak algoritmanın mevcut en iyi çözümü etkili bir şekilde kullanması ve arama yönünü yönlendirmesi sağlanır.
4. Uyarlanabilir özelliklere sahip davranış dönüşüm faktörü B'nin kurulması, keşif ve sömürü arasındaki dinamik ilişkiyi etkili bir şekilde dengeler.
5. APO algoritmasının performansı, CEC2017, CEC2019 ve CEC2022 fonksiyonları kullanılarak, toplamda 51 zorlu benchmark fonksiyonu ile doğrulanmış ve dokuz tanınmış ve ileri optimizasyon algoritması ile karşılaştırılarak analiz edilmiştir.
6. İki parametrik olmayan test olan Friedman ortalama sıralama testi ve Wilcoxon işaretli sıralama testi kullanılarak kapsamlı istatistiksel analizler yapılmış ve algoritmanın farklı problem alanlarındaki dayanıklılığı ve üstünlüğü doğrulanmıştır.
7. Dört alandaki on üç karmaşık gerçek dünya mühendislik optimizasyon problemi çözülerek, APO'nun geniş bir optimizasyon senaryosunda genelleştirilebilirliği ve dayanıklılığı değerlendirilmiş ve doğrulanmıştır.

**2. İlgili Çalışmalar**

Metasezgisel algoritmalar, özellikle polinom zaman karmaşıklığına sahip olmayan ve karmaşık, çözümü zor problemlerin çözümünde sıklıkla tercih edilen bir yöntem olarak ön plana çıkmaktadır. Bu algoritmalar genellikle doğal fenomenlere dayalı olarak tasarlanmaktadır ve iç mekanizmalarına göre evrimsel, fizik veya matematik temelli, insan davranışı temelli ve sürü zekası temelli olmak üzere farklı kategorilere ayrılabilmektedirler. Tablo 1'de gösterildiği gibi, bu kategoriler aşağıda kısaca özetlenmiştir.

**2.1. Evrimsel Tabanlı Algoritmalar**

Biyo-evrim prensiplerine dayalı olarak tasarlanmış evrimsel Metasezgisel optimizasyon algoritmaları, türlerin evrim süreçlerini ve genetik mirası simüle ederek en iyi çözümleri iteratif olarak optimize etmek ve aramak için kullanılır. Genetik algoritmalar (GA) [20], diferansiyel evrim algoritması (DE) [21] gibi temsilci algoritmaların yanı sıra bu kategori, evrimsel stratejiler (ES) [22], biyocoğrafya temelli optimizatörler (BBO) [23], evrimsel programlama (EP) [24] gibi yöntemleri içerir.

**2.2. Fizik Tabanlı Algoritmalar**

Bu kategorideki algoritmalar, fizik veya matematik ilkelerinin taklit edilmesinden ilham alır. Fizikteki hareket yasalarından ve matematik modellerinden esinlenen bu algoritmalar, karmaşık problemleri çözmek için içgörüler sağlar. Örnekler arasında denge optimizatörü (EO) [25], Arşimet optimizasyon algoritması (AOA) [26], aritmetik optimizasyon algoritması (AOA) [27], su döngüsü algoritması (WCA) [28], Kepler optimizasyon algoritması (KOA) [29], kara delik algoritması (BH) [30], yerçekimi arama algoritması (GSA) [31], Yang'ın Çift Yarık Algoritması (YDSE) [32] vb. bulunur.

**2.3. İnsan Davranışı Temelli Algoritmalar**

Bu kategoriye ait algoritmalar, insan davranışından ve sosyal etkileşimden ilham alır. Örneğin, harmony search (HS) [33] müziğin güzelliğini ve uyumunu simüle ederek nota ayarlamaları yaparak küresel optimaliteyi hedefler. Brainstorm Optimization (BSO), takım çalışması ve yaratıcılık uyarımından esinlenerek beyin fırtınası sürecini simüle eder ve arama sürecini optimize eder [34]. Ayrıca öğrenci psikolojisine dayalı optimizasyon algoritmaları (SPBO) [35], öğretme-öğrenme tabanlı optimizasyon (TLBO) [36], Politik Optimizasyon (PO) algoritması [37], İmparatorluk Rekabetçi Algoritması (ICA) [38] gibi yöntemler bulunmaktadır. Bu algoritmalar, insan sosyal davranışına dayalı olarak problem çözümü için benzersiz ilham sağlar.

**2.4. Sürü Zekası Temelli Algoritmalar**

Bu kategoriye ait algoritmalar, doğadaki kolektif davranışları taklit ederek hayvanların, böceklerin veya kuşların sosyal ve işbirlikçi davranışlarını simüle eder. Örneğin, particle swarm optimization (PSO) [39] kuşların yiyecek arama sürecini taklit eder; parçacıklar potansiyel çözümleri temsil eder ve performanslarına göre hareket eder. Gray Wolf Algorithm (GWO) avlanma ve av arama süreçlerinden esinlenerek geliştirilmiş optimize bir arama yöntemidir [6]. Ant colony optimization (ACO) karıncaların yiyecek ararken feromon salgılamalarına dayanır ve koloninin optimal yolunu bulmasını sağlar [40]. Whale Optimization Algorithm (WOA) gaga balinalarının beslenme davranışını taklit ederek aramayı ayarlar [41]. Ayrıca harris şahin algoritması (HHO) [42], beyaz balina optimizasyonu (BWO) [43], peygamberdeve arama algoritması (MSA) [44], yapay tavşan optimizasyonu (ARO) [45], altın kartal optimizasyonu (GEO) [46], sümüksü küf algoritması (SMA) [47], puma optimizasyon algoritması (PO) [48] gibi algoritmalar da bulunmaktadır. Bu algoritmalar, doğal biyolojik popülasyonların zekasından ilham alarak kolektif bir yaklaşım sunar ve problem çözümü için benzersiz bir yöntem sunar.

Yukarıdaki optimizasyon algoritmalarının özelliklerini daha net bir şekilde ortaya koymak için, Tablo 1 her bir algoritmanın avantajlarını ve dezavantajlarını incelemektedir. Bu genel bakışa dayanarak, her optimizasyon algoritmasının farklı avantajlara ve dezavantajlara sahip olduğu gözlemlenebilir. Bu aralarında, keşif ve geliştirme aşamaları arasındaki denge eksikliği, yerel optimum tuzağına düşme eğilimi ve yüksek boyutlu problemlerde performans düşüşü gibi sorunlar hala mevcut algoritmaların ortak sorunları olarak ortaya çıkmaktadır. Bu sorunlara daha iyi çözüm bulabilmek için, yeni optimizasyon algoritmaları önermekten başka, araştırmacılar algoritmaların diğer alanlarda geliştirilmiş versiyonlarını önermeye ve bu konuda incelemeler yapmaya çalışmaktadır [49-54]. Örneğin, Farhad Soleimanian Gharehchopogh, manta ray foraging optimization (MRFO) algoritmasını analiz ederek ve farklı akademik alanlarda entegrasyonunu inceleyerek [55]. Hoda Zamani, Moth Flame Optimization algoritmasını ve varyantlarını inceleyerek yapısal inceleme, performans değerlendirmesi ve istatistiksel analiz gerekliliklerini gösterir [56]. Bu süreçte, diğer araştırmacılar algoritmanın genel performansını artırmak için çeşitli etkili mekanizmalar önermiştir [57-64]. Bu çalışmalar temelinde, optimizasyon algoritmaları, optimizasyon arayış performansını artırmak için çeşitli etkili mekanizmalar sunabilir. Bu mekanizmalar, çözüm çeşitliliğini ve rastgeleliğini artırma, optimal çözüme hızlı bir şekilde yakınsama sağlama gibi stratejiler içermelidir. Bu nedenle, mükemmel optimizasyon algoritmaları, algoritmaların keşif ve geliştirme aşamalarının dengesini sağlamak için makul bir süreç içinde akıllı mekanizmalar ve stratejiler sunmalıdır.

Son yıllarda önerilen yeni Metasezgisel algoritmaların ve iyileştirilmiş versiyonlarının sentezlenmesi, karmaşık problemleri çözmede iyi performans göstermelerine rağmen, hala daha fazla gelişim potansiyeline sahiptirler. İlgili literatürü gözden geçirerek, bu çalışmalar bizi farklı alanlarda karmaşık mühendislik problemleriyle başa çıkmak ve daha etkili çözümler bulmak için daha yeni ve daha iyi performans gösteren bir algoritma önermeye teşvik etmiştir. Bu nedenle, bu makalede önerilen APO algoritması, mevcut algoritmaları tamamlamak ve yeni bir keşif yönü olarak, optimizasyon algoritmalarında daha derin gelişmeler ve çığır açacak buluşlar getirmesi beklenmektedir.

A diagram of a company

Description automatically generated with medium confidence

Şekil 2 - Metasezgisel Optimizasyon Algoritmaları

Tablo 1 - Popüler optimizasyon algoritmalarının bir incelemesi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tür** | **Algoritma** | **İlham Kaynağı** | **Avantaj** | **Dezavantaj** |
| Evrimsel tabanlı | GA [20] | Doğal seçilim ve genetik evrim. | Güçlü keşif aşaması. | Zayıf yararlanma aşaması. |
| Evrimsel tabanlı | DE [21] | Biyolojik evrim teorisi. | Üretim çözümleri mükemmel bir çeşitliliğe sahiptir. | Zayıf yoğunlaştırma bileşeni. |
| Evrimsel tabanlı | ES [22] | Biyolojik evrim teorisi. | İyi paralellik ve gerçek zamanlı performansa sahip olma. | Zayıf arama yeteneği. |
| Evrimsel tabanlı | BBO [23] | Biyocoğrafya teorisi. | Basitlik ve verimlilik. | Boyutsallık lanetine duyarlı. |
| Evrimsel tabanlı | EP [24] | Biyolojik evrim teorisi. | Çok amaçlı problemleri çözmek için güçlü arama yeteneği. | Zor parametre seçimi, uzun çalışma süresi. |
| İnsan davranışı tabanlı | HS [33] | Bir müzik çalar ile doğaçlama kompozisyon. | Basit uygulama, esnek. | Parametre ayarlaması zor. |
| İnsan davranışı tabanlı | BSO [34] | İnsan beyin fırtınası süreçleri. | Güçlü pratiklik. | Uyum eksikliği, problemden kolayca etkilenir. |
| İnsan davranışı tabanlı | SPBO [35] | Öğrencilerin psikolojisine dayalı. | Güçlü yakınsama yeteneği. | Yerel aramaya kolayca düşebilir. |
| İnsan davranışı tabanlı | TLBO [36] | Öğretmenlerin, öğrencilerin çıktılarına etkisi. | Güçlü yakınsama yeteneği. | Yerel aramaya kolayca düşebilir. |
| İnsan davranışı tabanlı | PO [37] | Politikanın çok aşamalı süreci. | Güçlü küresel arama yeteneği. | Büyük hesaplama yükü, yavaş yakınsama hızı. |
| Sürü zekası tabanlı | PSO [39] | Parçacık sürülerinin doğal davranışı. | Yoğunlaştırma bileşeninin iyi yeteneği. | Parametrelere duyarlılık. |
| Sürü zekası tabanlı | GWO [6] | Gri kurtların liderlik hiyerarşisi ve avlanma mekanizmaları. | Az sayıda parametre ile güçlü yakınsama performansı. | İlerleyen aşamalarda yavaş yakınsama hızı. |
| Sürü zekası tabanlı | ACO [40] | Karıncaların yiyecek ararken feromon salgılama davranışı. | Güçlü faydalanma aşaması. | Bazı problemler için yerel optimalliğe düşme. |
| Sürü zekası tabanlı | WOA [41] | Sperm balinalarının beslenme davranışı. | Basitlik, verimlilik ve statik parametre gereksizliği. | Yerel optimuma kolayca düşebilir. |
| Sürü zekası tabanlı | HHO [42] | Harris şahinlerinin işbirliği davranışı ve avlanma taktikleri. | Az sayıda parametre, basit ve kolay uygulama. | Düşük boyutlu problemlerde zayıf arama yeteneği. |
| Sürü zekası tabanlı | BWO [43] | Beyaz balinaların yaşam davranışları. | Yapısı basit ve kolay uygulanabilir. | Parametre ayarlama zorluğu. |
| Sürü zekası tabanlı | MSA [44] | Peygamberdevelerinin benzersiz avlanma ve yamyamlık davranışları. | Güçlü arama yeteneği. | Zayıf arama yeteneği. |
| Sürü zekası tabanlı | ARO [45] | Tavşanların hayatta kalma stratejileri. | Güçlü optimizasyon yeteneği ve hızlı yakınsama hızı. | Dayanıklılık eksikliği. |
| Sürü zekası tabanlı | GEO [46] | Altın kartalların avlanma içgüdüsü. | Güçlü küresel arama yeteneği. | Yavaş yakınsama hızı. |
| Sürü zekası tabanlı | SMA [47] | Doğal dünyada sümüksü kalıpların salınım deseni. | Güçlü doğal ilham. | Zor parametre seçimi, uzun çalışma süresi. |
| Sürü zekası tabanlı | PO [48] | Pumaların zekası ve yaşamı. | Yerel optimumlardan etkili kaçınma ve kabul edilebilir yakınsama hızı. | İlerleyen aşamalarda yavaş yakınsama hızı. |

**3. Deniz Papağanı Optimizasyonu (APO)**

**3.1. Fikrin Ortaya Çıkışı**

Deniz papağanı, Arktik'e endemik nadir ve küçük bir kuştur. Genellikle okyanuslarda yaşarlar ve bu kuşlar genellikle gruplar halinde veya sürüler halinde mükemmel avcılık becerileri ve uçuş yetenekleri ile avlanır ve okyanusta işbirliği yaparlar [65]. Deniz papağanıler, genellikle okyanusun kenarında küme halinde uçmayı tercih eder, yüzeyde yüzerler ve balık avlarlar. Diğer deniz kuşlarına kıyasla, Deniz papağanılerin kısa, tombul kanatları vardır ve küçük kanatlarını yüksek frekansta çırparak düşük irtifada uçuşlarını sürdürürler. Havada 88 kilometreye kadar hız yapabilirler, dakikada dört yüz kez kanat çırparlar ve uçarken yiyecek hedeflerine yönelip dalabilirler [66].

Ayrıca, Deniz papağanıler, Şekil 2'de gösterildiği gibi havada uçuş ile su altında avlanma arasında dikkate değer bir denge sağlarlar. Kanatları yüzme sırasında çift kürek görevi görerek su altında hızlı dalış için bir güç aracı olarak işlev görür. Deniz papağanıler, en az on küçük balığı dalış başına yakalayan olağanüstü güçlü avcılardır. Dalış öncesinde, diğer deniz papağanlarından gelen davranış sinyallerini izleyerek avlanma verimliliklerini artırmak için koordineli beslenme davranışları kullanırlar. Dalış sırasında, Deniz papağanıler, tırtıklı dil ve üst gagasında keskin dikenlerle kaplı olarak benzersiz bir şekilde tasarlanmış gagalarını kullanarak avı hızla yakalarlar. Avı ağızlarında dikenlerle sabitlerler ve ardından avlanmaya devam ederek dalış sürelerini uzatırlar. Ayrıca etraflarındaki yiyecek kaynakları azaldığında, Deniz papağanıler, av verimliliğini sağlamak için su altında pozisyonlarını esnek bir şekilde ayarlarlar.

Deniz papağanıler, denizde veya uçuş aktiviteleri sırasında genellikle gruplar halinde kalmayı tercih ederler. Bu kolektif ve büyük ölçekli karışık yaşam tarzı, onların Arktik'te daha iyi hayatta kalmasını sağlar. Arktik maceralarında, Deniz papağanıler kaçınılmaz olarak tehlike ile karşılaşırlar. Bir düşman fark edildiğinde, deniz papağanları uyarı sesleri çıkarır ve hızla havada sürü halinde uçarlar [67].

PO algoritması, doğadaki deniz papağanlarının hayatta kalma stratejilerinden kaynaklanmaktadır. Aşağıda kısaca özetlenmiştir:

* Deniz papağanları, uçuş sırasında formasyonlar veya gruplar oluşturarak uçuşlarını koordine eder ve uçuş verimliliğini artırır, işbirliği içinde avlama fırsatları yaratır.
* Uçuş sırasında, büyük miktarda gıda kaynağı tespit edildiğinde, deniz papağanları hızla uçuşlarını yeniden yönlendirir ve hızlı bir şekilde avı yakalamak için dalış-besleme stratejisi kullanır.
* Su yüzeyinde, deniz papağanları genellikle balık sürülerini grup yardımıyla yuvarlar ve avlanma verimliliğini artırır. Ayrıca diğer üyelerin davranışlarını gözlemleyerek dalış noktalarını veya gıda kaynaklarını belirler.
* Deniz papağanları, etraflarındaki gıda kaynakları tükendiğinde ek gıda kaynakları bulmak için su altı konumlarını dinamik olarak ayarlar.
* Yakınlarda bir tehlike, örneğin bir yırtıcı tespit edildiğinde, kutup puffini hızla konumunu değiştirir ve tehlikeden kaçınmak için eşlerine mesaj gönderir.
* Özellikle, deniz papağanlarının davranışlarının rastgeleliği, algoritmanın tasarımındaki motivasyonu vurgular; çünkü bu beş davranış her iterasyonda her bireyde rastgele olarak gerçekleşir.

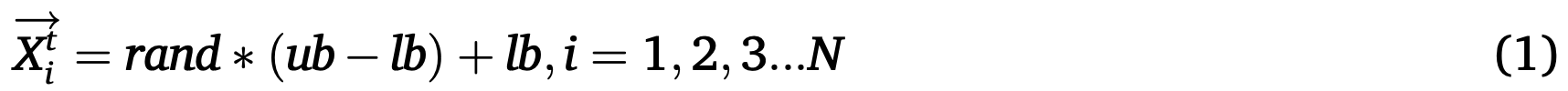
Özetle, deniz papağanlarının havada ve su yüzeyinde gösterdiği özel davranışlar, global optimizasyon problemlerini çözmek için APO algoritmalarının tasarımının temelini oluşturur.

**3.2. Matematiksel Model ve Temel Prensipler**

Bu bölümde, Arktik papaganın hayatta kalma davranışlarından ilham alınarak APO algoritması önerilmektedir. APO'nun matematiksel modeli üç ana aşamadan oluşur: nüfus başlangıcı, hava uçuş aşaması (keşif) ve su altı avcılık aşaması (istismar). Ayrıca, bu iki strateji arasında geçiş, papaganın davranışsal dönüşüm faktörü B tarafından tetiklenmektedir. Her aşama için matematiksel modeller aşağıda açıklanmaktadır.

**3.2.1. İlk Nüfus**

Arktik papaganlar, göç sırasında veya habitatlarında güçlü bir toplulukçuluk sergilerler, her zaman gruplar halinde hareket eder ve işbirliği yaparlar. Her Arktik papaganı, optimizasyona katılan potansiyel bir çözümü temsil eder. Nüfusun başlangıcı süreci aşağıdaki denklemle tanımlanmıştır:



Burada X, i. Deniz papağının pozisyonunu temsil eder; rand 0 ile 1 arasında rastgele bir sayı üretir; ub ve lb sırasıyla üst ve alt sınırları temsil eder; N ise popülasyondaki bireylerin sayısını ifade eder.

A diagram of a bird diagram

Description automatically generated

Şekil 3 - Arktik Deniz Papağanlarının Hava Keşfi Aşaması

**3.2.2. Hava Uçuş Aşaması (Keşif)**

Arktik deniz papağanları, zorlu yaşam koşullarında seyir ve avlanma için benzersiz stratejilere güvenirler. Günlük yaşamlarında, beslenme ihtiyaçlarını karşılamak ve çeşitli ortamlara uyum sağlamak için okyanus ve hava arasında esnek bir şekilde adapte olmak zorundadırlar. Hava seyri sırasında kuşlar, Şekil 3'te gösterildiği gibi, farklı durumları ele almak için iki temel strateji kullanırlar. İlk strateji, hava keşiflerini içerirken, ikinci strateji av için dalış yapmayı içerir. Bu iki aşamada sergilenen çeşitli davranış stratejileri, Arktik deniz kuşlarının çeşitli senaryolarda uyum sağlama yeteneğini gösterir, başarılı bir şekilde hayatta kalma ve üreme imkanı sağlar. Bundan sonra, bu makale Arktik deniz kuşlarının hayatında bu iki stratejinin detaylı bir şekilde incelenmesini ve önemini açıklayacaktır.

* İlk strateji hava aramasıdır.

Arktik deniz papağanları genellikle formasyonlar veya gruplar halinde koordineli uçuş yaparlar, bu işbirlikçi hareket uçuş verimliliğini artırır ve işbirlikçi avlanma fırsatları sunar. Potansiyel su altı gıda kaynaklarını yakalamayı kolaylaştırmak için genellikle düşük uçuş yüksekliklerinde kalırlar. Bu aşamada, potansiyel avları izlemeye odaklanırken yakındaki potansiyel yırtıcılarından da kaçınırız. Uygun koşullar altında veya yırtıcıların nadir olduğu ve balık popülasyonlarının bol olduğu durumlarda, avlarını daha iyi yakalamak için su yüzeyine hızla doğru hareket ederler. Bu stratejiyle ilişkili pozisyon güncelleme denklemleri aşağıdaki gibidir.

A black text on a white background

Description automatically generated



r, i hariç 1 ile N - 1 arasında rastgele bir tamsayıyı temsil eder;

Xti, populasyondaki mevcut i'nci aday çözümü temsil eder; Xtr, mevcut populasyondan rastgele seçilmiş bir aday çözümüdür ve Xti != Xtr koşulunu (eşit değil) sağlar;

L(D), Levy uçuşuyla üretilen rastgele bir sayıyı ifade eder; D boyut sayısını; α standart normal dağılımı takip eden rastgele bir sayıyı ifade eder.

Kutup penguenleri havadan arama stratejisi sırasında, güçlü kanatlarına benzer şekilde Levy uçuş katsayısını kullanarak pozisyonlarını değiştirirler. Levy uçuşu, uzak mesafeli sıçramalarıyla bilinir ve bu sayede penguenlerin yiyecek kaynaklarını etkili bir şekilde keşfetmelerini sağlar [68-71]. Bu uçuş arama stratejisi, penguenlerin geniş okyanus alanlarını hızla kapsamalarına, potansiyel yiyecek kaynaklarına sürekli olarak yakınlıklarını korumalarına ve yiyecek kaynaklarını etkin bir şekilde aramalarına yardımcı olur. Aynı zamanda, martılar gibi hava avcılarıyla karşılaştıklarında, penguenler saldırganlara karşı dönme uçuş taktiği kullanarak savunma yaparlar, bu da saldırganların bir zayıflık bulmasını zorlaştırır. Dolayısıyla, penguenler için havadan arama stratejisi sırasında önemli olan strateji, uygun balık avlama bölgelerini belirlemektir. Bu uçuş stratejisi, algoritmanın global arama yeteneklerini artırır, problem alanını daha iyi keşfetmesine ve global optimal çözümü keşfetme olasılıklarını artırmasına yardımcı olur.

* İkinci strateji, daldırarak avlama stratejisidir.

Daldırarak avlama, deniz papağanının avlanması sırasında kritik bir stratejidir çünkü hızla uçuşlarını yeniden yönlendirerek yiyecek yakalamayı hızlandırırlar. Bu daldırarak avlama stratejisi, diğer rakipler karşısında önemlidir çünkü hayatta kalmak için daha hızlı ve daha başarılı av yakalamalarını sağlamaları gerekmektedir. Bu süzülme davranışını modellemek için, puffin'in süzülme sırasında konum değişimini ayarlamak için bir hız katsayısı S tanıttık. Aşağıda pozisyon güncelleme denklemi verilmiştir.





Bu uçuş stratejisinde, Arktik deniz papağanı, π matematiksel sabiti yaklaşık olarak 3.14 olan bir hız katsayısı S tanıtarak ilk aşamada yer değiştirmesini ayarlar. S, Arktik deniz papağanının uçuş hızının hızını ve yönünü düzenlemede önemli bir rol oynar. Matematiksel olarak, S, deniz papağanının uçuş hızının büyüklüğünü ve yönünü ayarlayarak farklı beslenme ihtiyaçlarına esnek bir şekilde uyum sağlamasını sağlayan bir hız katsayısıdır. Parametre S ile algoritma, deniz papağanının hava davranışına daha yakın bir şekilde eşleşir, böylece rekabet ve belirsizlik karşısında daha esnek hale gelir ve daha karmaşık hava ortamlarına adapte olur. Ayrıca, parametre S'nin dağıtımsal özelliklerini tanıtarak algoritmanın rastgeleliğini ve çeşitliliğini artırır ve Arktik deniz papağanının keşif yeteneğini güçlendirir. Bu şekilde, algoritmanın çözüm uzayında arama verimliliği daha da artar. Algoritmanın farklı bağlamlarla uğraşırken adaptasyon ve arama yeteneği geliştirilir.

Genel olarak, APO'nun uçuş aşamasında, algoritmanın keşif sürecini artırmak için iki temel davranış stratejisi kullanılır. Algoritma, global arama için Levy uçuş katsayısına güvenir. Levy uçuş katsayısını tanıtarak, algoritma daha fazla rastgelelik ve çeşitlilik sağlar, keşif alanını genişletir. Bu strateji, APO'nun potansiyel çözümler için geniş bir arama yeteneği kazanmasını sağlar. Uygun av alanları keşfedildiğinde, algoritma hızlı aramalarla potansiyel çözümleri bulmak için swooping predation stratejisini kullanır. Bu strateji, kısa sürede potansiyel çözümlerin derinlemesine araştırılmasını vurgular ve algoritmanın keşif yeteneğini güçlendirir. APO, swooping predation ile olası optimal çözümlere hızlı ve doğru bir şekilde odaklanarak, arama hızını ve doğruluğunu artırır ve çözüm uzayında yüksek kaliteli çözümler arayışını kolaylaştırır.

Farklı senaryolarda optimal sonuçlar elde etmek için, algoritma her iki aşamada üretilen aday pozisyonları yeni bir çözüm haline getirir. Bu çözümler fitness'e göre sıralanır ve en iyi N birey yeni populasyonu oluşturmak için seçilir. Denklemler aşağıda açıklanmıştır:

A white background with black dots

Description automatically generated

Yeni populasyon Xti'i seçmek için, populasyonunun fitness değerlerine göre küçükten büyüğe sıralandığı yerdir.

A diagram of a duck diagram

Description automatically generated with medium confidence

Şekil 4 - Arktik Deniz Papağanlarının Su Altı Avlanma Aşaması

**3.2.3. Su Altı Avlanma Aşaması (Sömürü)**

Kutup pelikanının hayatta kalma stratejisi iki önemli yönü içermektedir: hava uçuşu ve sualtı avcılığı. Sualtı avcılığı, avlanma verimliliğini artırmak için belirli bağlamlarda kullanılan üç ana stratejiyi içermektedir. Bu üç strateji, avlanma toplama, arama yoğunlaştırma ve avcılardan kaçınma olarak Şekil 4'te gösterilmiştir. Aşağıdaki bölümler, bu aşamaların detaylı bir şekilde incelenmesini ve kutup pelikanının yaşamındaki hayati rollerini keşfetmeyi sağlayacaktır.

* Avlanma toplama, sualtı stratejisinin ilk adımıdır.

Kutup pelikanlarının avlanma davranışında, genellikle suyun yüzeyine yakın balık sürülerinin etrafında toplanarak kolektif bir strateji benimserler. Bu işbirlikçi avlanma davranışı, avcılık verimliliğini artırır ve başarı oranını yükseltir. İşbirlikçi avlanma, etkili bir şekilde işbirliği yapmalarını sağlar; balık sürülerini çevreleyerek ve yakalayarak başarılı avlanma şansını artırır. Ayrıca, su yüzeyinde kalan pelikanlar, dalış noktalarını veya yiyecek kaynaklarını belirlemek için diğer üyelerin davranışlarını gözlemleyebilirler.

A math equations with arrows

Description automatically generated with medium confidence

İki pozisyon denklemi, toplama avcılık stratejisini foraj davranışını tasvir etmek için kullanılmıştır. Bu denklemlerde, rand<0.5 olduğunda, Kutup penguenleri diğer penguenlerle işbirliği yaparak kooperatif foraj yaparlar ve çevredeki ortamı keşfetmek için rastgele hareketlerle işbirlikçi faktör F'yi kullanırlar, yiyecek veya diğer kaynakları ararlar. Diğer denklem ise, rand≥0.5 olduğunda daha karmaşık bir yiyecek arama stratejisini temsil eder. Toplama foraj sürecinde bir penguen, diğer üyeleri takip edebilir ve balık sürüsü tespit edildiğinde hızla yüzer, konumunu değiştirerek daha avantajlı bir avcılık grubuna katılır. Bu nedenle, Levy uçuş faktörü tekrar kullanılarak penguenin hızlı avcılık yeteneği artırılır ve etkili av yakalama sağlanır. Bu çift-strateji yaklaşımı optimizasyon sırasında keşif ve sömürüyü dengelemeye yardımcı olarak çözüm alanının daha kapsamlı bir aramasına katkıda bulunur.

APO algoritmasında, parametre F, Arctic puffinlerin foraj davranışlarında sergiledikleri işbirliği ve takım avcılığından esinlenerek sinerjik bir faktör olarak tasarlanmıştır. Arctic puffinler genellikle suyun yüzeyine yakın balıklar etrafında toplanır ve birleşik bir foraj stratejisi benimserler. Bu sinerjik foraj davranışı, foraj verimliliğini ve başarı oranını büyük ölçüde artırır. Dolayısıyla, F parametresinin tanıtılması, algoritmayı doğadaki hayvanların davranış kalıplarına daha yakın hale getirir ve bu kolektif, işbirlikçi davranışı simüle ederek algoritmanın çözüm çeşitliliğini ve geliştirme yeteneğini artırabilir. Aynı zamanda, F değerinin makul bir şekilde ayarlanması, algoritmanın performansını ve verimliliğini önemli ölçüde artırabilir, algoritmanın keşif ve geliştirme sürecini etkili bir şekilde dengeleyebilir. F'nin hassasiyet analizi sonucunda, F = 0.5'in daha optimal bir değer olduğu belirlenmiştir. Parametrelerin hassasiyet analizini içeriğin ilerleyen bölümlerinde detaylı olarak tartışacağız.

• Arama yoğunlaştırma, ikinci su altı stratejisidir.

Avcılık ilerledikçe, Arctic puffinler belirli bir süreden sonra mevcut foraj alanlarında yiyecek kaynaklarının azaldığını veya tükendiğini hissedebilirler. Beslenme ihtiyaçlarını karşılamak için daha fazla balık veya diğer su altı gıda kaynakları aramak için su altı pozisyonlarını değiştirmeleri gerekmektedir. Bu aşama için pozisyon güncelleme denklemi aşağıdaki gibidir:

A white background with black and white clouds

Description automatically generated

T, toplam iterasyon sayısını temsil ederken, t mevcut iterasyon sayısını belirtmektedir. Rand, f'ye bazı rastgelelik ekleyen rasgele bir sayıdır. Denklem (11), başlangıç pozisyonuna dayanan değişkenlikleri temsil eden bu aşamayı tanımlamaktadır. Geliştirilmiş arama sürecinin ikinci aşamasında, (1 + f) parametresi önemli bir rol oynamaktadır; burada f, Arktik deniz papağanının su içindeki pozisyonunu ayarlamak için kullanılan uyarlanabilir bir faktördür. Bu uyarlanabilir faktörün tasarımı, deniz kuşlarının avlanma sırasında çevrelerine uyum sağlama esnekliğinden ilham almaktadır. Özellikle, iterasyon sayısı arttıkça, parametre f giderek ayarlanır, böylece deniz papağanı, aramanın ilerlemesine ve rastgeleliğine bağlı olarak pozisyonunu değiştirip daha zengin besin kaynakları bulmak için karar verebilir.

Bu strateji, Arktik deniz kuşlarının çevrelerindeki yiyecek dağılımlarına esnek bir şekilde uyum sağlayabilmesini ve avlanma sırasında hayatta kalma ihtiyaçlarını karşılamasını sağlar. Algoritma, dinamik olarak arama alanını keşfeder ve yeni çözümler bulur, böylelikle değişen problemlere ve yeni çözümlere karşı uyum sağlama ve gelişme yeteneklerine katkıda bulunur.

* Yırtıcı hayvanlardan kaçınma, üçüncü su altı stratejisidir.

Bu strateji, Arktik deniz kuşlarının yırtıcıları yakınlarında algıladıklarında sergiledikleri davranışı tanımlamak için kullanılır. Yakınlardaki yırtıcıları algıladıklarında belirli bir ses veya çağrı kullanarak diğer deniz kuşlarını uyarır, tehlikenin varlığını işaret eder. Bu çağrı, diğer deniz kuşlarında uyanıklık uyandırarak onları tehlikeli bölgeden uzaklaşmaya teşvik eder. Aynı anda, bir yırtıcının yakınlarda olduğu algılandığında, Arktik deniz kuşları hızla pozisyonlarını değiştirir, tehlikeden kaçınmak için güvenli bir alana doğru daha büyük bir yol boyunca hızla yüzer. Bu strateji için kullanılan pozisyon güncelleme denklemi aşağıda verilmiştir:

A math equations with numbers

Description automatically generated with medium confidence

β, bu durumda 0 ile 1 arasında uniform olarak dağılmış rastgele bir sayıdır. Bu stratejide, özellikle tehlikeli durumlarla karşılaşıldığında, zeki bir dengeleme mekanizması bulunmaktadır. Bu mekanizma, iki farklı davranış modunu içerir: biri tehlikeden kaçınmayı yavaşça yaparken, diğeri hızlı bir şekilde tehlikeden sıyrılmayı sağlar. Algoritmada, bu çift-davranış stratejisi farklı durumlarda yerel optimumlardan kaçma şeklini simüle eder. rand ≥0.5 olduğunda bir yırtıcı yaklaştığını gösterirken, Kutup dalgıçları hemen kaçmayı tercih eder ve pozisyonlarını daha belirgin bir şekilde değiştirirler. Bu, bilinen bilgiyi kullanarak sıçrama yeteneğini artırmanın bir yoludur ve olası yerel optimumlarda daha iyi çözümler bulmak için algoritmaya yardımcı olur. Diğer yandan, rand < 0.5 olduğunda Kutup dalgıçları yırtıcıları proaktif bir şekilde kaçınmayı tercih eder ve davranışları daha dikkatli olma eğilimindedir. Potansiyel tehlikelerden rastgele pozisyonlarını değiştirerek kaçınırlar. Bu keşifsel bir yaklaşımdır ve algoritmanın çevresel ortamı daha hassas bir şekilde aramasına olanak tanır, üstün çözümler bulma umuduyla.

Bu çift-davranış stratejisi, algoritmanın farklı durumlarda dengeyi korumasına yardımcı olur. Bilinen bilgiye hızlı bir şekilde adapte olurken yeni alanların dikkatli keşfi için olanak tanır, böylece algoritmanın global arama kapasitesini artırır ve yerel optimumlardan kaçmasına yardımcı olur. Bu dengeleme mekanizması, algoritmanın sağlamlığını ve adaptasyon yeteneğini simüle eder.

Özetle, kutup dalgıçları su altında farklı stratejiler kullanır; bunlar arasında yem toplama, yoğun arama ve yırtıcıları kaçınma bulunur. Bu stratejiler farklı koşullar altında farklı yem arama sonuçlarına yol açabilir. Algoritma, aday pozisyonları üç farklı pozisyon denkleminden yeni bir çözüm olarak birleştirir ve çeşitli durumlarda optimal sonuçlar elde eder. Çözümler fitness'e göre sıralanır ve en iyi N birey seçilir. Denklem aşağıdaki gibidir:

A white background with black dots

Description automatically generated

Bu kapsamlı strateji, modelin birden fazla avlanma senaryosunu eş zamanlı olarak değerlendirmesine olanak tanır ve sonuç olarak en uygun konumu optimal çözüm olarak seçer. Bu, karabatakların farklı durumlarla karşılaştığında avlanma ve hayatta kalma yeteneklerini artırır.

**3.2.4. Davranış Dönüşüm Katsayısı B**

APO algoritmasında, Arktik karabataklar, iterasyonun ilk aşamasında genellikle global arama yapmak için sık sık havada uçuş gerçekleştirme eğilimindedir, iterasyonun sonraki aşamalarında ise sık sık besin aramak için dalış yapmayı tercih eder. Bu arama mekanizması, Arktik karabatağın yaşam alışkanlıklarından esinlenmiştir. İlk aşamada Arktik karabatak, uygun beslenme sulamalarını arama eğilimindedir, sonraki aşamalarda ise besin arayışına odaklanır. Bu davranışsal desene dayanarak, APO algoritması global aramadan yerel sömürüye sorunsuz bir geçiş sağlamak için davranışsal geçiş katsayısı B'yi tasarlar. Bu katsayı aşağıdaki şekilde tanımlanmıştır:



APO algoritmasında, rand (0, 1) arasında bir rastgele sayı, t ve T sırasıyla mevcut iterasyon numarası ve maksimum iterasyon numarasıdır. Algoritmadaki B parametresi, rastgele seçilemeyen ancak mevcut iterasyon numarası ile maksimum iterasyon numarası arasındaki orana dayalı olarak dinamik olarak belirlenen ve aynı zamanda Arktik deniz kuşlarının davranışındaki dinamik rastgeleliği simüle etmek için stokastisiteyi içeren adaptif olarak değişen bir parametredir. Bu tasarım, B'nin iterasyon ilerledikçe dinamik olarak ayarlanmasına izin verir, böylece farklı aşamalardaki arama ihtiyaçlarına uyum sağlar.

APO algoritmasında, mevcut iterasyon aşamasında hangi arama stratejisinin uygulanacağını belirlemek için B'nin C ile karşılaştırıldığı bir parametre olan C tanıtılmıştır. Bu çalışmada, C parametresi 0.5 olarak belirlenmiştir. Arktik deniz kuşlarının yaşam alışkanlıklarından esinlenerek, B'nin değeri C'den büyük olduğunda algoritma hava uçuşu aşamasına geçer ve geniş bir alanı keşfeder; C'ye eşit veya küçük olduğunda ise dalış ve avlanma aşamasına geçer ve yerel yakınsama stratejisini uygular. Davranışsal geçiş katsayısı B, algoritmanın keşif ve yakınsama arasında denge kurmasında kritik bir rol oynar. Bu tasarım, algoritmanın global arama ile yerel yakınsama arasında uygun bir denge sağlamasına ve performansını artırmasına yardımcı olur.

-- APO algoritması sözde kodu --

Girdi: N, T, D, F, ve C

Çıktı: en iyi Xti ve uygunluk değeri

N adet deniz papağanı, Xti (i = 1, 2, ..., N), Formül (1) i kullan.

Her Xti nin uygunluğunu değerlendirin ve Xti içinde en iyi uygunluğa sahip olanı bulun.

t = 1

while (t < T)

B davranış dönüşüm faktörünü Formül (10) a göre hesaplayın.

if B > C ; Hava uçuşu dönemi (Keşif aşaması)

for i = 1 to N

Formül (2) kullanılarak Yt+1i yi güncelleyin ; Hava araması

Formül (5) kullanılarak Zt+1i yi güncelleyin ; Dalış avcılığı

Formül (7)-(9) kullanılarak N mükemmel popülasyonu yeni popülasyon Xt+1i olarak seçin.

Deniz papağanlarını, Xt+1i, değerlendirin ve Xti nin yerine Xt+1i ile değiştirin eğer daha iyi ise.

t = t + 1

end for

else ; Dalış ve avlanma dönemi (Yakınsama aşaması)

for i = 1 to N

Formül (10) kullanılarak Wt+1i yi güncelleyin ; Yiyecek toplama

Formül (11) kullanılarak Yt+1i yi güncelleyin ; Arama güçlendirme

Formül (13) kullanılarak Zt+1i yi güncelleyin ; Yırtıcı hayvanlardan kaçınma

Eşitlik (14)-(16) kullanılarak N mükemmel popülasyonu yeni popülasyon Xt+1i olarak seçin.

Deniz kuşlarını, Xt+1i, değerlendirin ve Xti nin yerine Xt+1i ile değiştirin eğer daha iyi ise.

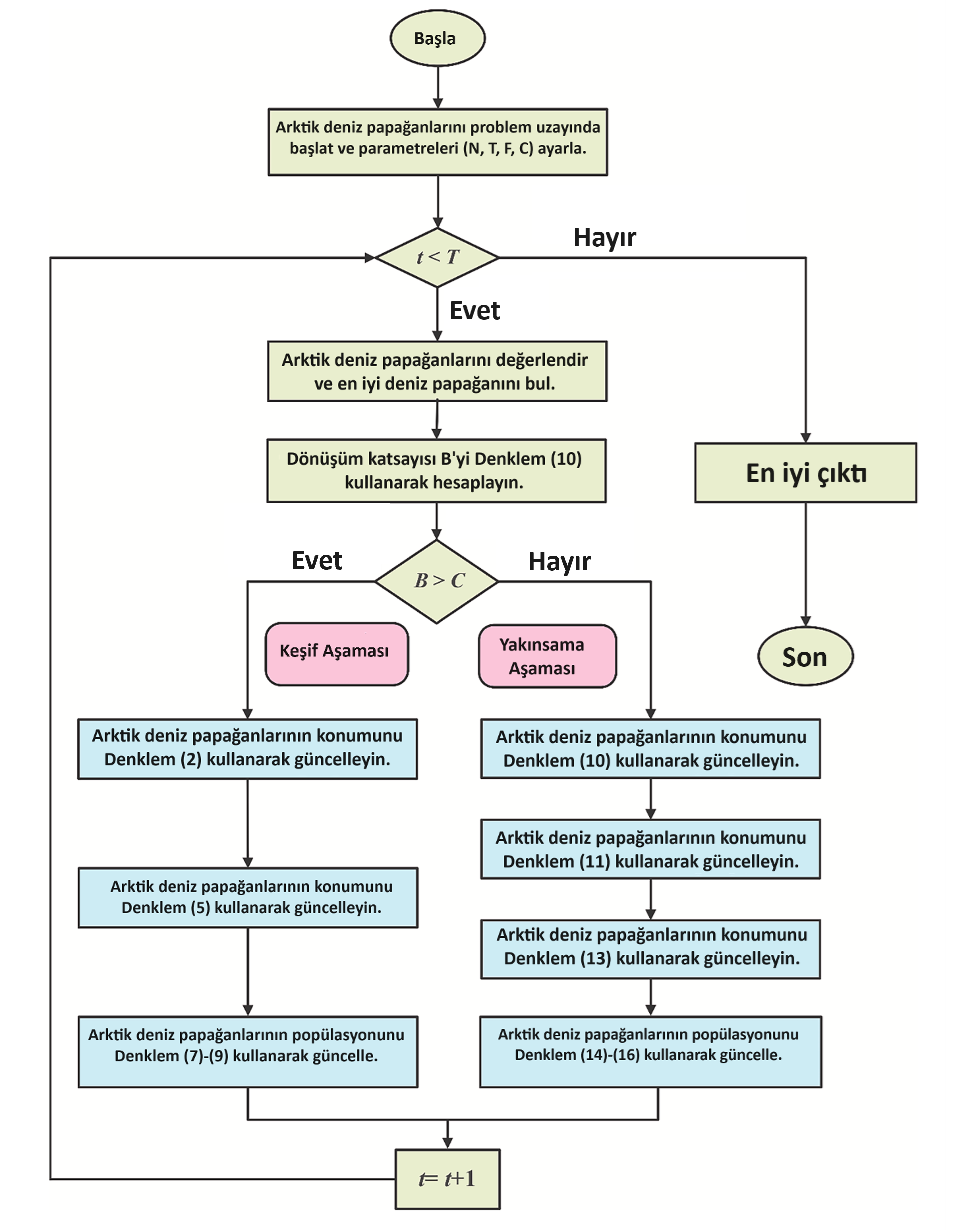
t = t + 1

end for

end if

end while

return Xti



Şekil 5 - APO Algoritması Akış Şeması

**APO.m**

function [BestF,BestX,curve]=APO(N,T,lb,ub,dim,fobj)

PopPos=zeros(N,dim); % Popülasyon pozisyonları

PopFit=zeros(N,1); % Popülasyon uygunlukları

for i=1:N

PopPos(i,:)=rand(1,dim).\*(ub-lb)+lb; % Rastgele popülasyon pozisyonları oluşturma

PopFit(i)=fobj(PopPos(i,:)); % Uygunlukları hesaplama

end

BestF=inf; % En iyi uygunluk başlangıçta sonsuz

BestX=[]; % En iyi pozisyon başlangıçta boş

for i=1:N

if PopFit(i)<=BestF

BestF=PopFit(i); % En iyi uygunluk güncelleme

BestX=PopPos(i,:); % En iyi pozisyon güncelleme

end

end

curve=zeros(T,1); % İterasyon eğrisi

%% -------------------İterasyona Başla------------------------------------%

for It=1:T

for i=1:N

theta1=(1-It/T); % İterasyon oranı

B=2\*log(1/rand)\*theta1;

%% -------------------*1.Hava* Uçuş Aşaması-------------------%

if B > 0.5

while true

K = [1:i-1, i+1:N];

RandInd = K(randi([1,N-1]));

step1 = PopPos(i, :) - PopPos(RandInd, :);

if norm(step1) ~= 0 && ~isequal(PopPos(i, :), PopPos(RandInd, :))

break;

end

end

%% -------------------1.1 Hava Arama-------------------%

Y=PopPos(i,:) + Levy(dim) .\* step1 +round(0.5\*(0.05+rand))\*randn;

%% -------------------1.2 Dalma Avlanma-------------------%

R=rand(1,dim);

step2=(R-0.5)\*pi;

S=tan(step2);

Z=Y.\*S;

Y = SpaceBound(Y, ub, lb); % Sınırları aşmama kontrolü

Z = SpaceBound(Z, ub, lb);

NewPop=[Y;Z];

NewPopfit=[fobj(Y);fobj(Z)];

[~,sorted\_indexes]=sort(NewPopfit);

newPopPos=NewPop(sorted\_indexes(1),:);

else

%% -------------------*2.Su* Altı Beslenme Aşaması-------------------%

F=0.5;

K = [1:i - 1, i + 1:N];

RandInd = K(randi([1, N - 1], 1, 3));

f=(0.1\*(rand-1)\*(T-It))/T;

while true

RandInd = K(randi([1 N-1], 1, 3));

step1 = PopPos(RandInd(2), :) - PopPos(RandInd(3), :);

if norm(step1) ~= 0 && RandInd(2) ~= RandInd(3)

break;

end

end

%% -------------------2.1 Beslenme Toplama-------------------%

if rand<0.5

W = PopPos(RandInd(1), :) + F .\* step1;

else

W = PopPos(RandInd(1), :) + F .\*Levy(dim).\* step1;

end

%% -------------------2.2 Aramayı Yoğunlaştırma-------------------%

Y=(1+f)\* W ;

%% -------------------2.3 Su Altı Beslenme Aşaması-------------------%

while true

rand\_leader\_index1 = floor(N \* rand() + 1);

rand\_leader\_index2 = floor(N \* rand() + 1);

X\_rand1 = PopPos(rand\_leader\_index1, :);

X\_rand2 = PopPos(rand\_leader\_index2, :);

step2 = X\_rand1 - X\_rand2;

if norm(step2) ~= 0 && ~isequal(X\_rand1, X\_rand2)

break;

end

end

Epsilon = unifrnd(0, 1);

if rand<0.5

Z = PopPos(i, :) + Epsilon .\* step2; % Eq.(11)4.3

else

Z = PopPos(i, :) + F .\* Levy(dim) .\* step2;

end

NewPop=[W;Y;Z];

NewPopfit=[fobj(W);fobj(Y);fobj(Z)];

[~,sorted\_indexes]=sort(NewPopfit);

newPopPos=NewPop(sorted\_indexes(1),:);

end

newPopPos = SpaceBound(newPopPos, ub, lb); % Sınır kontrolü

newPopFit = fobj(newPopPos);

if newPopFit < PopFit(i)

PopFit(i) = newPopFit; % Uygunluk güncelleme

PopPos(i, :) = newPopPos; % Pozisyon güncelleme

end

end

for i=1:N

if PopFit(i)<BestF

BestF=PopFit(i); % En iyi uygunluk güncelleme

BestX=PopPos(i,:); % En iyi pozisyon güncelleme

end

end

curve(It)=BestF; % Eğriyi güncelleme

end

end

function o=Levy(Dim)

beta=1.5;

sigma=(gamma(1+beta)\*sin(pi\*beta/2)/(gamma((1+beta)/2)\*beta\*2^((beta-1)/2)))^(1/beta);

u=randn(1,Dim)\*sigma;

v=randn(1,Dim);

step=u./abs(v).^(1/beta);

o=step;

end

**Functions\_details.m**

% lb alt sınırdır: lb=[lb\_1,lb\_2,...,lb\_d]

% ub üst sınırdır: ub=[ub\_1,ub\_2,...,ub\_d]

% dim değişken sayısıdır (problemin boyutu)

function [lb,ub,dim,fobj] = Functions\_details(F)

d=10; %%%dim=10,50,500

switch F

case 'F1'

fobj = @F1;

lb=-100;

ub=100;

dim=d;

case 'F2'

fobj = @F2;

lb=-10;

ub=10;

dim=d;

case 'F3'

fobj = @F3;

lb=-10;

ub=10;

dim=d;

case 'F4'

fobj = @F4;

lb=-10;

ub=10;

dim=d;

case 'F5'

fobj = @F5;

lb=-1.28;

ub=1.28;

dim=d;

case 'F6'

fobj = @F6;

lb=-1;

ub=1;

dim=d;

case 'F7'

fobj = @F7;

lb=-10;

ub=10;

dim=d;

case 'F8'

fobj = @F8;

lb=-5;

ub=10;

dim=d;

case 'F9'

fobj = @F9;

lb=-100;

ub=100;

dim=d;

case 'F10'

fobj = @F10;

lb=-5.12;

ub=5.12;

dim=d;

case 'F11'

fobj = @F11;

lb=-5.12;

ub=5.12;

dim=d;

case 'F12'

fobj = @F12;

lb=-50;

ub=50;

dim=d;

case 'F13'

fobj = @F13;

lb=-600;

ub=600;

dim=d;

case 'F14'

fobj = @F14;

lb=-10;

ub=10;

dim=d;

case 'F15'

fobj = @F15;

lb=-10;

ub=10;

dim=d;

case 'F16'

fobj = @F16;

lb=-5;

ub=5;

dim=d;

case 'F17'

fobj = @F17;

lb=-2;

ub=2;

dim=d;

case 'F18'

fobj = @F18;

lb=[-5,0];

ub=[10,15];

dim=2;

end

end

function o = F1(x)

o=sum(x.^2);

end

function o = F2(x)

o=sum(abs(x))+prod(abs(x));

end

function o = F3(x)

dim=size(x,2);

o=0;

for i=1:dim

o=o+sum(x(1:i))^2;

end

end

function o = F4(x)

o=max(abs(x));

end

function o = F5(x)

dim=size(x,2);

o=sum((1:dim).\*(x.^4))+rand;

end

function o = F6(x)

dim=size(x,2);

o=0;

for i=1:dim

o=o+(abs(x(i))^(i+1));

end

end

function o = F7(x)

dim=size(x,2);

o=0;

for i=1:dim

o=o+i\*x(i)^2;

end

end

function o = F8(x)

D = size(x,2);

o = sum(x.^2)+sum(0.5\*D\*(x.^2))+sum(0.5\*D\*(x.^4));

end

function o = F9(x)

dim=size(x,2);

o=sum(0.5\*dim\*x.^4)+rand;

end

function o = F10(x)

dim=size(x,2);

o=sum(x.^2-10\*cos(2\*pi.\*x))+10\*dim;

end

function o = F11(x)

dim=size(x,2);

o=0;

for i=1:dim

if abs(x(i))<0.5

o=o+x(i)^2-10\*cos(2\*pi\*x(i))+10;

else

o=o+(round(2\*x(i))/2)^2-10\*cos(2\*pi\*round(2\*x(i))/2)+10;

end

end

end

function o = F12(x)

dim=size(x,2);

o=-20\*exp(-.2\*sqrt(sum(x.^2)/dim))-exp(sum(cos(2\*pi.\*x))/dim)+20+exp(1);

end

function o = F13(x)

dim=size(x,2);

o=sum(x.^2)/4000-prod(cos(x./sqrt(1:dim)))+1;

end

function o = F14(x)

o=sum(abs(x.\*sin(x)+*0.1.*\*x));

end

function o = F15(x)

dim=size(x,2);

a = 0.5;

b = 3;

kmax = 20;

c1(1:kmax+1) = a.^(0:kmax);

c2(1:kmax+1) = 2\*pi\*b.^(0:kmax);

o=0;

for i=1:dim

o=o+w(x(:,i)',c1,c2);

end

function y = w(x,c1,c2)

y = zeros(length(x),1);

for k = 1:length(x)

y(k) = sum(c1 .\* cos(c2.\*(x(:,k)+0.5)))-k\*sum(c1 .\* cos(c2\*0.5));

end

end

end

function o = F16(x)

o=1-cos(2\*pi\*sqrt(sum(x.^2)))+0.1\*sum(x.^2);

end

function o = F17(x)

dim=size(x,2);

o=0;

for i=1:dim-1

o=o+x(i)^2+2.\*x(i+1)^2-0.3\*cos(2\*pi\*x(i))-0.4\*cos(4\*pi\*x(i+1))+0.7;

end

end

function o=Ufun(x,a,k,m)

o=k.\*((x-a).^m).\*(x>a)+k.\*((-x-a).^m).\*(x<(-a));

end

function o = F18(x)

o=(x(2)-(x(1)^2)\*5.1/(4\*(pi^2))+5/pi\*x(1)-6)^2+10\*(1-1/(8\*pi))\*cos(x(1))+10;

end

**initialization.m**

% Bu fonksiyon, arama ajanlarının ilk popülasyonunu başlatır

function Positions=initialization(SearchAgents\_no,dim,ub,lb)

Boundary\_no= size(ub,2); % sınırların sayısı

% Eğer tüm değişkenlerin sınırları eşitse ve kullanıcı hem ub hem de lb için tek bir sayı girerse

if Boundary\_no==1

Positions=rand(SearchAgents\_no,dim).\*(ub-lb)+lb;

end

% Eğer her değişkenin farklı bir alt ve üst sınırı varsa

if Boundary\_no>1

for i=1:dim

ub\_i=ub(i);

lb\_i=lb(i);

Positions(:,i)=rand(SearchAgents\_no,1).\*(ub\_i-lb\_i)+lb\_i;

end

end

**SpaceBound.m**

function X=SpaceBound(X,Up,Low)

Dim=length(X);

S=(X>Up)+(X<Low);

X=(rand(1,Dim).\*(Up-Low)+Low).\*S+X.\*(~S);

**Main.m**

% MATLAB R2022b de geliştirildi

% Kaynak kodları

% \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

clear % Çalışma alanını temizle

clc % Komut penceresini temizle

close all % Tüm grafik pencerelerini kapat

%% Parametrelerin Tanımlanması

N = 30; % Arama ajanlarının sayısı

T = 1000; % Maksimum iterasyon sayısı

F\_name = 'F13'; % Test fonksiyonunun adı

% Seçilen benchmark fonksiyonunun ayrıntılarını yükle

[lb, ub, D, fobj] = Functions\_details(F\_name);

% APO algoritması ile en iyi çözümü bul

[Best\_Fitness, Best\_Pos, Convergence\_curve] = APO(N, T, lb, ub, D, fobj);

%% Hesaplama Sonuçlarının Gösterilmesi

display(['En iyi uygunluk değeri:', num2str(Best\_Fitness)]);

display(['En iyi pozisyon:', num2str(Best\_Pos)]);

Kaynaklar

[1] Nelder JA, Mead R. A simplex method for function minimization. Comput J 1965;7:

308–13.

[2] Mandic DP. A generalized normalized gradient descent algorithm. IEEE Signal Proc

Lett 2004;11:115–8.

[3] Ezugwu AE, Shukla AK, Nath R, Akinyelu AA, Agushaka JO, Chiroma H, et al.

Metaheuristics: a comprehensive overview and classification along with

bibliometric analysis. Artif Intell Rev 2021;54:4237–316.

[4] Guan Z, Ren C, Niu J, Wang P, Shang Y. Great wall construction algorithm: a novel

meta-heuristic algorithm for engineer problems. Expert Syst Appl 2023;233:

120905.

[5] Goodarzimehr V, Shojaee S, Hamzehei-Javaran S, Talatahari S. Special relativity

search: a novel metaheuristic method based on special relativity physics. Knowl

Based Syst 2022;257:109484.

[6] Mirjalili S, Mirjalili SM, Lewis A. Grey Wolf Optimizer. Adv Engineer Software

2014;69:46–61.

[7] Dhiman G, Kumar V. Seagull optimization algorithm: theory and its applications

for large-scale industrial engineering problems. Knowl Based Syst 2019;165:

169–96.

[8] Alweshah M, Khalaileh SA, Gupta BB, Almomani A, Hammouri AI, Al-Betar MA.

The monarch butterfly optimization algorithm for solving feature selection

problems. Neural Comput Applicat 2022;34:11267–81.

[9] Wang W, Tian W, Chau K-w, Xue Y, Xu L, Zang H. An improved bald eagle search

algorithm with cauchy mutation and adaptive weight factor for engineering

optimization. Comp Model Engineer Sci 2023;136:1603–42.

[10] Wang W, Tian W, Chau K, Zang H, Ma M, Feng Z, et al. Multi-reservoir flood

control operation using improved bald eagle search algorithm with ε constraint

method. Water (Basel) 2023.

[11] W-c Wang, W-c Tian, d-m Xu, K-w Chau, Ma Q, C-j Liu. Muskingum models’

development and their parameter estimation: a state-of-the-art review. Water Res

Managem 2023;37:3129–50.

[12] Zhao S, Zhang T, Ma S, Wang M. Sea-horse optimizer: a novel nature-inspired

meta-heuristic for global optimization problems. Appl Intelligence 2023;53:

11833–60.

[13] Aydemir SB. Enhanced marine predator algorithm for global optimization and

engineering design problems. Adv Engineer Software 2023;184:103517.

[14] Gharehchopogh FS, Abdollahzadeh B, Barshandeh S, Arasteh B. A multi-objective

mutation-based dynamic Harris Hawks optimization for botnet detection in IoT. Int

Things 2023;24:100952.

[15]¨

Ozbay E. An active deep learning method for diabetic retinopathy detection in

segmented fundus images using artificial bee colony algorithm. Artif Intell Rev

2023;56:3291–318.

[16] Gharehchopogh FS. An improved harris hawks optimization algorithm with multi-

strategy for community detection in social network. J Bionic Eng 2023;20:

1175–97.

[17]¨

Ozbay E, ¨

Ozbay FA, Gharehchopogh FS. Peripheral blood smear images

classification for acute lymphoblastic leukemia diagnosis with an improved

convolutional neural network. J Bionic Eng 2023.

[18] Gharehchopogh FS, Ibrikci T. An improved African vultures optimization

algorithm using different fitness functions for multi-level thresholding image

segmentation. Multimed Tools Appl 2024;83:16929–75.

[19] Wolpert DH, Macready WG. No free lunch theorems for optimization. IEEE Transac

Evolution Computat 1997;1:67–82.

[20] Holland JH. Genetic Algorithms. Sci Am 1992;267:66–73.

[21] Storn R, Price K. Differential evolution – a simple and efficient heuristic for global

optimization over continuous spaces. J Glob Optimizat 1997;11:341–59.

[22] Beyer H-G, Schwefel H-P. Evolution strategies – A comprehensive introduction. Nat

Comput 2002;1:3–52.

[23] Simon D. Biogeography-Based Optimization. IEEE Transac Evolution Computat

2008;12:702–13.

[24] Zhang J-H, Xu X-H. An efficient evolutionary programming algorithm. Comput

Oper Res 1999;26:645–63.

[25] Faramarzi A, Heidarinejad M, Stephens B, Mirjalili S. Equilibrium optimizer: a

novel optimization algorithm. Knowl Based Syst 2020;191:105190.

[26] Hashim FA, Hussain K, Houssein EH, Mabrouk MS, Al-Atabany W. Archimedes

optimization algorithm: a new metaheuristic algorithm for solving optimization

problems. Appl Intelligence 2021;51:1531–51.

[27] Abualigah L, Diabat A, Mirjalili S, Abd Elaziz M, Gandomi AH. The arithmetic

optimization algorithm. Comput Methods Appl Mech Eng 2021;376:113609.

[28] Eskandar H, Sadollah A, Bahreininejad A, Hamdi M. Water cycle algorithm – A

novel metaheuristic optimization method for solving constrained engineering

optimization problems. Comput Struct 2012;110-111:151–66.

W.-c. Wang et al.Advances in Engineering Software 195 (2024) 103694

44

[29] Abdel-Basset M, Mohamed R, Azeem SAA, Jameel M, Abouhawwash M. Kepler

optimization algorithm: a new metaheuristic algorithm inspired by Kepler’s laws of

planetary motion. Knowl Based Syst 2023;268:110454.

[30] Hatamlou A. Black hole: a new heuristic optimization approach for data clustering.

Info Sci 2013;222:175–84.

[31] Rashedi E, Nezamabadi-pour H, Saryazdi SGSA. A gravitational search algorithm.

Info Sci 2009;179:2232–48.

[32] Abdel-Basset M, El-Shahat D, Jameel M, Abouhawwash M. Young’s double-slit

experiment optimizer: a novel metaheuristic optimization algorithm for global and

constraint optimization problems. Comput Methods Appl Mech Eng 2023;403:

115652.

[33] Zong Woo G, Joong Hoon K, Loganathan GV. A new heuristic optimization

algorithm: harmony search. Simulation 2001;76:60–8.

[34] Shi Y. Brain Storm Optimization Algorithm. In: Tan Y, Shi Y, Chai Y, Wang G,

editors. Advances in Swarm Intelligence Berlin. Heidelberg: Springer Berlin

Heidelberg; 2011. p. 303–9.

[35] Das B, Mukherjee V, Das D. Student psychology based optimization algorithm: a

new population based optimization algorithm for solving optimization problems.

Adv Engineer Software 2020:146.

[36] Rao RV, Savsani VJ, Vakharia DP. Teaching–learning-based optimization: a novel

method for constrained mechanical design optimization problems. Comp-Aid Des

2011;43:303–15.

[37] Askari Q, Younas I, Saeed M. Political Optimizer: a novel socio-inspired meta-

heuristic for global optimization. Knowl Based Syst 2020;195:105709.

[38] Atashpaz-Gargari E, Lucas C. Imperialist competitive algorithm: an algorithm for

optimization inspired by imperialistic competition. In: 2007 IEEE Congress on

Evolutionary Computation; 2007. p. 4661–7.

[39] Kennedy J, Eberhart R. Particle swarm optimization. In: Proceedings of ICNN’95 -

International Conference on Neural Networks. 4; 1995. p. 1942–8. vol.

[40] Dorigo M, Birattari M, Stutzle T. Ant colony optimization. IEEE Comput Intell Mag

2006;1:28–39.

[41] Mirjalili S, Lewis A. The whale optimization algorithm. Adv Engineer Software

2016;95:51–67.

[42] Heidari AA, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen H. Harris hawks

optimization: algorithm and applications. Future Generat Comp Sys 2019;97:

849–72.

[43] Zhong C, Li G, Meng Z. Beluga whale optimization: a novel nature-inspired

metaheuristic algorithm. Knowl Based Syst 2022;251:109215.

[44] Abdel-Basset M, Mohamed R, Zidan M, Jameel M, Abouhawwash M. Mantis search

algorithm: a novel bio-inspired algorithm for global optimization and engineering

design problems. Comput Methods Appl Mech Eng 2023;415:116200.

[45] Wang L, Cao Q, Zhang Z, Mirjalili S, Zhao W. Artificial rabbits optimization: a new

bio-inspired meta-heuristic algorithm for solving engineering optimization

problems. Eng Appl Artif Intell 2022;114:105082.

[46] Mohammadi-Balani A, Nayeri MD, Azar A, Taghizadeh-Yazdi M. Golden eagle

optimizer: a nature-inspired metaheuristic algorithm. Comput Ind Eng 2021:152.

[47] Li S, Chen H, Wang M, Heidari AA, Mirjalili S. Slime mould algorithm: a new

method for stochastic optimization. Future Generat Comp Sys 2020;111:300–23.

[48] Abdollahzadeh B, Khodadadi N, Barshandeh S, Trojovský P, Gharehchopogh FS, El-

kenawy E-SM, et al. Puma optimizer (PO): a novel metaheuristic optimization

algorithm and its application in machine learning. Cluster Comput 2024.

[49]¨

Ozbay FA. A modified seahorse optimization algorithm based on chaotic maps for

solving global optimization and engineering problems. Engineer Sci Technol, an Int

J 2023;41:101408.

[50] Gharehchopogh FS, Nadimi-Shahraki MH, Barshandeh S, Abdollahzadeh B,

Zamani H. CQFFA: a chaotic quasi-oppositional farmland fertility algorithm for

solving engineering optimization problems. J Bionic Eng 2023;20:158–83.

[51] Shen B, Khishe M, Mirjalili S. Evolving Marine Predators Algorithm by dynamic

foraging strategy for real-world engineering optimization problems. Eng Appl Artif

Intell 2023;123:106207.

[52] Khishe M. Greedy opposition-based learning for chimp optimization algorithm.

Artif Intell Rev 2023;56:7633–63.

[53] Saffari A, Zahiri SH, Khishe M. Fuzzy whale optimisation algorithm: a new hybrid

approach for automatic sonar target recognition. J Experim Theoret Artificial

Intelligence 2023;35:309–25.

[54] Bo Q, Cheng W, Khishe M. Evolving chimp optimization algorithm by weighted

opposition-based technique and greedy search for multimodal engineering

problems. Appl Soft Comput 2023;132:109869.

[55] Gharehchopogh FS, Ghafouri S, Namazi M, Arasteh B. Advances in manta ray

foraging optimization: a comprehensive survey. J Bionic Eng 2024;21:953–90.

[56] Zamani H, Nadimi-Shahraki MH, Mirjalili S, Soleimanian Gharehchopogh F,

Oliva D. A critical review of moth-flame optimization algorithm and its variants:

structural reviewing, performance evaluation, and statistical analysis. Arch

Computat Methods Engineer 2024.

[57] Khishe M, Orouji N, Mosavi MR. Multi-objective chimp optimizer: an innovative

algorithm for multi-objective problems. Exp Syst Appl 2023;211:118734.

[58] Wang B, Liu L, Li Y, Khishe M. Robust grey wolf optimizer for multimodal

optimizations: a cross-dimensional coordination approach. J Sci Comput 2022;92:

110.

[59] Liu L, Khishe M, Mohammadi M, Hussein Mohammed A. Optimization of constraint

engineering problems using robust universal learning chimp optimization. Adv

Engineer Inform 2022;53:101636.

[60] Gong S-P, Khishe M, Mohammadi M. Niching chimp optimization for constraint

multimodal engineering optimization problems. Exp Syst Appl 2022;198:116887.

[61] Kaidi W, Khishe M, Mohammadi M. Dynamic levy flight chimp optimization.

Knowl Based Syst 2022;235:107625.

[62] Khishe M, Nezhadshahbodaghi M, Mosavi MR, Martín D. A weighted chimp

optimization algorithm. IEEE Access 2021;9:158508–39.

[63] Wang J, Khishe M, Kaveh M, Mohammadi H. Binary chimp optimization algorithm

(BChOA): a new binary meta-heuristic for solving optimization problems. Cognit

Comput 2021;13:1297–316.

[64] Qian L, Khishe M, Huang Y, Mirjalili S. SEB-ChOA: an improved chimp

optimization algorithm using spiral exploitation behavior. Neural Comput Applicat

2024;36:4763–86.

[65] Burnham KK, Burnham JL, Johnson JA, Huffman A. Migratory movements of

Atlantic puffins Fratercula arctica naumanni from high Arctic Greenland. PLoS One

2021;16:e0252055.

[66] Guilford T, Freeman R, Boyle D, Dean B, Kirk H, Phillips R, et al. A dispersive

migration in the atlantic puffin and its implications for migratory navigation. PLoS

One 2011;6:e21336.

[67] Fayet AL, Clucas GV, Anker-Nilssen T, Syposz M, Hansen ES. Local prey shortages

drive foraging costs and breeding success in a declining seabird, the Atlantic puffin.

J Animal Ecol 2021;90:1152–64.

[68] Zhong C, Li G, Meng Z, He W. Opposition-based learning equilibrium optimizer

with Levy flight and evolutionary population dynamics for high-dimensional global

optimization problems. Exp Syst Appl 2023;215:119303.

[69] He Q, Liu H, Ding G, Tu L. A modified L´

evy flight distribution for solving high-

dimensional numerical optimization problems. Math Comput Simul 2023;204:

376–400.

[70] Wang Z, Chen Y, Ding S, Liang D, He H. A novel particle swarm optimization

algorithm with L´

evy flight and orthogonal learning. Swarm Evol Comput 2022;75:

101207.

[71] Wang Y, Ran S, Wang G-G. Role-oriented binary grey wolf optimizer using

foraging-following and L´

evy flight for feature selection. Appl Math Modelling

2024;126:310–26.

[72] Awad N.H., Ali M.Z., Suganthan P.N., Liang J.J., Qu B.Y. Problem definitions and

evaluation criteria for the CEC 2017 competition and special session on

constrained single objective real-parameter optimization. 2016.

[73] Kumar A., Price K.V., Mohamed A.W., Hadi A.A., Suganthan P.N. Problem

definitions and evaluation criteria for the CEC 2022 special session and

competition on single objective bound constrained numerical optimization. 2021.

[74] Abdollahzadeh B, Gharehchopogh FS, Mirjalili S. African vultures optimization

algorithm: a new nature-inspired metaheuristic algorithm for global optimization

problems. Comput Ind Eng 2021;158:107408.

[75] Abdollahzadeh B, Soleimanian Gharehchopogh F, Mirjalili S. Artificial gorilla

troops optimizer: a new nature-inspired metaheuristic algorithm for global

optimization problems. Int J Intell Syst 2021;36:5887–958.

[76] Abualigah L, Elaziz MA, Sumari P, Geem ZW, Gandomi AH. Reptile Search

Algorithm (RSA): a nature-inspired meta-heuristic optimizer. Exp Syst Appl 2022;

191:116158.

[77] Ahmadianfar I, Heidari AA, Gandomi AH, Chu X, Chen H. RUN beyond the

metaphor: an efficient optimization algorithm based on Runge Kutta method.

Expert Syst Appl 2021;181:115079.

[78] Hansen N, Müller SD, Koumoutsakos P. Reducing the time complexity of the

derandomized evolution strategy with covariance matrix adaptation (CMA-ES).

Evol Comput 2003;11:1–18.

[79] Martin L, Leblanc R, Toan NK. Tables for the Friedman rank test. Canadian J Stat

1993;21:39–43.

[80] Kumar A, Wu G, Ali MZ, Mallipeddi R, Suganthan PN, Das S. A test-suite of non-

convex constrained optimization problems from the real-world and some baseline

results. Swarm Evol Comput 2020;56:100693.

[81] Abdel-Basset M, Mohamed R, Jameel M, Abouhawwash M. Nutcracker optimizer: a

novel nature-inspired metaheuristic algorithm for global optimization and

engineering design problems. Knowl Based Syst 2023;262:110248.

[82] Zamani H, Nadimi-Shahraki MH, Gandomi AH. Starling murmuration optimizer: a

novel bio-inspired algorithm for global and engineering optimization. Comput

Methods Appl Mech Eng 2022;392:114616.

[83] Faramarzi A, Heidarinejad M, Mirjalili S, Gandomi AH. Marine Predators

Algorithm: a nature-inspired metaheuristic. Expert Syst Appl 2020;152:113377.

[84] Li K, Huang H, Fu S, Ma C, Fan Q, Zhu Y. A multi-strategy enhanced northern

goshawk optimization algorithm for global optimization and engineering design

problems. Comput Methods Appl Mech Eng 2023;415:116199.

[85] McCarthy GT. The unit hydrograph and flood routing. Proc Confer North Atlantic

Division, US Army Corps Engineers 1938;1938:608–9.

[86] Wang W, Tian W, Xu L, Liu C, Xu D. Mε-OIDE algorithm for solving constrained

optimization problems and its application in flood control operation of reservoir

group. Shuili Xuebao/J Hydraulic Engineer 2023;54:148–58.