

METASEZGİSEL ALGORİTMALAR VE ÇALIŞMA ALANLARINA GENEL BİR BAKIŞ

Cem Uğurcan ÖZDEMİR¹, Atanur İZ², İlker KILIÇ³, Mustafa NİL⁴

Accepted: 2024-12-27
DOI: 10.47118/somatbd.1593895

ÖZET

Bu makalede optimizasyon algoritmalarının farklı türleri ayrıntılı bir şekilde tanıtılmış ve son yıllarda kullanım oranlarında artış gözlenen bu algoritmaların çıkış mantığı, temel prensipleri ve optimizasyon sürecindeki adımları açıklanmıştır. Ayrıca yerel optimum noktaya takılma potansiyeline sahip olan algoritmalar vurgulanmış ve bu durumu iyileştirmeye çalışan çalışmalardan bazılarının yer verilmiştir. Optimizasyon algoritmalarının geniş bir yelpazeye yayılan uygulama alanlarına değinilmiştir. Algoritmaların hem teorik çerçevesi hem de pratikteki etkileri ele alınmış, bu alandaki yenilikler ve mevcut çalışmaları tanıtacak bilgiler sunulmuştur.

Anahtar Kelimeler: Metasezgisel optimizasyon algoritmaları, Gri kurt algoritması, Parçacık sürü optimizasyon algoritması, Yapay arı kolonisi, Meyve sineği algoritması

ABSTRACT

In this article, different types of optimization algorithms are introduced in detail and the output logic, basic principles and optimization process steps of these algorithms, which have been increasing in usage rates in recent years, are explained. In addition, algorithms that have the potential to get stuck at local optimum points are highlighted and some of the studies that try to improve this situation are given. A wide range of application areas of optimization algorithms are discussed. Both the theoretical framework and practical effects of the algorithms are discussed, and information is presented to introduce innovations and existing studies in this field.

Keywords: Metaheuristic optimization algorithms, Gray wolf algorithm, Particle swarm optimization algorithm, Artificial bee colony, Fruitfly algorithm

1. GİRİŞ

Optimizasyon varolan bir problemin çözüm havuzundan en olası çözümü bulmak demektir. Optimizasyon işlemi bu amaç doğrultusunda hareket eder. Bu amaca yönelik optimizasyon algoritmaları oluşturulmuştur. Algoritma kelimesi tanım olarak bir problemi çözme amacıyla

¹ Manisa Celal Bayar Üniversitesi, Mühendislik Fakültesi, Elektrik Elektronik Mühendisliği, Manisa, Elektrik Elektronik Mühendisi, ORCID : 0009-0005-0340-6369

² Volt Elektrik Motorları, Üretim Müdürü, ORCID : 0000-0001-5102-9792

³ Manisa Celal Bayar Üniversitesi, Mühendislik Fakültesi, Elektrik Elektronik Mühendisliği, Manisa, Dr. Öğretim Üyesi, ORCID : 0000-0003-3978-4829

⁴ Manisa Celal Bayar Üniversitesi, Mühendislik Fakültesi, Elektrik Elektronik Mühendisliği, Manisa, Dr. Öğretim Üyesi, ORCID : 0000-0003-0794-7603

tasarlanmış işlem adımları dizisidir. Optimizasyon algoritmaları kesin ve yaklaşık olmak üzere iki çeşittir.(Yaylacı vd., 2023) Kesin metotlar doğrusal, tamsayı ve dinamik programlama örnek verilebilir. Yaklaşık metotlar da iç yapısında sezgisel ve metasezgisel olarak çeşitlenmiştir. Sezgisel optimizasyon algoritmalarını çözüm kümesinin geniş olduğu yerlerde kullanmak çözüm maliyeti açısından daha karlıdır. Metasezgisel algoritmalar sonuca ulaşma konusunda sezgisel algoritmalara göre daha etkilidir. Metasezgisel optimizasyon algoritmaları global en iyiyi bulmayı amaçlar.(Eroğlu vd., 2024) Uygulama alanı açısından çok geniş bir alana sahiptirler. Karmaşık sistemlerin tasarım ve üretim süreçlerinde, verimlilikte, finans alanında, ticaret stratejilerinde, biyoinformatikte, yapay zeka ve makine öğrenmesinde kullanılırlar.

2. Metasezgisel Optimizasyon Algoritmaları ve Uygulama Alanları

2.1. Gri Kurt Optimizasyon Algoritması

Gri kurtlar, Canidae familyasından gelirler. Besin zincirinin en üst katmanında yer almaktadırlar ve sürü halinde yaşamayı benimsemişlerdir. Bir sürünün ortalama büyüklüğü 5 ila 12 arasındadır. Gri kurtların sürüdeki hiyerarşik düzeni çok baskıcıdır. Hiyerarşik düzenin tepesinde alfa diye hitap edilen bir adet erkek lider ve bir adet dişi lider bulunmaktadır. Bu liderler sürü için önemli kararları vermek ile yükümlüdürler. Aynı zamanda bu liderler sürüdeki diğer kurtların takibini de yürütürler.(Mirjalili vd., 2014)

Bu sürüde güç ikinci planda yer almaktadır çünkü sürünün liderleri en güçlüler arasından değil sürüyü en iyi yönetebilme kabiliyetine sahip olanlar arasından seçilir. Hiyerarşik düzenin ikinci basamağını betalar olarak adlandırılan kurtlar oluşturur. Betalar cinsiyet fark etmeksizin alfa kurtlar için geri bildirim görevini üstlenirler. Hiyerarşik düzenin en alt basamağında omega sınıfı yer alır. Bu kurtlar emirlere itaat etmekle yükümlüdürler. Bu sınıflardan herhangi birinde yer almayan kurtlara ast denir. Bu kurtlar omegalara söz geçirebilirler ancak alfa ve betaların emirlerini yerine getirmelidirler. Bu sınıfta iz sürücüler, muhafızlar, yaşlı kurtlar, avcılar ve bakıcılar yer alır.(Mirjalili vd., 2014)

Hiyerarşiye ek olarak gri kurtların avlanma stratejileri merak uyandırıcıdır. Önce avlarını takip ederler, daha sonrasında av hareket edemeyecek hale geldiği anda avın etrafını çevreler ve farklı yönlerden avı saldırırlar. Gri kurt optimizasyon algoritması da bu avlanma stratejisini ve hiyerarşiyi kullanır böylelikle küresel en iyiyi bulmayı amaçlar.(Mirjalili vd., 2014)

Gri kurt algoritması, özellikle mühendislik optimizasyonu ve karmaşık problemlerde son yıllarda popüleritesini oldukça arttırmıştır.

2.1.1. Gri Kurt Optimizasyon Algoritması Çalışma Yapısı ve Uygulama Alanları

Problemin çözüm kümesi alfa olarak kabul edilir. Ardından diğer çözümler sırası ile ikinci çözüm beta üçüncü çözüm delta ve diğer aday çözümler omegadır. Omega kurtları diğer üç çözümü takip ederler. Ve bu üç çözüm algoritmayı yönlendirir. Avı çevreleme stratejisinde (1) ve (2) 'den yararlanır.(Mirjalili vd., 2014)

$$\vec{D} = |\vec{C}\vec{X}_p(t) - \vec{X}(t)| \quad (1)$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D} \quad (2)$$

Burada t mevcut yinelemeyi, \vec{A} ve \vec{C} katsayı vektörlerini, \vec{x}_p avın pozisyon vektörünü ve \vec{X} gri kurdun pozisyon vektörünü temsil eder. Katsayı vektörü hesaplaması için (3) ve (4)'ten yararlanılır.(Mirjalili vd., 2014)

$$\vec{A} = 2 \cdot \vec{a} \cdot \vec{r}_1 - \vec{a} \quad (3)$$

$$\vec{C} = 2 \cdot \vec{r}_2 \quad (4)$$

\vec{a} 'nın bileşenleri yineleme sayısının artmasıyla beraber 2'den 0'a doğru doğrusal bir şekilde azaltılır. r vektörleri rastgele olarak seçilir. Bu denklemler ile bozkurt yeni pozisyonunu güncelleyebilir hale gelir. Avlanma stratejisinde arama uzayında en iyi hakkında bir bilgi birikimi yoktur. Alfa, beta ve deltanın değerlerinden elde edilen en uygun değer kaydedilir. Omega kurtlar da optimum çözümlere göre konum güncellemesi işlemine tabidirler. Konum güncelleme denklemleri (5)-(11)'de belirtilmiştir.(Mirjalili vd., 2014)

$$\vec{D}_\alpha = |\vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}| \quad (5) \quad \vec{X}_2 = \vec{X} - \vec{A}_2 \cdot (\vec{D}_\beta) \quad (9)$$

$$\vec{D}_\beta = |\vec{C}_2 \cdot \vec{X}_\beta - \vec{X}| \quad (6) \quad \vec{X}_3 = \vec{X}_\delta - \vec{A}_3 \cdot (\vec{D}_\delta) \quad (10)$$

$$\vec{D}_\delta = |\vec{C}_3 \cdot \vec{X}_\delta - \vec{X}| \quad (7) \quad \vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad (11)$$

$$\vec{X}_1 = \vec{X}_\alpha - A_1 \cdot (\vec{D}_\alpha) \quad (8)$$

Alfa, beta ve delta kurtları avlanacak canlının konumunu tahmin etmekle omegalar ise bu tahminler etrafında hareket etmekle yükümlüdürler. Kurtlar ava yaklaştıkça \vec{A} değeri düşürülür. Kurtlar avlarını ararlarken birbirlerinden ayrılırlar ve avlanmak için birleşirler. (Mirjalili vd., 2014)

Yerele takılmayı engelleme amacıyla \vec{A} değerine $(\infty, 1) \cup (-\infty, -1)$ aralığında değerler verilir. Böylelikle küreselde bir arama yapılmış olur. Algoritma adımlarını gri kurt popülasyonu rastgele bir şekilde arama uzayında oluşturulur. İterasyonlar sırasında alfa, beta ve delta kurtları konum tahmini yaparlar. Aday çözümler ortaya çıkar iterasyonlar sonucunda çözümlerin konumları ve ava olan mesafe de güncellenir. Sonuç olarak algoritmayı sonlandırma kriteri ile işlem biter.(Mirjalili vd., 2014)

Gri kurt algoritması farklı uygulama alanlarında da kullanılmıştır.

Örneğin transformatörün performansında etkili olan kütle ve boyut parametrelerini şekillendiren akım yoğunluğu, manyetik çekirdek kesit alanı ve uyumluluk katsayısı, transformatör tasarım kriterlerinin iyileştirilmesinde kritik bir rol üstlenmektedir. Bu parametrelerin birlikte ele alınarak optimize edilmesinde, doğadan esinlenen yöntemlerden biri olan gri kurt algoritması kullanılmıştır. Gri kurt algoritması, çoklu hedeflerin dengeli bir şekilde iyileştirilmesi için hızlı ve etkili bir yöntem sunmaktadır.(Toren vd., 2023) Bir başka çalışmada ise, düzensiz şekilli iki boyutlu ayakkabı kalıbının, belirli geometrik kısıtlamalar ve verimlilik hedefleri dikkate alınarak belirsiz şekilli deri yüzeyine optimum yerleştirilmesi işleminde faydalanılmıştır. Bu süreçte, deri yüzeyinin maksimum kullanımını sağlamak ve kaybı en aza

indirmek için çeşitli hesaplama yöntemleri ve algoritmalar uygulanmıştır.(Şenel vd., 2018) Diğer bir çalışmada araçtan şebekeye doğru enerji transferi yapan şarj istasyonlarının güç dönüşüm verimliliğini artırmak ve enerji kayıplarını en aza indirmek amacıyla gri kurt optimizasyon algoritmasından yararlanılmıştır. Bu süreçte, şarj istasyonlarının dönüştürücü devre tasarımı, akım ve gerilim kontrolü ile termal yönetim gibi kritik parametreler optimize edilerek sistem performansı iyileştirilmiştir. Gri kurt algoritması, çoklu hedeflerin dengelenmesi ve hızlı çözüm üretme kabiliyeti sayesinde bu tür karmaşık optimizasyon problemleri için etkili bir yaklaşım sunmuştur.(Sarıdağ vd., 2021) Gri kurt algoritmasını geliştirmek için de çok sayıda çalışma yapılarak kararsızlığı azaltmak, yüksek doğruluğa ulaşmasını sağlamak ve daha hızlı yakınsama hızına erişmesi için kaotik çadır haritalaması iyileştirilmiştir.(Hou vd., 2022)

2.2 Meyve Sineği Algoritması

Bu algoritma temelde meyve sineklerinin kendine özgü yiyecek arayışına çıkma ve yiyeceği tespit etme davranışlarından üretilmiştir. Meyve sineklerinin yapısında olan osfrezis organından dolayı çok iyi bir koku algılama hassasiyetine sahiptirler. Hatta havada yayılan kokuyu 40 kilometre uzaktan bile fark edebilirler.(Pan, 2012)

Meyve sineklerinin bir başka önemli özelliği olan koku algılama ile aynı hassaslıktaki görüşleridir. Belirli bir mesafeden sonra yiyecek kaynağını koklama duyuları ile tespit etme yerini görme duyuları ile tespit etmeye bırakır.(Pan, 2012)

2.2.1 Meyve Sineği Optimizasyon Algoritması Çalışma Yapısı ve Uygulama Alanları

Meyve sineği optimizasyon algoritmasının çalışma yapısı net anlaşılabilir düzeydedir. Meyve sineği popülasyonu mevcuttur ve buradaki her sineğin iterasyon sonundaki konumu çözüm kümesini oluşturur. Algoritmada hesaplanması gereken değerler koku, mesafe ve uygunluk değeri yani çözümdür.

Koku değeri mesafeye göre değişkenlik gösteren bir değerdir.(Pan, 2012) Bu iki değişkenin hesabı (12) ve (13)'de gösterilmiştir.(Pan, 2012)

$$D_i = \sqrt{x_i^2 + y_i^2} \quad (12)$$

$$S_i = \frac{1}{D_i} \quad (13)$$

Algoritmada öncelikle popülasyonda yer alan meyve sinekleri için rastgele belirlenmiş başlangıç pozisyonları oluşturulur. Sonrasında bu sineklere rastgele bir hareket vektörü tanımlanarak her bir sinek için yeni pozisyonlar elde edilir. Yiyeceğin nerede olduğunu bilemedikleri için işleme mesafeyi hesaplayarak devam edilir. Mesafe, her bir sineğin bireysel olarak orijine olan uzaklığıdır. Bireysel orijine uzaklıklar hesaplandıktan sonra bireye özgü koku değerleri hesaplanır. Değerler uygunluk fonksiyonunun içerisine yerleştirilerek, çözüm kümesi elde edilir.(Pan, 2012)

Eğer meyve sinekleri ilk iterasyonda yiyeceği bulamaz ise diğer iterasyona geçilir. İlk iterasyondan sonra meyve sineklerinin başlangıç pozisyonları rastgele olarak değil bir önceki yinelemedeki en iyi değer konumundan itibaren başlatılır. Meyve sinekleri yiyeceği buluncaya kadar iterasyonlar devam eder.

Meyve sineği algoritması da gri kurt algoritması gibi farklı uygulama alanlarında da kullanılmıştır.

Meyve sineği algoritması yerelde en iyiye takılmaya müsait bir algoritmadır. Araştırmacılar bu algoritmayı zincirlerinden koparıp küresele taşımak için gayret etmişlerdir.

Bir yaklaşımda hücre haberleşmesinden esinlenilerek algoritmanın karar verme aşaması modifiye edilmiştir ve bu yaklaşımın standart algoritmaya göre başarılı sonuçlar verdiği görülmüştür.(Xiao vd., 2015) Diğerinde ise algoritmanın karar verme aşamasından ziyade başlangıç aşamasında bir sürü ile değil birden fazla sürüyle tarama yapması çalışılmıştır.(Yuan vd., 2014) Diğer bir çalışmada algoritmaya Levy Uçuşu tekniği uygulanmış ve küresel optimaya ulaşması hedeflenmiştir.(Kılıç, 2022) Meyve sineği optimizasyon algoritmasını geliştirme çalışmalarına ek olarak algoritmayı gerçek hayata uygulama çalışmalarına da başvurulmuştur. Yapıların hasar derecesini değerlendirme amacıyla gelişmiş meyve sineği optimizasyon algoritması kullanılmıştır.(Xiong vd., 2021)

2.3 Parçacık Sürü Optimizasyonu Algoritması

Parçacık sürü optimizasyonu temelde iki ana uygulama yöntemine dayanır. Bunlar yapay yaşam ve sürü teorisi. Ayrıca evrimsel hesaplamalara da bağlıdır. Kendi türünde olan canlıların aralarında bilgi paylaşabilme özelliği algoritmanın temel prensiplerinden biridir. Aynı zamanda her bir iterasyonda yani evrimde avantajlı bir durum oluşturmaktadır.(Kennedy & Eberhart, t.y.)

Algoritma esasında sürüde yer alan bütün parçacıkların, sürüdeki en iyiye sahip olan bireye doğru yönelmesiyle ilintilidir. Yani popülasyonda bulunan üyeler her bir yinelemede kendilerine has olan bir önceki pozisyondan daha iyi olan bir konuma geçerek en sonunda istenen değere ulaşması prensibine dayanır.(Kennedy & Eberhart, t.y.)

2.3.1 Parçacık Sürü Optimizasyonu Algoritması Çalışma Yapısı ve Uygulama Alanları

Başlangıç aşamasında rastgele olarak sürüdeki her bir elemanın başlangıç konumu ve hızı belirlenir. Bu parametreler oluşturulduktan sonra her bir bireye özel uygunluk değeri hesaplanır. Hesaplanan uygunluk değerleri arasından kişisel en iyi değerler seçilir. Bu değerler arasından küresel en iyi değer seçilir. Yinelemeler devam ettikçe bireylerin pozisyonları ve hızları rastgele oluşturulmak için (14) ve (15)'ten yararlanır.(Özsağlam & Çunkaş, 2008)

$$V_{id} = W \cdot V_{id} + c_1 \cdot rand_1 \cdot (P_{id} - X_{id}) + c_2 \cdot rand_2 \cdot (P_{gd} - X_{id}) \quad (14)$$

$$X_{id} = X_{id} + V_{id} \quad (15)$$

Bu denklemlerde rand parametreleri rastgele üretilmiş sayılardır. W parametresi atalet momentidir. C değerleri ölçeklendirme için kullanılır. Yinelemeler belirli bir durdurma kuralına ulaşınca işlem sonlandırılır.(Özsağlam & Çunkaş, 2008)

Parçacık sürü optimizasyonu algoritması da diğer çalışmalardaki gibi geniş bir yelpazede kullanım alanına sahiptir. Bir çalışmada, mobil robotların belirlenen rotasında bulunan statik

veya dinamik engellere çarpmadan en verimli şekilde ilerlemesini sağlayan bir rota bulucu algoritma geliştirilmiştir. Parçacık sürü optimizasyonu (PSO) metodu kullanılarak, robotların güvenli ve hızlı bir şekilde hedeflerine ulaşmaları istenir. Parçacık sürü optimizasyonu, çok boyutlu çözüm uzayında etkili şekilde yol bulma yeteneği sayesinde mobil robotların çevresel zorluklarla başa çıkmasını desteklemektedir.(Garip vd., 2021) Diğer bir çalışmada, güç sistemlerinde yük dağıtımını ekonomikleştirmek amacıyla parçacık sürü optimizasyonu algoritmasından faydalanılmıştır.(Eke vd., 2023)

Uygulama çalışmalarının yanında algoritmayı geliştirmeye önceliklendiren çalışmalar da vardır. Sürekli optimizasyon problemlerini çözecek bir algoritma geliştirilmiştir. Bu geliştirmede parçacık sürü optimizasyon algoritması geri izleme tarama algoritması birleştirilerek algoritmanın küresel en iyiyi bulma ve isabetlilik özelliklerinin artırılması hedeflenmiştir.(Zaman & Gharehchopogh, 2022)

2.4. Yapay Arı Kolonisi Algoritması

Algoritma temelde bal arısı sürüsünün akıllı davranışına dayanır. Bu algortmada bal arıları üç gruba ayrılır. Bunlar işçi arılar, seçici arılar ve izcilerdir. İşçi arılar daha önce gittikleri bir kaynağa giderlerken, seçici arılar ise hangi yemek kaynağına gideceklerini seçerler. Bir arı kolonisinin yarısını işçi arılar diğer yarısını da seçici arılar oluşturur. Her yiyecek noktası için bir adet işçi arı görevlendirilmiştir. İzci arılar ise aslında yiyecek noktası diğer arılar tarafından tüketilmiş işçi arılardır.(Karaboga & Basturk, 2007)

2.4.1 Yapay Arı Kolonisi Algoritması Çalışma Yapısı ve Uygulama Alanları

Algoritmadaki yiyecek noktası bulma yapısı temelde üç aşamalıdır. Birinci aşamada işçi arılar yiyecek noktalarına gönderilir ve gönderildikleri her bir nokta için nektar miktarını ölçerler. Daha sonrasında ölçtükleri nektar miktarlarını seçici arılara rapor ederler. Seçici arılar yiyecek noktalarını seçtikten sonra izci arılar seçilen noktalara yönlendirilir. Yiyecek noktasının nektar miktarı arttıkça seçilme olasılığı da artar. (Karaboga & Basturk, 2007)

Algoritmada bir yiyecek noktasının konumu aslında problemin çözüm kümesidir. Nektar yoğunluğu ise bu olası çözümün doğruluğudur. İşçi arıların sayısı, yiyecek noktalarının sayısına eşittir çünkü bir arı bir yiyecek noktasına gitmekle görevlidir. Bu durumda bize işçi arı sayısının olası çözüm sayısına eşit olduğunu göstermektedir.(Karaboga & Basturk, 2007)

Algoritmada kullanılacak formüller ve eşitlikler (16)-(20)'de verilmiştir.(Akay ,2009)

- 1- Başlangıç aşamasında yiyecek noktalarının rastgele üretilmesi için gerekli olan denklem (16)'da verilmiştir.

$$x_{ij} = x_j^{min} + rand(0,1)(x_j^{max} - x_j^{min}) \quad (16)$$

Bu denklemde i çözüm sayısını, j değeri de denklemin boyutunu ifade etmektedir. Üretilen yiyecek noktalarının tüketilmiş olup olmadığını limit parametresi belirler bu parametreyi başlangıçta doğal olarak sıfır kabul ederiz.(Karaboga & Basturk, 2007)

- 2- İlk yiyecek noktası bulunduktan sonra işçi arılar farklı noktaları bulma amacındadır. Bulunan yakın yiyecek noktalarının kalite kontrolü yapılır ve yiyecek noktasının nektar miktarı daha fazla ise artık bu kaynak birincil yiyecek noktası olur. Bu işlemde yeni yiyecek noktası bulmak için (17)'den yararlanılmıştır.

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \quad (17)$$

Bu denklemde ϕ_{ij} aralık değeri $[-1,1]$ 'dir.(Karaboga & Basturk, 2007)

- 3- Kaynağa uygunluk değerinin etiketlenmesi gerekir. Bunun için (18) numaralı eşitlik kullanılır.

$$fitness_i = \begin{cases} \frac{1}{(1+|f_i|)}, f_i \geq 0 \\ 1 + |f_i|, f_i < 0 \end{cases} \quad (18)$$

f_i amaç fonksiyonu değeridir. Yeni yiyecek noktası ile halihazırda olan yiyecek noktasında hangisinin daha iyi olduğunun seçilmesi gerekir. Mevcut yinelemedeki en iyiyi seçme karar yapısı uygulanır. (Akay ,2009)

- 4- Uygunluk değeri etiketlenen kaynak için seçici arılar tarafından bir olasılık değeri hesaplaması yapılır. Bu değer mevcut değerden daha iyi ise seçici arılar yeni yiyecek noktasını denklem (19) ile üretme emri verirler.(Karaboga & Basturk, 2007)

$$p_i = \frac{fitness_i}{\sum_{i=1}^{SN} fitness_i} \quad (19)$$

Algoritma sırasıyla şu şekilde ilerler. Rastgele başlangıç yiyecek noktaları oluşturulur. Her yiyecek noktası için amaç fonksiyonu hesaplanır. Yakın yiyecek noktaları üretilir. Yiyecek noktalarının olasılıkları belirlenir. Bu olasık değerleri karşılaştırılır ve yeni yiyecek noktasına geçilip geçilmeyeceği kararı verilir. Eğer yiyecek noktası tükenir ise yeni noktalar üretilir. En iyi çözüm değeri diğer iterasyonlarda kıyaslama işlemi için saklanır. Algoritma durdurma kriterine ulaşırsa işlem sonlandırılır.(Çavuş & Tuncer, 2017)

Bu algoritma birçok uygulama alanında kullanılmıştır. Bir uygulamada ders programının belirlenmesi aşamasında çizelgeleme yöntemini desteklemek amacıyla bu algoritmanın kullanıldığı gözlenmiştir.(Küçükşille & Tokmak, 2014) Diğer bir çalışmada, insansız hava araçları (İHA) için gidiş yolu planlamasında, Yapay Arı Kolonisi (ABC) algoritması kullanılmıştır. Bu algoritma, İHA'ların belirlenen rotada güvenli ve verimli bir şekilde hareket etmelerini sağlamak amacıyla geliştirilmiştir. ABC algoritması, uçuş sırasında engellerden kaçınma, optimum rotaları keşfetme ve enerji verimliliğini artırma yetenekleriyle İHA'ların hassas görevlerini başarılı bir şekilde yerine getirmesini sağlamıştır. Bu uygulamada, Yapay Arı Kolonisi algoritması sayesinde başarılı sonuçlar elde edilmiştir.(Çavuş & Tuncer, 2017) Başka bir uygulamada görüntü iyileştirme yöntemlerine ek olarak yapay arı kolonisi algoritması eklenmiştir.(Öztürk & Öztürk, 2016)

3. SONUÇ

Sonuç olarak, metasezgisel optimizasyon algoritmalarının kullanım alanlarının son derece geniş olduğu açık bir şekilde gözlemlenmiştir. Bu algoritmaların, küresel optimum çözümü bulma yeteneği, yapılan bilimsel ve mühendislik çalışmalarında büyük bir önem taşımaktadır. Araştırmacılar, bu algoritmaların farklı türdeki uygulamalara entegre edilmesinin yanı sıra, algoritmaların performansını artırmak ve daha etkili hale getirmek amacıyla çeşitli geliştirme çalışmaları yapmışlar ve bu süreçte önemli başarılarla imza atmışlardır. Bununla birlikte, metasezgisel algoritmaların bazı zayıf yönleri, yani konverjans hızı, hassas parametre ayarları ve hedef fonksiyon karmaşıklığı gibi sorunlar, bu algoritmaların geliştirilmesinde daha fazla dikkat gerektirmektedir. Ayrıca, meta-sezgisel algoritmaların gelişim süreci devam ederken, gelecekte yapılacak olan çalışmalarla bu algoritmaların başarı oranlarının daha da yükseleceği ve daha karmaşık problemlere uygulanabilir hale geleceği düşünülmektedir. Bu nedenle, bu algoritmaların hem teorik hem de pratik açıdan önemi giderek artmaktadır.

KAYNAKLAR

- Akay, B., 2009. Nümerik Optimizasyon Problemlerinde Yapay Arı Kolonisi (Artificial Bee Colony) Algoritmasının Performans Analizi, Doktora tezi, Erciyes Üniversitesi, 2009.
- Çavuş, V., & Tuncer, A. (2017). İnsansız Hava Araçları İçin Yapay Arı Kolonisi Algoritması Kullanarak. *Karaelmas Science and Engineering Journal*, 7(1), 259-265. <https://dergipark.org.tr/en/pub/karaelmasfen/issue/57120/805826>
- Eke, İ., Saka, M., & Tezcan, S. S. (2023). KAOTİK PARÇACIK SÜRÜ OPTİMİZASYONU KULLANARAK EKONOMİK YÜK DAĞITIMI PROBLEMİNİN ÇÖZÜMÜ. *Journal of Engineering Sciences and Design*, 11(3), 957-965. <https://doi.org/10.21923/JESD.1293964>
- Eroğlu, H., Akay, V., Seyyarer, E., Çağrı Merkezi İletişim Şirketi, P., Yüzüncü Yıl Üniversitesi, V., Teknolojileri Bölümü, B., & Mühendisliği Bölümü, B. (2024). Kolektif Karar Optimizasyonu Algoritması ile I-Kesitli Kirişin Düşey Deplasman Minimasyonu. *Doğu Fen Bilimleri Dergisi*, 7(1), 11-22. <https://doi.org/10.57244/DFBD.1494064>
- Garip, Z., Karayel, D., & Çimen, M. E. (2021). Parçacık Sürü Optimizasyon Tabanlı Mobil Robotlarda Global Yol Planlama. *Journal of Smart Systems Research*, 2(1), 18-26. <https://dergipark.org.tr/en/pub/joinsr/issue/64451/980306>
- Hou, Y., Gao, H., Wang, Z., & Du, C. (2022). Improved Grey Wolf Optimization Algorithm and Application. *Sensors* 2022, Vol. 22, Page 3810, 22(10), 3810. <https://doi.org/10.3390/S22103810>
- Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *Journal of Global Optimization*, 39(3), 459-471. <https://doi.org/10.1007/S10898-007-9149-X/METRICS>
- Kennedy, J., & Eberhart, R. (t.y.). Particle swarm optimization. *Proceedings of ICNN'95 - International Conference on Neural Networks*, 4, 1942-1948. <https://doi.org/10.1109/ICNN.1995.488968>
- Kılıç, İ. (2022). LEVY UÇUŞLU MEYVE SİNEĞİ ALGORİTMASI İLE GÖRÜNTÜ SIKIŞTIRMA. *Adıyaman Üniversitesi Mühendislik Bilimleri Dergisi*, 9(16), 37-48. <https://doi.org/10.54365/ADYUMBD.980173>
- Küçüksille, E., & Tokmak, M. (2014). Yapay Arı Kolonisi Algoritması Kullanarak Otomatik Ders Çizelgeleme. *Süleyman Demirel University Journal of Natural and Applied Sciences*, 15(3), 203-210. <https://doi.org/10.19113/SDUFBED.79047>
- Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey Wolf Optimizer. *Advances in Engineering Software*, 69, 46-61. <https://doi.org/10.1016/J.ADVENGSOFT.2013.12.007>
- Özsağlam, M. Y., & Çunkaş, M. (2008). Optimizasyon Problemlerinin Çözümü için Parçacık Sürü Optimizasyonu Algoritması. *Politeknik Dergisi*, 11(4), 299-305. <https://dergipark.org.tr/tr/pub/politeknik/issue/33033/367241>
- Öztürk, S., & Öztürk, N. (2016). YAPAY ARI KOLONİ ALGORİTMASI KULLANILARAK GÖRÜNTÜ İYİLEŞTİRME YÖNTEMİNİN GELİŞTİRİLMESİ. *Gazi University Journal of Science Part C: Design and Technology*, 4(4), 173-183. <https://dergipark.org.tr/en/pub/gujsc/issue/45198/565979>
- Pan, W. T. (2012). A new Fruit Fly Optimization Algorithm: Taking the financial distress model as an example. *Knowledge-Based Systems*, 26, 69-74. <https://doi.org/10.1016/J.KNOSYS.2011.07.001>
- Sarıdağ, Ö., & Doğan, A. (2021). V2G Şarj İstasyonlarının Optimum Konum ve Kapasitelerinin Gri Kurt Optimizasyon Algoritması ile Belirlenmesi. *Bilecik Şeyh Edebali Üniversitesi Fen Bilimleri Dergisi*, 8(2), 622-635. <https://doi.org/10.35193/BSEUFBD.916804>

- Şenel, F. A., Yüksel, A. S., Gökçe, F., & Yiğit, T. (2018). Gri kurt optimizasyon algoritması ile iki boyutlu dizilim yazılımının geliştirilmesi. *Balikesir Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 20(2), 293-306. <https://doi.org/10.25092/BAUNFBED.433321>
- Toren, M., & Mollahasanoğlu, H. (2023). Gri Kurt optimizasyon algoritması ile güç ve dağıtım türü transformatörlerin ağırlık optimizasyonu. *Nigde Omer Halisdemir University Journal of Engineering Sciences*, 12(1), 96-103. <https://doi.org/10.28948/NGUMUH.1127837>
- Xiao, C., Hao, K., & Ding, Y. (2015). An improved fruit fly optimization algorithm inspired from cell communication mechanism. *Mathematical Problems in Engineering*, 2015. <https://doi.org/10.1155/2015/492195>
- Xiong, C., & Lian, S. (2021). Structural Damage Identification Based on Improved Fruit Fly Optimization Algorithm. *KSCE Journal of Civil Engineering*, 25(3), 985-1007. <https://doi.org/10.1007/S12205-021-1115-5/METRICS>
- Yaylacı, E. K., Özdeş, H. N., Erdem Yılmaz, A., Üniversitesi, K., Fakültesi, M., Mühendisliği, E.-E., & Karabük, T. (2023). Geliştirilmiş Kızıl Tilki Optimizasyon Algoritması ile Da-Da Alçaltıcı Tip Dönüştürücü PI Katsayılarının Optimizasyonu. *Karaelmas Fen ve Mühendislik Dergisi*, 13(1), 1-11. <https://doi.org/10.7212/KARAEMLASFEN.1136404>
- Yuan, X., Dai, X., Zhao, J., & He, Q. (2014). On a novel multi-swarm fruit fly optimization algorithm and its application. *Applied Mathematics and Computation*, 233, 260-271. <https://doi.org/10.1016/J.AMC.2014.02.005>
- Zaman, H. R. R., & Gharehchopogh, F. S. (2022). An improved particle swarm optimization with backtracking search optimization algorithm for solving continuous optimization problems. *Engineering with Computers*, 38(4), 2797-2831. <https://doi.org/10.1007/S00366-021-01431-6/TABLES/12>