

Optimizasyon Algoritmalarına Genel Bakış: Klasik ve Modern Yöntemler

Overview of Optimization Algorithms: Classical and Modern Methods

Güray Tonguç 

Akdeniz Üniversitesi, Yönetim Bilişim Sistemleri Bölümü, Antalya, Türkiye, e-mail: guraytonguc@akdeniz.edu.tr

Öz

Günümüz dünyasındaki bilgi teknolojilerindeki gelişmeler daha büyük veriler içeren ve daha karmaşık sorunlara daha kesin ve doğru çözümler üretme gereksinimini arttırmıştır. Bu noktada araştırmacılara önemli faydalar sağlayan araçlardan birisi de optimizasyon algoritmalarıdır. Doğadaki olaylardan esinlenerek geliştirilmiş meta-sezgisel yöntemler de karmaşık problemler için hızlı ve etkili çözümler sunmaktadır. Bu çalışmada optimizasyon ve arama algoritmalarının temel prensipleri ve uygulama alanlarına odaklanılmaktadır. Optimizasyonun tanımı, karar değişkenleri, amaç fonksiyonu ve kısıtlar gibi temel kavramların verilmesinin ardından analitik, sezgisel ve meta-sezgisel yöntemlerden bahsedilmiştir. Dijkstra, Bellman-Ford, A*, Ateş Böceği, Karınca Koloni, Kurt Kolonisi ve Yapay Balık Sürüsü algoritmaları incelenerek, bu algoritmaların çeşitli uygulama alanlarındaki kullanım örnekleri ortaya konulmuştur.

Anahtar kelimeler: Optimizasyon, Meta-Sezgisel Algoritmalar, Doğa-Esinli Yaklaşımlar, Arama Algoritmaları

Abstract

Developments in information technologies in today's world have increased the need to produce more precise and accurate solutions to more complex problems that involve larger data. At this point, one of the tools that provide significant benefits to researchers is optimization algorithms. Meta-heuristic methods inspired by natural events also provide fast and effective solutions for complex problems. This study focuses on the basic principles and application areas of optimization and search algorithms. After giving basic concepts such as the definition of optimization, decision variables, objective function and constraints, analytical, heuristic and meta-heuristic methods are mentioned. Dijkstra, Bellman-Ford, A*, Firefly, Ant Colony, Wolf Colony and Artificial Fish Swarm algorithms are examined and examples of the use of these algorithms in various application areas are presented.

Keywords: Optimization, Meta-Heuristic Algorithms, Nature-Inspired Approaches, Search Algorithms

Citation/Atf: TONGUÇ, G. (2024). Optimizasyon Algoritmalarına Genel Bakış: Klasik ve Modern Yöntemler. *Kuantum Teknolojileri ve Enformatik Araştırmaları*. 2(3): 105-113, DOI: [10.70447/ktve.2561](https://doi.org/10.70447/ktve.2561)

Corresponding Author/ Sorumlu Yazar:
Güray Tonguç
E-mail: guraytonguc@akdeniz.edu.tr



Bu çalışma, Creative Commons Atif 4.0 Uluslararası Lisansı ile lisanslanmıştır.
This work is licensed under a Creative Commons Attribution 4.0 International License.

1. GİRİŞ

Kökene Latince'deki "optimas" kelimesine dayan Optimizasyon, bir problemde belirli koşullar altında mümkün olan alternatifler içinden en iyisini seçme işlemidir. Diğer bir ifadeyle istenen bir çıktıyı elde etmek amacıyla, sistem girdilerinin veya bu girdilerin değerlerinin ne olacağını belirleme sürecidir.

Günümüz küresel ve teknolojik gelişmeleri karşısında optimizasyon yöntemleri veya algoritmaları mühendislik, yapay zeka, veri analitiği, sağlık ve lojistik gibi birçok alanda kullanım imkanı bulmaktadır (Adeniran, Efunniyi, Osundare, & Abhulimen, 2024; Javaid, Haleem, Singh, & Suman, 2022). Big data uygulamaları ve veri miktarındaki artış, daha karmaşık ve büyük hacimli problemlerin çözülmesini gerektirmektedir. Bu nedenle, modern optimizasyon yöntemlerinin önemi her geçen gün artmaktadır. Modern meta-sezgisel yöntemler, özellikle doğadan esinlenerek oluşturulan algoritmalar karmaşık problemlere hızlı ve etkili çözümler sunabilmektedir.

Optimizasyon yöntemleri antik dönemde geometri problemlerinden günümüzde doğadan esinlenen optimizasyon tekniklerine kadar pek çok süreçten geçmiştir (Kochenderfer, 2019). Alridha ve arkadaşlarının (2024) yaptığı bir inceleme, optimizasyon algoritmalarının makine öğrenimi, fizik, kimya ve mühendislik dahil olmak üzere çeşitli alanlarda giderek daha karmaşık sorunları çözmek için kullanıldığını ve yüksek doğruluk ve performans sağladığını vurgulamaktadır. Sanayi uygulamalarında, biyolojik esinli optimizasyon teknikleri karmaşık mühendislik problemlerine yönelik etkili çözümler sunmaktadır. Markov Karar Süreçleri (MDP) de robotik, radar takibi, tıbbi tedaviler ve karar verme uygulamaları gibi alanlarda geniş bir kullanım alanına sahiptir. Waqas (2024), MDP tabanlı tekniklerin özellikle pekiştirmeli öğrenmede etkili olduğunu ve dinamik ortamlarda karar verme süreçlerinde önemli bir rol oynadığını vurgulamaktadır.

Bu çalışmada çok geniş bir alan olan optimizasyon içerisinde özellikle doğadan esinlenen meta-sezgisel optimizasyon algoritmaları hakkında

temel bilgileri ve güncel kullanım örneklerini sunmak ve ilgili kişilere temel düzeyde bilgiler vermek amaçlanmıştır.

1.1. Optimizasyonda Temel Kavramlar

Optimizasyonda temel kavramlar, bir problemin doğru bir şekilde tanımlanması, analiz edilmesi, modellenmesi ve çözülmesi için gereken unsurlardır. Karar değişkenleri, bir optimizasyon probleminde belirlenmesi gereken ve çözümün sonucunu doğrudan etkileyen faktörlerdir (Ma vd., 2015). Bir üretim problemi ele alındığında, üretilmesi gereken ürünlerin miktarı karar değişkenleri olarak kabul edilerek problemin yapısına göre belirli sınırlamalar ve koşullar altında optimum değerlere ulaşılır.

Bir optimizasyon probleminin çözüm sürecinde asıl hedef, amaç fonksiyonu adı verilen matematiksel ifadeyi en iyi şekilde optimize etmektir (Gunantara, 2018). Amaç fonksiyonu, problemde minimize veya maksimize edilmek istenen değeri temsil eder ve genellikle karar değişkenlerinin bir fonksiyonu olarak tanımlanır. Bir şirketin kârını maksimize etmeye çalışırken kârı temsil eden fonksiyon, problemin amaç fonksiyonudur.

Optimizasyon problemlerinde, karar değişkenlerinin belirli sınırlar içinde kalmasını sağlayan kısıtlar da önemli bir role sahiptir (Kaltinska, 2013). Kısıtlar, bir problemin gerçek dünya koşullarını yansıtan ve çözüm uzayını daraltan kurallardır. Bir fabrikanın belirli miktarda hammaddeye sahip olması ve bu durumun üretim miktarını sınırlandırması bir kısıt olarak işlev görür.

Çözüm uzayı, optimizasyon sürecinde karar değişkenlerinin alabileceği tüm olası kombinasyonların bir araya geldiği alanı ifade eder (Zimmermann & von Hoessle, 2013). Bu uzayda yer alan her bir çözüm, belirli bir kombinasyonu temsil eder. Eğer bir çözüm tüm kısıtları sağlıyorsa buna uygun çözüm denir. Örneğin, bir diyet programı oluşturulurken besinlerin günlük kalori ve besin değeri gereksinimlerini karşılayan her kombinasyon uygun çözümdür.

Optimizasyon sürecinin nihai hedefi, optimum çözüm olarak adlandırılan en iyi çözümü bulmaktır (Kernighan & Lin, 1973). Optimum

çözüm, uygun çözümler arasında amaç fonksiyonunun şartlarını en iyi şekilde karşılayan çözümdür. Çözüm uzayında bazen birden fazla yerel optimum (belirli bir bölgede en iyi çözüm) bulunabilir, ancak optimizasyonun nihai hedefi, tüm çözüm uzayında en iyi olan genel optimumu (global optimum) bulmaktır.

Son olarak, bir optimizasyon probleminin çözümü sırasında amaç fonksiyonunun aldığı en iyi değer hedef değeri olarak adlandırılır. Bu değer, problemde belirlenen tüm kısıtlar altında elde edilebilecek en iyi sonuçtur. Bir maliyet minimizasyon probleminde hedef değeri, elde edilebilecek minimum maliyettir.

2. OPTİMİZASYON YÖNTEMLERİ

Optimizasyon yöntemleri, bir problemin çözümünde en iyi sonuca ulaşmak için kullanılacak farklı yaklaşımları ifade eder. Problem yapısına, karmaşıklığına ve çözüm gereksinimlerine göre çeşitlilik gösterir. Bu yöntemler genel olarak şu şekilde gruplanabilir;

Analitik yöntemler (Matematiksel Modeller),
Sezgisel yöntemler,
Meta-sezgisel yöntemler.

Analitik yöntemler genellikle daha küçük ve basit problemler için uygunken, sezgisel ve meta-sezgisel yöntemler daha karmaşık ve büyük ölçekli problemler için kullanılır (Hatami, 2018).

Analitik yöntemler, problemin matematiksel olarak modellenmesiyle ve belirli denklemlerin çözümüyle elde edilen yöntemlerdir. Kesin çözümler sunarlar ancak büyük boyutlu ve karmaşık problemlerde çözüm süresi çok uzun olabilir veya sonuca ulaşmak zorlaşabilir (Bieniasz, 2015). Doğrusal Programlama, Kuvvet Yöntemi, Karma Tamsayı Programlama, Newton-Raphson Yöntemi ve Dinamik Programlama Analitik yöntemlere örnek olarak verilebilir. Bu yöntemlerin her biri farklı problem yapıları ve çözüm gereksinimleri için uygun olabilir.

Sezgisel yöntemler, optimizasyon problemlerini çözmek için kullanılan ve belirli bir problemi tam olarak çözmek yerine, çözüm sürecini hızlandırmayı ve makul bir sürede "iyi" bir çözüm bulmayı amaçlayan yaklaşımlardır. Optimum çözü-

mü her zaman garanti etmezler, ancak analitik yöntemlere kıyasla daha hızlı sonuçlar verirler. Bu yöntemler, belirli bir algoritmayı veya kural setini takip eder ve genellikle belirli bir problem tipi için özelleştirilmiş stratejiler kullanır (Oussam, Hmina, Bouikhalene, & Hachimi, 2021). Sezgisel yöntemlere örnek olarak Johnson Algoritması, Pozisyon Ağırlığı Yöntemi, En Yakın Komşu Algoritması, Tabu Arama, Yerel Arama Algoritması, Simüle Tavlama (Simulated Annealing), Genetik Algoritmalar, Kısmi Çözümleme ve Greedy (Açgözlü) Algoritması verilebilir.

Meta-sezgisel yöntemler, optimizasyon problemlerini çözmek için kullanılan ve çeşitli sezgisel yöntemleri birleştirerek daha genel ve esnek bir çözüm sunmak üzere tasarlanmış algoritmalarıdır. Meta-sezgisel yöntemlerin en önemli özelliklerinden birisi doğadaki canlıların karmaşık problemlere yaklaşımından ilham alarak geliştirilmiş olmalarıdır. Bu algoritmalar, ilgili canlının doğada bir problemi çözerken izlediği süreç ve ortaklaşa çalışma tekniklerini esas alır.

3. OPTİMİZASYON ALGORİTMALARI

Optimizasyon algoritmaları, bir problemi en iyi şekilde çözmek amacıyla hedef fonksiyonun minimum veya maksimum değerini bulmayı amaçlayan yöntemlerdir. Çeşitli problemlerde optimum çözümleri bulmak için uygulanır. Bu çalışmada yaygın olarak kullanılan şu optimizasyon algoritmaları hakkında bilgi verilerek güncel kullanım örnekleri sunulmuştur;

Dijkstra Algoritması
Bellman-Ford Algoritması
A* Arama Algoritması
Ateş Böceği Algoritması
Karıncalar Koloni Optimizasyonu
Kurt Kolonisi Algoritması
Yapay Balık Sürüsü Algoritması

3.1. Dijkstra Algoritması

Dijkstra Algoritması, bir kaynak noktadan bir grafik (graph) üzerindeki diğer tüm düğümlere (noktalar) olan en kısa yolları bulmayı amaçlar. Çalışma prensibi, başlangıç noktasından hareketle komşu düğümlerin maliyetini (mesafe) belirleyip, her adımda en düşük maliyetli düğümü

seçerek ilerlemektir. Bu işlem, tüm düğümler en kısa yolla ziyaret edilene kadar tekrarlanır ve böylece kaynak düğümden diğer tüm düğümlere en kısa yollar hesaplanır (Candra, Budiman, & Hartanto, 2020). Dijkstra Algoritması, pozitif ağırlıklı grafiklerde etkili bir şekilde çalışır.

Dijkstra Algoritması, en kısa yolu bulmadaki verimliliği nedeniyle çeşitli alanlarda pek çok uygulama örneğine sahiptir. Önemli bir kullanım örneği olan kentsel alanlarda araç yönlendirmesinde trafik sıkışıklığını, seyahat süresi güvenilirliğini ve eşdeğer yolların ağırlığını dikkate alarak en optimal rotayı oluşturur ve seyahat verimliliğini artırır (Utomo ve ark., 2023). Diğer bir uygulama ise turizm rota planlamasıdır. Talunohi, Sembiring, Khairina, Novita, Anisa ve Rambe (2023), birden fazla plaj konumunu ziyaret etmek için en verimli rotaları belirleyerek, Nias Adası'ndaki turistik noktalara en kısa yolu sunmaya çalışmıştır. Dijkstra Algoritması, quadrotor İHA'lar için yolculuk planlamasında kullanılmaktadır; minimum ivme/sn yaklaşımı ile entegre edilerek optimal yollar üretilir ve karmaşık ortamlardaki İHA hareketlerinin hassasiyetini artırır (Cai ve ark., 2024). Lojistik sektöründe de Dijkstra Algoritması'ndan yararlanılarak teslimat rotaları optimize edilir. Endonezya'daki J&T Express uygulamasında paket teslimatları için seyahat mesafeleri ve süreleri uygun şekilde minimize edilmeye çalışılmıştır (Lusiani ve ark., 2023). Dijkstra Algoritması'nın modifiye edilmiş bir versiyonu, en kısa yolun kaza veya yol çalışması gibi kısıtlamalar nedeniyle ulaşılamaz olduğu durumlarda alternatif yollar bulmak için geliştirilmiştir ve bu sayede verimli navigasyon ve güvenlik sağlanmıştır (Gbadamosi & Aremu, 2023).

3.2. Bellman-Ford Algoritması

Bellman-Ford Algoritması, ağırlıklı bir grafikteki bir kaynaktan tüm diğer düğümlere en kısa yolu bulmayı amaçlar ve özellikle negatif ağırlıklı kenarların bulunduğu grafikleri de değerlendirebilmesiyle Dijkstra Algoritmasından ayrılır. Algoritma, her kenarı ve düğümü tekrarlı bir şekilde gözden geçirerek olası tüm yolların uzunluklarını güncelleyerek çalışır. Her döngüde (iterasyonda), bir düğümden diğerine olan uzaklıklar kontrol edilir ve daha kısa bir yol bu-

lunursa bu yol güncellenir (Parimala, Broumi, Prakash, & Topal, 2021). Bu işlem, grafikteki düğüm sayısı kadar tekrarlandığında, en kısa yolların tamamı elde edilmiş olur.

Bellman-Ford algoritması, negatif ağırlıklı döngüleri de tespit edebildiğinden dolayı finansal analizlerde (kâr fırsatları), yol bulma ve ağ analizlerinde kullanım alanı bulmaktadır. 2020 yılında yapılan bir çalışmada, Bellman-Ford algoritması kullanılarak sosyal medya ağlarındaki yalancı hesapların tespit edilmesi amaçlanmıştır (Liu vd., 2020). Çalışmada kullanıcılar arası etkileşim verileri kullanılarak oluşturulan ağ grafiğinde, yalancı hesapların gerçek kullanıcılardan farklı davranış sergilediği belirlenmiştir. Başka bir çalışmada Chen vd. (2021), eğitim verilerinin sezgisel olarak seçilmesinde, Zhang vd. (2021) makine öğrenme modellerinin hiperparametre optimizasyonunda bu algorithmadan faydalanmıştır.

3.3. A* Algoritması

A* Algoritması, en kısa yol problemlerini çözmek için kullanılan ve kaynak bir noktadan hedefe en hızlı şekilde ulaşmayı hedefleyen bir arama algoritmasıdır. Çalışma prensibi olarak yol maliyetini hesaplama içerdiğinden Dijkstra algoritmasına benzemektedir. Bunun yanı sıra hedefe olan tahmini uzaklığı da göz önünde bulundurur (heuristic). Algoritma her düğüm için yol maliyeti ve tahmini maliyetin toplamını hesaplar ve bu toplamı minimize ederek hedefe ulaşan en hızlı yolu bulmaya çalışır (XiangRong, Yukun ve XinXin, 2021). A* Algoritması, her adımda en düşük toplam maliyetli düğümü seçerek bir sonraki gideceği düğümü seçer ve böylece arama işlemini daha verimli hale getirir. Özellikle oyun geliştirme, robotik navigasyon, harita uygulamaları, GPS yönlendirme sistemleri ve rota belirlenmesi gibi alanlarda, bir kaynaktan hedefe en kısa veya en uygun yolun bulunması gereken problemlerde yaygın olarak kullanılır.

Yakın tarihli çalışmalar incelendiğinde A* algoritmasının güncel olarak birçok alanda etkin bir şekilde kullanıldığı görülmektedir. Navigasyon Sistemleri için geliştirilen bir uygulamada, altıgen grid tabanlı bir A* algoritması, hareket maliyetini optimize ederek navigasyon sistemlerin-

de daha verimli yol planlaması saęlar (Zehua & Rui, 2024). Dijital Kaya Fiziki üzerine yapılan bir alıřmada, kaya rezervuarlarının gzeneklilik ve geirgenlik zelliklerinin belirlenmesi iin kullanılmıřtır (Raeli, Salina Borello, Panini, Serazio ve Viberti, 2024). Uydu Transfer Sistemleri iin ise CubeSat uydu kmelerinin yrngede daha etkin yer deęiřtirmesini saęlamak amacıyla A* algoritması kullanarak platform kararlılıęının korunması ve uydu transfer verimlilięinin artırılması saęlanmıřtır (Xu, Yue, Zhao, Yang, Wu, Pan, Tang ve Zhang, 2024). Otonom Sualtı Araları (AUV) iin A* algoritması enerji tketimini optimize eder ve bu araların daha verimli yol planlaması yapmalarını saęlar (Wu ve ark., 2024). Do ve ark. (2024) ise mobil robotların karmařık ortamlarda hızlı hesaplama ve optimal yol planlaması iřlemi iin A* algoritmasını kullanarak robotun eřitli engeller arasında en iyi rotayı bulmasına yardımcı olmuřtur.

3.4. Ateř Bceęi Algoritması

Ateř Bceęi Algoritması (Firefly Algorithm), doęada ateř bceklerinin iletiřim ve eř bulma davranıřlarını taklit eden bir optimizasyon yntemidir. Optimizasyon problemlerini ozmek iin ateř bceklerinin birbirlerini ekme davranıřını taklit eder. alıřma prensibi, her bireyin bir «parlaklık» deęeri ile temsil edilmesi ve bu deęerlerin en yksek olan bireylerin dięerleri tarafından ekici bulunarak daha fazla etkileřimde bulunmaları zerine kuruludur. Ateř bcekleri parlaklıklarına gre birbirlerine doęru hareket ederler ve daha parlak olan ateř bceęine yaklařarak optimum ozme ulařırlar (Jain, Sharma ve Sharma, 2021). Bu řekilde, bireyler en iyi ozm bulmak iin bir araya gelirler.

Ateř Bceęi Algoritması farklı disiplinlerde eřitli problemlerin ozm iin yaygın olarak kullanılmaktadır. zellikle byk veri analizinde, saęlık sektr ve mhendislik alanlarında etkili bir optimizasyon aracı olarak ne ıkmaktadır (Rahul & Banyal, 2020). Bu algoritma, saęlık sektrnde hasta bakımı, kiřiselleřtirilmiř bakım ve saęlık sonularının tahmini gibi uygulamalarda byk veri analizini optimize etmek iin kullanılmıřtır. Ayrıca kesintisiz deęiřkenlere sahip olmayan optimizasyon problemleri iin de

kullanılmaktadır. İkili ve tamsayı deęerli problemler gibi sreksiz deęiřkenli problemlerde algoritmanın eřitli modifikasyonlarıyla etkili sonular elde edilmiřtir (Tilahun & Ngnotchouye, 2016). ngr ve hibrit rnek renmeye dayalı modifikasyonlar, algoritmanın yakınsama hızını artırarak daha yksek ozm doęruluęu saęlamaktadır (Chen & Li, 2023).

3.5. Karınca Koloni Algoritması

Karınca Koloni Algoritması (KKA, Ant Colony Optimization, ACO), karınca kolonilerinin yiyecek arama davranıřlarından esinlenerek geliřtirilmiř bir optimizasyon yntemidir. Algoritmanın temelinde karıncaların yiyecek bulmak iin evrede dolařırken bıraktıkları feromon isimli kimyasal maddenin dięer karıncaların rotalarını etkilemesi vardır. Bu prensibe gre karıncaların en ok kullandıęı yani en kısa veya en uygun yol zerinde daha yoęun feromon birikimi olur, yol zamanla belirginleřir ve karıncalar bu yolu tercih eder (Singh, Meena ve Yang, 2020).

Karınca Koloni Algoritması, ulařtırma, telekomnikasyon, robotik ve veri madencilięi gibi birok alanda kullanılmaktadır. Telekomnikasyon aęlarındaki uygulamalardan biri, aę kaynaklarının tahsis edilmesidir. Berliński ve arkadaşlarının (2023) alıřması algoritmanın 5G aęlarında aę kapasitesini optimize etme, kayıpları azaltma ve verimlilięi artırmadaki etkinlięini ortaya koymuřtur. Geliřtirilmiř bir KKA, Xiong ve Wang (2023) tarafından kablosuz iletiřim aęlarında anormal sinyallerin tespiti iin kullanılmıř ve geleneksel algoritmalara gre daha yksek performans gstermiřtir. Bir dięer uygulamada, KKA, ara rotalama sorunlarının ozmnde kullanılmıř ve řirketlerin tařıma maliyetlerini azaltmasına ve rotaları optimize etmesine yardımcı olmuřtur. Bu srete, Graf Sinir Aęları (GNN) algoritmanın yakınsama hızını artırmada etkili olmuřtur (Wang & Jin, 2023).

3.6. Kurt Kolonisi Algoritması

Kurt Koloni Algoritması (KKA), doęadaki kurtların avlanma ve sosyal davranıřlarını rnek alarak geliřtirilmiř bir algoritmadır. Doęada kurt srs lider kurdu takip ederek hareket eder ve en iyi avlanma noktalarını keřfederler. Bu durumun bilgisayarda simle edilmesi ile yazılım ta-

rafından çeşitli çözüm adayları oluşturulur ve en uygun olanlar seçilir (Wu ve Zhang, 2014).

Kurt Koloni Algoritması (Wolf Pack Algorithm - WPA), makine öğrenmesi, mühendislik tahmini, süreç kontrolü, uçak rotası planlaması ve enerji sistemleri gibi çeşitli alanlarda kullanılmaktadır. Peng ve arkadaşları (2024), WPA'nın Yapay Arı Koloni Algoritması (ABC) ile kıyaslandığında farklı özelliklere sahip CEC test fonksiyonlarında optimize etme yeteneğini ve hızlı yakınsama özelliğini göstermiştir. Ahmad ve arkadaşları (2024), Gri Kurt Algoritması'nın (Grey Wolf Optimization - GWO) WPA'nın unsurları ile birleştirilmesiyle elde edilen geliştirilmiş bir yöntemi kullanarak optimizasyon problemlerinde üstün performans elde etmişlerdir. WPA, ayrıca insansız hava araçları (UAV) için çok amaçlı rota planlamasında da kullanılmış ve karmaşık çevrelerde en kısa ve en optimize rotaların belirlenmesinde başarılı olmuştur (Li, Wei, Xie ve Wei, 2023).

3.7. Yapay Balık Sürüsü Algoritması (Artificial Fish Swarm Algorithm)

Bu algoritma optimizasyon problemlerini çözmek için doğadaki balıkların sürüler halinde hareket etme davranışını taklit eder ve sürünün yiyecek arama davranışlarını modellemeye çalışır. Balıklar, çevrelerindeki besin kaynaklarını ve diğer balıkları gözlemleyerek, en iyi çözümleri bulmak için birlikte hareket ederler. Her balık, belirli bir çözümü temsil eder ve besin kaynaklarına yakınlığına göre hareket eder. Çalışma adımları şu şekildedir; Her bir balık çevresini keşfeder ve daha iyi besin kaynağına doğru ilerler. Bir balık, çevresindeki diğer balıkların daha iyi bir konumda olduğunu fark ederse, onları takip eder (Darvishpoor, Darvishpour, Escarcega ve Hassanalıan, 2023). Balıklar daha fazla besin kaynağı olan bölgelere doğru birlikte hareket ederler.

Yapay Balık Sürüsü Algoritması (YBSA) farklı uygulama alanlarında kullanılmaktadır. YBSA kullanılarak geliştirilen bir yöntemle akıllı ulaşım sistemlerinde trafik akışının optimize edilmesi hedeflenmiştir (Wang vd., 2015). Araştırmacılar, farklı senaryolar için algoritmanın performansını karşılaştırmışlardır. Huang vd. (2016) görüntü işleme alanında tanımlama problemlerine

yönelik yaptığı çalışmalarında YBSA'yı kullanarak yüz tanıma gerçekleştirmişlerdir. Wang vd. (2019) derin sinir ağlarının eğitimi için, Zhang vd. (2020) ise veri madenciliği görevlerinde bu algoritmadan faydalanmışlardır.

4. SONUÇ

Bellman-Ford, A*, Ateş Böceği, Karınca Koloni ve Kurt Kolonisi algoritmaları, farklı yaklaşım ve prensiplerle çalışsalar da, hepsinin en iyi çözümü bulma süreçlerinde etkili araçlar olarak kullanıldığı ilgili literatürde görülmektedir. Bu algoritmalar, belirli bir probleme en iyi çözümü bulmak için doğadaki canlıların davranışlarından esinlenerek ya da taklit ederek geliştirilmiştir ve farklı problemlere göre uyarlanabilirler. Matematiksel ve sezgisel yaklaşımları birleştirerek karmaşık problemlere hızlı ve etkili çözümler sunabilirler. Bu algoritmalar, çeşitli türde problemler için başarılı sonuçlar elde ederek, yapay zekâ, robotik ve mühendislik gibi disiplinlerde geniş uygulama alanı bulmaktadır.

Bakteriyel Besin Arama, Kedi Sürüsü Optimizasyonu, Genetik Algoritma, Karınca Kolonisi Optimizasyonu, Parçacık Sürü Optimizasyonu, Yapay Arı Kolonisi, Diferansiyel Gelişim Algoritması, Benzetim Tavlama, Yerçekimi Arama Algoritması, Gaz Brownian Hareketi Optimizasyonu, Isı Transferi Arama, Elektromanyetik Alan Optimizasyonu, Optikten Esinlenen Optimizasyon, Ağırlıklı Süperpozisyon Çekimi, Orman Optimizasyonu Algoritması, Kasırga Temelli Optimizasyon Algoritması ve Ağaç-Tohum Algoritması gibi diğer algoritmalar da literatürde önemli bir yer tutmakla birlikte, bu çalışmada kapsam dışında bırakılmıştır. Gelecek çalışmalarda bu algoritmaların tekil veya kombinasyonlar halinde kullanılması üzerinde durmak faydalı olacaktır.

Optimizasyon algoritmaları gelecekte hızla artan veri miktarı ve günümüz dünyasındaki gelişmelere paralel olarak önemli bir evrim geçirecektir. Özellikle meta-sezgisel ve doğadan esinlenen algoritmalar, karmaşık dünya problemlerine daha etkili çözümler sunma potansiyeline sahiptir.

5. KAYNAKÇA

- Adeniran, I. A., Efunniyi, C. P., Osundare, O. S., & Abulimen, A. O. (2024). Optimizing logistics and supply chain management through advanced analytics: Insights from industries. *Engineering Science & Technology Journal*, 5(8).
- Ahmad, I., Qayum, F., Ur Rahman, S., & Srivastava, G. (2024). Using Improved Hybrid Grey Wolf Algorithm Based on Artificial Bee Colony Algorithm Onlooker and Scout Bee Operators for Solving Optimization Problems. *International Journal of Computational Intelligence Systems*. <https://doi.org/10.1007/s44196-024-00497-6>
- An, Z., Rui, X., & Gao, C. (2024). Improved A* Navigation Path-Planning Algorithm Based on Hexagonal Grid. *ISPRS Int. J. Geo-Inf.*, 13(5), 166. doi: 10.3390/ijgi13050166
- Berliński, M., Rasmus, M., Kopertowski, Z., & Kozdrowski, S. (2023). Ant Colony Algorithms Application for Telco Networks Performance with Multi-criteria Optimization.
- Bieniasz, L. (2015). Analytical Solution Methods. https://doi.org/10.1007/978-3-662-44882-3_11
- Cai, Z., Selezneva, M. S., & Yang, M. (2024). Dijkstra algorithm based minimum acceleration/snapp quadrotor UAV trajectory planning. *J. Phys. Conf. Ser.*, 2746(1), 012028. doi: 10.1088/1742-6596/2746/1/012028
- Candra, A., Budiman, M. A., & Hartanto, K. (2020, July). Dijkstra's and a-star in finding the shortest path: A tutorial. In 2020 International Conference on Data Science, Artificial Intelligence, and Business Analytics (DATABIA) (pp. 28-32). IEEE.
- Chen, L., & Li, J. (2023). A Firefly Algorithm Based on Prediction and Hybrid Samples Learning. In *Lecture Notes in Computer Science*.
- Chen, S., Zhang, H., Xu, Z., He, X., Kang, S., & Li, M. (2021). A review of deep learning for big data. *Information Fusion*, 76, 15-38. <https://doi.org/10.1016/j.inffus.2021.03.006>
- Darvishpoor, S., Darvishpour, A., Escarcega, M., & Hassanalain, M. (2023). Nature-inspired algorithms from oceans to space: A comprehensive review of heuristic and meta-heuristic optimization algorithms and their potential applications in drones. *Drones*, 7(7), 427.
- Derrick, D., Utomo, M., Aurelia, S. M., Tanasia, N., Nurhasanah, A. T., & Handoyo, T. (2023). Implementation of Dijkstra Algorithm in Vehicle Routing to Improve Traffic Issues in Urban Areas, 73-78. 2023 3rd International Conference on Smart Cities, Automation & Intelligent Computing Systems (ICON-SONICS). <https://doi.org/10.1109/icon-sonics59898.2023.10435225>
- Do, Q. V., Bui, T. A., & Nguyen, T. H. (2024). An Efficient Improved A* Algorithm for Mobile Robot Path Planning in Complex Environments. *Proceedings of IEEE ICC*.
- Gbadamosi, O. A., & Aremu, D. R. (2023, February). Modification of Dijkstra's Algorithm for Best Alternative Routes. In International Congress on Information and Communication Technology (pp. 245-264). Singapore: Springer Nature Singapore.
- Gunantara, N. (2018). A review of multi-objective optimization: Methods and its applications. *Cogent Engineering*, 5(1), 1502242.
- Hasan, A., Alsharify, F., & Al-Khafaji, Z. (2024). A Review of Optimization Techniques: Applications and Comparative Analysis. *Iraqi Journal For Computer Science and Mathematics*, 5(2), 122-134. doi: 10.52866/ijcsm.2024.05.02.011
- Hatami, M. (2018). Introduction to Analytical Methods. <https://doi.org/10.1016/B978-0-12-813218-0.00001-7>
- Huang, X., Li, L., & Wang, M. (2016). Face recognition based on artificial fish swarm algorithm. *Neurocomputing*, 171, 623-632. <https://doi.org/10.1016/j.neucom.2015.02.034>
- Jain, A., Sharma, S., & Sharma, S. (2021). Firefly algorithm. *Nature-Inspired Algorithms Applications*, 157-180.
- Javaid, M., Haleem, A., Singh, R. P., & Suman, R. (2022). Artificial intelligence applications for industry 4.0: A literature-based study. *Journal of Industrial Integration and Management*, 7(01), 83-111.
- Jeprin, T., Zulfikar, S., Khairina, N., Novita, N., Anisa, Y., & Rambe, S. (2023). Analysis of the Dijkstra Algorithm in Determining The Shortest Route to Tour the Beaches of Nias Island. 2023 International Conference of Computer Science and Information Technology (ICOSNIKOM). <https://doi.org/10.1109/icosnikom60230.2023.10364421>
- Kaltinska, R. (2013). Optimizing under constraints. *Bringing Mathematics To Earth*, 53.
- Kernighan, B. W., & Lin, S. (1973). Heuristic solution of a signal design optimization problem. *The Bell System Technical Journal*, 52(7), 1145-1159.

- Khan, Q. W. (2024). Exploring Markov Decision Processes: A Comprehensive Survey of Optimization Applications and Techniques. *Igmin Research*, 2(7), 508–517. https://www.igmin-research.com/articles/html/igmin210_adresinden_alinmistir.
- Kochenderfer, M. J. (2019). *Algorithms for Optimization*. The MIT Press Cambridge.
- Li, G. G., Wei, J. J., Xie, F. X., & Wei, S. S. (2023). Multi-objective UAV Trajectory Planning Based on Improved Wolf Pack Algorithm Improved Wolf Pack Algorithm. *ACM Other conferences*. Association for Computing Machinery. doi: 10.1145/3625403.3625429
- Liu, L.-s., Lin, J.-f., Yao, J.-x., He, D.-w., Zheng, J.-s., Huang, J., & Shi, P. (2021). Path Planning for Smart Car Based on Dijkstra Algorithm and Dynamic Window Approach. *Wireless Communications and Mobile Computing*, 2021(1), 8881684. <https://doi.org/10.1155/2021/8881684>
- Liu, Q., Chen, E., Chen, C., & Jiang, X. (2020). Detecting fake accounts in online social networks using graph topological features. *Information Processing & Management*, 57(4), 102236. <https://doi.org/10.1016/j.ipm.2020.102236>
- Lusiani, A., Purwaningsih, S. S., & Sartika, E. (2023). Dijkstra Algorithm In Determining The Shortest Route For Delivery Service By J&T Express In Bandung. *Jurnal Lebesgue: Jurnal Ilmiah Pendidikan Matematika, Matematika dan Statistika*, 4(2), 940-948.
- Ma, X., Liu, F., Qi, Y., Wang, X., Li, L., Jiao, L., ... & Gong, M. (2015). A multiobjective evolutionary algorithm based on decision variable analyses for multiobjective optimization problems with large-scale variables. *IEEE Transactions on Evolutionary Computation*, 20(2), 275-298.
- Ouassam, E., Hmina, N., Bouikhalene, B., & Hachimi, H. (2021). Heuristic Methods: Application to Complex Systems. <https://doi.org/10.1109/ICOA51614.2021.9442647>
- Parimala, M., Broumi, S., Prakash, K., & Topal, S. (2021). Bellman-Ford algorithm for solving shortest path problem of a network under picture fuzzy environment. *Complex & Intelligent Systems*, 7, 2373-2381.
- Peng, Q., Zhan, R., Wu, H., & Shi, M. (2024). Comparative Study of Wolf Pack Algorithm and Artificial Bee Colony Algorithm. *International Journal of Swarm Intelligence Research*. <https://doi.org/10.4018/ijcir.352061>
- Raeli, A., Salina Borello, E., Panini, F., Serazio, C., & Viberti, D. (2024). A parallel programming application of the A* algorithm in digital rock physics. *Comput. Geosci.*, 187, 105578. doi: 10.1016/j.cageo.2024.105578
- Rahul, K., & Banyal, R. K. (2020). Firefly algorithm: an optimization solution in big data processing for the healthcare and engineering sector. *Int. J. Speech Technol.*, 24(3), 581–592. doi: 10.1007/s10772-020-09783-y
- Singh, P., Meena, N. K., & Yang, J. (2020). Ant colony optimization, modifications, and application. In *Swarm Intelligence Algorithms* (pp. 1-14). CRC Press.
- Talunohi, J., Sembiring, Z., Khairina, N., Novita, N., Anisa, Y., & Rambe, Y. S. (2023, November). Analysis of the Dijkstra Algorithm in Determining The Shortest Route to Tour the Beaches of Nias Island. In *2023 International Conference of Computer Science and Information Technology (ICOSNIKOM)* (pp. 1-5). IEEE.
- Tilahun, S. L., & Ngnotchouye, J. M. T. (2016). Firefly Algorithm for optimization problems with non-continuous variables: A Review and Analysis. *arXiv*, 1602.07884. <https://arxiv.org/abs/1602.07884v1> adresinden alınmıştır.
- Wang, G., Zhang, X., & Wang, J. (2019). Artificial fish swarm algorithm optimized deep belief networks for intrusion detection. *Knowledge-Based Systems*, 163, 848-861. <https://doi.org/10.1016/j.knosys.2018.08.017>
- Wang, J., Liang, Y., & Huang, H. J. (2015). Traffic signal optimization selection using artificial fish swarm algorithm. *Transportation Research Part C: Emerging Technologies*, 55, 460-473. <https://doi.org/10.1016/j.trc.2015.03.014>
- Wang, X., & Jin, Y. (2023). An Ant Colony Algorithm Assisted by Graph Neural Networks for Solving Vehicle Routing Problems. *ACM Conferences*. Association for Computing Machinery. doi: 10.1145/3583133.3596424
- Wu, G., Li, W., Li, J., Jiang, G., & Cheng, Y. . Improved A* Algorithm AUV Path Planning Based on Multi-Thread Parallelism and CUDA Optimization. *2024 5th International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT)*. IEEE. doi: 10.1109/AINIT61980.2024.10581438
- Wu, H. S., & Zhang, F. M. (2014). Wolf pack algorithm for unconstrained global optimization. *Mathematical Problems in Engineering*, 2014(1),

465082.

- XiangRong, T., Yukun, Z., & XinXin, J. (2021, February). Improved A-star algorithm for robot path planning in static environment. In *Journal of Physics: Conference Series* (Vol. 1792, No. 1, p. 012067). IOP Publishing.
- Xu, D., Yue, H., Zhao, Y., Yang, F., Wu, J., Pan, X., ...Zhang, Y. (2024). Improved A* Algorithm for Path Planning Based on CubeSats In-Orbit Electromagnetic Transfer System. *Aerospace*, 11(5), 394. doi: 10.3390/aerospace11050394
- Zhang, H., Cui, L., Neumann, M., & Chen, Z. (2021). An overview of deep learning in big data. *Information Fusion*, 76, 87-106. <https://doi.org/10.1016/j.inffus.2021.03.006>
- Zhang, H., Liu, S., Liu, Y., & Wang, X. (2020). Artificial fish swarm algorithm optimized support vector machine for data classification. *Knowledge-Based Systems*, 195, 105708. <https://doi.org/10.1016/j.knosys.2020.105708>
- Zimmermann, M., & von Hoessle, J. E. (2013). Computing solution spaces for robust design. *International Journal for Numerical Methods in Engineering*, 94(3), 290-307.