

ANKARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

YAPAY ZEKA OPTİMİZASYON ALGORİTMALARI İLE DOĞRUSAL
OLMAYAN REGRESYON MODELLERİNDE PARAMETRE TAHMİNİ

Murat AÇIKGÖZ

İSTATİSTİK ANABİLİM DALI

ANKARA
2025

Her hakkı saklıdır

ÖZET

Yüksek Lisans Tezi

YAPAY ZEKA OPTİMİZASYON ALGORİTMALARI İLE DOĞRUSAL OLMAYAN REGRESYON MODELLERİNDE PARAMETRE TAHMİNİ

Murat AÇIKGÖZ

Ankara Üniversitesi
Fen Bilimleri Enstitüsü
İstatistik Anabilim Dalı

Danışman: Prof. Dr. Özlem TÜRKŞEN

Doğrusal olmayan regresyon modelleri, gerçek dünya problemlerinin modellenmesi konusunda birçok bilimsel alanda yaygın olarak kullanılmaktadır. Bu modellerde parametre tahmini, hata kareler toplamının minimizasyonuna dayalı bir optimizasyon probleminin çözümü olarak gerçekleştirilmektedir. Amaç fonksiyonunun türevlenebilir olmaması, yerel minimum tuzaklarına takılma, hesaplama süresinin uzaması gibi nedenlerden dolayı geleneksel yöntemlerle modelin optimizasyon süreci zorlaşmaktadır. Bu çalışmada, yapay zeka tabanlı optimizasyon algoritmaları ile doğrusal olmayan regresyon modellerinde parametre tahmini konusu ele alınmıştır. Çalışmada evrim tabanlı algoritmalar arasında Genetik Algoritma, Diferansiyel Evrim Algoritması ve Çiçek Tozlaşması Algoritması yer alırken, sürü tabanlı algoritmalar arasında Yapay Arı Koloni Algoritması, Karınca Koloni Optimizasyonu, Yarasa Algoritması, Guguk Kuşu Algoritması, Gri Kurt Optimizasyonu, Jaya Algoritması ve Parçacık Sürü Optimizasyonu bulunmaktadır. Ayrıca müzik esinli algoritmalarından Armoni Arama Algoritması, insan davranışından ilham alan Öğretme-Öğrenme Tabanlı Optimizasyonu olmak üzere on iki farklı yapay zeka optimizasyon algoritması kullanılarak literatürde tanınmış üç farklı doğrusal olmayan regresyon modeli üzerinde performans değerlendirmesi yapılmıştır. Algoritmaların performansları, hata metrikleri ve hesaplama süreleri gibi kriterlere dayalı olarak IDOCRIW yöntemiyle ağırlıklandırılarak, Çok Kriterli Karar Verme yöntemleri olan COPRAS, EDAS ve MABAC ile sıralanmış ve sonuçlar Copeland yöntemi ile bütünleştirilerek bir sonuç kararı elde edilmiştir. Yapay zeka tabanlı optimizasyon algoritmalarının doğrusal olmayan regresyon modellerinde parametre tahmini sürecine yönelik uygulamalarda elde edilen bulgulara göre Diferansiyel Evrim Algoritması, Parçacık Sürü Optimizasyonu ile Çiçek Tozlaşması Algoritması yöntemleri üstün performans gösteren yapay zeka tabanlı optimizasyon algoritması olarak belirlenmiştir.

Şubat 2025, 111 sayfa

Anahtar Kelimeler: Yapay Zeka Optimizasyon Algoritmaları, Çok Kriterli Karar Verme Yöntemleri, Doğrusal Olmayan Regresyon Modelleri, Parametre Tahmini

ABSTRACT

Master Thesis

PARAMETER ESTIMATION OF NONLINEAR REGRESSION MODELS WITH ARTIFICIAL INTELLIGENCE OPTIMIZATION ALGORITHMS

Murat AÇIKGÖZ

Ankara University
Graduate School of Natural and Applied Sciences
Department of Statistics

Supervisor: Prof. Dr. Özlem TÜRKŞEN

Nonlinear regression models are widely used across various scientific fields for modeling real-world problems. In these models, parameter estimation is carried out as a solution to an optimization problem based on the minimization of the sum of squared errors. However, challenges such as the non-differentiability of the objective function, susceptibility to local minima, and increased computational time complicate the optimization process using traditional methods. This study addresses parameter estimation in nonlinear regression models using artificial intelligence-based optimization algorithms. Among the evolutionary algorithms, Genetic Algorithm, Differential Evolution Algorithm, and Flower Pollination Algorithm were employed, while the swarm intelligence-based algorithms included Artificial Bee Colony, Ant Colony Optimization, Bat Algorithm, Cuckoo Search Algorithm, Grey Wolf Optimization, Jaya Algorithm, and Particle Swarm Optimization. Additionally, Harmony Search from music-inspired algorithms and Teaching-Learning-Based Optimization from human behavior-inspired methods were utilized, making a total of twelve different artificial intelligence optimization algorithms evaluated on three well-known nonlinear regression models from the literature. The performance of the algorithms was assessed based on criteria such as error metrics and computational time, and these were weighted using the IDOCRIW method. The algorithms were then ranked using multi-criteria decision-making methods including COPRAS, EDAS, and MABAC, and the final decision was synthesized using the Copeland method. According to the findings from the applications of artificial intelligence-based optimization algorithms in parameter estimation for nonlinear regression models, Differential Evolution, Particle Swarm Optimization, and Flower Pollination Algorithm were identified as the top-performing methods.

February 2025, 111 pages

Key Words: Artificial Intelligence Optimization Algorithms, Multi-Criteria Decision Making Methods, Nonlinear Regression Models, Parameter Estimation

TEŞEKKÜR

Bir oyuncunun simülasyon dünyasında gizli mekanikleri keşfetmesi misali, ben de bu araştırmada istatistiğin görünmeyen dinamiklerini, varlığın kendisini sorgulayan bir yolculukla irdeledim. Her adımda hem geçmişin izlerini hem de geleceğin umutlarını araladım.

Bu keşif serüveninde, düşünsel ufkumu genişleterek yeni disiplinlerin kapılarını aralayan rehberim Prof. Dr. Özlem Türkşen'e ve Ankara Üniversitesi İstatistik Bölümü'nün değerli akademisyenlerine yürekten minnettarım. Zorlu bölümlerin üstesinden gelmemde yanımda olan aileme ve tüm dostlarıma da içten teşekkürlerimi sunarım.

Gerçek dünyadaki en zor level olan 'bilim', varlığın özüne ulaşma arzusunu yansıtır nitelikte. Umarım bu çalışma, bilimin ışığıyla hem gölgeleri aydınlatır hem de istatistiksel derinliklerin saklı olan anlamını gün yüzüne çıkarır.

Murat AÇIKGÖZ
Ankara, Şubat 2025

İÇİNDEKİLER

TEZ ONAYI

ETİK.....	i
ÖZET.....	ii
ABSTRACT	iii
TEŞEKKÜR	iv
KISALTMALAR DİZİNİ	vii
ŞEKİLLER DİZİNİ	viii
ÇİZELGELER DİZİNİ	ix
1. GİRİŞ	1
2. DOĞRUSAL OLMAYAN REGRESYON ANALİZİ	13
2.1 Doğrusal Olmayan Regresyon Modeli	13
2.2 Model Örnekleri ve Uygulama Alanları	14
2.3 Model Parametrelerinin Nokta Tahmini	16
2.4 Model Performans Metrikleri.....	19
3. YAPAY ZEKA OPTİMİZASYON ALGORİTMALARI	21
3.1 Evrimsel Tabanlı Yöntemler	22
3.1.1 Genetik Algoritma.....	22
3.1.2 Diferansiyel Evrim Algoritması	26
3.1.3 Çiçek Tozlaşması Algoritması.....	29
3.2 Sürü Tabanlı Yöntemler	32
3.2.1 Yapay Arı Koloni Algoritması	32
3.2.2 Karınca Koloni Optimizasyonu	36
3.2.3 Yarasa Algoritması	39
3.2.4 Guguk Kuşu Algoritması.....	42
3.2.5 Gri Kurt Optimizasyonu	45
3.2.6 Jaya Algoritması	49
3.2.7 Parçacık Sürü Optimizasyonu	52
3.3 Müzik Esinli Yöntemler.....	55
3.3.1 Armoni Arama Algoritması	56
3.4 İnsan Davranışında Etkilenen Yöntemler	59
3.4.1 Öğretme-Öğrenme Tabanlı Optimizasyon	59

4. ÇOK KRİTERLİ KARAR VERME YÖNTEMLERİ	62
4.1 Ağırlıklandırma.....	63
4.1.1 IDOCRIW	64
4.2 Sıralama	67
4.2.1 COPRAS	68
4.2.2 EDAS	70
4.2.3 MABAC.....	73
4.3 Genel Sıralama	75
4.3.1 Copeland Yöntemi.....	75
5. UYGULAMA.....	78
5.1 Uygulama 1 (BoxBOD Modeli)	83
5.2 Uygulama 2 (Meyer-1 Modeli)	88
5.3 Uygulama 3 (İkili Kompartman Modeli)	94
6. SONUÇ.....	100
KAYNAKLAR	104
ÖZGEÇMİŞ.....	111

KISALTMALAR DİZİNİ

ABC	Artificial Bee Colony Algorithm (Yapay Arı Koloni Algoritması)
ACOR	Ant Colony Optimization for Continuous (Karıncı Koloni Optimizasyonu)
AI	Artificial Intelligence (Yapay Zeka)
BA	Bat Algorithm (Yarasa Algoritması)
CSA	Cuckoo Search Algorithm (Guguk Kuşu Algoritması)
COPRAS	Complex Proportional Assessment
ÇKKV	Çok Kriterli Karar Verme
DE	Differential Evolution Algorithm (Diferansiyel Evrim Algoritması)
EDAS	Evaluation based on Distance from Average Solution
FPA	Flower Pollination Algorithm (Çiçek Tozlaşma Algoritması)
GA	Genetic Algorithm (Genetik Algoritma)
GWO	Grey Wolf Optimization (Gri Kurt Optimizasyonu)
HS	Harmony Search Algorithm (Armoni Arama Algoritması)
IDOCRIW	Integrated Determination of Objective Criteria Weights
JA	Jaya Algorithm (Jaya Algoritması)
MABAC	Multi-Attributive Border Approximation area Comparison
PSO	Particle Swarm Optimization (Parçacık Sürü Optimizasyonu)
TLO	Teaching-Learning Based Optimization (Öğretme-Öğrenme Tabanlı Optimizasyon)

ŞEKİLLER DİZİNİ

Şekil 3.1 GA'nın işleyiş diyagramı	24
Şekil 3.2 DE'nin işleyiş diyagramı	27
Şekil 3.3 FPA'nın işleyiş diyagramı	30
Şekil 3.4 ABC'nin işleyiş diyagramı	34
Şekil 3.5 ACOR'un işleyiş diyagramı	37
Şekil 3.6 BA'nın işleyiş diyagramı.....	40
Şekil 3.7 CSA'nın işleyiş diyagramı	44
Şekil 3.8 GWO'nun işleyiş diyagramı.....	47
Şekil 3.9 JA'nın işleyiş diyagramı.....	51
Şekil 3.10 PSO'nun işleyiş diyagramı	54
Şekil 3.11 HS'nin işleyiş diyagramı	57
Şekil 3.12 TLO'nun işleyiş diyagramı.....	60
Şekil 5.1 Metrikler üzerinden yürütülen ÇKKV yöntemlerinin diyagramı	81
Şekil 5.2 Veri setinin saçılım grafiği	84
Şekil 5.3 Kriterlerin IDOCRIW yöntemine göre ağırlıkları	87
Şekil 5.4 ÇKKV yöntemlerine göre algoritmaların sıralaması	88
Şekil 5.5 AI optimizasyon algoritmalarının Copeland yöntemine göre sıralamaları.....	88
Şekil 5.6 Veri setinin dağılım saçılım grafiği	89
Şekil 5.7 Kriterlerin IDOCRIW yöntemine göre ağırlıkları	91
Şekil 5.8 ÇKKV yöntemlerine göre algoritmaların sıralaması	93
Şekil 5.9 AI optimizasyon algoritmalarının Copeland yöntemine göre sıralamaları.....	93
Şekil 5.10 Veri setinin dağılım saçılım grafiği	95
Şekil 5.11 Kriterlerin IDOCRIW yöntemine göre ağırlıkları	98
Şekil 5.12 ÇKKV yöntemlerine göre algoritmaların sıralaması.....	99
Şekil 5.13 AI optimizasyon algoritmalarının Copeland yöntemine göre sıralamaları.....	99

ÇİZELGELER DİZİNİ

Çizelge 2.1 Sıklıkla kullanılan doğrusal olmayan regresyon model örnekleri	15
Çizelge 2.2 Doğrusal olmayan regresyon modellerinin uygulama alanları	16
Çizelge 2.3 Doğrusal olmayan regresyon modellerinin çözümü için kullanılan türeve dayalı sayısal optimizasyon yöntemleri	18
Çizelge 2.4 Doğrusal olmayan regresyon modellerinde kullanılan performans metrikleri	20
Çizelge 5.1 AI optimizasyon algoritmalarının ayarlanabilir parametre değerleri.....	79
Çizelge 5.2 Mikroorganizmaların zamana göre biyokimyasal oksijen ihtiyacı.....	83
Çizelge 5.3 Model parametrelerinin arandığı alt ve üst sınırlar.....	84
Çizelge 5.4 AI optimizasyon algoritmalarının en iyi performans gösterdiği ayarlanabilir parametre değerleri	85
Çizelge 5.5 AI optimizasyon algoritmalarının performans çıktıları ve model parametre tahminleri	86
Çizelge 5.6 Modelde kullanılan değişkenler ve veri seti	89
Çizelge 5.7 Model parametrelerinin arandığı alt ve üst sınırlar.....	90
Çizelge 5.8 AI optimizasyon algoritmalarının en iyi performans gösterdiği ayarlanabilir parametre değerleri	90
Çizelge 5.9 AI optimizasyon algoritmalarının performans çıktıları ve model parametre tahminleri	92
Çizelge 5.10 Vücuda alınan bir ilacın zamana göre plazma konsantrasyonu	94
Çizelge 5.11 Model parametrelerinin arandığı alt ve üst sınırlar	94
Çizelge 5.12 AI optimizasyon algoritmalarının en iyi performans gösterdiği ayarlanabilir parametre değerleri.....	96
Çizelge 5.13 AI optimizasyon algoritmalarının performans çıktıları ve model parametre tahminleri.....	97

1. GİRİŞ

Doğrusal olmayan regresyon modelleri, mühendislik, ekonomi, sağlık, sosyal ve fen bilimleri gibi pek çok alanda bağımlı ve bağımsız değişkenler arasındaki doğrusal olmayan ilişkileri doğrusal varsayımlarla modellemenin ötesine taşıyarak bu alanların gelişmesinde önemli bir rol oynamaktadır. Gerçek dünyadaki çok değişkenli ve çoğunlukla doğrusal olmayan ilişkileri daha doğru ve gerçekçi bir biçimde temsil eden bu modeller, araştırmacılara daha hassas tahminler yapma ve derinlemesine analizler gerçekleştirme imkânı sunmaktadır. Ancak, bu modellerin etkin bir biçimde kullanılabilmesi, model parametrelerinin nokta tahminindeki zorluklar nedeniyle sınırlanmaktadır. Modellerin parametrelerine göre doğrusal olmaması, geleneksel yöntemlerin uygulamasını zorlaştırmakta ve bu durum alternatif yöntemlerin geliştirilmesine olan ihtiyacı artırmaktadır.

Doğrusal olmayan regresyon modellerinin sunduğu avantajlar, bu modelleri pek çok alanda vazgeçilmez hale getirmiştir. Gerçek dünya verilerinin genellikle doğrusal olmayan özellikler göstermesi, doğrusal olmayan regresyon modellerinin önemini daha da öne çıkarmaktadır. Örneğin, bir ilacın kan dolaşımındaki etkisinin zamana bağlı olarak azalması ya da bir ürünün talebinin fiyat değişikliklerine tepkisi, genellikle doğrusal olmayan fonksiyonlarla açıklanabilmektedir. Bu gibi durumlarda, doğrusal olmayan regresyon modelleri çok değişkenli dinamikleri modellemede etkili bir araç olurken, doğrusal regresyon modellerinin yetersiz kaldığı durumlarda, gerçek dünya olgularını daha iyi temsil ederek çözüm sunmaktadır. Literatürde yaygın bir biçimde kullanılan doğrusal olmayan regresyon modelleri üstel ve rasyonel fonksiyonlu olmak üzere iki başlıkta incelenebilir. Doğrusal olmayan regresyon modellerin parametre nokta tahminlerinin türeve dayalı geleneksel yöntemlerle elde edilmesi konusunda farklı çıkarımlar yapılmış olup konuya ilişkin çeşitli çalışmalar mevcuttur. Doğrusal olmayan regresyon modelleri, biyolojik süreçlerden finansal piyasalara, jeolojik oluşumlardan epidemiyolojik modellere kadar geniş bir yelpazede, özgün sistemlerin dinamiklerini çözümlemek ve geleceğe yönelik öngörülerde bulunmak için güçlü bir araç olarak kullanılmaktadır.

Welagedara vd. (2019) çalışmalarında, Sri Lanka nüfusunun gelecekteki büyüklüğünü tahmin etmek için Gompertz ve Verhulst lojistik büyüme modellerini kullanmışlardır. Veriler 1990-2018 yılları arasında nüfus sayımı, doğum ve ölüm oranlarından elde edilmiştir. Modellerin tahmin doğruluğu, Hata Kareler Ortalamasının Karekökü, Ortalama Mutlak Yüzde Sapma ve Simetrik Ortalama Mutlak Yüzde Hata ile değerlendirilmiştir. Gompertz modelinin daha yüksek doğruluk sağladığı ve gelecekteki nüfus büyüklüğünü tahmin etmekte Verhulst modeline göre daha uygun olduğu sonucuna varılmıştır. Çalışma, sınırlı kaynaklara sahip bir ülke için nüfus tahmininde önemli bir araç sunmaktadır.

Miyoshi vd. (1996) çalışmalarında, tavukların yumurta üretim eğrilerini modellemek için dört farklı doğrusal olmayan matematiksel model (kompartıman, modifiye kompartıman, Wood ve Adams-Bell) kullanılmıştır. Modeller, yumurta üretim verilerindeki altı farklı desene uygulanmış ve uyum kriterleri (R^2 ve AIC) ile karşılaştırılmıştır. Çalışma, modifiye kompartıman ve Adams-Bell modellerinin, genel yumurta üretim eğrilerine iyi uyum sağladığını ancak hızlı üretim düşüşleri içeren desenlerde tüm modellerin yetersiz kaldığını ortaya koymaktadır. Bununla birlikte, Wood modeli her desene uygulanabilmiş ancak diğer modellere kıyasla daha düşük uyum göstermiştir. Çalışma, büyük sürüler için yıllık yumurta üretimi tahmininde bu modellerin uygulanabilir olduğunu vurgulamıştır.

Grossman-Ponemon vd. (2020) çalışmalarında, bir magmatik odadan yayılan kayaç dikeylerinin büyümesini modellemek için tam bağlı iki boyutlu simülasyonlar gerçekleştirilmiştir. Çalışma, magmatik odanın basınç kaybını, sıvı akışını, çatlak oluşumunu ve elastik deformasyonu bir araya getiren sistemlerin dinamiklerini anlamaya odaklanmıştır. Dik uzunluğunun zamanla logaritmik olarak arttığını ifade eden basit bir model önerilmiş ve bu modelin İzlanda ve Etiyopya'daki kayaç dikeylerinden elde edilen sismik verilerle uyumlu olduğu belirlenmiştir. Ayrıca, magmatik oda basıncının dik uzunluğu ile nasıl değiştiğine dair bir model türetilmiştir. Araştırma Sonuçları, kayaç dikeyinin büyümesi ve magmatik oda basınç kaybının öngörülmesinde bu basit modellerin kullanılabilir olduğunu göstermektedir.

Girdzijauskas ve Streimikiene (2008) çalışmalarında, lojistik büyüme modellerini kullanarak hisse senedi piyasası balonlarının oluşum mekanizmalarını incelemişlerdir. Çalışma, sınırlı kaynaklar altında büyümenin doygunluğunun balonların oluşmasına nasıl yol açtığını açıklamayı hedeflemiştir. Hisse senedi piyasalarındaki balonların temel özelliklerini ve nedenlerini analiz eden bu çalışmada 1920 ile 1990'daki balonlar örnek olarak incelenmiştir. Araştırmacılar, lojistik büyüme modellerinin, sistemin kararlılık derecesini göstermek ve piyasa balonlarını öngörmek için uygulanabilir olduğunu ortaya koymuşlardır. Bu yaklaşım, yatırım değerlendirmeleri ve piyasa doyum düzeyi hesaplanmasında daha hassas analizler yapılmasını olanak tanımaktadır.

Murugan (2024) çalışmasında, Michaelis-Menten tipi enzim kinetiğinin tam ve kısmi rekabetçi inhibisyon durumlarını incelemek için farklı hız denklemleri geliştirilmiştir. Çalışma, enzimin substrat ve inhibitör bağlanma dinamiklerine bağlı olarak iki farklı kararlı duruma geçebileceği gösterilmiştir. Geleneksel yarı kararlı durum yaklaşımının, substrat ve inhibitör bağlanma oranlarının farklı olduğu durumlarda hatalara yol açabileceği sonucuna ulaşılmıştır. Geliştirilen modeller, özellikle enzimin substrat ve inhibitör konsantrasyonlarının benzer düzeylerde olduğu durumlarda kinetik parametre tahminlerinde mevcut yöntemlere kıyasla daha doğru sonuçlar vermiştir. Çalışma, rekabetçi inhibisyonun enzim kinetiği üzerindeki etkilerini anlamada önemli katkılar sunmaktadır.

Gupta vd. (2023) çalışmalarında, epidemiyolojik bölmeli modellerin kalibrasyonu amacıyla optimizasyon ve güçlendirme öğrenimi yöntemlerini karşılaştırmıştır. Çalışmada, mevcut veri ve model dinamiklerine bağlı olarak en uygun parametreleri belirlemek için gradyan bazlı ve gradyan içermeyen çeşitli algoritmaları kullanılmıştır. Nelder-Mead, Powell ve en küçük kareler yöntemleri düşük veri rejiminde en iyi performansı gösterirken, yüksek veri rejiminde Nelder-Mead ve tröst bölgesi optimizasyon yöntemleri öne çıkmıştır. Gürültülü verilerle yapılan deneyler, bu yöntemlerin dayanıklılığını doğrulamış ve güçlendirme öğrenimi kullanılarak parametre tahminlerinin daha da iyileştirilebileceğini ortaya koymuştur. Epidemik model analizlerinde kullanılan bu yöntemlerin karar alıcılar için model doğruluğunu artırma potansiyeline sahip olduğu vurgulanmıştır.

Doğrusal olmayan regresyon model parametrelerinin nokta tahmininin elde edilmesi aşamasında kullanılan optimizasyon sürecinde geleneksel yöntemler sıklıkla başvurulmaktadır. Bu yöntemler arasında Gauss-Newton, Levenberg-Marquardt ve Newton-Raphson öne çıkmaktadır. Ancak, bu yöntemler belirli varsayımlar altında etkili olabilmektedir. Optimize edilecek amaç fonksiyonunun sürekli, türevlenebilir ve konveks olması durumlarında geleneksel optimizasyon yöntemlerinin başarılı sonuçlar verebileceği gözlenmiştir. Bu durum, söz konusu yöntemlerin iyi tanımlanmış matematiksel altyapılarının bulunması ve belirli koşullar altında hızlı çözümler üretebilmesinin bir sonucudur. Özellikle Gauss-Newton ve Levenberg-Marquardt yöntemleri, parametrelerin türevlerinin alınabildiği durumlarda oldukça etkili çalışmaktadır (Yonar ve Yonar, 2022). Bununla birlikte, bu yöntemler parametrelerin başlangıç değerlerine duyarlı olup, yerel minimumlara takılma riski taşımakta ve global optimuma ulaşmada zorluk yaşayabilmektedir (Sabastine vd., 2022). Özellikle yüksek boyutlu arama uzaylarında, geleneksel yöntemlerin etkinliği azalabilmektedir. Bu sınırlamaların üstesinden gelmek amacıyla, literatürde yapay zeka (Artificial Intelligence - AI) tabanlı meta-sezgisel algoritmaların potansiyeli giderek daha fazla dikkat çekmektedir. AI optimizasyon algoritmaları olarak da adlandırılan yöntemler, minimizasyon ya da maksimizasyon yapılması istenen bir amaç fonksiyonunu optimize ederek model parametrelerinin nokta tahmininin elde edilmesi amacıyla kullanılmaktadır (Türkşen ve Tez, 2016). Bu algoritmalar, özellikle büyük veri kümeleri ve zorlu problem yapılarına sahip sistemlerde geleneksel optimizasyon yöntemlerine göre daha hızlı ve etkili çözümler sunabilmektedir.

AI tabanlı optimizasyon algoritmaları geniş arama uzaylarını keşfetme ve yerel minimumlara takılma sorununu minimize etme yetenekleriyle ön plana çıkmaktadır. Bu algoritmalar, özellikle zor optimizasyon problemlerin çözümünde etkili bir biçimde kullanılmaktadır (Türkşen, 2023). Günümüzde modern yaklaşımlar, AI ve sezgisel algoritmalar üzerine yoğunlaşmıştır. Bu kapsamda, Genetik Algoritma (Genetic Algorithm - GA) ve Parçacık Sürü Optimizasyonu (Particle Swarm Optimization - PSO) gibi sezgisel algoritmalar doğrusal olmayan modellerdeki parametre nokta tahmini için etkili araçlar olarak dikkat çekmektedir (Martinelli vd., 2024). Bu AI algoritmaları, geniş bir arama uzayında çalışabilmeleri ve parametrelerin başlangıç değerlerine duyarlı

olmama gibi avantajları sayesinde, geleneksel yöntemlere alternatif sunmaktadır (Lai vd., 2020). Ayrıca, bu algoritmaların tek işlemci yerine eş zamanlı olarak birden fazla işlemci kullanan paralel hesaplama yetenekleri, büyük ölçekli problemlerde hesaplama süresini önemli ölçüde azaltarak pratik uygulanabilirliği artırmaktadır.

Zhang vd. (2013) çalışmalarında, Yapay Arı Kolonisi (Artificial Bee Colony - ABC) algoritmasını kullanarak Ağırlıklı Alt Uzay Uyumlama yöntemine dayalı varış yönü ve Doppler frekansı tahmini için bir yöntem geliştirmiştir. Çalışmada, parametre nokta tahmini problemini doğrusal olmayan çok boyutlu bir optimizasyon problemine dönüştürmek için Ağırlıklı Alt Uzay Uyumlama yöntemi kullanılmıştır. ABC algoritması, daha düşük hesaplama yüküyle ve yüksek doğrulukta asimptotik yansız tahminler sağlamak için kullanılmıştır. Geliştirilen yöntem, düşük sinyal gürültü oranı değerlerinde ve anlık sayılarda diğer yöntemlerden daha iyi performans göstermiştir.

Nobahari ve Sharifi (2019) yapmış olduğu çalışmada, doğrusal olmayan sistemler için durum tahmini amacıyla genişletilmiş bir kalman filtresi ile sürekli karınca kolonisi algoritmasını birleştiren yeni bir algoritma önermişlerdir. Bu algoritma, ölçüm anında karıncaların konumlarını kalman filtresi ile güncellerken, durum uzayında optimum tahmini aramak için karınca kolonisi optimizasyonunu kullanmıştır. Yeni hibrit algoritma, yüksek doğruluk ve kararlı performans göstererek mevcut tekniklerden daha üstün sonuçlar sağlamıştır. Algoritmanın insansız hava araçlarının durumu ve rüzgar bileşenlerini tahmin etmekte başarılı olduğu ve gerçek zamanlı uygulamalar için uygun olduğu kanıtlanmıştır.

Ramsundram vd. (2023), doğrusal olmayan Muskingum sel yönlendirme modelinin parametre nokta tahmini için yarası optimizasyon algoritmasını (Bat Algorithm - BA) kullanmışlardır. Çalışmada, model parametreleri minimum hata ölçütlerine dayalı olarak optimize edilmiştir. BA, diğer AI optimizasyon yöntemleriyle karşılaştırıldığında daha düşük *RMSE* ve artık hata ile üstün performans göstermiştir. BA'nın, rastgele uçuşlar ve geniş çözüm alanını keşfetme yeteneği sayesinde yerel optimallere takılmadan global optimuma ulaşmayı sağladığı belirtilmiştir. Çalışma, bu algoritmanın su kaynakları yönetimi ve sel yönlendirme modellerinde kullanılabilirliğini vurgulamaktadır.

Maciel vd. (2016), ABD hazine tahvilleri için getiri eğrilerini tahmin etmek amacıyla Diferansiyel Evrim (Differential Evolution - DE) algoritması önermiştir. Nelson-Siegel ve Svensson fonksiyonları kullanılarak, geleneksel doğrusal olmayan en küçük kareler yöntemini evrimsel algoritmalarla karşılaştırılmıştır. Çalışma, DE algoritmasının başlangıç parametre değerlerine bağımlılığı azaltarak, daha yüksek doğruluk ve daha düşük işlem süresi ile diğer yöntemleri geride bıraktığını göstermiştir. DE'nin basitliği ve kontrol parametrelerinin azlığı, finansal optimizasyon problemlerinde pratik bir araç olarak kullanımını öne çıkarmaktadır.

Roeva ve Trenkova (2012), GA ve Ateşböceği Algoritması (Firefly Algorithm - FA) kullanarak doğrusal olmayan biyoproses modelinde parametre nokta tahmini gerçekleştirmiştir. E. Coli mikroorganizmaları fed-batch kültür yöntemine dayalı olarak, maksimum özgül büyüme hızı, doyumluk sabitleri ve verim katsayılarını içeren model parametreleri optimize edilmiştir. FA'nın daha hızlı yakınsadığı ve daha düşük amaç fonksiyonu değerleri sağladığından dolayı GA'ya göre daha doğru ve tutarlı sonuçlar sunduğu görülmüştür. FA'nın, özellikle biyolojik sistemlerin çok değişkenli ve doğrusal olmayan yapısını ele almada güçlü bir meta-sezgisel yöntem olduğu sonucuna varılmıştır.

Lara ve Guerra (2023), yapmış olduğu çalışmada doğrusal olmayan titreşim sistemlerindeki Bouc-Wen modeli parametrelerini tahmin etmek için GA kullanmıştır. Çalışmada, yayların histerezis davranışını modellemek için Bouc-Wen modelinin ayırık bir yaklaşımı uygulanmış ve Euler geriye doğru ayırıklaştırma yöntemi kullanılmıştır. Algoritma, deneysel histerezis döngü verilerini kullanarak Bouc-Wen model parametrelerini optimize etmiştir. Algoritma, yüksek doğruluk sağlarken yerel optimallere takılmadan etkili bir yakınsama göstermiştir. Deneysel sonuçlar ile algoritmanın tahminleri arasında yüksek bir uyum gözlenmiş, bu durum yöntemin doğruluğunu ve uygulanabilirliğini ortaya koymuştur.

Gopal vd. (2021), COVID-19 pandemisine ilişkin Susceptible-Infectious-Removed modeli parametrelerini tahmin etmek için Armoni Arama (Harmony Search - HS) algoritması ve on varyantını kullanmışlardır. Beş farklı ülkeye ait veri setleri üzerinde yapılan çalışmada, HS algoritmalarının parametre nokta tahminindeki etkinliği

değerlendirilmiştir. Simülasyonlar ve projeksiyonlar, varyantların arasında bazılarının daha düşük hata oranları sağladığını göstermiştir. HS algoritmalarının, doğrusal olmayan regresyon model için doğru tahminler üreterek, salgının dinamiklerini modelleme ve öngörme kapasitesine katkı sağladığı belirtilmiştir.

Kang vd. (2021), beton barajların dinamik parametre ters analizinde Gaussian süreç regresyonu ve Jaya algoritmasını birleştiren bir yöntem önermişlerdir. Gaussian süreç regresyonu, model parametreler ile dinamik elastik modül arasında bir yanıt yüzeyi oluşturmak için kullanılmış, model, parametre tahminlemede Jaya algoritması, PSO ve GA ile karşılaştırılmış ve modelin daha hızlı çalıştığı, aynı zamanda daha düşük hata oranları ile üstün performans sergilediği görülmüştür.

Fu vd. (2024), Çin hamsteri yumurtalık hücrelerinin kültür sürecinde mekanistik model parametrelerini tahmin etmek için iyileştirilmiş bir PSO algoritması önermiştir. PSO, GA ve tavlama benzetimi (Simulated Annealing - SA) ile karşılaştırılmış ve daha yüksek doğruluk ve daha hızlı yakınsama sağladığı gösterilmiştir. İkinci dereceden salınım ve parçacık yenileme stratejileri ile geliştirilen PSO algoritması, parametre tahmin performansını %35 artırmış hem deneysel hem de model tahmini verilerinde daha düşük *RMSE* değerleri ile daha yüksek uyum sağlamıştır. Algoritma, farklı klonlar ve koşullarda yüksek tahmin doğruluğu sergilemiş, başlangıç değerlerine ihtiyaç duymadan dahi etkili sonuçlar sunarak hücre kültürü modellemesinde geniş bir uygulama potansiyeli ortaya koymuştur.

Sezgisel algoritmalar olarak da adlandırılan AI optimizasyon algoritmaları, doğrusal olmayan regresyon modellerinin parametre nokta tahminine yönelik daha esnek çözümler sunmaktadır. GA, evrimsel süreçlerden ilham alarak parametre nokta tahmini yaparken, PSO popülasyon tabanlı bir yaklaşımla çalışmakta ve bir grup bireyin hareketlerinden faydalanmaktadır. ABC algoritması ise doğal sistemlerden esinlenerek çok değişkenli optimizasyon problemlerini çözmede kullanılan etkili bir araçtır. Bu algoritmalar, yerel minimumlara takılmadan geniş bir arama uzayını keşfedebilmekte ve başlangıç tahminlerine bağımlı olmadan çözüm üretebilmektedir (Erdoğan ve Ekiz, 2017). Ayrıca bu yöntemler, farklı problem türlerine uyarlanabilir yapıları sayesinde farklı disiplinlerde

başarılı bir biçimde uygulanabilmekte ve optimizasyon süreçlerinde esneklik sağlamaktadır.

Doğrusal olmayan regresyon modellerinin parametre tahmininde kullanılan AI tabanlı optimizasyon algoritmaları, çok sayıda modelin ve algoritmanın karşılaştırmalı olarak değerlendirilmesini gerekli kılmaktadır. Bu noktada, yalnızca tahmin doğruluğu değil, aynı zamanda hesaplama süresi, yakınsama hızı, başlangıç değerlerine duyarlılık gibi çeşitli kriterler de önem arz etmektedir. Farklı algoritmaların performanslarının çok boyutlu ve çok kriterli bir yapıda değerlendirilmesi gerekliliği, çalışmanın üçüncü boyutunu oluşturan Çok Kriterli Karar Verme (ÇKKV) yaklaşımlarını gündeme getirmektedir. ÇKKV yöntemleri, alternatif algoritmaların farklı kriterler ışığında sistematik biçimde değerlendirilmesine olanak sağlayarak, hangi algoritmanın hangi koşullarda daha üstün olduğunu ortaya koymak adına güçlü bir çerçeve sunmaktadır.

Karar verme süreçleri, artan belirsizliği ve çok boyutluluk gibi unsurlar nedeniyle geleneksel yöntemlerle yönetilmesi güç bir hal almıştır. Karar vericiler; ekonomik, sosyal, teknik ve çevresel gibi birbirinden farklı boyutlardaki kriterleri dikkate alarak en uygun alternatifi belirlemeye çalışmaktadır. Bu noktada, ÇKKV yöntemleri, karar problemlerinin sistematik, tutarlı ve bütüncül bir biçimde ele alınmasını mümkün kılan güçlü araçlar olarak öne çıkmaktadır (Alinezhad ve Khalili, 2019). ÇKKV yöntemleri, karar vericilerin hem nicel hem de nitel verilere dayalı olarak alternatifleri karşılaştırmasına, kriterlerin göreceli önemini belirlemesine ve en uygun seçeneği ortaya koymasına olanak sağlamaktadır. Söz konusu yöntemler, karar süreçlerine bilimsel bir çerçeve kazandırmakta, subjektif yargıların etkisini minimize ederken, veriye dayalı analizlerle desteklenen daha güvenilir sonuçlara ulaşılmasını mümkün kılmaktadır.

ÇKKV süreçlerinin temel yapı taşlarını kriter ağırlıklandırma ve alternatif sıralama yaklaşımları oluşturmaktadır. Kriter ağırlıklandırma süreci, hangi kriterin karar sürecinde ne derece etkili olduğunu belirlemeye yönelik olup, karar probleminin doğasına uygun ağırlıkların tespit edilmesini amaçlamaktadır. Buna göre, hem karar vericilerin sezgisel ve deneyimsel bilgisine dayanan subjektif yöntemler hem de veri setinin sunduğu bilgileri esas alan objektif teknikler kullanılmaktadır. Sıklıkla tercih edilen bir yaklaşım ise bu iki

yöntemin entegrasyonuna dayanan hibrit modellerdir. Alternatif sıralama yöntemleri ise belirlenen kriter ağırlıkları doğrultusunda her bir seçeneğin performansını analiz ederek karar vericiye rasyonel bir sıralama sunar. Kendi matematiksel yapıları ve varsayımları çerçevesinde sıralamayı gerçekleştiren bu yöntemlerin, birlikte veya karşılaştırmalı olarak kullanılması, karar sürecinin şeffaflığını ve tutarlılığını artırmakta, böylece çok boyutlu problemler daha etkin bir biçimde yönetilebilmektedir (Keshavarz Ghorabae vd., 2015).

Favardin vd. (2002) çalışmalarında, üç alternatifli oylama durumlarında Borda kuralı ile Copeland yönteminin bireysel ve koalisyonel stratejik manipülasyona ne ölçüde açık olduğu analitik olarak incelemişlerdir. Yapılan karakterizasyon ve hesaplamalar sonucunda, Borda kuralının her iki türdeki stratejik oylamaya Copeland yöntemine kıyasla daha açık olduğu gösterilmiştir. Koalisyonel manipülasyonda ise, seçmen sayısı arttıkça Borda'nın manipülasyon oranı %50'nin üzerine çıkarken, Copeland yöntemi için bu oran %25 civarında kalmaktadır. Sonuçlar, Copeland yönteminin stratejik oylamaya karşı Borda kuralına kıyasla daha dirençli olduğunu ortaya koymaktadır.

Çakır (2017) çalışmasında, çok sayıda karar verici ile kriter ağırlıklandırmasında yaşanan tutarsızlık problemini aşmak amacıyla SWARA – Copeland yöntemi geliştirilmiştir. Bu yöntemde, karar vericilerin kriter sıralamaları Copeland yöntemi ile tek bir sıralamaya dönüştürülmekte, ardından SWARA yönteminin adımlarıyla kriter ağırlıkları hesaplanmaktadır. Aydın Nazilli'de bir dişli üretim işletmesinde uygulanan çalışmada, CNC makinesi seçiminde yedi kriter değerlendirilmiş ve sonuçlar klasik SWARA yöntemi ile karşılaştırılmıştır. SWARA – Copeland yöntemi karar verici etkisini azaltması ve zamandan tasarruf sağlaması açısından literatüre katkı sağlayıcı bir model olarak sunulmuştur.

Yakut (2020) çalışmasında, OECD ülkelerinin 2017-2019 dönemine ait bilgi ve iletişim teknolojileri gelişmişlik düzeylerini değerlendirmek amacıyla Entropi yöntemiyle kriter ağırlıkları belirlenmiş, MOORA ve WASPAS yöntemleriyle ülke sıralamaları yapılmış ve bu sıralamalar Copeland yöntemiyle karşılaştırılarak bütünleşik bir sıralama elde edilmiştir. Çalışmada 17 farklı bilgi ve iletişim teknolojisi göstergesi üç ana başlık altında

incelenmiş. Bulgular, İsveç, Norveç ve İngiltere'nin en yüksek bilgi ve iletişim teknolojileri gelişmişliğine; Romanya, Bulgaristan ve Hırvatistan'ın ise en düşük düzeye sahip olduğunu göstermektedir. MOORA ve WASPAS yöntemlerinin sonuçlarının Copeland yöntemiyle yüksek düzeyde uyum gösterdiği, sadece birkaç ülkenin sıralamasında küçük farklılıkların olduğu tespit edilmiştir.

Çiftaslan ve Rençber (2022) çalışmalarında, Türkiye'de sistemik öneme sahip sekiz bankanın 2010-2020 yılları arasındaki finansal performanslarını CAMELS analizine dayalı olarak değerlendirmişlerdir. Kriter ağırlıklandırması için Entropi tabanlı IDOCRIW yöntemi, performans sıralaması için ise CoCoSo yöntemi kullanılmıştır. Analiz sonucunda, Ziraat Bankası, İş Bankası ve Garanti Bankası'nın performanslarında artış eğilimi gözlemlenirken, Halkbank'ın performansında düşüş eğilimi tespit edilmiştir. Çalışma, sistemik bankaların uzun dönemli değerlendirilmesine olanak tanınması ve bütünsel ÇKKV yöntemlerinin birlikte kullanılması açısından literatüre katkı sağlamaktadır.

Ayan vd. (2023) çalışmalarında, ÇKKV problemlerinde ağırlık belirleme yöntemlerinin kritik rolünü vurgulayarak CILOS, IDOCRIW, FUCOM, LBWA, SAPEVO-M ve MEREC gibi altı yeni yöntemi kapsamlı bir literatür taramasıyla incelemişlerdir. Her bir yöntemin özellikleri, adımları, uygulama alanları, bulanık yaklaşımlar, hibrit kullanımları ve yazılım destekleri detaylı olarak analiz edilmiştir. FUCOM ve MEREC yöntemlerinin son yıllarda öne çıktığı; SAPEVO-M'nin ise daha sınırlı ülkelerde uygulandığı ve LBWA ile FUCOM için henüz yazılım aracı geliştirilmediği belirtilmiştir. Çalışma, karar vericilere uygun yöntem seçimi konusunda rehberlik sağlamak ve özellikle objektif ve subjektif yöntemlerin birlikte kullanımının güncel bir eğilim olduğunu ortaya koymaktadır.

Çilek (2023) çalışmasında, 2017-2020 döneminde Türkiye'de Hazine ve Maliye Bakanlığı tarafından belirlenen piyasa yapıcı 11 bankanın finansal performanslarını değerlendirmek amacıyla CILOS, MARCOS ve Copeland yöntemlerini bir arada kullanan bütünsel bir model önerilmiştir. Öncelikle CILOS yöntemi ile sekiz finansal değerlendirme kriterinin ağırlıkları belirlenmiş, ardından MARCOS yöntemi ile her yıl

için performans sıralamaları yapılmış ve son aşamada ise bu sıralamalar Copeland yöntemiyle uzlaşık bir sıralamaya dönüştürülmüştür. Analiz sonucunda en yüksek performansa sahip bankalar sırasıyla Akbank, Garanti ve HSBC olurken; Yapı Kredi, VakıfBank ve Halkbank en düşük performansa sahip bankalar olarak belirlenmiştir. Ayrıca MARCOS ve Copeland yöntemleriyle elde edilen sıralamalar arasında yüksek düzeyde pozitif korelasyon saptanmıştır.

Işıldak vd. (2023) çalışmalarında, Tekirdağ Çorlu Atatürk Havalimanı'nın 2017-2021 yılları arasındaki performansını değerlendirmek amacıyla IDOCRIW ve WEDBA yöntemlerini birlikte kullanmışlardır. Kriter ağırlıklarının belirlenmesinde IDOCRIW, yıllara göre performans analizinde ise WEDBA yöntemi tercih edilmiştir. Çalışmada toplam uçak, yolcu ve yük sayısı olmak üzere üç kriter esas alınmış, en önemli kriterin toplam yolcu sayısı olduğu tespit edilmiştir. WEDBA yöntemiyle yapılan değerlendirmede, 2018 yılı en yüksek performansa sahip yıl olurken, 2021 yılı en düşük performans sergileyen yıl olarak belirlenmiştir. Covid-19 pandemisinin 2020 ve 2021 yıllarındaki performans düşüşlerinde etkili olduğu vurgulanmıştır. Bu çalışmanın IDOCRIW ve WEDBA yöntemlerinin birlikte kullanıldığı ilk havalimanı performans değerlendirmesi olması bakımından literatüre katkı sağladığı belirtilmiştir.

Literatürde yer alan farklı uygulama alanlarına sahip doğrusal olmayan regresyon model parametrelerinin nokta tahminine yönelik çalışmalar detaylı bir biçimde incelenmiş ve mevcut yöntemlerin belirli sınırlamaları olduğu sonucuna ulaşılmıştır. Bu tez çalışmasında, gerçek yaşamdan ilham alınarak oluşturulan on iki farklı AI optimizasyon algoritması kullanılarak yalnızca tek bir algoritmanın değerlendirilmesinin ötesinde evrimsel süreçler, müziksel yapılar ve sürü davranışlarından esinlenen yaklaşımların entegrasyonu ile geniş bir perspektif sunulması hedeflenmektedir. Buna göre AI optimizasyon algoritmalarının başlangıç değerlerinden bağımsız olarak geniş arama uzaylarını keşfetme kapasitesi, doğruluk, hesaplama maliyeti ve hız gibi performans kriterleri açısından algoritmaların etkinliğinin değerlendirilmesinde kritik bir rol oynaması beklenmektedir. Ayrıca farklı veri setleri ve problem türleri üzerinde gerçekleştirilecek detaylı performans analizleri sonucunda, her bir algoritmanın hangi problem tiplerinde üstün sonuçlar verdiği ortaya konulacaktır. Bu algoritmaların

performansları ÇKKV yöntemleri ile analiz edilecektir. Böylece, tek boyutlu performans değerlendirmelerinin ötesine geçilerek, çok boyutlu analizler üzerinden algoritmaların güçlü ve zayıf yönleri ortaya konulacaktır. Önerilen yaklaşım, yalnızca en iyi performans gösteren algoritmanın belirlenmesine odaklanmayıp, aynı zamanda çeşitli problem türleri için en uygun algoritmaların seçilmesinde rehberlik edecek bilgileri sağlamayı amaçlamaktadır. Tez çalışmasının ikinci bölümünde doğrusal olmayan regresyon modellerine ve model örneklerine yer verilecektir. Çalışmanın üçüncü bölümünde AI optimizasyon algoritmaları ve çalışma süreçleri hakkında detaylı bilgilendirme yapılacaktır. Dördüncü bölümde, ÇKKV yöntemleri ile kriter ağırlıklandırma yöntemleri ve sıralama yaklaşımları teorik çerçevede incelenecektir. Beşinci bölümde ise doğrusal olmayan regresyon modelleri üzerinde AI optimizasyon algoritmalarının parametre tahmini ve ÇKKV uygulaması yapılacaktır.

2. DOĞRUSAL OLMAYAN REGRESYON ANALİZİ

Doğrusal olmayan regresyon analizi, bir veri setindeki bağımlı değişken ile bir veya birden fazla bağımsız değişken arasındaki ilişkinin doğrusal olmaması durumunda uygulanan istatistiksel modelleme yöntemidir. Bu yaklaşımla gerçek dünyada sıklıkla karşılaşılan ve doğrusallık varsayımının geçerli olmadığı pek çok olgunun modellenmesi mümkün olmaktadır.

2.1 Doğrusal Olmayan Regresyon Modeli

Bir bağımlı değişken, p tane bağımsız değişken ve n gözlemlili bir veri seti için oluşturulabilecek doğrusal olmayan regresyon modeli

$$Y_i = f(\mathbf{X}, \boldsymbol{\beta}) + \varepsilon_i \quad i = 1, 2, \dots, n \quad (2.1)$$

biçiminde tanımlanır. Burada

Y_i : Bağımlı değişken

$\mathbf{X}$: Bağımsız değişkenlerden oluşan vektör $\mathbf{X} = [X_1 \ X_2 \ \dots \ X_p]$

$\boldsymbol{\beta}$: Tahmin edilecek parametreler vektörü $\boldsymbol{\beta} = [\beta_1 \ \beta_2 \ \dots \ \beta_p]^T$

ε_i : Hata terimi

dır. f fonksiyonu model parametrelerini doğrusal olmayan bir fonksiyona göre ifade eder. f 'nin biçimsel olarak bilindiği ve parametre vektörlerine göre en az iki defa türevinin alındığı varsayılır (Türkşen, 2023).

Doğrusal olmayan regresyon modellerinde rastgele değişken olarak kabul edilen hatalar sıfır ortalamalı, sabit varyanslı ve ilişkisiz olduğu kabul edilir. Matematiksel olarak bu varsayımlar sırasıyla

- (i) $E(\varepsilon_i) = 0$
- (ii) $Cov(\varepsilon_i) = \sigma^2 I$
- (iii) $\varepsilon_i \sim (0, \sigma^2)$

biçiminde ifade edilir, $i = 1, 2, \dots, n$.

Modelin analitik olarak çözülebilmesi veya en azından parametrelerinin nokta tahmininin güvenilir bir biçimde yapılabilmesi için, f fonksiyonunun model parametreleri bakımından en az ikinci türevin alınabilmesi optimizasyon algoritmalarının yakınsama hızını ve doğruluğunu artırmaktadır. Burada amaç, model parametrelerinin $\beta_1, \beta_2, \dots, \beta_p$ nokta tahminlerini optimal biçimde elde etmektir.

2.2 Model Örnekleri ve Uygulama Alanları

Doğrusal olmayan regresyon modelleri, gerçek dünya problemlerinin modellemesinde oldukça geniş bir uygulama alanına sahiptir. Örneğin, biyoloji alanında bir canlının büyüme eğrisini yalnızca doğrusal yaklaşımlarla açıklamak çoğu zaman mümkün değildir. Benzer biçimde kimyada, reaksiyon kinetiği veya enzim hızlarına ilişkin süreçlerde doğrusal modellerin yetersiz kalabildiği durumlarda, doğrusal olmayan modellere başvurmak zorunlu hale gelir. Zorlu dinamik sistemlerin anlaşılması için doğrusal olmayan regresyon, daha doğru ve gerçeğe yakın tahminler sunarak, anlamlı bilgiler sağlamaktadır.

Farklı disiplinlerden örneklerin sunulduğu sıklıkla kullanılan model örnekleri Çizelge 2.1’de yer verilmiştir.

Çizelge 2.1 Sıklıkla kullanılan doğrusal olmayan regresyon model örnekleri

<i>Model</i>	<i>Parametre Sayısı</i>	<i>Fonksiyon</i>	<i>Model Parametreleri</i>
Üstel	2	$f(x, \boldsymbol{\beta}) = \beta_1 e^{\beta_2 x}$	$[\beta_1 \ \beta_2]^T$
Logaritmik	2	$f(x, \boldsymbol{\beta}) = \beta_1 + \beta_2 \ln(x)$	$[\beta_1 \ \beta_2]^T$
Gauss Dağılım	2	$f(x, \boldsymbol{\beta}) = \beta_1 e^{-\frac{(x-\beta_2)^2}{2\beta_2^2}}$	$[\beta_1 \ \beta_2]^T$
Michaelis-Menten	2	$f(x, \boldsymbol{\beta}) = \frac{\beta_1 x}{\beta_2 + x}$	$[\beta_1 \ \beta_2]^T$
Gompertz	3	$f(x, \boldsymbol{\beta}) = \beta_1 e^{-\beta_2 e^{-\beta_3 x}}$	$[\beta_1 \ \beta_2 \ \beta_3]^T$
Kompartman	4	$f(x, \boldsymbol{\beta}) = \beta_1 e^{-\beta_2 t} + \beta_3 e^{-\beta_4 t}$	$[\beta_1 \ \beta_2 \ \beta_3 \ \beta_4]^T$

Çizelge 2.1’de görülen üstel model, büyüme analizlerinde, canlı organizmaların büyüme eğrileri veya bitkilerde verim tahminleri gibi zaman içinde değişen süreçlerin modellenmesinde kullanılır. Enzim kinetiği ve farmakokinetik çalışmalarda ise Michaelis-Menten gibi fonksiyonlar aracılığıyla ilaç etkileşimleri ve enzim-substrat ilişkilerindeki doğrusal olmayan yapılar yansıtılır. Nüfus projeksiyonlarında, Gompertz tipi modeller kullanılarak belirli bir popülasyonun zaman içindeki değişimleri tahmin edilmeye çalışılır. Ekonomik modellemelerde, piyasa talebindeki doygunluk noktalarının belirlenmesi veya teknoloji benimseme eğrilerinde sıklıkla doğrusal olmayan yapılar gözlemlenir. Mühendislik uygulamalarında ise imalat sürecinde malzemelerin dayanıklılığı, korozyon davranışları veya sistem dinamikleri çoğunlukla doğrusal olmayan eşitliklerle ifade edilmektedir. Çizelge 2.2’de, doğrusal olmayan regresyon modellerinin uygulama alanları, literatür çalışmaları ile birlikte sunulmuştur.

Bu farklı çalışma alanlarındaki doğrusal olmayan regresyon modellerinin kullanımı, esneklik ve gerçek süreçleri yansıtabilme kapasitesinin uygulama alanında önemli olduğunu göstermektedir.

Çizelge 2.2 Doğrusal olmayan regresyon modellerinin uygulama alanları

Model	Kullanıldığı Alanlar	Yazarlar
Üstel	Nüfus Projeksiyonu Ekonomik Büyüme Analizleri Kimyasal Reaksiyon Hızları	Anggreini vd. (2023) Dym (2004)
Logaritmik	Jeoloji ve Yer Bilimleri Ekonomik Modeller Biyolojik Büyüme Analizleri	Grossman-Ponemon vd. (2020) Ray (2023) Paltanea vd. (2010)
Gauss Dağılımı	Sinyal İşleme Fiziksel Ölçümler İstatistiksel Dağılım Analizleri	Dudek ve Baranowski (2022) Chen vd. (2023)
Michaelis-Menten	Enzim Kinetiği Biyokimyasal Reaksiyonlar Farmakokinetik	Murugan (2024) Saganuwan (2021)
Kompartman	Epidemiyoloji Farmakokinetik Çevresel Bilimler	Gupta vd. (2023) Türkşen ve Akgün (2018) Miyoshi vd. (1996)
Gompertz	Tümör Büyüme Modelleri Nüfus Büyümesi Teknoloji Benimseme Eğrileri	Jabbar ve Al-saedi. (2022) Welagedara vd. (2019) Lartey (2020)

2.3 Model Parametrelerinin Nokta Tahmini

Doğrusal olmayan regresyonda parametre tahmini, genellikle hataların dağılımı üzerinde varsayıma ihtiyaç duymayan en küçük kareler (EKK) yöntemi ile yapılmaktadır. Bu çalışmada amaç fonksiyonu

$$\min \Phi(\boldsymbol{\beta}) = \sum_{i=1}^n [Y_i - f(\mathbf{X}, \boldsymbol{\beta})]^2 \quad (2.2)$$

biçiminde ele alınmıştır. Bu fonksiyonun minimize edilmesi, modelin veriye en iyi uyumu sağlaması için kritik öneme sahiptir (Khuri ve Cornell, 1996). Parametre vektörünün EKK tahminlerini elde edebilmek için, öncelikle yukarıdaki eşitlik kapsamında tanımlanmış olan amaç fonksiyonunun minimum noktasında, söz konusu parametrelerin her birine göre birinci dereceden kısmi türevlerin sıfır olması ve ikinci dereceden kısmi türevlerden oluşan Hessian matrisinin pozitif tanımlı bulunması gerekir. Bu koşullar altında, sürekli ve türevlenebilir olduğu varsayılan amaç fonksiyonunun model parametreleri üzerindeki türevleri alınarak

$$\frac{\partial \phi}{\partial \beta_j} = \sum_{i=1}^n [Y_i - f(\mathbf{X}, \boldsymbol{\beta})] \left(\frac{\partial f(\mathbf{X}, \boldsymbol{\beta})}{\partial \beta_j} \right) = 0, \quad j = 1, 2, \dots, p \quad (2.3)$$

denklem sistemi oluşturulur. f fonksiyonunun parametrelere göre doğrusal olmayan bir formda olduğundan, eşitlik (2.3)'te tanımlanan denklem sisteminin çözümü kolayca sağlanamamaktadır. Denklem sisteminin çözümü ile elde edilecek olan parametrelerin nokta tahmini türevelere dayalı sayısal hesaplamalar içeren iteratif yöntemler aracılığıyla gerçekleştirilir. Bu iteratif yöntemler, literatür uygulamalarında sıklıkla kullanılan türevelere dayalı sayısal optimizasyon yöntemleri ve hesaplamalarına ilişkin açıklayıcı özet bilgiler Çizelge 2.3'te yer verilmiştir.

Çizelge 2.3 Doğrusal olmayan regresyon modellerinin çözümü için kullanılan türev dayalı sayısal optimizasyon yöntemleri

<i>Yöntemler</i>	<i>Açıklamalar</i>
Gauss-Newton	Hızlı yakınsama özelliklerine sahip olup, hata fonksiyonunun türevlerine dayalı olarak parametreleri günceller. Ancak başlangıç değerlerine duyarlıdır ve yerel minimum sorunları yaşayabilir.
Levenberg-Marquardt	Gauss-Newton ve En Dik İniş yönteminin özelliklerini birleştiren hibrit bir yaklaşım sunar. Parametre güncellemelerinde, adaptif bir adım büyüklüğü kullanarak, zorlu topolojilerde dahi yakınsamayı iyileştirir. Yerel minimumların çokluğunda bile kararlı sonuçlar verebilmektedir.
Gradient Descent	Hata fonksiyonunun gradyanını izleyerek parametreleri adım adım günceller; basit ve genel amaçlı bir yöntemdir, ancak yakınsama hızı diğer yöntemlere göre daha düşüktür.

Geleneksel yöntemler, matematiksel olarak sağlam temellere dayansa da doğrusal olmayan ve zorlu problemlerde bazı sınırlamalara sahiptir. Yerel minimumlara takılma, başlangıç değerlerine duyarlılık ve türev hesaplamalarının zorluğu gibi sorunlar, bu yöntemlerin etkinliğini azaltmaktadır.

Doğrusal olmayan regresyon modellerinde parametre tahmin sürecinin başarısı büyük ölçüde algoritmanın yakınsama davranışına ve global optimuma ulaşabilme yeteneğine bağlıdır. Optimizasyon sürecinde model parametrelerinin başlangıç değerleri belirleyici rol oynar ve kötü seçilmiş model parametre başlangıç değerleri algoritmayı global optimal yerine yerel minimuma hapsedebilir. Modelin fonksiyonel yapısı, türev alınabilme kolaylığı ve fonksiyonun sürekliliği de yakınsamayı etkilemektedir. Özellikle çok sayıda yerel nokta içeren fonksiyonlarda yerel minimum problemleri daha sık gözlenir. Veri setinin boyutu ve kalitesi arttıkça parametre tahminleri genellikle daha

istikrarlı hale gelse de yetersiz örneklem ya da gürültülü veriler, algoritma kullanarak minimum hatalı tahmin yapılmasını güçleştirir.

Yapay zeka temelli optimizasyon yaklaşımları, zorlu ve çok tepe/çukur içeren problemlerde yakınsamayı hızlandırmanın yanı sıra global çözümlere ulaşmada esnek ve güçlü bir çerçeve sunarlar. Bu yöntemler, fonksiyonun türev bilgisine bağımlı olmadan geniş bir arama uzayında global optimuma yönelmesi sayesinde yerel minimumlara takılma olasılığını azaltarak daha istikrarlı ve iyi parametre tahminleri elde etmeye olanak tanır. Ayrıca, parametrik olmayan ve farklı boyutlardaki veri setleriyle çalışabilme kabiliyetleri sayesinde, doğrusal olmayan regresyon modellerinin uygulanabilirliğini ve başarısını önemli ölçüde artırır.

2.4 Model Performans Metrikleri

Doğrusal olmayan regresyon modelinin başarımını değerlendirmek için çeşitli istatistiksel ölçütler kullanılır. Bu metrikler, tahmin edilen modellerin gerçek verilere ne kadar iyi uyum sağladığını ve algoritmaların parametre tahminindeki başarısını ortaya koyacaktır. Bu çalışmada, tahmin edilen modellerin gerçek verilere ne kadar iyi uyduğunu ve algoritmaların parametre tahminindeki başarısını ölçümlemek için en yaygın kullanılan metrikler, *RMSE* (Root Mean Squared Error - Kare Hatalar Ortalamasının Karekökü), *MAE* (Mean Absolute Error - Ortalama Mutlak Hata), *MAPE* (Mean Absolute Percentage Error - Ortalama Mutlak Yüzde Hatası), ve R^2 (Coefficient of Determination - Belirleme Katsayısı)'dir. Çizelge 2.4'te gösterilen bu metriklerin her biri hataları farklı bir perspektiften ele alarak modelin güçlü ve zayıf yönlerini anlamaya yardımcı olmaktadır.

Çizelge 2.4 Doğrusal olmayan regresyon modellerinde kullanılan performans metrikleri

<i>Performans Metrikleri</i>	<i>Formüller</i>
<i>RMSE</i>	$\sqrt{\frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2}$
<i>MAE</i>	$\frac{1}{n} \sum_{i=1}^n Y_i - \hat{Y}_i $
<i>MAPE</i>	$\left(\frac{100\%}{n}\right) \sum_{i=1}^n \left \frac{Y_i - \hat{Y}_i}{Y_i} \right $
<i>R²</i>	$1 - \frac{\sum_{i=1}^n (Y_i - \hat{Y}_i)^2}{\sum_{i=1}^n (Y_i - \bar{Y})^2}$

Model başarımını değerlendirmek amacıyla seçilen metrikler, analizin hedefleri ve veri özellikleri doğrultusunda belirlenmelidir. Doğrusal olmayan regresyon problemlerinde karşılaşılabilecek olası sorunları en aza indirmek için, model parametrelerinin çoklu başlangıç değerleriyle denemeler yapmak, veri ön işleme süreçlerine özen göstermek ve türevden bağımsız optimizasyon yöntemlerini içeren stratejiler uygulamak etkili bir yaklaşımdır.

3. YAPAY ZEKA OPTİMİZASYON ALGORİTMALARI

Günümüzde yapay zeka, insan benzeri zekaya sahip sistemlerin geliştirilmesini hedefleyen ve bilimsel araştırmalardan endüstriyel uygulamalara kadar geniş bir yelpazede çığır açıcı gelişmelere ortam hazırlayan disiplinler arası bir alandır. Yapay zeka yöntemleri sayesinde zorlu problemlerin çözümünde insan müdahalesine ihtiyaç duymadan öğrenme, tahmin ve karar verme mekanizmaları geliştirilebilmektedir. Optimizasyon, belirli bir hedef doğrultusunda sınırlı kaynakların en verimli biçimde kullanılması sürecini ifade etmekte olup yapay zeka ile optimizasyonun kesişimi ise fen, sağlık, sosyal, ekonomi ve mühendislik alanlarındaki bilimsel sorunların çözümü için kritik önem taşımaktadır. Bu kapsamda ele alınan yapay zeka tabanlı optimizasyon yöntemleri, doğadan ve çeşitli doğal süreçlerden ilham alarak farklı problem senaryolarında güçlü ve esnek çözümler üretilmesini amaçlar. Bu yöntemlerin temelinde, doğadaki canlıların veya süreçlerin keşif (exploration) ve sömürü (exploitation) dengesi gözetilerek taklit edilmesi yer almaktadır. Böylece, problemi tanımlamak için oluşturulan matematiksel modellere küresel ölçekte etkili çözümler sunulabilmektedir.

AI ile optimizasyonun bulunduğu noktada doğadan ve çeşitli süreçlerden ilham alan pek çok yeni yaklaşım ortaya çıkmaktadır. Bu yaklaşımlar, evrimsel tabanlı (evolutionary inspired), sürü temelli (swarm inspired), insan davranışından esinlenen (human inspired), müzik esinli (music inspired), fizik tabanlı (physics inspired), biyoloji tabanlı (biology inspired) ve matematiksel tabanlı (mathematical inspired) gibi farklı alanlardan esinlenen birçok algoritmayı kapsamaktadır. Tez çalışmasının bu bölümünde, AI ve optimizasyon ilişkisi detaylı bir biçimde irdelenerek çeşitli alanlarda kullanılan ve her biri özgün ilkelere dayanan AI tabanlı optimizasyon algoritmaları ele alınmıştır. Bu algoritmalar, tez çalışması kapsamında AI optimizasyon algoritmaları olarak adlandırılmıştır.

AI optimizasyon algoritmaları, mühendislik tasarımı, veri madenciliği, enerji sistemleri, rota planlama, makine öğrenmesi ve daha pek çok alanda etkin bir biçimde kullanılmaktadır. Artan hesaplama gücü ve AI teknolojilerinin gelişmesi, bu algoritmaların daha zorlu problemlere uygulanmasına zemin hazırlarken, aynı zamanda farklı yöntemlerin bir araya gelmesiyle hibrit çözümler de ortaya çıkmaktadır. Evrimsel

tabanlı, sürü temelli, müzik ilhamlı veya insan davranışı odaklı olsun, bu algoritmaların ortak noktası, keşif ve sömürü süreçlerini dengeli biçimde kullanarak zorlu problemlere yenilikçi ve etkili çözümler sunmalarıdır.

AI tabanlı optimizasyon algoritmalarının sunduğu hız, esneklik ve küresel çözüm elde etme becerisi, çağımızın zorlu problem senaryolarına kapsamlı ve yenilikçi yaklaşımlar getirmektedir. Bu tez çalışmasında ele alınan AI optimizasyon algoritmaları, ilham alınan yaklaşım başlıklarına göre incelenerek hem teorik hem de uygulamaları bakımından incelenerek kapsamlı bir bakış açısı sunulması hedeflenmektedir.

3.1 Evrimsel Tabanlı Yöntemler

Evrimsel tabanlı yöntemler arasında Genetik Algoritma (Genetic Algorithm – GA), Diferansiyel Evrim Algoritması (Differential Evolution Algorithm – DE) ve Çiçek Tozlaşması Algoritması (Flower Pollination Algorithm – FPA) özellikle dikkat çekmektedir. Söz konusu algoritmalar, doğal seçilim ve popülasyon temelli arama mekanizmalarını kullanarak ilerleyen nesillerde daha iyi çözümlere doğru evrimleşir.

3.1.1 Genetik Algoritma

Genetik Algoritma (Genetic Algorithm - GA), John H. Holland'ın “Genetic Algorithms” adlı makalesine göre, doğadaki doğal seçilim ve üreme mekanizmalarından esinlenilerek geliştirilmiştir. Canlı organizmaların çevresel koşullara adaptasyonu, doğal seçilim ve genetik kombinasyonlarla gerçekleşir; bu süreçlerde en iyi uyum sağlayan bireyler hayatta kalırken, genetik materyalleri sonraki nesillere aktarılır. Holland, bu evrimsel sürecin problem çözme yeteneğini taklit etme amacıyla genetik algoritmaları geliştirmiştir. Genetik algoritmalar, popülasyonlar arasında çaprazlama ve mutasyon operatörlerini kullanarak potansiyel çözümleri yeniden şekillendirir ve her adımda en iyi performansı gösteren çözümleri seçer. Bu süreç, optimizasyon problemlerini çözmek için evrimsel bir yaklaşım sunar. Doğal seçilimin “en iyinin hayatta kalması” prensibi gibi,

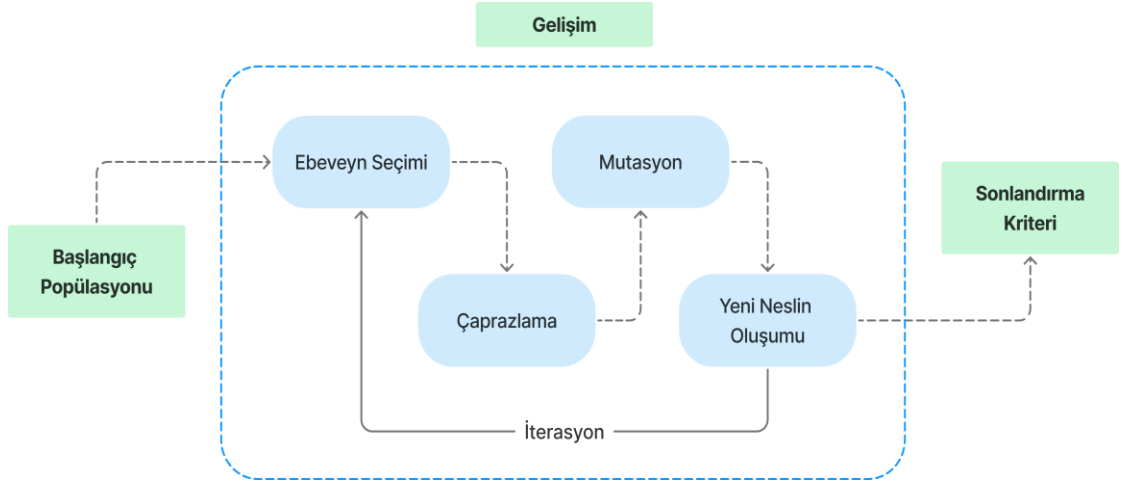
genetik algoritmalar da en uygun çözüm stratejilerini belirleyip çoğaltarak sürekli gelişim sağlar.

Holland (1992), GA algoritması ile biyolojik evrim ve doğal seçim süreçlerinden esinlenerek problemlerin çözümü için nasıl kullanılabileceğini incelemiştir. GA'lar, popülasyonlar içinde rastgele çözüm dizilerini değerlendirir, yüksek performanslı dizileri seçer ve çaprazlama ile yeni kombinasyonlar üretir. Çalışma, GA'ların yerel çözümlerden kaçma ve geniş bir çözüm uzayını etkili bir biçimde tarama kabiliyeti ile diğer yöntemlere kıyasla daha avantajlı olduğunu göstermiştir. Ayrıca, jet motoru tasarımı gibi mühendislik problemlerinde başarıyla uygulanarak daha hızlı ve optimize çözümler elde edilmesini sağlamıştır.

Kalinina vd. (2023), GA kullanarak bir lojistik fonksiyon olan Richards fonksiyonunun parametrelerini tahmin etmişlerdir. Çalışmada, *trees* adlı veri seti kullanılarak ağaç yaşının odun hacmi üzerindeki etkisi modellenmiştir. GA'nın rastgele seçim, mutasyon ve tek noktalı çaprazlama operatörleri uygulanmıştır. Sonuçlar, fonksiyon parametrelerinin yüksek doğrulukla tahmin edildiğini ve R^2 değerinin %99,96 gibi yüksek bir seviyeye ulaştığını göstermiştir. Bu yöntem, doğrusal olmayan regresyon problemlerinde genetik algoritmanın etkinliğini ortaya koymuştur.

GA, başlangıçta rastgele bir popülasyon oluşturarak başlar ve bu popülasyonu değerlendirme, ebeveyn seçimi, çaprazlama ve mutasyon gibi operatörler sürekli olarak geliştirir. Bu adımlar, yineleme sayısı parametresi ile belirlenen döngü sayısına ulaşana kadar devam eder. Kodda verilen farklı seçim, çaprazlama ve mutasyon operatörleri, algoritmanın çeşitli optimizasyon problemlerine uygun olarak adapte edilmesini sağlar. Bu adaptasyon, algoritmanın daha hızlı ve daha doğru bir biçimde global en iyi çözüme ulaşmasına yardımcı olabilir.

Şekil 3.1'de GA'nın işleyiş diyagramı görülmektedir. Buna göre GA'nın başlangıç popülasyonu oluşumu, gelişim süreci ve sonlandırma kriteri adımlarına ilişkin açıklamalara aşağıda yer verilmiştir.



Şekil 3.1 GA'nın işleyiş diyagramı

Adım 1: Başlangıç Popülasyonu

GA, çözüm uzayında rastgele çözümlerden oluşan bir popülasyon ile başlatılır. Her çözüm, popülasyondaki bir bireyi temsil eder. Bu bireyler, algoritmanın üzerinde çalıştığı problem için birer olası çözümden oluşur.

Adım 2: Gelişim

Başlangıç popülasyonundaki her birey bir uygunluk (fitness) değeri ile değerlendirilir. Bu uygunluk değeri, bireyin problemin çözümüne olan katkısını ifade eder.

- **Ebeveyn Seçimi**

Ebeveynler, çaprazlama işlemi ile yeni bireylerin oluşturulması için seçilir. Farklı seçim stratejileri kullanılır. Rulet Çarkı (Roulette), daha yüksek uygunluk değerine sahip bireylerin seçilme olasılığı daha yüksektir. Turnuva (Tournament), rastgele seçilen bir alt kümeden en uygun bireyler seçilir. k_way parametresi, turnuvadaki alt kümenin büyüklüğünü belirler. Rastgele (Random), popülasyondan tamamen rastgele birey seçilir. Sıralı (Rank), popülasyondan uygunluk değeri sıralamasına göre birey seçilir.

- **Çaprazlama**

İki ebeveyn arasındaki genetik bilgilerin karıştırılmasıyla yeni çocuk bireyler oluşturulur. Kullanılabilecek farklı çaprazlama yöntemleri şunlardır: Tek Nokta (One

Point) Çaprazlama, ebeveynlerin genetik bilgileri, belirli bir noktada kesilerek karıştırılır. Çok Noktalı (Multi Points) Çaprazlama, birden fazla noktada kesim yapılır ve ebeveynlerin genetik bilgileri bu noktalarda karıştırılır. Uniform Çaprazlama, ebeveynlerden her gen için rastgele seçim yapılarak çocuk oluşturulur. Aritmetik (Arithmetic) Çaprazlama, ebeveynlerin genetik bilgileri arasındaki aritmetik bir ortalama alınarak çocuklar oluşturulur.

- **Mutasyon**

Çocuk bireylerin bazı genlerinde rastgele değişiklikler yapılır. Bu işlem, popülasyonun çeşitliğini artırmak ve yerel optimumlardan kaçınmak amacıyla kullanılır. Belirli bir olasılıkla her bireyin genlerinde değişiklik yapılır. Kullanılabilecek farklı mutasyon türleri şunlardır. Flip, rastgele bir genin değeri değiştirilir. Swap, iki genin yerleri değiştirilir. Scramble, genlerin bir kısmı rastgele karıştırılır. Inversion, belirli bir aralıktaki genler ters çevrilir.

- **Yeni Neslin Oluşumu**

Çaprazlama ve mutasyon operatörlerinden sonra oluşturulan yeni çocuk bireyler, mevcut popülasyondaki bireylerle kıyaslanır. Daha uygun olan bireyler yeni popülasyona alınır.

Adım 3: Sonlandırma Kriteri

Algoritma belirli bir sayıda döngü boyunca tekrarlanır. Her döngüde yeni nesiller oluşturulmaya devam edilir.

GA algoritması işletilirken gelişim adımı söz edilen ayarlanabilir parametreler; popülasyon sayısı (pop_size), iterasyon sayısı (epoch), çaprazlama olasılığı (pc) ve mutasyon olasılığı (pm) olarak tanımlanmaktadır.

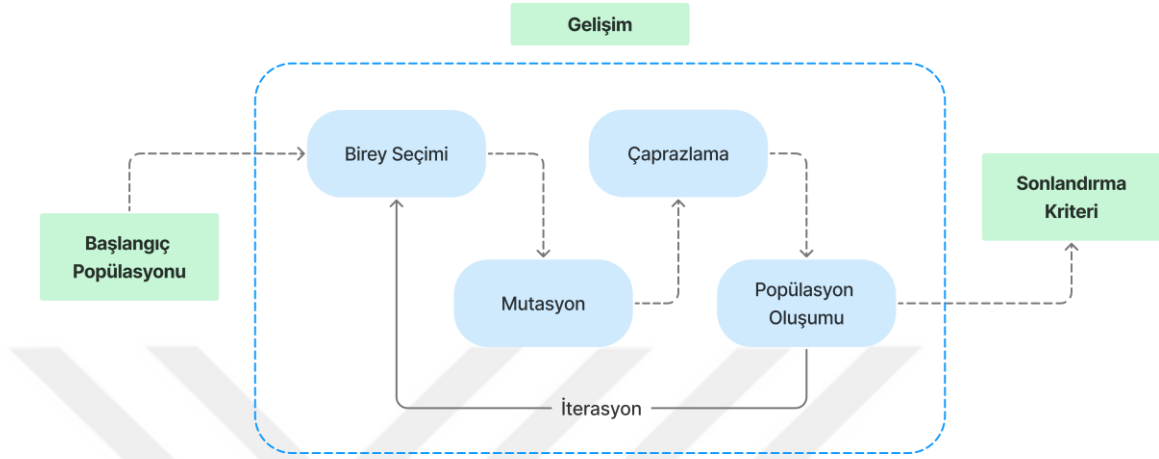
3.1.2 Diferansiyel Evrim Algoritması

Diferansiyel Evrim Algoritması (Differential Evolution Algorithm - DE), doğada bulunan evrimsel stratejilerden esinlenerek geliştirilmiştir. Storn ve Price (1997) tarafından oluşturulan DE algoritması, biyolojik evrimden gelen varyasyon ve seçim süreçlerini temel alır ve sürekli parametre optimizasyonunda güçlü bir araç olarak öne çıkar. DE algoritmasının temel prensibi, popülasyonda bulunan çözüm adaylarını rastgele bir biçimde seçerek bu adaylar arasında farklar yaratmak ve bu farkları kullanarak yeni çözümler üretmektir. Doğada olduğu gibi, popülasyon içinde çeşitlilik sağlayan bu tür farklılıklar sayesinde, algoritma yerel minimumlardan kurtulma ve daha iyi global çözümler bulma yeteneği kazanır. DE, özellikle büyük ve zorlu arama alanlarında çoklu yerel çözümlerin olduğu durumlarda, biyolojik evrimin sağladığı esneklik ve çeşitliliği taklit ederek optimum çözümlere ulaşmada etkin bir performans sergilerler.

Storn ve Price (1997), sürekli alanlar üzerinde küresel optimizasyon için DE algoritmasını önermiştir. DE, popülasyondaki bireyler arası farklılıkları kullanarak yeni çözümler üretir ve bu çözümleri seçim süreciyle iyileştirir. Çalışma, DE'nin az sayıda ayarlanabilir parametre gerektirmesi, yüksek başarı oranlarıyla küresel minimumlara ulaşması ve diğer popüler yöntemlere kıyasla daha hızlı yakınsama sağladığını göstermiştir. Algoritmanın paralel işleme uygun yapısı, kullanım kolaylığı ve sağlamlığı vurgulanmıştır.

DE algoritması, özellikle nümerik optimizasyon problemlerinde etkili bir arama stratejisi sunar. Algoritmanın farklı stratejileri ve parametreleri, çözüm uzayının daha geniş bir bölümünü keşfetmeyi ve daha hızlı bir biçimde küresel optimuma ulaşmayı sağlar. Algoritmada belirlenen popülasyon boyutunda rastgele bireylerden oluşan bir başlangıç popülasyonu oluştururlar. Her birey için popülasyondan farklı bireyler seçilerek mutasyon işlemi yapılır. Bu işlem, çözüm uzayında farklı bölgeleri keşfetmeyi sağlar. Çaprazlama operatörleri ile mevcut çözüm ve mutasyon sonucu elde edilen çözüm birleştirilir. Her bireyin yeni versiyonu, uygunluk değerine göre eski versiyonu ile karşılaştırılır ve daha uygun olan popülasyonda tutulur. Bu adımlar, epoch parametresi ile belirlenen maksimum iterasyon sayısına ulaşana kadar veya belirli bir durdurma kriteri sağlanana kadar tekrarlanır.

Şekil 3.2’de DE’nin işleyiş diyagramı görülmektedir. Buna göre DE’nin başlangıç popülasyonu oluşumu, gelişim süreci ve sonlandırma kriteri adımlarına ilişkin açıklamalara aşağıda yer verilmiştir.



Şekil 3.2 DE’nin işleyiş diyagramı

Adım 1: Başlangıç Popülasyonu

DE algoritması rastgele bir başlangıç popülasyonu oluşturur. Bu popülasyon, üzerinde çalışılan problemin çözüm uzayında yer alan rastgele bireylerden oluşur ve her birey bir olası çözümü temsil eder. Popülasyondaki her bir bireyin boyutu, problemdeki çözüm vektörünün boyutuyla aynıdır.

Adım 2: Gelişim

- **Birey Seçimi**

Başlangıç popülasyonundaki her birey bir uygunluk (fitness) değeri ile değerlendirilir. Bu uygunluk değeri, bireyin problemin çözümüne olan katkısını ifade eder.

- **Mutasyon**

DE’nin temel farkı, mutasyon sürecinde üç farklı çözümü kullanarak yeni bir çözüm oluşturmasıdır. Her birey için, popülasyondaki diğer bireylerden üçü rastgele seçilir. Bu üç bireyin kullanılmasıyla mevcut çözümün yeni bir mutasyon vektörü oluşturulur. Weighting Factor, mutasyon sırasında kullanılan ağırlık faktörünü belirler. Ağırlık faktörü, popülasyondan rastgele seçilen iki birey arasındaki farkın

rastgele çözüm vektörüne eklenmesinde kullanılır. Kullanılan stratejiye göre farklılık gösterebilmektedir.

- **Çaprazlama**

Mutasyon sonucu elde edilen yeni çözüm, mevcut çözüm ile karıştırılarak bir çaprazlama işlemi yapılır. Bu sayede çözümdeki bazı genler mevcut çözümden, bazıları ise mutasyon sonucundan gelir. Çaprazlama olasılığı, çaprazlama sırasında her bir genin mutasyon sonucu elde edilen çözümden mi yoksa mevcut çözümden mi alınacağını belirleyen bir olasılık değeridir. DE algoritmasında çeşitli stratejiler kullanılabilir ve bu stratejiler, bireylerin nasıl mutasyon ve çaprazlama yapılacağını belirler. Kullanılan kütüphanede, 6 farklı strateji tanımlanmıştır: Strategy 0 (DE/current-to-rand/1/bin): Mevcut çözümden başlayarak, rastgele seçilen üç birey kullanılarak mutasyon yapılır. Strategy 1 (DE/best/1/bin): En iyi çözümden başlayarak, popülasyondan rastgele iki birey kullanılarak mutasyon yapılır. Strategy 2 (DE/best/2/bin): En iyi çözümden başlayarak, dört rastgele birey kullanılarak mutasyon yapılır. Strategy 3 (DE/rand/2/bin): Rastgele seçilen iki bireyden başlayarak beş birey kullanılarak mutasyon yapılır. Strategy 4 (DE/current-to-best/1/bin): Mevcut çözümden başlayarak, en iyi çözüm ve rastgele iki birey kullanılarak mutasyon yapılır. Strategy 5 (DE/current-to-rand/1/bin): Mevcut çözümden başlayarak, üç rastgele birey kullanılarak mutasyon yapılır.

- **Popülasyon Oluşumu**

Çaprazlama ve mutasyon sonucunda elde edilen yeni bireyler, mevcut popülasyondaki bireylerle karşılaştırılır ve daha uygun olanlar popülasyonda kalır. Yeni bireyin uygunluk değeri eski bireyden daha iyiye, popülasyondaki birey güncellenir.

Adım 3: Sonlandırma Kriteri

Algoritma belirli bir sayıda döngü boyunca tekrarlanır. Her döngüde yeni nesiller oluşturulmaya devam edilir.

DE algoritması işletilirken gelişim adımı söz edilen ayarlanabilir parametreler; popülasyon sayısı (pop_size), iterasyon sayısı (epoch), weighting factor (wf), çaprazlama olasılığı (cr) ve strateji (strategy) olarak tanımlanmaktadır.

3.1.3 Çiçek Tozlaşması Algoritması

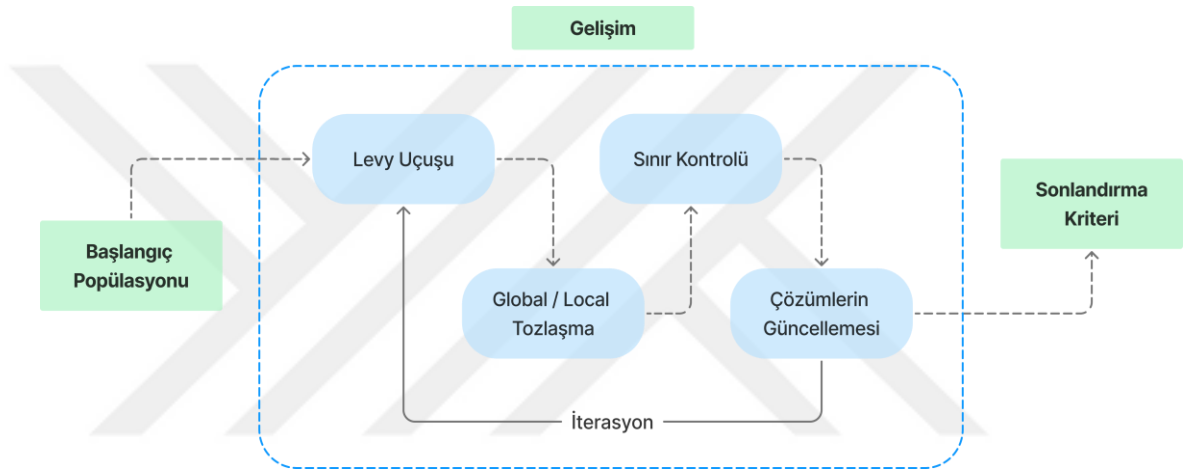
Çiçek Tozlaşması Algoritması (Flower Pollination Algorithm - FPA), çiçeklerin tozlaşma sürecinden esinlenilerek geliştirilmiş bir algoritmadır. Çiçek tozlaşması, bitkilerin hayatta kalması ve üremesi açısından önemli bir süreçtir ve bu süreçte biyolojik faktörler optimal tozlaşmayı sağlar. Çiçeklerin tozlaşması, biyotik ve abiyotik tozlaşma olmak üzere iki biçimde gerçekleşir. Biyotik tozlaşmada böcekler, kuşlar ve yarasalar gibi tozlaştırıcılar, polenleri taşıyarak genetik çeşitliliği destekler. Bu tür tozlaşma, polenlerin uzun mesafelere taşınmasıyla global bir süreç olarak düşünülebilir. Abiyotik tozlaşma ise rüzgar veya su gibi çevresel etkenlerle gerçekleşir ve daha yerel bir etkiye sahiptir. Yang (2012), çiçek tozlaşma sürecini taklit eden FPA algoritmasını önermiştir. Algoritma, biyotik ve abiyotik tozlaşma süreçlerini küresel ve yerel optimizasyon adımlarıyla modellemiştir. On test fonksiyonu üzerinde yapılan simülasyonlar, FPA'nın GA ve PSO'ya göre daha etkili olduğunu göstermiştir. Ayrıca, basınçlı kap tasarımı gibi mühendislik optimizasyon problemlerinde hızlı yakınsama ile üstün performans sergilediği vurgulanmıştır.

Widodo vd. (2022), FPA algoritması ve elitizm stratejisiyle zenginleştirilmiş bir türevini zaman alanı elektromanyetik verilerin ters çözümünde kullanmıştır. Çalışma, homojen ve çok katmanlı modellerle hem sentetik hem de saha verileri üzerinde algoritmaların performansını değerlendirmiştir. Sonuçlar, daha hızlı yakınsama sağladığını ve daha düşük hata oranlarıyla daha iyi ters çözüm sonuçları ürettiğini göstermiştir. Algoritmaların, jeofizik modelleme ve saha verilerinin değerlendirilmesinde uygulanabilir olduğu vurgulanmıştır.

FPA algoritması, bu doğa olaylarını taklit eden iki temel aşamaya dayanır: global tozlaşma ve yerel tozlaşma. Global tozlaşmada, polen taşıyıcılarının uzun mesafeler kat etmesi, Lévy uçuşu gibi matematiksel modellerle simüle edilir. Bu süreç, çözüm alanında

geniş çapta keşif sağlar. Yerel tozlaşma ise aynı tür çiçekler arasında gerçekleşir ve daha küçük ölçekli, yakın etkileşimleri içerir. Bu aşamalar, algoritmanın keşif ve sömürü yeteneklerini dengeli bir biçimde kullanarak hızlı ve verimli bir optimizasyon sağlamasına olanak tanır.

Şekil 3.3'te FPA'nın işleyiş diyagramı görülmektedir. Buna göre FPA'nın başlangıç popülasyonu oluşumu, gelişim süreci ve sonlandırma kriteri adımlarına ilişkin açıklamalara aşağıda yer verilmiştir.



Şekil 3.3 FPA'nın işleyiş diyagramı

Adım 1: Başlangıç Popülasyonu

Algoritma, belirli bir popülasyon büyüklüğüne bağlı olarak rastgele çözümlerden oluşan bir başlangıç popülasyonu oluşturur. Her birey, çözüm uzayında bir potansiyel çözümü temsil eder.

Adım 2: Gelişim

- **Levy uçuşu**

Elde edilen adım ile popülasyondaki çözümünden küresel en iyi çözüme doğru bir adım atarak yeni bir pozisyon oluşturulur. Girilen çarpan değeri bu uçuşun büyüklüğünü belirler. Eğer bir birey, lokal tozlaşma yöntemini kullanacaksa, mevcut çözüm komşu bireylerin çözümleri ile değiştirilerek yerel arama yapılır. Bu işlemde, mevcut

çözüm, popülasyondan rastgele seçilen iki birey arasındaki fark eklenir ve böylece lokal bir adım atılır. Bu yöntem, çözümün yerel bölgelerde iyileştirilmesini sağlar.

- ***Yerel ve Global Tozlaşma***

FPA’de, çiçeklerin tozlaşma süreci global tozlaşma ve lokal tozlaşma olmak üzere iki temel biçimde gerçekleşir. Algoritmanın her iterasyonunda, bireylerin hangi tozlaşma yöntemini kullanacağı “switch probability” parametresi ile belirlenir. Global tozlaşma, Levy uçuşu ile modellenir. Bu, çözümlerin geniş bir arama alanında arama yapmasını sağlar ve yerel optimumlardan kaçınmaya yardımcı olur.

- ***Sınır Kontrolü***

Oluşturulan yeni çözüm, problem sınırları dışında kalabilir. Bu durumda çözüm düzeltilir ve sınırlar içinde kalması sağlanır. Bu, algoritmanın çözümlerin geçerli bir çözüm aralığında kalmasını sağlamasına yardımcı olur.

- ***Çözümlerin Güncellenmesi***

Güncellenen pozisyonların uygunluk değerleri hesaplanır ve her birey için bu yeni pozisyonun eski pozisyonla karşılaştırılması yapılır. Eğer yeni pozisyon daha iyi bir çözüm sunuyorsa, birey bu yeni pozisyonla güncellenir. Yeni popülasyon, eski popülasyonla birleştirilir ve uygunluk değerine göre sıralanarak en iyi çözümler seçilir.

Adım 3: Sonlandırma Kriteri

Algoritma, epoch parametresi ile belirlenen iterasyon sayısına kadar devam eder. Her iterasyonda, popülasyonun her bireyi için global veya lokal tozlaşma uygulanarak yeni çözümler oluşturulur.

FPA algoritması işletilirken gelişim adımı söz edilen ayarlanabilir parametreler; popülasyon sayısı (pop_size), iterasyon sayısı (epoch), switch probability (p_s) ve Levy çarpanı (levy_multiplier) olarak tanımlanmaktadır.

3.2 Sürü Tabanlı Yöntemler

Sürü tabanlı yöntemler, doğadaki toplu veya koloni davranışlarını taklit ederek küresel optimuma yakın çözümler üretilmesini amaçlar. Yapay Arı Koloni Algoritması (Artificial Bee Colony – ABC), Yarasa Algoritması (Bat Algorithm – BA), Karınca Koloni Optimizasyonu (Ant Colony Optimization for Continuous – ACOR), Guguk Kuşu Algoritması (Cuckoo Search Algorithm – CSA), Gri Kurt Optimizasyonu (Grey Wolf Optimization – GWO), Parçacık Sürü Optimizasyonu (Particle Swarm Optimization – PSO) ve Jaya Algoritması (Jaya Algorithm – JA) dahil edilebilir. Arıların besin arayışı, karıncaların feromon izleri, yarasaların ekolokasyonu, guguk kuşlarının üreme stratejileri veya kurtların avlanma taktikleri gibi doğal süreçler, bu algoritmaların keşif ve sömürü dengelerini kurmasında temel rol oynar.

3.2.1 Yapay Arı Koloni Algoritması

Yapay Arı Koloni Algoritması (Artificial Bee Colony – ABC), bal arılarının koloni içindeki davranışlarından esinlenerek geliştirilmiştir. Bal arıları, yiyecek kaynaklarını bulma, sömürme ve yeni kaynaklar keşfetme süreçlerinde toplu bir biçimde çalışarak yüksek bir organizasyonel zeka sergilemektedir. ABC algoritması, bu doğal davranışları matematiksel olarak modelleyerek, optimizasyon problemlerine uygulanabilir bir çözüm yöntemi sunmaktadır.

Karaboğa (2005), ABC algoritmasını geliştirerek, arıların besin arama davranışını simüle eden bir optimizasyon yöntemi sunmuştur. Bu yöntem, çok modlu ve çok boyutlu sayısal optimizasyon problemlerinde kullanılmak üzere tasarlanmıştır. Çalışma, algoritmanın Sphere, Rosenbrock ve Rastrigin gibi test fonksiyonlarında uygulanabilirliğini göstermiş ve algoritmanın esnek, basit ve sağlam bir yapı sunduğunu ortaya koymuştur. ABC algoritmasının yerel minimumlardan çıkabilme yeteneğine sahip olduğu ve keşif ile yerel arama süreçlerini etkin bir biçimde dengeleyebildiği gösterilmiştir.

ABC algoritmasının başarısı, farklı alanlarda çeşitli geliştirmelere yol açmıştır. Tefek (2021), ABC algoritmasını yerel arama prosedürü ile genişleterek, fotovoltaik sistemlerinin parametre tahminine yönelik yeni bir ABC-yerel arama yöntemi önermiştir. Bu yöntemin, klasik ABC algoritmasına kıyasla yerel arama kabiliyetini artırdığı ve küresel keşif ile yerel sömürü dengesi sağladığı gösterilmiştir. ABC-yerel arama yöntemi, 15 test fonksiyonunda başarıyla test edilmiş ve diğer algoritmalarından üstün performans göstermiştir. Ayrıca, tek-diyot, çift-diyot ve Photowatt-PWP-201 fotovoltaik modüllerinde parametre tahminlerinde daha düşük *RMSE* değerleri ile diğer yöntemlerden daha başarılı sonuçlar elde etmiştir.

Kumar vd. (2023), uçakların boylamasına aerodinamik parametrelerini tahmin etmek için yapay sinir ağları ile ABC algoritmasını birleştirerek yeni bir sinirsel yapay arı kolonisi algoritması geliştirmiştir. Çalışma, bu yöntemin gürültü içeren simüle edilmiş ve gerçek uçuş verilerinde geleneksel yöntemlere kıyasla daha iyi performans gösterdiğini ortaya koymuştur. Maliyet fonksiyonlarını optimize etmek amacıyla kullanılan algoritma, yüksek doğruluk ve hızlı yakınsama sağlamıştır. ATTAS ve HANSA-3 uçaklarından elde edilen verilerle yapılan deneyler, algoritmanın ölçüm belirsizliği içeren durumlarda dahi etkili olduğunu göstermiştir.

Algoritmanın yapısı çalışan arılar, gözlemci arılar ve izci arılar olmak üzere üç gruba ayrılır. Çalışan arılar, mevcut yiyecek kaynaklarını ziyaret eder ve nektar miktarını ölçer. Gözlemci arılar, çalışan arıların paylaştığı bilgileri kullanarak en iyi kaynakları seçer. İzci arılar ise eskiyen veya terk edilen kaynakların yerine yenilerini bulmak için rastgele arama yapar. Bu süreç, optimizasyon problemlerinde keşif (exploration) ve sömürü (exploitation) arasındaki dengeyi sağlar. Bu süreç, algoritmanın global optimuma ulaşmak için çözüm uzayında etkili bir biçimde arama yapmasına olanak tanımaktadır. Ayrıca algoritma, popülasyondaki çeşitliliği koruyarak yerel optimumlardan kaçınmayı hedeflemektedir.

Bir yiyecek kaynağının pozisyonu, bir çözüm adayını temsil etmektedir ve nektar miktarı, bu çözümün uygunluğunu gösterir. Arılar, mevcut çözümü iyileştirmek için küçük değişiklikler yaparak veya tamamen yeni çözümler üreterek daha iyi sonuçlar elde etmeye

kullanılarak oluşturulur. Bu aşamada kullanılan ϕ , $[-1, 1]$ aralığında rastgele bir sayıdır ve yeni çözümün arama sürecindeki yönünü ve adım büyüklüğünü belirler. Eğer yeni çözüm, mevcut çözümden daha iyiye (yani uygunluk değeri daha yüksekse), mevcut çözüm yeni çözümle değiştirilir ve deneme sayısı sıfırlanır. Aksi takdirde, deneme sayısı artırılır.

- **Gözcü Arı Aşaması**

Gözcü arılar, çalışan arılar tarafından keşfedilen yiyecek kaynaklarına yönelir ve en iyi kaynakları seçmek için çalışan arılar arasında seçim yaparlar. Gözcü arılar, her bir çalışan arının uygunluk değerine dayalı olarak yiyecek kaynağını seçmek için rulet tekerleği seçimi kullanır.

- **Çözümlerin Güncellemesi**

Seçilen çalışan arıya yakın yeni bir çözüm oluşturulur ve bu çözüm, mevcut çözümle karşılaştırılarak iyileştirme yapılır. Eğer yeni çözüm daha iyiye, mevcut çözüm güncellenir ve deneme sayısı sıfırlanır.

- **İzci Arı Aşaması**

Eğer bir yiyecek kaynağı belirli bir süre boyunca (deneme sayısı, n_limits değerini aştığında) iyileştirilemezse, bu yiyecek kaynağı terk edilir ve yerine rastgele bir çözüm atanır. Bu süreç, izci arıların yeni yiyecek kaynakları aramasıyla temsil edilir. Bu aşama, algoritmanın çözüm uzayında sıkışıp kalmasını önlemek ve çeşitliliği artırmak için kullanılır. n_limits , bir yiyecek kaynağının terk edilmesi için gereken maksimum deneme sayısını belirler. Yüksek n_limits değeri, bir kaynağın daha uzun süre iyileştirilmeye çalışılmasını sağlar; düşük değer ise kaynakları daha hızlı terk eder ve çeşitliliği artırır.

Adım 3: Sonlandırma Kriteri

Algoritma, epoch parametresi ile belirlenen iterasyon sayısına kadar devam eder. Her iterasyonda çalışan arılar, gözcü arılar ve izci arılar çözüm uzayında yeni aramalar yapar. İstenilen iterasyon sayısına ulaşıldığında veya belirli bir durdurma kriteri sağlandığında algoritma sona erer ve en iyi çözüm döndürülür.

ABC algoritması işletilirken gelişim adımıda söz edilen ayarlanabilir parametreler; popülasyon sayısı (pop_size), iterasyon sayısı (epoch), besin kaynakları terk etme limiti (n_limits) olarak tanımlanmaktadır.

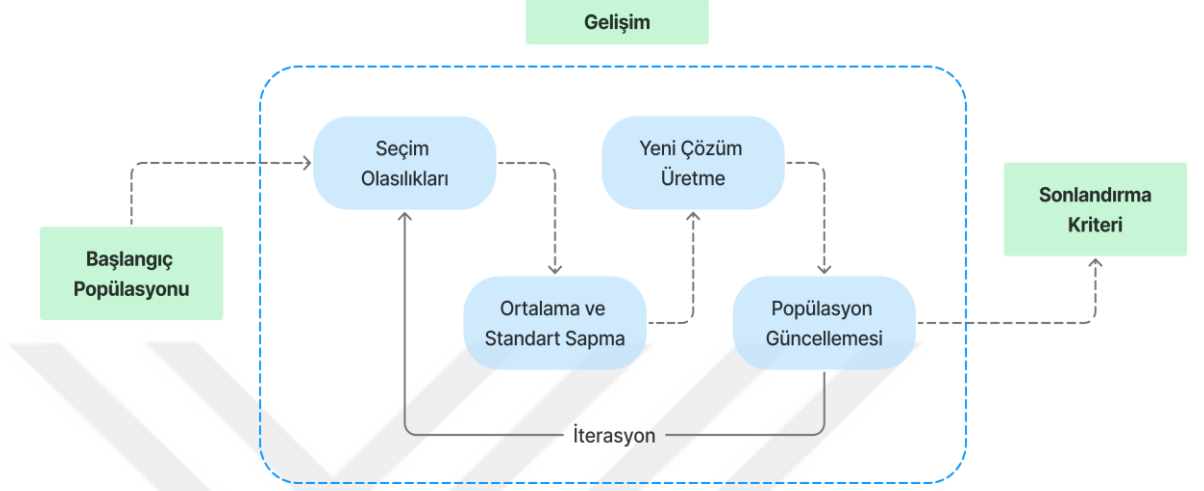
3.2.2 Karınca Koloni Optimizasyonu

Karınca Koloni Optimizasyonu (Ant Colony Optimization for Continuous - ACOR), karıncaların doğadaki yiyecek arama davranışlarından esinlenerek geliştirilmiştir. Geleneksel Ant Colony Optimization (ACO) algoritması, kombinatoriyal optimizasyon problemleri için tasarlanmış olmasına rağmen, ACOR algoritmasında bu yaklaşımı sürekli değişkenler üzerinde çalışan optimizasyon problemlerine uyarlanmıştır. Socha ve Dorigo (2008), ACO algoritmasının sürekli alanlara uyarlanmasını, yani ACOR algoritmasını tanıtmıştır. Çalışma, ACO'nun temel yapısını değiştirmeden sürekli optimizasyon problemlerinde nasıl etkili bir biçimde uygulanabileceğini göstermektedir. Sürekli değişkenlerle çalışmak için kesikli olasılık dağılımı yerine olasılık yoğunluk fonksiyonları kullanan ACOR, diğer yöntemlerle karşılaştırılmıştır. Deneyler, ACOR'un hem doğruluk hem de kararlılık açısından güçlü performans sergilediğini ve özellikle sürekli optimizasyon problemleri için etkili bir çözüm sağladığını ortaya koymuştur.

Doğadaki karıncalar, yiyecek kaynaklarına en kısa yolu bulmak için feromon izleri bırakır ve bu izler diğer karıncaları yönlendirir. Algoritma, bu davranışı sürekli alanlarda çözüm arama stratejisi olarak modellemektedir. Algoritmada, çözüm uzayı bir olasılık yoğunluk fonksiyonu ile temsil edilir. Bu olasılık yoğunluk fonksiyonu, Gaussian kernel fonksiyonları kullanılarak oluşturulur ve her adımda dinamik olarak güncellenir. Böylece, karıncalar bir önceki çözüm adımlarına bağlı olarak olasılık yoğunluk fonksiyonlarından örnekleme yapar ve en iyi çözüm bölgelerine odaklanır.

Algoritmanın bu tasarımı hem sürekli değişkenlerle çalışabilme hem de karıncaların doğal arama mekanizmalarına sadık kalma özelliğini taşır. Bu yapı, hem keşif (yeni bölgelerin araştırılması) hem de sömürü (bulunan iyi çözümlerin detaylandırılması) arasında etkili bir denge sağlar ve optimizasyon süreçlerinde güçlü bir performans sergiler.

Şekil 3.5'te ACOR'un işleyiş diyagramı görülmektedir. Buna göre ACOR'un başlangıç popülasyonu oluşumu, gelişim süreci ve sonlandırma kriteri adımlarına ilişkin açıklamalara aşağıda yer verilmiştir.



Şekil 3.5 ACOR'un işleyiş diyagramı

Adım 1: Başlangıç Popülasyonu

Algoritma, belirli bir popülasyon büyüklüğüne bağlı olarak rastgele çözümlerden oluşan bir başlangıç popülasyonu oluşturur. Bu çözümler, karıncaların başlangıçtaki “yiyecek kaynaklarını” temsil eder. Her iterasyonda, bu çözümler iyileştirilir ve yeni çözümler üretilerek küresel optimuma yakınsama sağlanır.

Adım 2: Gelişim

- ***Seçim Olasılıklarının Hesaplanması***

ACOR algoritmasında, her bireyin seçilme olasılığı hesaplanır. Bu olasılıklar, popülasyonun sıralamasına ve “intensification factor” adı verilen parametreye bağlı olarak belirlenir. Popülasyondaki her çözüm bir sıralama ile numaralandırılır. Intensification Factor kullanılarak seçim baskısı hesaplanır. Matrix ağırlığı, her bireyin seçilme ağırlığını belirtirken, matrix olasılığı ile bu ağırlıkları normalleştirerek seçim olasılıklarını hesaplar. Intensification Factor, seçim baskısını belirler. Yüksek değerler, popülasyondaki en iyi çözümlere daha fazla odaklanılmasına neden olur, düşük değerler ise çeşitliliği artırır.

- ***Ortalama ve Standart Sapmaların Hesaplanması***

Her birey için standart sapmalar hesaplanır. Bu, bireyler arasındaki uzaklık ve zeta (sapma-mesafe oranı) kullanılarak belirlenir. Bu işlem, her bireyin diğer bireylerden ne kadar uzak olduğunu ve bu uzaklığın yeni çözümler oluşturulurken kullanılacak standart sapmalar üzerindeki etkisini belirler. Zeta, sapma-mesafe oranıdır ve yeni çözümler oluşturulurken kullanılan standart sapmaları belirler. Yüksek zeta değerleri, daha büyük arama adımlarına neden olur ve çeşitliliği artırır.

- ***Yeni Çözüm Üretilme***

Algoritmanın her iterasyonunda yeni çözümler üretilir. Her çözüm için matrix olasılığı kullanılarak rulet tekerleği seçimi yapılır ve mevcut popülasyondan bir birey seçilir. Yeni çözüm, seçilen bireyin çözümüne rastgele bir Gaussian dağılımı (normal dağılım) eklenerek oluşturulur. Yeni çözümler oluşturulurken, belirlenen standart sapmalar kullanılır. Bu, algoritmanın çözüm uzayını daha etkili bir biçimde keşfetmesini sağlar. Ayrıca yeni çözümlerin sınırların dışına çıkması durumunda geçerli çözüm aralığında kalmasını sağlanır. Sample Count, her iterasyonda oluşturulacak yeni çözüm sayısını belirler. Yüksek değerler, çözüm uzayında daha geniş bir arama yapılmasını sağlar.

- ***Popülasyonun Güncellenmesi***

Üretilen yeni çözümler, mevcut popülasyona eklenir ve uygunluk değerine göre sıralanır. En iyi çözümler, yeni popülasyon olarak seçilir ve sonraki iterasyona geçilir. Bu süreç, popülasyonun her iterasyonda iyileştirilmesini ve küresel optimuma doğru yakınsamasını sağlar.

Adım 3: Sonlandırma Kriteri

Algoritma, epoch parametresi ile belirlenen iterasyon sayısına kadar devam eder. Her iterasyonda popülasyonun uygunluk değeri iyileştirilir ve küresel optimuma yakınsama sağlanır. İstenilen iterasyon sayısına ulaşıldığında veya belirli bir durdurma kriteri sağlandığında algoritma sona erer ve en iyi çözüm döndürülür.

ACOR algoritması işletilirken gelişim adımında söz edilen ayarlanabilir parametreler; popülasyon sayısı (pop_size), iterasyon sayısı (epoch), sample count (sample_count), mutasyon olasılığı (intent_factor) ve zeta (zeta) olarak tanımlanmaktadır.

3.2.3 Yarasa Algoritması

Yarasa Algoritması (Bat Algorithm - BA), doğadaki mikro yarasaların ekolokasyon (sesle yer belirleme) davranışlarından esinlenerek geliştirilmiştir. Mikro yarasalar, yüksek frekanslı ultrasonik ses dalgaları yayıp yankıları dinleyerek avlarını tespit eder, engellerden kaçınır ve yönlerini belirler. Bu süreçte yarasalar, sesin yankılanma süresini, iki kulak arasındaki zaman farkını ve yankıların şiddetini analiz ederek çevrelerinde üç boyutlu bir harita oluşturabilirler. Yang (2010), çalışmasında yarasaların ekolokasyon davranışlarından esinlenerek BA yöntemini önermiştir. Algoritma, mevcut yöntemlerin avantajlarını birleştirmeyi ve üstün performans sunmayı amaçlamaktadır. Çeşitli test fonksiyonları üzerinde yapılan karşılaştırmalarda, BA'nın genetik algoritmalar ve parçacık sürü optimizasyonuna kıyasla daha yüksek doğruluk ve verimlilik sunduğu görülmüştür. Özellikle gürültü seviyesi ve darbe yayılım oranı gibi parametrelerin dinamik ayarlanması, algoritmanın global en iyi çözüme yakınsamadaki başarısını artırmaktadır.

Asokan vd. (2020), uydu görüntülerinin kontrastını artırmak amacıyla BA'ya dayalı bir yöntem önermişlerdir. Çalışmada, BA'nın uyumlu parametre seçimi sayesinde, uydu görüntülerinde doğrusal olmayan kontrast germe yöntemiyle parlaklık seviyelerini koruyarak iyileştirme sağladığı belirtilmiştir. BA'nın performans metriklerinde ACOR ve PSO gibi diğer yöntemlerden üstün olduğu, daha yüksek doğruluk ve daha hızlı yakınsama sağladığı gözlemlenmiştir.

Bu ekolokasyon mekanizması, yarasa algoritmasının temelini oluşturur. Algoritmada, yarasaların farklı frekanslarda ses dalgaları yayması ve yankılardan bilgi alması, optimizasyon problemleri için çözümler arayan “sanal yarasalar”ın davranışlarıyla modellenir. Yarasaların ses yayma frekansları, yankıların yüksekliği ve yankı oranları, algoritmanın keşif ve sömürü aşamalarında dengeli bir biçimde kullanılır. Bu sayede

Her ajan için frekans, belirlenen minimum ve maksimum sınırlar arasında rastgele bir deęer olarak atanır.

Adım 2: Gelişim

- ***Frekans, Hız ve Pozisyon Güncellemeleri***

Her iterasyonda, ajanın hızını ve pozisyonunu güncelleriz. Frekans deęeri, ajanın çözüm uzayında ne kadar ve hangi yöne doğru hareket edeceğini kontrol eder. Her ajanın pozisyonu, global en iyi çözüm referans alınarak güncellenir. Hız Güncelleme: Velocity deęişkeni, mevcut hız ve ajanın çözümünün en iyi çözümden ne kadar uzakta olduğunu belirten bir bileşen kullanılarak güncellenir. Pozisyon Güncelleme: Ajanın pozisyonu, mevcut pozisyona hız eklenerek güncellenir. Darbe Frekansı: Ajanın hızını ve arama davranışını etkileyen bir parametredir. Daha yüksek frekanslar, daha geniş bir arama yapılmasını sağlar.

- ***Yerel Arama***

Eđer rastgele bir sayı pulse rate deęerinden büyükse, ajan mevcut en iyi çözümün çevresinde küçük bir yerel arama yapar. Bu işlem, ajanın mevcut en iyi çözümün yakınında yerel bir iyileştirme yapmasını sağlar ve daha ince bir arama sürecine olanak tanır. Darbe Hızı (Pulse Rate), ajanların, global en iyi çözümün çevresinde yerel arama yapma olasılığını belirler.

- ***Sınır Kontrolü***

Güncellenen pozisyonun problem sınırları içerisinde olup olmadığını kontrol etmek ve gerekirse düzeltme uygulanır. Bu adım, algoritmanın geçerli çözüm aralığında kalmasını ve sınırların dışına çıkmamasını sağlar.

- ***Çözümlerin Güncellenmesi***

Güncellenen çözümün uygunluk deęeri hesaplanır ve popülasyondaki çözümle karşılaştırılır. Eđer yeni çözüm daha iyi ise, mevcut çözüm yeni çözümle deęiştirilir. Bu süreçte loudness deęeri kullanılır. Yeni çözüm, eski çözümden daha iyi ise ve bir rastgele sayı loudness deęerinden küçükse, mevcut ajan güncellenir. Loudness, ajanın bir çözümü ne kadar sıklıkla kabul edeceğini belirler. Daha düşük bir loudness deęeri,

sadece daha iyi çözümlerin kabul edilmesini sağlar. Yüksek loudness değeri, daha fazla çeşitlilik sağlar.

Adım 3: Sonlandırma Kriteri

Algoritma, epoch parametresi ile belirlenen iterasyon sayısına kadar devam eder. Her iterasyonda popülasyondaki ajanın hızı, pozisyonu ve uygunluk değeri güncellenir. İstenilen iterasyon sayısına ulaşıldığında veya belirli bir durdurma kriteri sağlandığında algoritma sona erer ve en iyi çözüm döndürülür.

BA algoritması işletilirken gelişim adımında söz edilen ayarlanabilir parametreler; popülasyon sayısı (pop_size), iterasyon sayısı (epoch), ses şiddeti (loudness), darbe hızı (pulse_rate) ve darbe frekansı sınır aralığı (pf_min, pf_max) olarak tanımlanmaktadır.

3.2.4 Guguk Kuşu Algoritması

Guguk Kuşu Algoritması (Cuckoo Search Algorithm – CSA), bazı guguk kuşu türlerinin parazit üreme davranışları ve Levy uçuşları gibi doğadan esinlenen mekanizmaları temel alır. Guguk kuşları, yumurtalarını diğer kuş türlerinin yuvalarına bırakma (zorunlu yuva parazitizmi) davranışı sergiler. Bu davranış, kendi yumurtalarının diğer kuşlar tarafından büyütülmesine olanak tanır ve üreme başarılarını artırır. Bazı guguk kuşları, ev sahibi kuşların yumurtalarını taklit eden renk ve desenlere sahip yumurtalar üretir ve böylece yumurtalarının reddedilme olasılığını azaltır. Yang ve Deb (2009), CSA algoritmasını Levy uçuşlarına dayalı bir optimizasyon yöntemi olarak sunmuşlardır. Algoritma, guguk kuşlarının zorunlu kuluçka parazitliği davranışları ile bazı kuş ve meyve sineklerinin Levy uçuşu özelliklerini birleştirir. Test fonksiyonları üzerinde yapılan deneyler, algoritmanın GA ve PSO gibi yöntemlere kıyasla daha az parametre ile daha yüksek başarı oranlarına ulaştığını göstermiştir. CSA, özellikle çok modlu ve çok hedefli optimizasyon problemleri için etkili bir strateji olarak değerlendirilmiştir.

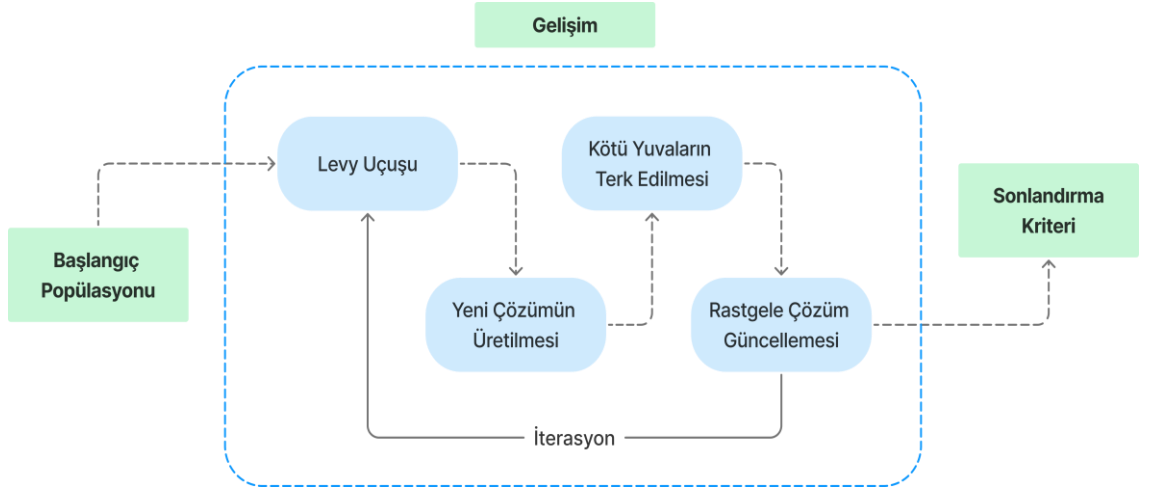
Bala vd. (2018) çalışmasında, CSA algoritmasını kullanarak Echo State Network parametrelerinin zaman serisi tahmini için optimize edilmesi incelenmiştir. Çalışmada,

Mackey-Glass zaman serisinin tahmini sırasında Echo State Network'in rezervuar boyutu, spektral yarıçapı ve bağlanabilirlik parametreleri optimize edilmiştir. Sonuçlar, CSA tabanlı optimizasyon yönteminin klasik Echo State Network'e kıyasla daha düşük MSE ve hızlı yakınsama ile üstün performans gösterdiğini ortaya koymuştur.

Turan-Karaoğlu ve Göktürkler (2021) çalışmalarında, CSA'nın kendi kendine potansiyel anomali verilerinin ters çözümünde model parametrelerini tahmin etmek için kullanmışlardır. Çalışmada hem sentetik hem de saha verileri üzerinde algoritmanın başarımı değerlendirilmiştir. Sonuçlar, CSA'nın diğer AI tabanlı optimizasyon yöntemlerine kıyasla benzer veya daha iyi sonuçlar sağladığını ve az ayarlanabilir parametre gereksinimi ile pratik bir alternatif sunduğunu göstermiştir.

CSA algoritması bu biyolojik stratejiyi matematiksel olarak modelleyerek, optimizasyon problemlerinde kullanılabilir bir yöntem sunar. Algoritmada, bir guguk yumurtası yeni bir çözümü temsil ederken, ev sahibi kuşun yuvası mevcut bir çözümün bulunduğu yer olarak modellenir. Yeni çözümler, Levy uçuşlarına dayalı rastgele bir yürüyüşle oluşturulur, böylece uzun mesafeli adımlarla arama alanının etkin bir biçimde keşfedilmesi sağlanır. Ayrıca, düşük kaliteli yuvalar belirli bir olasılıkla terk edilerek yerlerine yeni rastgele çözümler eklenir. Bu mekanizma, keşif ve sömürü arasında etkili bir denge sağlar ve algoritmayı global optimum çözümler bulmada güçlü kılar.

Şekil 3.7'de CSA'nın işleyiş diyagramı görülmektedir. Buna göre CSA'nın başlangıç popülasyonu oluşumu, gelişim süreci ve sonlandırma kriteri adımlarına ilişkin açıklamalara aşağıda yer verilmiştir.



Şekil 3.7 CSA'nın işleyiş diyagramı

Adım 1: Başlangıç Popülasyonu

Algoritma, belirli bir popülasyon büyüklüğüne bağlı olarak rastgele çözümlerden oluşan bir başlangıç popülasyonu oluşturur. Bu çözümler, guguk kuşlarının yuvalarını temsil eder. Her iterasyonda popülasyon, yeni çözümler oluşturularak ve mevcut çözümler iyileştirilerek güncellenir.

Adım 2: Gelişim

- **Levy Uçuşu**

Her iterasyonda, mevcut çözümler (yuvalar) üzerinde Levy uçuşu kullanılarak yeni çözümler üretilir. Levy uçuşu, büyük sıçramalar yaparak çözüm uzayının daha geniş bir bölümünü keşfetmeyi sağlar. Yeni çözüm, mevcut çözüm ve global en iyi çözüm arasındaki fark kullanılarak oluşturulur. Levy Adımı ile algoritmanın çözüm uzayında daha geniş bir bölümünü aramasına olanak tanır. Bu işlem, her bir çözümün en iyi çözüme yakınsamasını sağlamak için uygulanır ve rastgeleliği artırarak yerel optimumlardan kaçınılmasına yardımcı olur.

- **Yeni Çözümlerin Üretilmesi**

Yeni oluşturulan çözümler, problem sınırları dışında kalmışsa düzeltilir. Her yeni çözümün uygunluk değeri (fitness) hesaplanır ve mevcut çözümle karşılaştırılır. Eğer yeni çözüm daha iyiye, mevcut çözüm yeni çözümle değiştirilir. Bu adım,

popülasyonun her iterasyonda iyileştirilmesini ve en iyi çözüme doğru yakınsamasını sağlar.

- ***En Kötü Yuvaların Terk Edilmesi***

Algoritmanın bir diğer adımı da belirli bir olasılıkla (p_a) en kötü yuvaların terk edilmesidir. Bu işlem, çeşitliliği artırarak algoritmanın çözüm uzayında daha etkili bir biçimde arama yapmasını sağlar. Belirli bir sayıda en kötü yuva terk edilir ve bu yuvalar yerine rastgele yeni çözümler üretilir. Bu işlem, popülasyondaki çeşitliliği koruyarak yerel optimumlara sıkışmayı önler ve daha geniş bir çözüm uzayı aramasına olanak tanır. Tahrip Olma Olasılığı, terk etme olasılığıdır. Bu parametre, popülasyonun belirli bir kısmının terk edilip yerine rastgele yeni çözümler eklenme olasılığını belirler. Yüksek p_a değeri, daha fazla çözümün terk edilmesine neden olur ve bu da çözüm uzayında daha fazla çeşitlilik sağlar.

Adım 3: Sonlandırma Kriteri

Algoritma, epoch parametresi ile belirlenen iterasyon sayısına kadar devam eder. Her iterasyonda popülasyonun uygunluk değeri iyileştirilir ve küresel optimuma yakınsama sağlanır. İstenilen iterasyon sayısına ulaşıldığında veya belirli bir durdurma kriteri sağlandığında algoritma sona erer ve en iyi çözüm döndürülür.

CSA algoritması işletilirken gelişim adımı söz edilen ayarlanabilir parametreler; popülasyon sayısı (pop_size), iterasyon sayısı (epoch) ve tahrip olma olasılığı (p_a) olarak tanımlanmaktadır.

3.2.5 Gri Kurt Optimizasyonu

Gri Kurt Optimizasyonu (Grey Wolf Optimization - GWO), doğadaki gri kurtların (Canis lupus) liderlik hiyerarşisi ve avlanma davranışlarından esinlenerek geliştirilmiştir. Gri kurtlar, güçlü bir sosyal yapı içinde avlanırken, hiyerarşilerinde dört ana grup bulunur: alfa, beta, delta ve omega. Alfa kurtları liderlik yaparken, beta ve delta kurtları destek görevleri üstlenir, omega kurtları ise en alttaki sıradadır. Bu hiyerarşi, algoritmada en

uygun çözümün alfa, ikinci en uygun çözümün beta, üçüncü en uygun çözümün delta olarak atanmasıyla modellenir ve diğer kurtlar (omega) bu çözüm adaylarını takip eder.

Mirjalili vd. (2014), gri kurtların sosyal hiyerarşi ve avlanma davranışlarından esinlenerek GWO algoritmasını önermişlerdir. Algoritma, alfa, beta, delta ve omega gibi farklı liderlik seviyeleri ve avın izlenmesi, kuşatılması ve saldırılması gibi üç ana avlanma aşamasını modellemektedir. 29 test fonksiyonu üzerinde yapılan deneyler, GWO'nun PSO, DE ve GA gibi popüler metasezgisel algoritmalara kıyasla rekabetçi sonuçlar sağladığını göstermiştir. Ayrıca, gerilme/yay yay tasarımı, kaynaklı giriş tasarımı ve basınçlı kap tasarımı gibi mühendislik problemlerinde yüksek performans sergilediği doğrulanmıştır.

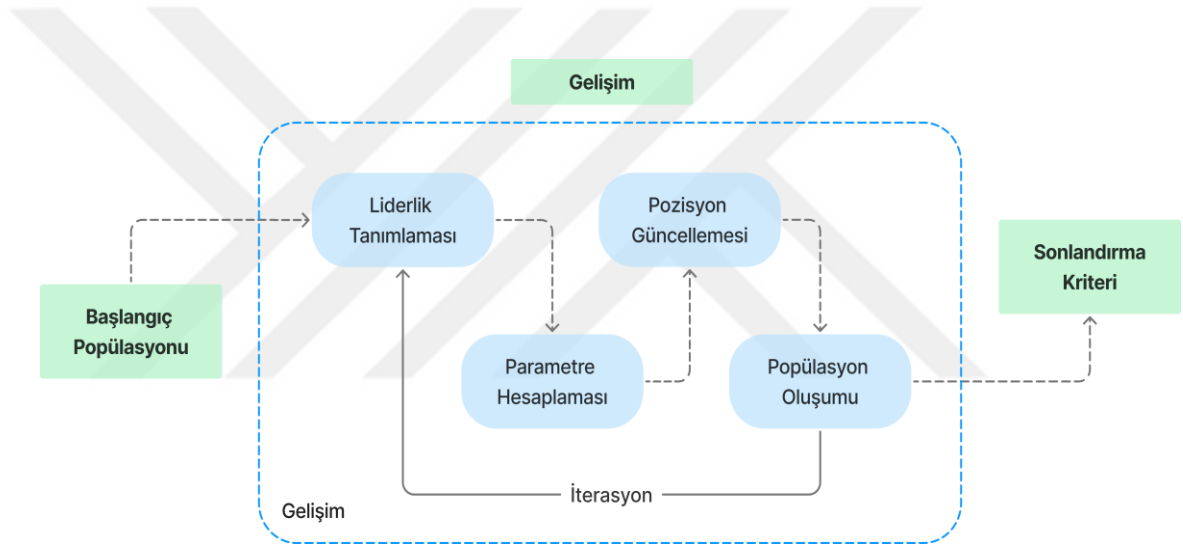
Abdelwanis (2022), üç fazlı doğrusal indüksiyon motorlarının parametre tahmini için GWO algoritmasını kullanmışlardır. Çalışmada, algoritma ile elde edilen parametreler, GA, DE ve PSO ile karşılaştırılmıştır. Sonuçlar, GWO'nun daha düşük hata oranları ve daha yüksek yakınsama performansı ile diğer yöntemlerden üstün olduğunu ve performans tahminini doğru olarak sağladığını göstermiştir. GWO, parametre tahmini ve performans analizi için etkili bir yöntem olarak değerlendirilmiştir.

Cheng ve Liu (2024), GWO algoritması geliştirmek amacıyla Tent kaotik haritalama, altın madenciliği optimizasyon algoritması, yatay ve dikey çaprazlama stratejileri ve Gauss mutasyonunu entegre ederek karma bir algoritma önermiştir. Çalışma, algoritmanın küresel arama yeteneğini artırdığını, yerel optimuma takılmayı önlediğini ve daha hızlı yakınsama sağladığını göstermiştir. 23 standart test fonksiyonunda GWO'nun, diğer optimizasyon algoritmalarından daha yüksek doğruluk ve kararlılık sergilemiş, ayrıca yol planlama problemlerinde daha kısa yollar ve daha az hesaplama süresi ile üstün performans sağlamıştır.

Avlanma sürecinde kurtlar, avı çevreleyerek ve aşamalı olarak ona yaklaşarak avlarını yakalar. GWO algoritmasında, kurtların avı çevreleme, yaklaşma ve saldırma davranışları matematiksel olarak modellenmiştir. Bu süreç, algoritmanın geniş bir arama (keşif) ve daha iyi çözümlere odaklanma (sömürü) yeteneklerini dengeli bir biçimde kullanmasını

sağlar. Kurtlar, popülasyondaki en iyi üç çözüm olan alpha, beta ve delta liderlerine göre pozisyonlarını güncelleyerek çözüm uzayında arama yaparlar. Bu süreç, kontrol parametreleri kullanılarak gerçekleştirilir ve algoritma her iterasyonda avı (küresel optimum çözümü) yakalamaya çalışır. Kodda tanımlanan parametreler ve adımlar, kurtların çözüm uzayındaki arama davranışını belirler ve küresel optimuma ulaşmayı hedefler.

Şekil 3.8’de GWO’nun işleyiş diyagramı görülmektedir. Buna göre GWO’nun başlangıç popülasyonu oluşumu, gelişim süreci ve sonlandırma kriteri adımlarına ilişkin açıklamalara aşağıda yer verilmiştir.



Şekil 3.8 GWO’nun işleyiş diyagramı

Adım 1: Başlangıç Popülasyonu

Algoritma, belirli bir popülasyon büyüklüğüne bağlı olarak rastgele çözümlerden oluşan bir başlangıç popülasyonu oluşturur. Her birey çözüm uzayında bir potansiyel çözümü temsil eder. Bu kurt popülasyonu, alpha, beta ve delta olarak adlandırılan üç lider bireyi içerir. Bu liderler, popülasyondaki en iyi çözümleri temsil eder ve diğer bireylerin hareketini yönlendirir.

Adım 2: Gelişim

- ***Liderlik Tanımlanması***

Popülasyonun en iyi üç bireyi sırasıyla alpha, beta ve delta olarak belirlenir. Bu bireyler, sürüyü yönetir ve diğer kurtların (omega kurtları) pozisyonlarını güncellemelerine rehberlik eder. Bu liste, popülasyondaki en iyi üç çözümü (alpha, beta, delta) içerir.

- ***Parametre Hesaplanması***

GWO algoritmasında, kurtların avını yakalama sürecini simüle etmek için bazı koordinat faktörleri hesaplanır. a: iterasyonlar boyunca 2'den 0'a doğru lineer olarak azalır. Bu, kurtların avlarına yakınsamalarını ve daha hassas arama yapmalarını sağlar. A ve C: Kurtların avını bulma ve avın konumuna doğru hareket etmeleri için kullanılan iki vektördür. A, arama sürecinin keşif veya sömürü olduğunu belirlerken, C avın (en iyi çözümün) konumuna göre diğer kurtların pozisyonlarını güncellemeleri için kullanılır.

- ***Pozisyon Güncellemesi***

Her kurt, popülasyondaki alpha, beta ve delta konumlarına göre hareket eder. Her kurt, bu üç liderin pozisyonuna göre kendi pozisyonunu günceller. Yeni pozisyon, bu üç lider bireyin konumlarına göre hesaplanarak, bu üç pozisyonun ortalaması olarak belirlenir. X_1 : alpha kurtuna göre hesaplanan pozisyon. X_2 : beta kurtuna göre hesaplanan pozisyon. X_3 : delta kurtuna göre hesaplanan pozisyon.

- ***Popülasyon Oluşumu***

Güncellenen pozisyonların uygunluk değerleri hesaplanır ve her kurt için bu yeni pozisyonun eski pozisyonla karşılaştırılması yapılır. Eğer yeni pozisyon daha iyi bir çözüm sunuyorsa, ajan bu yeni pozisyonla güncellenir. Yeni popülasyon, eski popülasyonla birleştirilir ve uygunluk değerine göre sıralanarak daha iyi çözümler popülasyonda tutulur.

Adım 3: Sonlandırma Kriteri

Algoritma, belirlenen iterasyon sayısına kadar çalışır. Her iterasyonda kurtların pozisyonları güncellenir ve popülasyondaki en iyi çözüm bulunmaya çalışılır. İstenilen iterasyon sayısına ulaşıldığında algoritma sona erer.

GWO algoritması işletilirken gelişim adımında söz edilen ayarlanabilir parametreler; popülasyon sayısı (pop_size) ve iterasyon sayısı (epoch) olarak tanımlanmaktadır.

3.2.6 Jaya Algoritması

Jaya Algoritması (Jaya Algorithm - JA), en iyi çözüme ulaşmayı ve en kötü çözümden uzaklaşmayı hedefleyen basit ama güçlü bir optimizasyon yöntemidir. Bu algoritma, diğer doğa esinli algoritmalar gibi zorlu ayar parametrelerine ihtiyaç duymaz ve başarıya ulaşmayı, başarısızlıktan kaçınmayı esas alır. “Jaya” kelimesi Sanskritçe’de “zafer” anlamına gelir ve algoritmanın temel amacı her iterasyonda çözümleri daha iyiye yönlendirerek en iyi çözümü bulmaya çalışmaktır. Bu yönüyle JA, başarıya ulaşma ve hedefe yönelik ilerlemeyi temsil eden bir zafer arayışını simgeler.

Rao (2016), genetik ve sürü zekası tabanlı algoritmalara alternatif olarak Jaya isimli, algoritma-spesifik kontrol parametresi gerektirmeyen bir optimizasyon yöntemi geliştirmiştir. Çalışmada, bu algoritma kısıtlı ve kısıtsız 54 farklı benchmark fonksiyonunda test edilmiş ve diğer popüler yöntemler (GA, PSO, DE, ABC, vb.) ile karşılaştırılmıştır. JA, özellikle düşük hesaplama maliyetiyle yüksek doğruluk ve yakınsama başarısı sergilemiş hem Friedman sıralama testi hem de Holm-Sidak testi sonuçlarına göre üstün performans göstermiştir. Bu algoritma, mühendislik problemlerinde etkin ve uygulanabilir bir çözüm sunmaktadır.

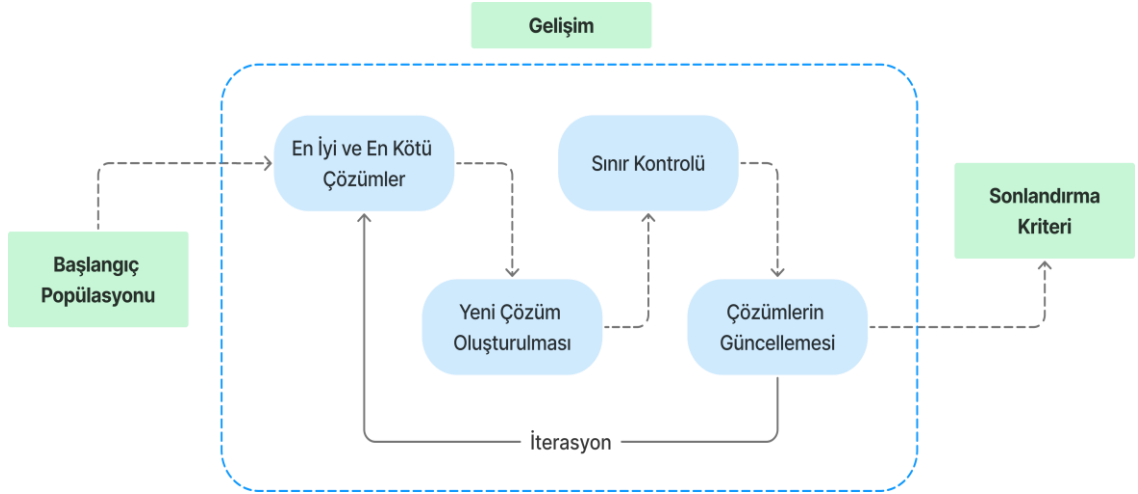
Sepulchro ve Encarnação (2024), güç sistemlerindeki harmonik bozulmayı azaltmak için Harmonik Durum Tahmini metodolojisinde JA’yı önermiştir. Algoritma, IEEE 14 ve 30 otobüs sistemlerinde test edilmiş. JA, daha az parametre gereksinimi ve basit yapısı sayesinde diğerlerine kıyasla daha düşük işlem süreleri ile aynı doğrulukta sonuçlar

sunmuştur. Çalışma, önerilen metodolojinin harmonik kaynakları tespit etmek için daha az sayıda güç kalitesi ölçer kullanarak etkili olduğunu göstermiştir.

Amavasai ve Dijkstra (2024), parçacık filtreleme yönteminin zayıflıklarını gidermek amacıyla JA ile birleşik bir hibrit algoritma önermişlerdir. Algoritma, geoteknik mühendislik problemlerinde hem durum hem de parametre tahmininde kullanılmıştır. Parçacıkların yüksek olasılık bölgelerine taşınmasını sağlayarak doğruluk ve yakınsama hızını artırırken, parçacık çeşitliliğini koruyarak yoksullaşma riskini azaltmıştır. Analitik ve elastoplastik modelleme dahil olmak üzere çeşitli örneklerle test edilen algoritma, PF-SIR ve EnKF gibi geleneksel yöntemlerden daha üstün performans göstermiştir. Ayrıca, algoritmanın gözlem verilerinin seyrek olduğu durumlarda bile etkin olduğu ortaya konulmuştur.

JA, optimize etmeye çalıştığı problemi basit bir biçimde ele alır. Her iterasyonda popülasyondaki bireyler, en iyi çözüme doğru yakınsar ve en kötü çözümden uzaklaşmaya çalışır. Bu sürecin amacı, bireylerin her iterasyonda daha iyi çözümler bulması ve küresel optimuma ulaşmasıdır. JA'nın avantajı, ek parametreler gerektirmemesi ve bu nedenle ayarlanması gereken parametrelerin olmamasıdır. Bu, algoritmanın kullanımı ve uygulanmasını kolaylaştırır.

Şekil 3.9'da JA'nın işleyiş diyagramı görülmektedir. Buna göre JA'nın başlangıç popülasyonu oluşumu, gelişim süreci ve sonlandırma kriteri adımlarına ilişkin açıklamalara aşağıda yer verilmiştir.



Şekil 3.9 JA'nın işleyiş diyagramı

Adım 1: Başlangıç Popülasyonu

Algoritma, belirli bir popülasyon büyüklüğüne bağlı olarak rastgele çözümlerden oluşan bir başlangıç popülasyonu oluşturur. Her birey, çözüm uzayında bir potansiyel çözümü temsil eder. Bu popülasyon, problemin çözüm uzayındaki rastgele noktalarla başlatılır ve her birey için uygunluk değeri hesaplanır.

Adım 2: Gelişim

- **En İyi ve En Kötü Çözümlerin Seçilmesi**

Her iterasyonda, popülasyonun en iyi ve en kötü çözümleri seçilir. Bu çözümler, popülasyonun geri kalanının performansını yönlendirmek için kullanılır. Bu iki çözüm, diğer bireylerin iyileştirilmesi sürecinde kullanılır.

- **Yeni Çözüm Oluşturulması**

JA'nın tema adımı, mevcut çözümleri iyileştirmek için en iyi ve en kötü çözümler kullanılarak yeni çözümler oluşturulmasıdır. Bu adım, bireylerin hem en iyi çözüme doğru ilerlemesini hem de en kötü çözümden uzaklaşmasını sağlayarak daha iyi çözümler bulunmasını hedefler.

- **Çözümlerin Güncellenmesi**

Oluşturulan yeni çözümler, problem sınırları dışında kalabilir. Bu durumda çözümler düzeltilir ve sınırlar içinde olacak biçimde ayarlanır. Yeni pozisyonun uygunluk

değeri (fitness) hesaplanır ve popülasyona eklenir. Yeni popülasyon, uygunluk değerlerine göre en iyi çözümleri popülasyonda tutarak güncellenir.

Adım 3: Sonlandırma Kriteri

Algoritma, epoch parametresi ile belirlenen iterasyon sayısına kadar devam eder. Her iterasyonda popülasyonun en iyi ve en kötü çözümleri belirlenir ve bu çözümler kullanılarak popülasyonun diğer bireyleri iyileştirilir. İstenilen iterasyon sayısına ulaşıldığında algoritma sona erer ve en iyi çözüm döndürülür.

JA algoritması işletilirken gelişim adımında söz edilen ayarlanabilir parametreler; popülasyon sayısı (pop_size) ve iterasyon sayısı (epoch) olarak tanımlanmaktadır.

3.2.7 Parçacık Sürü Optimizasyonu

Parçacık Sürü Optimizasyonu (Particle Swarm Optimization - PSO), kuş sürüsü ve balık sürüsü gibi doğal grupların davranışlarını taklit etmek amacıyla geliştirilmiştir. Kennedy ve Eberhart'ın bu algoritmayı geliştirirken temel motivasyonları, sosyal organizmaların çevresel uyaranlara topluca ve organize bir biçimde tepki verme biçimlerini modellemek olmuştur. Bu sosyal modelde bireyler, çevrelerinden bilgi toplayarak ve diğer bireylerin deneyimlerinden faydalananarak kendilerini en iyi duruma getirmeye çalışırlar. Kennedy ve Eberhart (1995), kuş sürüsü davranışından esinlenerek PSO algoritmasını geliştirmişlerdir. Bu algoritma, bireylerin yerel ve küresel en iyi konumlarına doğru hızlarını ayarlayarak çözümleri optimize etmelerini sağlar. PSO, yapısal olarak basit olup az parametre gerektirir ve çok boyutlu, doğrusal olmayan optimizasyon problemlerinde etkili sonuçlar verir. Özellikle yapay sinir ağlarının ağırlıklarını eğitmede ve genetik algoritmalar için zorlu test fonksiyonlarını optimize etmede başarılı olduğu gösterilmiştir.

Feng vd. (2023), kalıcı mıknatıslı senkron jeneratörlerin parametre tahmini için, Levy uçuşu ile geliştirilmiş kendi kendine öğrenme PSO algoritmasını önermiştir. PSO, popülasyon yoğunluğuna dayalı bir dağılım stratejisi, kaotik eylemsizlik ağırlığı ve bellek tavlama anı algoritması gibi yenilikçi teknikler kullanarak PSO'nun yerel maksimumlara takılma sorununu azaltmıştır. Çalışma, PSO'nun %2'nin altında hata oranıyla yüksek

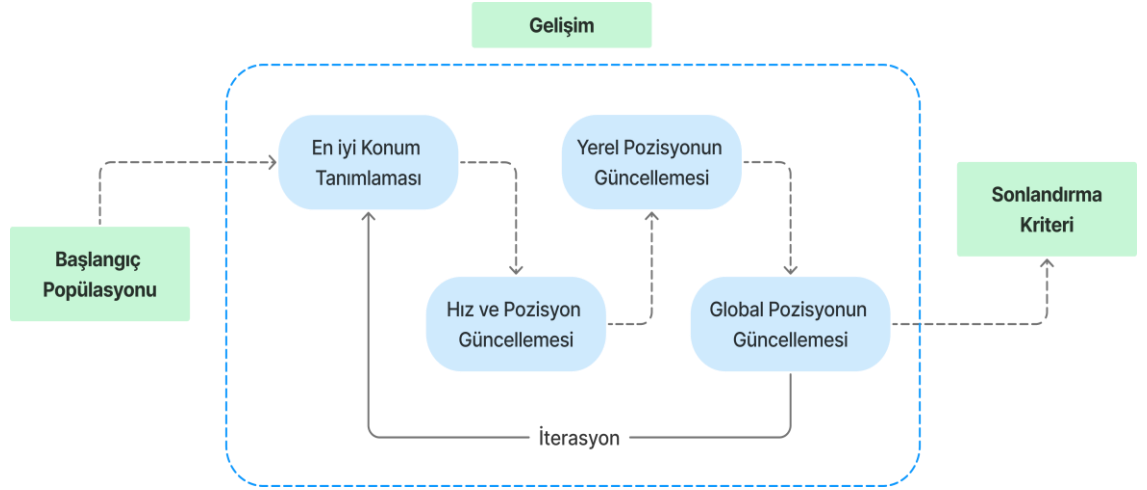
doğruluk sağladığını ve değişken çalışma koşulları altında diğer yöntemlerden daha hızlı ve kararlı olduğunu göstermiştir.

Yang vd. (2023), fotovoltaik hücre model parametrelerinin tahmini için geliştirilmiş bir PSO algoritması önermiştir. Bu algoritma, sıkışma faktörü yöntemi ve deneyime dayalı öğrenme stratejisi gibi yenilikler ile PSO'nun çok boyutlu problemler ve yerel çözümlere takılma sorunlarını iyileştirmeyi amaçlamıştır. Tek diyot modeli üzerinde yapılan deneyler, geliştirilmiş algoritmanın daha yüksek doğruluk ve hızlı yakınsama sağladığını, böylece standart PSO'ya kıyasla parametre tahmininde daha üstün performans gösterdiğini ortaya koymuştur.

Kuş sürüleri veya balık sürüleri örneğinde olduğu gibi, her bir “parçacık” (birey), kendi en iyi pozisyonunu (pbest) ve sürünün en iyi pozisyonunu (gbest) bilerek hareket eder. Parçacıklar, sürünün en iyi bulduğu pozisyona doğru hareket ederken kendi en iyi deneyimlerini de dikkate alır ve bu süreç boyunca hedeflerini aşarak çevreyi keşfetme eğilimindedirler. Bu basit ama etkili davranış, PSO algoritmasına çözüm arama sürecinde hem çeşitlilik hem de uyum sağlar ve sonuç olarak optimal çözümlere hızla ulaşma yeteneği sunar.

PSO algoritması, sürü zekasından ilham alan bir optimizasyon algoritmasıdır. Her parçacık, çözüm uzayında hareket ederek daha iyi bir çözüm bulmayı hedefler. Parçacıkların çözüm uzayında global optimuma yakınsamak amacıyla hız ve pozisyonlarını bilişsel ve sosyal bileşenlere göre güncellemelerine dayanan bir algoritmadır. Her parçacık kendi yerel en iyi çözümünü ve popülasyonun global en iyi çözümünü takip ederek, çözüm uzayında arama yapar ve bu süreç, en uygun çözümü bulmak için tekrarlanır. Kullanılan parametreler ile parçacıkların arama uzayındaki davranışını kontrol edilir ve algoritmanın performansı belirlenir.

Şekil 3.10'da PSO'nun işleyiş diyagramı görülmektedir. Buna göre PSO'nun başlangıç popülasyonu oluşumu, gelişim süreci ve sonlandırma kriteri adımlarına ilişkin açıklamalara aşağıda yer verilmiştir.



Şekil 3.10 PSO'nun işleyiş diyagramı

Adım 1: Başlangıç Popülasyonu

Algoritma, popülasyon büyüklüğüne bağlı olarak rastgele çözümlerden oluşan bir başlangıç popülasyonu (parçacıklar) oluşturur. Her parçacık, çözüm uzayında rastgele bir noktadan başlar ve rastgele bir başlangıç hızına sahiptir. Her parçacığın pozisyonu (solution) ve hızı (velocity) vardır. Hız, çözümün çözüm uzayında ne kadar ve hangi yöne hareket edeceğini belirler.

Adım 2: Gelişim

- **En İyi Konumların Tanımlanması**

Her parçacığın bir yerel en iyi (pbest) konumu vardır. Bu, parçacığın o ana kadar elde ettiği en iyi çözümü ifade eder. Ayrıca tüm popülasyonun genelinde küresel en iyi konum (gbest) tanımlanır. Bu, tüm parçacıkların en iyi çözümüdür ve popülasyonun hedefi bu en iyi çözümü daha da geliştirmektir.

- **Hız ve Pozisyon Güncelleme**

Her iterasyon boyunca, parçacıklar hızlarını ve pozisyonlarını güncellerler. Bilişsel bileşen ve sosyal bileşen, her iterasyonda rastgele bir sayı ile çarpılarak parçacıkların arama alanında farklı yöne gitmeleri sağlanır. Parçacığın hızı güncellenirken eylemsizlik, bilişsel ve sosyal katsayılar kullanılır. Momentum Katsayısı (Eylemsizlik): Önceki hızın bir kısmını korur ve parçacığın mevcut hareket yönünü sürdürmesini sağlar. Momentum katsayısı hızın ağırlığını belirtir. Bilişsel Katsayı

(Local): Parçacığın kendi yerel en iyi konumuna yönelmesini sağlar. Bu bileşen, parçacığın kendi en iyi çözümüne doğru çekilmesini sağlar. Sosyal Katsayı (Global): Parçacığın küresel en iyi konuma doğru yönelmesini sağlar. Bu bileşen, popülasyondaki diğer parçacıklardan öğrenmeyi ve popülasyonun genelinde en iyi konuma yakınsamayı ifade eder.

- ***Yerel Pozisyonun Güncellenmesi***

Güncellenen hız, parçacığın pozisyonunu güncellemek için kullanılır. Parçacık, hızına bağlı olarak çözüm uzayında yeni bir konuma hareket eder. Yeni pozisyon, problem sınırları içinde olup olmadığı kontrol edilerek düzeltilir. Yeni pozisyonun uygunluğu (fitness) hesaplanır ve bu uygunluk, parçacığın önceki çözümüyle karşılaştırılır. Eğer yeni çözüm (pbest) daha iyiye, parçacığın çözümü güncellenir.

- ***Global Pozisyonun Güncellenmesi***

Her iterasyonun sonunda, popülasyondaki tüm parçacıkların uygunluk değerleri değerlendirilir ve küresel en iyi çözüm (gbest) güncellenir. Bu, algoritmanın tüm popülasyonun en iyi çözümüne yakınsamasını sağlar.

Adım 3: Sonlandırma Kriteri

Algoritma, belirlenen iterasyon sayısına (epoch) ulaşana kadar devam eder. Her iterasyonda parçacıkların hızları ve pozisyonları güncellenir.

PSO algoritması işletilirken gelişim adımı söz edilen ayarlanabilir parametreler; popülasyon sayısı (pop_size), iterasyon sayısı (epoch), momentum katsayısı (w), yerel katsayı (c1) ve global katsayı (c2) olarak tanımlanmaktadır.

3.3 Müzik Esinli Yöntemler

Müzik esinli Armoni Arama (Harmony Search – HS) Algoritması, müzisyenlerin doğaçlama yaparken yeni nota kombinasyonlarını arama mantığını modelleyerek çözüme ulaşır.

3.3.1 Armoni Arama Algoritması

Armoni Arama (Harmony Search – HS) Algoritması, müziksel doğaçlama sürecinden esinlenerek geliştirilmiştir. Müzik performanslarında, özellikle bir caz grubunda olduğu gibi, müzisyenler, en iyi uyumu yakalamak için çeşitli notaları deneyerek bir uyum yakalamaya çalışırlar. Bu süreçte müzikal estetik açısından en tatmin edici ses birleşimini (harmoni) bulmaya yönelik yapılan doğaçlama, HS algoritmasının esin kaynağını oluşturmuştur.

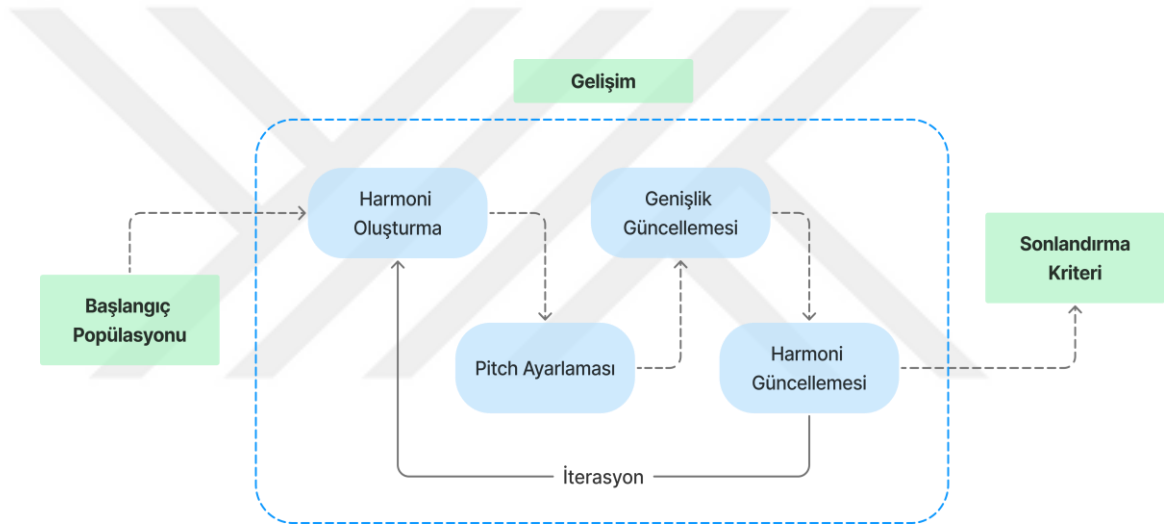
Geem vd. (2001), müziksel doğaçlama sürecinden esinlenerek HS algoritmasını önermişlerdir. Bu algoritma, optimizasyon problemlerine yeni bir bakış açısı getirerek müziksel uyumu iyileştirme prensibini matematiksel modelleme ile birleştirmiştir. Çalışma, HS'nin Gezgin Satıcı Problemi, sürekli değişkenli optimizasyon ve boru şebekesi tasarımı gibi çeşitli problemlerde etkili sonuçlar sağladığını göstermiştir. Özellikle, HS'nin az sayıda iterasyonla daha hızlı ve daha iyi çözümler üretebildiği vurgulanmış ve diğer yöntemlere kıyasla daha esnek ve uyarlanabilir olduğu belirtilmiştir.

Al-Betar (2021), ekonomik yük dağıtım problemlerini çözmek için ada tabanlı HS algoritmasını önermiştir. HS, nüfusu adalara bölerek çeşitliliği kontrol altına alır ve HS algoritmasını her adada bağımsız olarak uygular. Göç süreçleri sayesinde adalar arasında bilgi paylaşımı sağlanır. Çalışmada, HS'nin 3, 13, 40 ve 80 jeneratör birimine sahip sistemler için üstün sonuçlar ürettiği, özellikle zorlu kısıtlarla başa çıkmada etkili olduğu gösterilmiştir. HS, düşük yakıt maliyeti ve hızlı yakınsama ile diğer optimizasyon yöntemlerine karşı rekabetçi bir alternatif sunmaktadır.

HS algoritmasında, bir çözüm havuzu olan “Hafıza” (Harmony Memory) ilk adımda rastgele çözümlerle doldurulur ve her döngüde bu hafızadan esinlenilerek yeni çözümler oluşturulur. Her iterasyonda yeni oluşturulan çözüm, mevcut en kötü çözümle karşılaştırılır; daha iyi ise hafızaya eklenir. Bu süreç, müzikal bir doğaçlama gibi, daha iyi bir çözüm yakalanana kadar devam eder. Böylelikle HS algoritması, müziğin uyum arayışına benzer bir biçimde optimizasyon problemlerinde en iyi çözümü arar.

HS algoritması, müzik besteleme sürecine benzer bir biçimde yeni harmoniler (yeni çözümler) oluşturur ve mevcut popülasyonla bu çözümleri birleştirerek uygunluğu en yüksek olan çözümleri tutar. Algoritma, mevcut iyi çözümlerden yararlanarak ve ince ayar yaparak çözüm uzayında en iyi uyumu (global optimum) arar. Parametrik değerler olan Harmony Memory Consideration Rate ve Pitch Adjustment Rate'i kullanarak HS algoritmasının bu süreci yönetmesini sağlar.

Şekil 3.11'de HS'nin işleyiş diyagramı görülmektedir. Buna göre HS'nin başlangıç popülasyonu oluşumu, gelişim süreci ve sonlandırma kriteri adımlarına ilişkin açıklamalara aşağıda yer verilmiştir.



Şekil 3.11 HS'nin işleyiş diyagramı

Adım 1: Başlangıç Popülasyonu

HS algoritması, belirli bir sayıda çözümden (bireylerden) oluşan bir başlangıç popülasyonu oluşturur. Bu bireyler, problemin çözüm uzayında rastgele noktalardan seçilir ve her biri bir olası çözümü temsil eder.

Adım 2: Gelişim

- **Harmoni Oluşturma**

HS algoritmasının en önemli adımı, mevcut çözümlerden (harmony memory) esinlenerek yeni bir çözüm (yeni harmoni) oluşturmaktır. Yeni harmoni oluşturulurken, her çözüm bileşeni için belirli bir olasılık (Harmony Memory

Consideration Rate) ile mevcut popülasyondan esinlenilir. Eğer rastgele bir sayı bu parametre değerinden küçükse, popülasyondan rastgele seçilen bir bireyin ilgili bileşeni kullanılır. Harmony Memory Consideration Rate: Bu parametre, yeni çözümün mevcut popülasyondan nasıl oluşturulacağını belirler. Eğer rastgele bir sayı bu değerden küçükse, popülasyondan rastgele seçilen bir birey alınır. Bu, mevcut iyi çözümlerin gelecekteki çözümlerde de kullanılmasını sağlar.

- ***Pitch Ayarlaması***

Yeni oluşturulan çözümde, bazı bileşenler üzerinde ince ayar yapılabilir. Bu işlem, çözüm uzayının daha detaylı bir biçimde keşfedilmesine olanak tanır ve mevcut çözümün çevresinde arama yapılmasını sağlar. Pitch Adjustment Rate: Bu parametre, yeni çözümün ince ayar yapılarak değiştirilip değiştirilmeyeceğini belirler. Eğer rastgele bir sayı bu değerden küçükse, çözümün ilgili bileşeni üzerinde bir değişiklik yapılır. Bu işlem, Gauss dağılımına dayalı bir değişiklik yapılması ile gerçekleştirilir. Gauss dağılımı, çözüm uzayının daha iyi araştırılması için bir ortalama ve standart sapma kullanılarak bireylerin değiştirilmesini sağlar.

- ***Genişlik Güncellenmesi***

Oluşturulan yeni harmoniler, mevcut popülasyon ile birleştirilir ve uygunluk değerlerine göre sıralanır. Daha sonra, popülasyon boyutunu aşan çözümler kesilir ve yalnızca en iyi çözümler popülasyonda tutulur.

- ***Harmoni Güncellemesi***

Perde genişliği (fret width), çözüm uzayında yapılacak değişikliklerin büyüklüğünü kontrol eder. Her iterasyonda bu değer azaltılması, algoritmanın son iterasyonlara doğru daha hassas bir biçimde arama yapmasına olanak tanır.

Adım 3: Sonlandırma Kriteri

Algoritma, belirlenen iterasyon sayısı kadar tekrarlanır. Her iterasyonda yeni harmoniler oluşturulur, ince ayar yapılır ve popülasyon güncellenir.

HS algoritması işletilirken gelişim adımında söz edilen ayarlanabilir parametreler; popülasyon sayısı (pop_size), iterasyon sayısı (epoch), çaprazlama olasılığı (c_r) ve mutasyon olasılığı (pa_r) olarak tanımlanmaktadır.

3.4 İnsan Davranışında Etkilenen Yöntemler

İnsan davranışlarından ilham alan Öğrenme-Öğretme Tabanlı Optimizasyon (Teaching-Learning based Optimization – TLO) ise, eğitim sürecinden yola çıkarak öğretmen-öğrenci etkileşimini taklit eder ve en başarılı bireylerin bilgisini tüm topluluğa aktararak popülasyonun genel performansını yükseltmeyi hedefler.

3.4.1 Öğretme-Öğrenme Tabanlı Optimizasyon

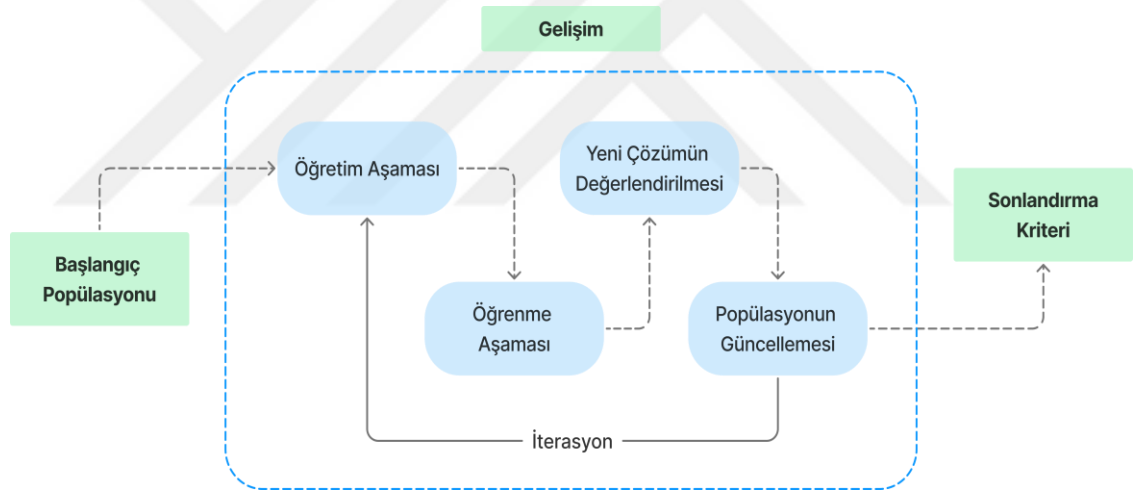
Öğrenme-Öğretme Tabanlı Optimizasyon (Teaching-Learning based Optimization - TLO), bir sınıf ortamındaki öğretmen-öğrenci etkileşiminden esinlenmiştir. TLO, bir öğretmenin öğrenciler üzerindeki etkisini ve öğrencilerin birbirleriyle etkileşimlerinden nasıl öğrendiklerini modelleyen bir optimizasyon yöntemidir. Bu algoritma, popülasyon tabanlı bir yöntemdir ve problem çözme sürecini iki aşamaya böler. Öğretmen aşaması, sınıfın en bilgili bireyi olan öğretmen (veya mevcut en iyi çözüm), öğrencilerin bilgi seviyesini artırmaya çalışır. Bu süreç, sınıfın genel ortalamasını (çözüm uzayındaki popülasyonun ortalaması) en iyi çözüme doğru yaklaştırmayı hedefler. Öğrenci Aşamasında, öğrenciler birbirleriyle etkileşime girerek öğrenme sürecine devam ederler. Daha bilgili bir öğrenciden öğrenme fırsatı bulan diğer öğrenciler, mevcut çözümlerini geliştirir.

Rao vd. (2011), TLO algoritmasını, öğretmenin öğrenciler üzerindeki etkisinden esinlenerek, mekanik tasarım problemlerinin çözümü için önermiştir. Algoritma, öğretmen ve öğrenciler arasındaki bilgi aktarımını “Öğretmen Aşaması” ve “Öğrenci Aşaması” olarak iki aşamada modellemektedir. Çeşitli kısıtlı test fonksiyonları ve mekanik tasarım problemleri üzerinde yapılan deneylerde TLO’nun, PSO, GA ve ABC gibi diğer yöntemlere kıyasla daha az işlem maliyetiyle daha iyi sonuçlar verdiği

görülmüştür. TLO, algoritma parametre ayarlarına ihtiyaç duymadığı için uygulamada basit ve etkili bir yöntem olarak vurgulanmıştır.

TLO'nun ana esin kaynağı, bir öğretmenin öğrencilerine bilgi aktarımı sırasında ortaya çıkan öğrenme etkisi ve öğrencilerin grup etkileşimlerinden kazandıkları öğrenme yeteneğidir. Algoritma, optimizasyon süreçlerinde herhangi bir parametre ayarı gerektirmeden çalışmasıyla öne çıkar ve hem küresel hem de yerel optimum çözümleri keşfetmede etkili bir araçtır.

Şekil 3.12’de TLO’nun işleyiş diyagramı görülmektedir. Buna göre TLO’nun başlangıç popülasyonu oluşumu, gelişim süreci ve sonlandırma kriteri adımlarına ilişkin açıklamalara aşağıda yer verilmiştir.



Şekil 3.12 TLO’nun işleyiş diyagramı

Adım 1: Başlangıç Popülasyonu

Algoritma, belirli bir popülasyon büyüklüğüne göre rastgele çözümler oluşturur. Her çözüm, bir öğrenciyi temsil eder. Her bireyin uygunluk değeri hesaplanır ve popülasyonun en iyi çözümü belirlenir.

Adım 2: Gelişim

- ***Öğretim Aşaması***

Bu aşamada, popülasyondaki en iyi çözüm, bir öğretmen gibi hareket ederek bireylerin performansını artırmayı hedefler. Öğretim faktörü, rastgele olarak 1 veya 2 olarak atanır. Bu, öğretmenin öğrenciler üzerindeki etkisinin derecesini belirler. Popülasyondaki tüm bireylerin çözüm vektörleri hesaplanır. Popülasyonun ortalama çözümü, öğretmenin etkisini hesaba katarak bireylerin çözümlerini güncellemek için kullanılır. Yeni değerler sınırların dışına çıkarsa, düzeltilir. Eğer yeni çözüm, mevcut çözümden daha iyi ise birey güncellenir.

- ***Öğrenme Aşaması***

Bu aşamada, bireyler rastgele seçilen bir partnerden (diğer bir birey) öğrenir. Öğrenci ile partner arasındaki uygunluk değerine göre bilgi alışverişi yapar. Eğer mevcut birey daha iyiye, partner bireyden öğrenme gerçekleşir. Aksi takdirde, birey partnerden etkilenir. Yeni çözüm, mevcut çözümden daha iyi ise birey güncellenir.

Adım 3: Sonlandırma Kriteri

Algoritma, belirtilen iterasyon sayısına kadar yukarıdaki adımları tekrarlar. Her iterasyonda bireylerin çözümleri güncellenir ve en iyi çözüm belirlenir. İstenilen iterasyon sayısına ulaşıldığında algoritma sonlanır ve en iyi çözüm döndürülür.

TLO algoritması işletilirken gelişim adımında söz edilen ayarlanabilir parametreler; popülasyon sayısı (pop_size) ve iterasyon sayısı (epoch) olarak tanımlanmaktadır.

4. ÇOK KRİTERLİ KARAR VERME YÖNTEMLERİ

Çok Kriterli Karar Verme (ÇKKV) yöntemleri, karar süreçlerinde birden fazla kriterin dikkate alınmasını sağlayan sistematik yaklaşımlar sunmaktadır. Bu yöntemler, alternatifler arasından en uygun seçeneğin belirlenmesine yardımcı olarak, karar vericilere daha rasyonel, nesnel ve tutarlı seçimler yapma imkânı tanımaktadır. ÇKKV yöntemleri, özellikle çok sayıda kriterin etkilerinin değerlendirilmesi gereken durumlarda önemli bir rol oynamaktadır (Sel ve Demir, 2020).

Bu tez kapsamında, karar verme sürecinde ilk olarak IDOCRIW yöntemi (Özçalıcı, 2021) kullanılarak kriter ağırlıkları belirlenecek, ardından alternatiflerin sıralanması için COPRAS, EDAS ve MABAC yöntemleri uygulanacaktır. Son aşamada ise sıralama yöntemlerinin karşılaştırılması ve nihai kararın belirlenmesi için Copeland yöntemi kullanılacaktır. Kullanılan yöntemlerin temel dayanakları, normalizasyon tekniklerine dayanmaktadır. Toplama dayalı normalleştirme içeren COPRAS yöntemi, alternatiflerin pozitif ve negatif kriterlere dayalı olarak değerlendirilmesini sağlayarak, performanslarının ağırlıklı toplamlar üzerinden hesaplanmasını mümkün kılmaktadır. Orana dayalı normalleştirme içeren EDAS yöntemi, alternatiflerin ortalama çözüme olan uzaklıklarını dikkate alarak sıralama yapmakta ve her alternatifin ortalama çözümden pozitif ve negatif sapmalarını göz önünde bulundurarak değerlendirilmesini sağlamaktadır. Minimum ve maksimum değerlere dayalı normalleştirme içeren MABAC yöntemi ise alternatiflerin minimum ve maksimum değerler arasındaki konumlarını belirleyerek sıralama yapmaktadır. Alternatiflerin sınır değerlerle olan ilişkisini analiz eden bu yöntem, karar sürecinin daha kapsamlı ve sağlıklı bir biçimde yürütülmesine katkı sunmaktadır. Son aşamada kullanılan Copeland yöntemi ise farklı sıralama yöntemlerinin sonuçlarını bütünleştirerek daha güvenilir bir son sıralama elde etmeyi amaçlamaktadır. Bu yöntemde, alternatifler diğer alternatiflerle birebir karşılaştırılarak galibiyet ve mağlubiyet sayıları belirlenmekte ve en yüksek skora sahip alternatif en iyi seçenek olarak kabul edilmektedir.

Bu tez çalışmasının temel amacı, ÇKKV yöntemlerini kullanarak, performans metrikleri ile belirlenen kriterlere göre on iki AI optimizasyon algoritmasını sistematik ve nesnel bir

biçimde değerlendirmek ve bu algoritmaları alternatifler olarak sıralamaktır. ÇKKV yöntemleri, özellikle birden fazla kriterin dikkate alınması gereken karar problemlerinde karar vericilere büyük kolaylık sağlamaktadır. Tez kapsamında, farklı kriterlerin objektif olarak ağırlıklandırılması, alternatiflerin belirlenen kriterlere göre performanslarının karşılaştırılması ve farklı normalizasyon tekniklerinin sıralama üzerindeki etkilerinin incelenmesi amaçlanmaktadır. Aynı zamanda, karar süreçlerinde belirsizliği azaltarak daha güvenilir ve sistematik sonuçlar elde edilmesi hedeflenmektedir. Çalışma, farklı ÇKKV yöntemlerinin karar süreçlerine nasıl katkı sağladığını analiz ederek bu yöntemlerin sonuçlarının ne derece tutarlı olduğunu ortaya koymayı amaçlamaktadır. Bu bağlamda, ÇKKV yöntemlerinin karar vericilere daha güvenilir ve rasyonel sonuçlar sunarak alternatifler arasındaki en uygun seçeneği belirleme sürecini optimize ettiği gösterilecektir (Alinezhad ve Khalili, 2019).

Yöntemlerin başlangıç noktası, karar matrisidir. Bu $n \times m$ boyutlu D matrisinin matrisi $i = 1, 2, \dots, n$ ve $j = 1, 2, \dots, m$ olmak üzere

$$D = [d_{ij}]_{n \times m} = \begin{bmatrix} d_{11} & d_{12} & \dots & d_{1m} \\ d_{21} & d_{22} & \dots & d_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ d_{n1} & d_{n2} & \dots & d_{nm} \end{bmatrix} \quad (4.1)$$

biçiminde tanımlanır. Burada d_{ij} , i -inci alternatifin j -inci karar vericiye ilişkin değeridir. n , alternatif sayısını ve m , karar verici sayısını göstermektedir (Ulu Metin, 2024).

4.1 Ağırlıklandırma

Kriter ağırlıklandırma işlemi, her bir ölçütün karar sürecine olan katkısını nicel olarak ifade etmeyi amaçlar. Bu kapsamda, veri tabanlı objektiflik ile kriterler arası etkileşimleri bir araya getiren bütünleşmiş bir yaklaşım sunarak daha dengeli ve hassas ağırlıkların elde edilmesini sağlamaktadır.

4.1.1 IDOCRIW

IDOCRIW (Integrated Determination of Objective CRiteria Weights) yöntemi, karar verme sürecinde kriterlerin ağırlıklarını belirlemek için Entropi ve CILOS (Criterion Impact Loss) yöntemlerini entegre ederek daha hassas sonuçlar elde etmeye yaramaktadır. Zavadskas ve Podvezko (2016) çalışmalarında, ÇKKV süreçlerinde kriter ağırlıklarının belirlenmesinin önemini vurgulamaktadır. Çalışmada, genellikle uzman görüşleriyle belirlenen öznel ağırlıklar yerine, verinin yapısını dikkate alan nesnel ağırlık belirleme yöntemlerine odaklanılmıştır (Çiftaslan ve Rençber, 2022). Bu bağlamda, yaygın olarak kullanılan Entropi yöntemi ile CILOS yöntemi birleştirilerek IDOCRIW yöntemi önerilmiştir. IDOCRIW, Entropi yönteminin kriterler arasındaki baskınlık ilişkisini ve CILOS'un kriterlerin önem kaybını dikkate alarak daha dengeli ağırlıklar üretmesini sağlamaktadır. Bu yöntemin uygulanma süreci adımları aşağıdaki biçimde açıklanabilir.

Adım 1: Normalleştirilmiş Karar Matrisi

İlk adımda, karar matrisindeki veriler normalize edilir. Bu işlem, her bir kriter için alternatif değerlerinin toplamına bölünerek gerçekleştirilir. Normalizasyon işlemi

$$v_{ij} = \frac{d_{ij}}{\sum_{i=1}^n d} , i = 1, 2, \dots, n , j = 1, 2, \dots, m \quad (4.2)$$

ve

$$V = [v_{ij}]_{n \times m} = \begin{bmatrix} v_{11} & v_{12} & \dots & v_{1m} \\ v_{21} & v_{22} & \dots & v_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ v_{n1} & v_{n2} & \dots & v_{nm} \end{bmatrix} \quad (4.3)$$

formülüyle ifade edilir. Burada v_{ij} , i -inci alternatife j -inci kritere göre normalize edilmiş değerini göstermektedir. Bu işlem, farklı ölçeklerdeki verilerin karşılaştırılabilir hale gelmesini sağlamaktadır.

Adım 2: Entropi Derecesi

Entropi derecesi, her bir kriterin karar sürecindeki belirsizlik seviyesini ölçmektedir.

Entropi değeri

$$E_j = -\frac{1}{\ln n} \sum_{i=1}^n v \times \ln(v), \quad i = 1, 2, \dots, n, \quad j = 1, 2, \dots, m \quad (4.4)$$

biçiminde hesaplanır. Burada E_j , j -inci kriterin entropisini, n ise alternatif sayısını temsil eder. Entropi değeri düşük olan kriterler, karar verme sürecinde daha belirleyici kabul edilmektedir.

Adım 3: Entropi Ağırlıkları

Entropi derecelerinden yola çıkarak kriterlerin ağırlıkları belirlenir. Öncelikle her bir kriter için sapma oranı

$$r_j = 1 - E_j, \quad j = 1, 2, \dots, m \quad (4.5)$$

eşitliğiyle hesaplanır. Daha sonra bu sapma oranları kullanılarak entropi ağırlıkları hesaplanır.

$$w_j = \frac{r_j}{\sum_{j=1}^m r_j}, \quad j = 1, 2, \dots, m \quad (4.6)$$

Bu adımda elde edilen ağırlıklar, kriterlerin karar üzerindeki göreceli önemini temsil eder.

Adım 4: Dönüştürülmüş Matris

Negatif kriterleri pozitif hale getirmek için kare matris oluşturulur. Bu matris

$$v_{ij}^* = \frac{\min(d_{ij})}{d_{ij}}, \quad i = 1, 2, \dots, n, \quad j = 1, 2, \dots, m \quad (4.7)$$

biçiminde hesaplanır. Bu işlem, karar matrisindeki minimum değerlerle karşılaştırarak, kriterlerin olumsuz etkilerini normalize etmeye yaramaktadır.

Adım 5: Göreceli Etki Kaybı Matrisi

Bu adımda, her bir kriterin diğer kriterler üzerindeki göreceli etkisini ölçen matris oluşturulur. Göreceli etki kaybı matrisi

$$p_{ij} = \frac{a_{jj} - a_{ij}}{a_{jj}}, \quad i = 1, 2, \dots, n, \quad j = 1, 2, \dots, m \quad (4.8)$$

ve

$$a_{jj} = \max(d_{ij}), \quad i = 1, 2, \dots, n, \quad j = 1, 2, \dots, m \quad (4.9)$$

eşitliğiyle hesaplanır. Burada a_{jj} , kriterin maksimum değerini, a_{ij} ise diğer alternatiflerin aynı kritere ilişkin değerlerini ifade eder. Bu matris, her kriter seçildiğinde diğer kriterlerde oluşan kaybı göstermektedir.

Adım 6: Ağırlık Sistemi Matrisi

Ağırlık sistemi matrisi, kriterlerin birbirleri üzerindeki göreceli etkilerini topluca gösterir. Ağırlık sistem matrisi

$$F_{n \times m} = \begin{bmatrix} \sum P_{i1} & P_{12} & \dots & P_{1m} \\ P_{21} & \sum P_{i2} & \dots & P_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ P_{m1} & P_{m2} & \dots & \sum P_{im} \end{bmatrix} \quad (4.10)$$

biçiminde ifade edilir. Bu matris, göreceli etki kaybı değerlerinden türetilerek kriterler arası ilişkileri analiz etmeye yardımcı olmaktadır.

Adım 7: Kriter Etki Kaybı Ağırlıkları

Kriter etki kaybı ağırlıkları, ağırlık sistemi matrisinden elde edilir. Bu adımda aşağıdaki lineer denklem çözülür.

$$F \times q^T = 0 \quad (4.11)$$

Burada q , kriter ağırlıklarını temsil eden vektördür. Bu denklem sisteminin çözülmesiyle, her bir kriter için göreceli etki ağırlıkları belirlenir.

Adım 8: Birleşik Ağırlıkların Hesaplanması

Entropi ve CILOS yöntemlerinden elde edilen ağırlıklar birleştirilerek nihai ağırlıklar hesaplanır. Bu hesaplama

$$x_j = \frac{q_j \times w_j}{\sum_{i=1}^n (q_i \times w_i)}, \quad i = 1, 2, \dots, n \quad (4.12)$$

biçiminde yapılır. Burada x_j , j -inci kriterin birleşik ağırlığını gösterir. Bu birleşik ağırlıklar, her iki yöntemin sonuçlarını harmanlayarak daha güvenilir bir değerlendirme sağlar.

Adım 9: Nihai Sıralama

Elde edilen birleşik ağırlıklar büyükten küçüğe sıralanarak kriterler önceliklendirilir. En yüksek ağırlığa sahip kriter, karar verme sürecinde en önemli kriter olarak kabul edilmektedir.

Bu süreç sayesinde IDOCRIW yöntemi hem entropi tabanlı objektif bilgiler hem de kriterler arası etkileşimler dikkate almasından dolayı karar verme süreçlerinde daha dengeli ve hassas sonuçlar sunmaktadır.

4.2 Sıralama

Sıralama yöntemleri, her bir alternatifin çok boyutlu performansını bütüncül bir bakış açısıyla analiz ederek, karar vericiye sistematik bir tercih sıralaması sunar. Kriter ağırlıkları belirlendikten sonra, alternatiflerin performanslarının değerlendirilerek sıralanmasını sağlamaktadır.

4.2.1 COPRAS

COPRAS (COmplex PROportional ASsessment) yöntemi, karar verme süreçlerinde alternatiflerin değerlendirilmesi için kullanılan ÇKKV yöntemlerinden biridir. Bu yöntem, maksimum ve minimum endeks değerlerini değerlendirerek alternatifleri sıralamaktadır. Zavadskas vd. (2007), COPRAS yöntemini tanıtarak, ÇKKV problemlerinde alternatiflerin değerlendirilmesi için oransal değerlendirme temelli bir yaklaşım sunmuştur. Çalışmada, COPRAS yönteminin, pozitif ve negatif kriterleri dikkate alarak alternatifleri sıralamada etkili olduğu gösterilmiştir. Yöntem, kriter ağırlıklarını ve alternatiflerin performanslarını orantılı olarak değerlendirerek karar vericilere daha şeffaf bir analiz imkânı sağlamaktadır. Bu yöntemin uygulanma sürecindeki adımlar aşağıda yer almaktadır.

Adım 1: Normalleştirilmiş Karar Matrisi

İlk adımda karar matrisi normalize edilir. Bu işlem, her bir kriterin değerini o kriterin toplamına bölerek yapılır. Normalizasyon işlemi

$$v_{ij} = \frac{d_{ij}}{\sum_{j=1}^m d_{ij}}, \quad i = 1, 2, \dots, n, \quad j = 1, 2, \dots, m \quad (4.13)$$

ve

$$V = [v_{ij}]_{n \times m} = \begin{bmatrix} v_{11} & v_{12} & \dots & v_{1m} \\ v_{21} & v_{22} & \dots & v_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ v_{n1} & v_{n2} & \dots & v_{nm} \end{bmatrix} \quad (4.14)$$

formülüyle ifade edilir. Burada v_{ij} , i -inci alternatifin j -inci kritere göre normalize edilmiş değerini göstermektedir. Bu işlem, farklı ölçeklerdeki verilerin karşılaştırılabilir hale gelmesini sağlamaktadır.

Adım 2: Ağırlıklı Normalleştirilmiş Karar Matrisi

Normalleştirilmiş değerler, kriter ağırlıkları ile çarpılarak ağırlıklı normalleştirilmiş karar matrisi elde edilir. Bu işlem

$$v_{ij}^* = v_{ij} \times w_j, \quad i = 1, 2, \dots, n, \quad j = 1, 2, \dots, m \quad (4.15)$$

formülüyle yapılır. Burada w_j , j -inci kriterin ağırlığını temsil eder. Bu adımda kriterlerin göreceli önem dereceleri dikkate alınmaktadır.

Adım 3: Maksimizasyon ve Minimizasyon Endeksleri

Her bir alternatif için maksimizasyon (S_i^+) ve minimizasyon (S_i^-) endeksleri hesaplanır. Bu endeksler

$$S_i^+ = \sum_{j=1}^g v_{ij}^*, \quad i = 1, 2, \dots, n, \quad j = 1, 2, \dots, m \quad (4.16)$$

ve

$$S_i^- = \sum_{j=g+1}^n v_{ij}^*, \quad i = 1, 2, \dots, n, \quad j = 1, 2, \dots, m \quad (4.17)$$

formülleriyle belirlenir. Burada g , pozitif (maksimize edilmesi gereken) kriterlerin sayısını, $n - g$ ise negatif (minimize edilmesi gereken) kriterlerin sayısını ifade etmektedir. Bu endeksler, alternatiflerin güçlü ve zayıf yönlerini ortaya koyar.

Adım 4: Göreli Önem Değeri

Her alternatifin göreli önem değeri (Q_i) hesaplanır. Bu değer, alternatiflerin genel performansını temsil eder ve şu formüllerden

$$Q_i = S^+ + \frac{\min(S^-)}{\sum_{i=1}^m \min(S^-)} \times \frac{\sum_{i=1}^m S^-}{S^-}, \quad i = 1, 2, \dots, n \quad (4.18)$$

veya alternatif olarak;

$$Q_i = S^+ + \frac{\sum_{i=1}^m S^-}{S^-} \times \frac{1}{\sum_{i=1}^m \frac{1}{S^-}}, \quad i = 1, 2, \dots, n \quad (4.19)$$

biriyle hesaplanabilir. Bu adımda hesaplanan değerler, alternatiflerin göreceli olarak ne kadar avantajlı olduğunu göstermektedir.

Adım 5: Nihai Sıralama

Son adımda, alternatifler göreceli önem değerlerine göre sıralanır. En yüksek Q_i değerine sahip alternatif en iyi seçenek olarak kabul edilmektedir. Alternatifler büyükten küçüğe doğru sıralanarak nihai karar verilir.

Bu süreç sayesinde COPRAS yöntemi, karar verme süreçlerinde hem pozitif hem de negatif kriterleri dikkate alarak daha dengeli ve kapsamlı bir analiz sunmaktadır.

4.2.2 EDAS

EDAS (Evaluation based on Distance from Average Solution) yöntemi, alternatiflerin performanslarını ortalama çözüme olan mesafelerine göre değerlendiren ÇKKV yöntemlerinden biridir. Bu yöntem, alternatiflerin ortalama çözüme olan pozitif ve negatif mesafelerini hesaplayarak en iyi seçeneği belirler. Keshavarz Ghorabae vd. (2015), EDAS yöntemini tanıtarak, çok kriterli envanter sınıflandırma problemleri için yeni bir karar verme yaklaşımı sunmuştur. Çalışmada, geleneksel sınırları aşmak amacıyla, pozitif ve negatif ortalama çözüm mesafeleri kullanılarak alternatiflerin değerlendirilmesi önerilmiştir. Envanter yönetimi, tedarikçi seçimi ve proje değerlendirme gibi alanlarda uygulanabilirliği vurgulanan yöntem, farklı kriter ağırlıkları altında tutarlı performans sergilemiştir. Bu yöntemin uygulanma sürecindeki adımlar aşağıda yer almaktadır.

Adım 1: Ortalama Çözümün Hesaplanması

Her bir kriter için ortalama değer

$$AV_j = \frac{\sum_{i=1}^m d_{ij}}{m}, i = 1, 2, \dots, n, j = 1, 2, \dots, m \quad (4.20)$$

formülüyle hesaplanır. Bu değer, alternatiflerin o kritere göre genel performans düzeyini gösterir.

Adım 2: Pozitif ve Negatif Mesafelerin Hesaplanması

Her bir alternatif için ortalama çözüme göre pozitif (PDA_{ij}) ve negatif (NDA_{ij}) mesafeler belirlenir. Minimize edilmesi gereken kriterler için pozitif mesafe

$$PDA_{ij} = \frac{\max(0, d_{ij} - AV_j)}{AV_j}, i = 1, 2, \dots, n, j = 1, 2, \dots, m \quad (4.21)$$

minimize edilmesi gereken kriterler için negatif mesafe ise

$$NDA_{ij} = \frac{\max(0, AV_j - d_{ij})}{AV_j}, i = 1, 2, \dots, n, j = 1, 2, \dots, m \quad (4.22)$$

formülleriyle hesaplanır. Bu hesaplamalar, alternatiflerin ortalama performansa göre ne kadar iyi veya kötü olduğunu göstermektedir.

Adım 3: Ağırlıklı Pozitif ve Negatif Mesafelerin Hesaplanması

Her bir kriter için belirlenen ağırlıklar (w_j) kullanılarak

$$SP_i = \sum_{j=1}^n PDA_{ij} \times w_j, i = 1, 2, \dots, n, j = 1, 2, \dots, m \quad (4.23)$$

ve

$$SN_i = \sum_{j=1}^n NDA_{ij} \times w_j, i = 1, 2, \dots, n, j = 1, 2, \dots, m \quad (4.24)$$

Formülleriyle pozitif ve negatif mesafeler ağırlıklandırılır. Burada SP_i , i -inci alternatifin ağırlıklı pozitif mesafesini, SN_i ise ağırlıklı negatif mesafesini gösterir.

Adım 4: Normalleştirilmiş Pozitif ve Negatif Mesafeler

Ağırlıklı pozitif ve negatif mesafeler

$$NSP_i = \frac{SP_i}{\max(SP_i)} , i = 1, 2, \dots, n \quad (4.25)$$

ve

$$NSN_i = \frac{SN_i}{\max(SN_i)} , i = 1, 2, \dots, n \quad (4.26)$$

formülleriyle normalize edilir. Bu adım, farklı ölçeklerdeki değerleri karşılaştırılabilir hale getirmektedir.

Adım 5: Değerlendirme Puanının Hesaplanması

Her bir alternatif için değerlendirme puanı

$$AS_i = \frac{NSP_i + NSN_i}{2} , i = 1, 2, \dots, n \quad (4.27)$$

formülüyle hesaplanır. Bu puan, alternatifin genel performansını göstermektedir.

Adım 6: Nihai Sıralama

Son adımda, alternatifler değerlendirme puanlarına göre büyükten küçüğe sıralanmaktadır. En yüksek AS_i değerine sahip alternatif, en iyi seçenek olarak kabul edilmektedir.

Bu süreç sayesinde EDAS yöntemi hem pozitif hem de negatif kriterleri dikkate alarak alternatifler arasında dengeli bir değerlendirme yapılmasını sağlamaktadır.

4.2.3 MABAC

MABAC (Multi-Attributive Border Approximation area Comparison) yöntemi, alternatiflerin sınır yaklaşık alanına olan mesafelerine göre değerlendirilmesini sağlayan ÇKKV yöntemlerinden biridir. Bu yöntem, alternatiflerin sınır yaklaşık alanına olan mesafelerini hesaplayarak en iyi alternatifi belirler. Pamučar ve Ćirović (2015), MABAC yöntemi ile karar alternatiflerinin sınır yaklaşık alanına olan mesafelerini hesaplayarak sıralama yapmaktadır. Duyarlılık analizleri, kriter ağırlıkları ve ölçek değişimlerine karşı yöntemin tutarlılığını test edilmiştir. Bu yöntemin uygulanma sürecindeki adımları aşağıda yer almaktadır.

Adım 1: Normalleştirilmiş Karar Matrisi

İlk adımda karar matrisi normalize edilir. Pozitif ve negatif kriterler için farklı normalizasyon formülleri kullanılır. Pozitif kriterler için

$$v_{ij} = \frac{d_{ij} - d_i^-}{d_i^+ - d_i^-}, \quad i = 1, 2, \dots, n, \quad j = 1, 2, \dots, m \quad (4.28)$$

formülü kullanılmaktadır. Negatif kriterler için ise

$$v_{ij} = \frac{d_{ij} - d_i^+}{d_i^- - d_i^+}, \quad i = 1, 2, \dots, n, \quad j = 1, 2, \dots, m \quad (4.29)$$

formülüyle hesaplanmaktadır. Burada v_{ij} , i -inci alternatifin j -inci kritere göre normalize edilmiş değerini, d_i^+ maksimum değeri, d_i^- ise minimum değeri göstermektedir.

Adım 2: Ağırlıklı Normalleştirilmiş Karar Matrisi

Normalleştirilmiş değerler, kriterlerin ağırlıkları (w_j) ile çarpılarak ağırlıklı normalleştirilmiş karar matrisi

$$v_{ij}^* = w_j + v_{ij} \times w_j, \quad i = 1, 2, \dots, n, \quad j = 1, 2, \dots, m \quad (4.30)$$

formülüyle elde edilmektedir. Bu adımda, kriterlerin göreceli önem dereceleri dikkate alınmaktadır.

Adım 3: Yaklaşık Sınır Alan Matrisi

Sınır yaklaşık alan matrisi, her bir kriter için geometrik ortalama kullanılarak

$$g_j = (\prod_{i=1}^n v^*)^{1/m}, j = 1, 2, \dots, m \quad (4.31)$$

formülüyle hesaplanmaktadır. Bu matris, alternatiflerin sınır değerlere olan ortalama mesafelerini göstermektedir.

Adım 4: Sınır Yaklaşık Alanına Olan Mesafe

Her bir alternatif için, sınır yaklaşık alanına olan mesafe

$$q_{ij} = v_{ij}^* - g_j, i = 1, 2, \dots, n, j = 1, 2, \dots, m \quad (4.32)$$

formülüyle hesaplanmaktadır. Bu değer, alternatifin belirli bir kriterde sınır değere göre ne kadar sapma gösterdiğini belirtmektedir.

Adım 5: Toplam Mesafelerin Hesaplanması

Alternatiflerin tüm kriterler için toplam mesafeleri

$$S_i = \sum_{j=1}^m q_{ij}, i = 1, 2, \dots, n, j = 1, 2, \dots, m \quad (4.33)$$

biçiminde hesaplanır. Bu toplam mesafe, alternatifin genel performansını sınır değerlere göre ölçmektedir.

Adım 6: Nihai Sıralama

Son adımda, toplam mesafeler büyükten küçüğe sıralanır. En yüksek toplam mesafeye sahip alternatif en iyi seçenek olarak kabul edilir.

MABAC yöntemi, karar verme süreçlerinde alternatiflerin sınır değerlere olan mesafelerini dikkate alarak daha objektif ve dengeli bir değerlendirme sağlar. Bu yöntem özellikle karar kriterlerinin birbirinden bağımsız olduğu durumlarda etkili bir biçimde kullanılmaktadır.

4.3 Genel Sıralama

Yöntemler arası olası tutarsızlıkları en aza indirerek karar sürecinin güvenilirliğini artırılması hedeflenmektedir. Bu bağlamda kullanılan yöntem, alternatiflerin ikili karşılaştırmalarına dayalı olarak kapsamlı bir sentez sunmakta ve nihai sıralamayı bütüncül bir yaklaşımla belirlemektedir.

4.3.1 Copeland Yöntemi

Copeland yöntemi, alternatiflerin birbirleriyle yapılan ikili karşılaştırmalara dayalı olarak sıralandığı bir yöntemdir. Bu yöntemde, her alternatif diğer alternatiflerle kıyaslanarak, galibiyet veya mağlubiyet durumlarına göre puanlar verilir. Saari ve Merlin (1996), Copeland yöntemi üzerine yaptıkları çalışmalarında, seçim sonuçlarının adayların dahil edilmesi veya çıkarılmasıyla nasıl değişebileceğini analiz etmişlerdir. Çalışmada, Copeland yöntemi ile konumsal oylama yöntemleri arasındaki ilişkiler incelenmiş ve yöntemin güçlü ve zayıf yönleri geometrik bir yaklaşımla ortaya konmuştur (Lestari vd., 2018). Ayrıca, Copeland sıralamalarının, aday sayısı arttıkça daha fazla belirsizlik içerebileceği ve konumsal oylama yöntemleriyle uyumsuzluk gösterebileceği vurgulanmıştır. Aşağıda yer alan adımlarda Copeland yönteminin nasıl çalıştığı detaylı olarak açıklanmıştır.

Adım 1: İkili Karşılaştırma Matrisi Oluşturulması

Copeland yönteminin temel mantığı, her alternatifin diğer tüm alternatiflerle karşılaştırılmasıdır. Bu karşılaştırmalar sonucunda bir ikili karşılaştırma matrisi

$$f_k(i, j) = \begin{cases} 1, & r_k(D_i) < r_k(D_j) \text{ ve } i \neq j \\ 0, & r_k(D_i) > r_k(D_j) \text{ ve } i \neq j, i = 1, 2, \dots, n, j = 1, 2, \dots, m \\ \emptyset, & r_k(D_i) = r_k(D_j) \text{ ve } i = j \end{cases} \quad (4.34)$$

eşitliğiyle oluşturulur. Burada D_i ve D_j karşılaştırılan iki alternatifi temsil ederken, $f_k(i, j)$, D_i ve D_j arasındaki karşılaştırmaların sonucunu gösterir. k ise karar verici indeksini ifade eder. $r_k(D_i)$ ifadesi D_i alternatifinin k 'ıncı karar vermeye göre sıralamadaki yerini göstermektedir. Bu matris, alternatifler arasında hangi alternatifin daha üstün olduğunu belirlemek için kullanılır.

Adım 2: Toplam Oyların Hesaplanması

İkili karşılaştırmalar sonucunda her alternatifin diğer alternatiflere karşı aldığı toplam oy sayısı

$$S(i, j) = \sum_{k=1}^m f_k(i, j) \text{ ve } i \neq j, i = 1, 2, \dots, n, j = 1, 2, \dots, m \quad (4.35)$$

formülüyle hesaplanır. Burada $S(i, j)$, D_i 'nin D_j 'ye karşı aldığı toplam oy sayısını ifade ederken, m karar verici sayısını gösterir. Bu adım, hangi alternatifin daha fazla üstünlük elde ettiğini belirlemek için önemlidir.

Adım 3: Galibiyet, Beraberlik ve Yenilgi Durumlarının Belirlenmesi

Alternatiflerin birbirlerine karşı olan galibiyet, beraberlik ve yenilgi durumları belirlenir. Bu durumlar

$$G(i, j) = \begin{cases} 1, & S(i, j) > (m - S(i, j)) \text{ ve } i \neq j \\ 0.5, & S(i, j) = (m - S(i, j)) \text{ ve } i \neq j, i = 1, 2, \dots, n, j = 1, 2, \dots, m \\ -1, & S(i, j) < (m - S(i, j)) \text{ ve } i \neq j \end{cases} \quad (4.36)$$

formülüyle hesaplanır.

Adım 4: Galibiyet ve Yenilgi Puanlarının Hesaplanması

Galibiyet puanı

$$GP_i = \sum_{j=1}^n G(i, j) , i = 1, 2, \dots, n , j = 1, 2, \dots, m \quad (4.37)$$

$G(i, j) > 0$ olması durumunda, eşitlik (4.37) ile hesaplanır. Bu formül, D_i 'nin kazandığı tüm karşılaştırmalardan aldığı puanları toplar. Burada n alternatiflerin toplam sayısını gösterir. Yenilgi puanı ise

$$YP_i = \sum_{j=1}^n G(i, j) , i = 1, 2, \dots, n , j = 1, 2, \dots, m \quad (4.38)$$

$G(i, j) < 0$ olması durumunda, eşitlik (4.38) ile hesaplanır. Bu da D_i 'nin kaybettiği karşılaştırmalardan aldığı (negatif) puanların toplamıdır.

Adım 5: Copeland Puanının Hesaplanması

Copeland puanı, alternatifin elde ettiği galibiyet ve yenilgi puanlarının toplamıyla

$$CP_i = GP_i + YP_i , i = 1, 2, \dots, n \quad (4.39)$$

formülüyle hesaplanmaktadır. Burada CP_i , D_i 'nin nihai Copeland puanını gösterir. Pozitif bir CP_i değeri, alternatifin genel olarak daha iyi performans gösterdiğini; negatif değer ise daha zayıf olduğunu ifade etmektedir.

Adım 6: Alternatiflerin Sıralanması

Son olarak, alternatifler Copeland puanlarına göre büyükten küçüğe doğru sıralanır. En yüksek puana sahip alternatif en iyi tercih olarak kabul edilir. Eğer iki alternatif aynı Copeland puanına sahipse, eşitliği bozmak için ikincil kriterler veya daha detaylı analizler yapılabilir. Ayrıca, karar verici tercihlerine göre ek değerlendirmeler de yapılabilir.

Bu yöntem, karar verme süreçlerinde objektif, sistematik ve adil sonuçlar sağlaması açısından oldukça etkilidir. Alternatiflerin birbirleriyle doğrudan karşılaştırılması sayesinde güçlü bir sıralama yöntemi olarak öne çıkar.

5. UYGULAMA

Tez çalışmasının uygulama bölümünde doğrusal olmayan regresyon modelleri üzerinde AI tabanlı optimizasyon algoritmalarının performanslarının incelenmesine ilişkin deneysel çalışmalar içeren veri setleri ve bu verilere ilişkin doğrusal olmayan regresyon modellerinde parametre tahminlerini içeren sonuçlar detaylı bir biçimde sunulmaktadır. Çalışmada kullanılan doğrusal olmayan regresyon modelleri, iki, üç ve dört parametrelili olmak üzere sırasıyla BoxBOD (Biyokimyasal), Meyer-1 ve İkili Kompartman modelleri olarak belirlenmiştir. Her bir model, *Python 3.12.7* ortamında uygulanmış, AI optimizasyon algoritmalarının yürütülmesinde *MEALPY 3.0.1*, performans metriklerinin hesaplanmasında *PerMetrics 2.0.0* ve ÇKKV yöntemleri ile ağırlıklandırma yöntemlerinin kullanılmasında *PyMCDM 1.3.0* ile *PyDecision 4.7.1* kütüphaneleri tercih edilmiştir. AI optimizasyon algoritmalarının ayarlanabilir parametrelerinin belirlenmesi aşaması her AI algoritması için 10 kez tekrarlanmış, minimum En Küçük Kareler (EKK) değerine sahip ayarlanabilir parametre vektörü belirlenerek optimizasyon süreçlerinde kullanılmıştır. Ayrıca, her algoritma 100 bireyden oluşan bir popülasyon ile 100 iterasyonda çalıştırılmış ve bu süreç 50 defa tekrarlanarak istatistiksel olarak güvenilir sonuçlar elde edilmesi hedeflenmiştir. Çalışma kapsamında, algoritmaların yalnızca optimizasyon yeteneklerini değil, aynı zamanda hesaplama maliyetlerini ve istatistiksel olarak anlamlılıklarını da değerlendirebilmek amacıyla çeşitli performans metrikleri dikkate alınmıştır.

Kullanılan Python kütüphanelerinden olan *MEALPY*, Nguyen ve Mirjalili (2023) tarafından, açık kaynaklı bir kütüphane olarak derlemiştir. *MEALPY*, doğadan ilham alan optimizasyon algoritmalarını geniş bir yelpazede sunarak kullanıcı dostu bir yapı sunmaktadır. 200'den fazla klasik ve yenilikçi AI optimizasyon algoritmasını içeren *MEALPY*, basit arayüzü ve minimum bağımlılıkları ile öne çıkmaktadır. Kütüphane, makine öğrenmesi modellerinin hiper-parametre optimizasyonu gibi uygulamalarda kullanılabilir ve paralel hesaplama desteği ile diğer optimizasyon kütüphanelerine kıyasla daha esnek ve verimli bir yapı sunmaktadır.

Çizelge 5.1 AI optimizasyon algoritmalarının ayarlanabilir parametre değerleri

<i>Algoritmalar</i>	<i>Ayarlanabilir Parametreler</i>
ABC	epoch: {100}, pop_size: {100}, n_limits: {15, 25, 45, 75}
ACOR	poch: {100}, pop_size: {100}, sample_count: {20, 50, 100}, intent_factor: {0.3, 0.5, 0.7}, zeta: {1, 2, 3}
BA	epoch: {100}, pop_size: {100}, loudness: {0.85, 0.90, 0.95}, pulse_rate: {0.025, 0.1}, pf_min: {0}, pf_max: {10}
CSA	epoch: {100}, pop_size: {100}, p_a: {0.1, 0.3, 0.5}
DE	epoch: {100}, pop_size: {100}, wf: {-0.1, 0.1, 0.4}, cr: {0.7, 0.9}, strategy: {0, 4}
FPA	epoch: {100}, pop_size: {100}, p_s: {0.7, 0.8, 0.9}, levy_multiplier: {0.2, 5, 100}
GA	epoch: {100}, pop_size: {100}, pc: {0.85, 0.90, 0.95}, pm: {0.025, 0.1}
GWO	epoch: {100}, pop_size: {100}
HS	epoch: {100}, pop_size: {100}, c_r: {0.05, 0.15, 0.45}, pa_r: {0.3, 0.5, 0.7}
JA	epoch: {100}, pop_size: {100}
PSO	epoch: {100}, pop_size: {100}, w: {0.4, 0.7}, c1: {0.7, 2.1, 4.2}, c2: {0.7, 2.1, 4.2}
TLO	epoch: {100}, pop_size: {100}

Bu çalışmada, doğadan esinlenmiş AI optimizasyon algoritmaları için ayarlanabilir parametre aralıklarında Çizelge 5.1’de yer alan denemeler yapılmıştır. Belirlenen ayarlanabilir parametreler kullanılarak algoritmaların performans analizleri gerçekleştirilmiştir.

Gerçekleştirilen denemeler ve iterasyonlar sonucunda elde edilen metrik değerleri, ağırlıklandırma yaklaşımları kullanılarak ÇKKV yöntemleriyle analiz edilmiştir. Bu doğrultuda, IDOCRIW objektif ağırlıklandırma yöntemi ile ağırlıklandırma yaklaşımı benimsenmiştir. Böylece, metriklerin hangi ölçüde önemli kabul edildiğine dair senaryosu incelenmiştir. Ağırlıklandırılan metrikler üzerinden COPRAS, EDAS, MABAC ve Copeland’in yer aldığı ÇKKV yöntemleri uygulanarak AI optimizasyon algoritmalarının genel sıralamaları elde edilmiştir. Bu yöntemlerin her biri, hem kriterleri fayda veya maliyet perspektifinde ele alması hem de alternatifleri sıralama biçimi yönüyle farklı hesaplama prensiplerine dayanmaktadır. Dolayısıyla, elde edilen algoritma sıralamaları, farklı değerlendirme çerçeveleri altında hangi yöntemin daha başarılı olduğunu çok boyutlu bir bakış açısıyla ortaya koymaktadır. Buna göre, kriter ağırlıklandırma ve ÇKKV yöntemlerinin nasıl işlediğinin açıklanması, elde edilen

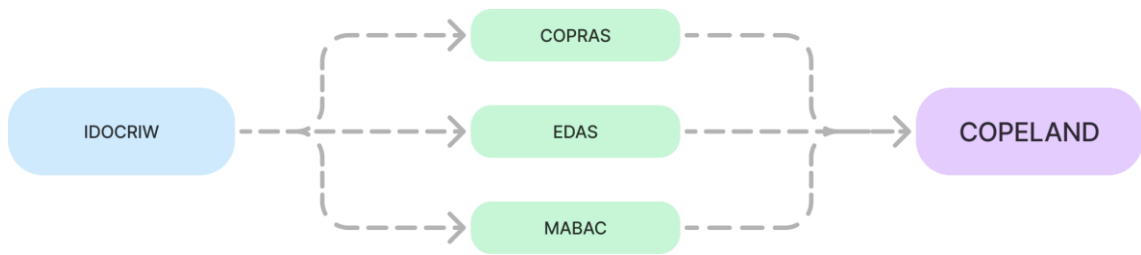
sıralamaların çeşitliliğini ve bütüncüllüğünü anlamak bakımından oldukça önem taşımaktadır.

Ağırlıklandırma sürecinde kullanılan IDOCRIW yöntemi çeşitli objektif ağırlıklandırma yöntemlerini bir arada kullanarak bütüncül bir karar vermeyi amaçlamakta, farklı yöntemlerin sonuçlarını bütünleştirerek daha istikrarlı ve yaygın bir kullanım potansiyeline sahip ağırlık değerleri üretmektedir.

Ağırlıklandırma işlemi sonucunda elde edilen kriter ağırlıkları, ÇKKV yöntemleri olan COPRAS, EDAS ve MABAC uygulanırken kullanılmaktadır. COPRAS, alternatiflerin pozitif ve negatif ideal çözüme olan uzaklıklarını temel alarak bir görelî önem derecesi hesaplamakta, bu dereceye göre alternatifleri sıralamaktadır. EDAS yönteminde, her bir alternatifin ortalama çözüme olan uzaklığı pozitif ve negatif performans ölçütleri şeklinde değerlendirilmektedir. Bu sayede, ortalamanın üzerinde performans gösteren alternatiflerin pozitif değerlendirme skorları yükselmekte, ortalamanın altında kalanların ise negatif değerlendirme skorları artmaktadır. MABAC yöntemi, karar verme sürecinde sınır yaklaşım bölgesi kullanarak alternatifleri değerlendirmektedir. Bu yöntemde, kriter ağırlıkları dikkate alınarak alternatiflerin karar matrisi üzerinde bir yaklaşım sınırı belirlenir ve alternatiflerin bu sınıra olan uzaklıkları hesaplanır. Alternatiflerin sınır bölgesine olan mesafeleri, pozitif ya da negatif performanslarını belirler ve buna göre sıralama yapılır. Son aşamada sıralama için kullanılan Copeland yöntemi ise çift yönlü karşılaştırmalara dayalı bir oylama yöntemi olup, alternatiflerin birbirine karşı kazandığı ve kaybettiği durumları analiz ederek bir net skor belirler. Farklı ÇKKV yöntemleri tarafından üretilen sıralama sonuçları tek bir nihai sıralamaya indirgenerek Copeland yöntemi ile objektif bir biçimde bütünleştirilmektedir. Her alternatifin diğer alternatiflerle yaptığı ikili karşılaştırmalarda kazandığı oy sayısı ile kaybettiği oy sayısı arasındaki fark hesaplanarak Copeland skoru oluşturulur. En yüksek Copeland skoruna sahip alternatif en iyi seçenek olarak belirlenir. Bu sayede, farklı ÇKKV yöntemlerinin çeşitliliğinden kaynaklanabilecek belirsizlik en aza indirilmekte ve algoritmalar arasında daha bütüncül ve güvenilir bir kıyaslama yapılabilmektedir.

Söz konusu yaklaşımlar sayesinde, tek bir metriğe odaklanan analizlerin ötesine geçerek, bir algoritmanın hem hata performansı hem çalışma süresi hem de açıklama gücü yönünden kapsamlı bir değerlendirmeye tabi tutulmasına olanak sağlamaktadır. Her bir algoritma, farklı ağırlıklandırma senaryoları altında söz konusu metriklerin tamamı ışığında değerlendirilmekte ve farklı ÇKKV yöntemleri ile sıralama yapılmaktadır. Bu işlemler sonucunda, farklı perspektiflerden bakıldığında hangi algoritmanın öne çıktığına ilişkin geniş bir tablo oluşmaktadır.

Gerçekleştirilen analiz sürecinde, değerlendirme metrikleri olarak *RMSE*, *MAE*, *MAPE*, R^2 , *İterasyon Sayısı* ve *Toplam Hesap Süresi* kullanılmıştır. Her bir metriğin 50 farklı deneme sonucunda elde edilen en iyi değerlerinin ortalama, standart sapma ve medyan değerleri hesaplanarak toplamda 18 (3x6) farklı metrik oluşturulmuştur. Bu metrikler, ağırlıklandırma yöntemi olan IDOCRIW aracılığıyla kriter önem düzeyleri belirlenerek işlenmiş ve ardından COPRAS, EDAS ve MABAC'ın yer aldığı ÇKKV yöntemlerine entegre edilmiştir. Bu sistematik olarak alınan sıralama sonuçları objektif olarak değerlendirilmek için Copeland yöntemi üzerinden incelenmiştir. Böylece, yalnızca tek bir hata metriğine bağlı değerlendirme yerine, farklı hata türleri ile hesap süresinin bir arada dikkate alındığı çok boyutlu bir analiz gerçekleştirilmiştir. Bu yöntem sayesinde, kullanılan ağırlıklandırma stratejisi altında hangi algoritmanın daha dengeli bir performans sunduğu belirlenmiş olup, hem hata hem de verimlilik açısından optimize edilen sonuçlar elde edilmiştir.



Şekil 5.1 Metrikler üzerinden yürütülen ÇKKV yöntemlerinin diyagramı

Bu doğrultuda, izleyen alt bölümlerde üç farklı modelde üzerinden deneysel çalışmalar incelenmiştir. Basit yapılı bir doğrusal olmayan model olan BoxBOD, biyokimyasal oksijen talebi konusundaki ilişkileri yakalamaya yönelik kullanılmaktadır. İki parametrelili

olması nedeniyle arama uzayı dardır. Bununla birlikte, yerel minimumlara sıkışma veya başlangıç değerine duyarlılık gibi potansiyel zorluklar belirli algoritmaların performans farklılıklarını net biçimde göstermektedir. Meyer-1 Modeli, önceki modele göre bir ek parametre içerdiğinden daha özgün bir yapı sergilemekte ve optimizasyon gücünü artırmaktadır. Üç parametrenin varlığı, algoritmaların yerel minimumlara takılma riskini artırmakta ve arama sürecinde popülasyonun çeşitliliğini korumayı zorunlu kılmaktadır. Dört parametrelilikli Kompartman Modeli, en geniş arama uzayı nedeniyle optimizasyon sürecindeki zorlukların açık biçimde ortaya çıktığı örneği temsil etmektedir. Çok sayıda yerel minimumun olması, başlangıç popülasyonunun çeşitliliği ve iterasyon boyunca bu çeşitliliğin korunması gibi faktörleri kritik hale getirmektedir. AI optimizasyon algoritmalarında ayarlanabilir parametrelerinin belirlenmesi, bu modellerde daha da büyük önem kazanmakta, seçilen en iyi parametre kombinasyonuna rağmen bazı algoritmaların farklı tekrarlar arasında kararlı sonuçlar üretmediği gözlenmektedir. Söz konusu performans farklılıkları, çalışma süresi ve iterasyon maliyeti gibi yönlerden de belirginleşmekte, dolayısıyla bu model, optimizasyon algoritmalarının güçlü ve zayıf yönlerini en açık biçimde görmeyi sağlamaktadır.

Tüm bu modellerden elde edilen sonuçlar, metrik tabanlı çok yönlü bir karar matrisine dönüştürülerek, IDOCRIW yöntemiyle ağırlık setleri oluşturulmuş ve her ağırlık seti altında COPRAS, EDAS ve MABAC yöntemleri aracılığıyla AI optimizasyon algoritmaları sıralanmıştır. Ağırlıklandırma yöntemlerinin kendine özgü hesaplama temelleri sayesinde farklı kriterlerin önem dereceleri değiştirildiğinde, bazı algoritmaların sıralamada yükseldiği veya gerilediği görülmektedir. Benzer olarak, ÇKKV yöntemleri arasındaki metodolojik farklılıklar, özellikle net skor hesaplama aşamalarında değişikliklere yol açmakta ve elde edilen sonuç sıralamalarında kısmen farklılık yaratmaktadır. Bu durum, çok kriterli değerlendirmelerde karar vericinin ana amacının ne olduğuna (hata performansı, hesaplama hızı, vb.) bağlı olarak hangi algoritmanın daha uygun olacağının değişebileceğini göstermektedir. Dolayısıyla, optimizasyon yöntemlerinin seçimi sadece tek bir hata ölçütü üzerinden değil, aynı zamanda zaman ve hesaplama kaynaklarının sınırlılığı veya modelin açıklama gücü gibi diğer faktörlerin de dikkate alındığı bütüncül bir perspektif gerektirmektedir. Bu açıdan, belirtilen ağırlıklandırma ve ÇKKV yöntemlerinin kullanımı, karar vericilere ve araştırmacılara

esnek bir araç seti sunmakta, elde edilen sonuçları farklı senaryolar altında inceleme olanağı vermektedir. Bu kapsamlı analizler sonucunda elde edilen çıkarımlar, model ve algoritma seçiminde yol gösterici niteliğiyle birlikte sunulmaktadır.

5.1 Uygulama 1 (BoxBOD Modeli)

Mikroorganizmaların biyokimyasal oksijen ihtiyacı (Biochemical Oxygen Demand - BOD) üzerindeki zaman etkisi analiz edilmiştir. BOD, organik madde içeren suyun biyolojik olarak ayrışması sırasında mikroorganizmalar tarafından tüketilen oksijen miktarını ifade eder ve su kalitesinin önemli bir göstergesidir. Bu bağlamda, BOD'nin zamanla değişimini modellemek amacıyla BoxBOD modeli kullanılmıştır.

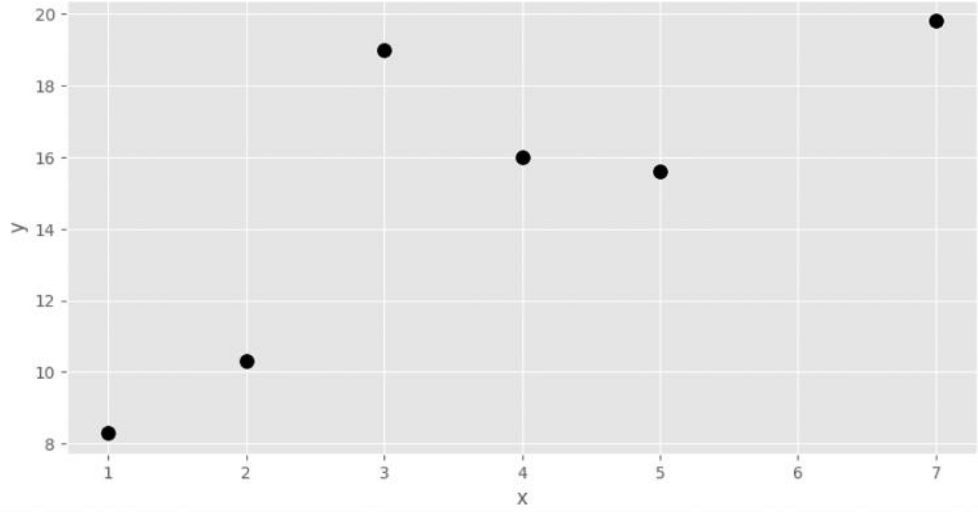
$$Y = \beta_1(1 - e^{(-\beta_2 x)}) + \varepsilon \quad (5.1)$$

Eşitliği (5.1)'de verilen fonksiyon, zaman (x) ilerledikçe biyokimyasal oksijen ihtiyacının (Y) asimptotik bir limite ulaştığını ifade eden doğrusal olmayan bir regresyon modelidir. Model parametreleri (β_1 ve β_2) tahmin edilerek sürecin matematiksel olarak açıklanması amaçlanmıştır.

Çizelge 5.2 Mikroorganizmaların zamana göre biyokimyasal oksijen ihtiyacı

x	1	2	3	4	5	7
Y	8.3	10.3	19	16	15.6	19.8

Modelin doğruluğunu değerlendirmek için Çizelge 5.2'de verilen veri seti kullanılarak görselleştirme ve model uyumu testleri yapılmıştır.



Şekil 5.2 Veri setinin saçılım grafiği

Çalışma kapsamında, veri setine dayalı olarak Şekil 5.2’de bir dağılım grafiği oluşturulmuş, sınır aralıkları Çizelge 5.3’te belirlenmiş, fonksiyon modellenmiş ve kod çıktıları analiz edilmiştir. Elde edilen sonuçlar, BOD’nin zamanla nasıl değiştiğini anlamamıza ve su kalitesi yönetimi açısından önemli çıkarımlar yapmamıza olanak sağlamaktadır.

Çizelge 5.3 Model parametrelerinin arandığı alt ve üst sınırlar

	β_1	β_2
Alt Sınır	15	0.1
Üst Sınır	25	1

Performans metrikleri ile farklı optimizasyon algoritmalarının belirli bir modele uygulanması sonucunda elde edilen sonuçları karşılaştırmaktadır. Her bir satır, belirli bir algoritmanın performansını gösterirken, farklı sütunlar bu algoritmaların çalıştırma süresi, öğrenme süreci, elde edilen parametre değerleri ve performans metriklerini içermektedir. Çizelge 5.5’te, farklı algoritmaların aynı model üzerinde nasıl çalıştığını ve birbirlerine göre nasıl bir performans sergilediğini görmek mümkündür. Bu karşılaştırma, en verimli ve en uygun optimizasyon yönteminin belirlenmesine yardımcı olmaktadır.

Çizelge 5.4 AI optimizasyon algoritmalarının en iyi performans gösterdiği ayarlanabilir parametre değerleri

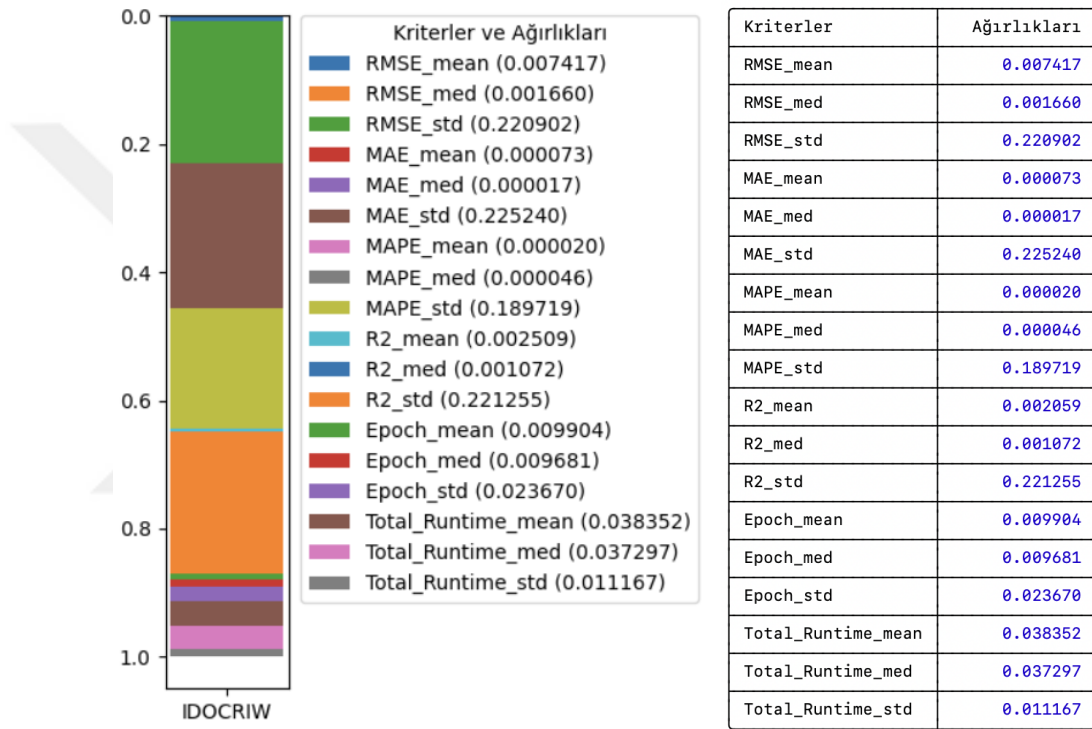
<i>Algoritmalar</i>	<i>Ayarlanabilir Parametreler</i>
ABC	{'epoch': 100, 'pop_size': 100, 'n_limits': 75}
ACOR	{'epoch': 100, 'pop_size': 100, 'sample_count': 100, 'intent_factor': 0.7, 'zeta': 1}
BA	{'epoch': 100, 'pop_size': 100, 'loudness': 0.8, 'pulse_rate': 0.8, 'pf_min': 0, 'pf_max': 10}
CSA	{'epoch': 100, 'pop_size': 100, 'p_a': 0.3}
DE	{'epoch': 100, 'pop_size': 100, 'wf': 0.4, 'cr': 0.9, 'strategy': 4}
FPA	{'epoch': 100, 'pop_size': 100, 'p_s': 0.7, 'levy_multiplier': 5}
GA	{'epoch': 100, 'pop_size': 100, 'pc': 0.9, 'pm': 0.1}
GWO	{'epoch': 100, 'pop_size': 100}
HS	{'epoch': 100, 'pop_size': 100, 'c_r': 0.45, 'pa_r': 0.3}
JA	{'epoch': 100, 'pop_size': 100}
PSO	{'epoch': 100, 'pop_size': 100, 'c1': 0.7, 'c2': 0.7, 'w': 0.4}
TLO	{'epoch': 100, 'pop_size': 100}

Üzerinde çalışılan algoritmalar, kendine özgü ayarlanabilir parametreler ile tasarlanmış olup, her biri belirli stratejilere dayanarak optimizasyon sürecini yürütmektedir. Bu algoritmalar, belirlenen ayarlanabilir parametreler doğrultusunda en uygun çözüme ulaşmayı hedeflemektedir. En iyi algoritma ayarlanabilir parametrelerinin performansına göre aldığı değerler Çizelge 5.4'te yer almaktadır. İncelenen metriklerde en düşük değerler koyu renk ile belirtilmiştir, sadece R^2 metriğinde ise en yüksek olanlar koyu renk ile belirtilmiştir.

Çizelge 5.5 AI optimizasyon algoritmalarının performans çıktıları ve model parametre tahminleri

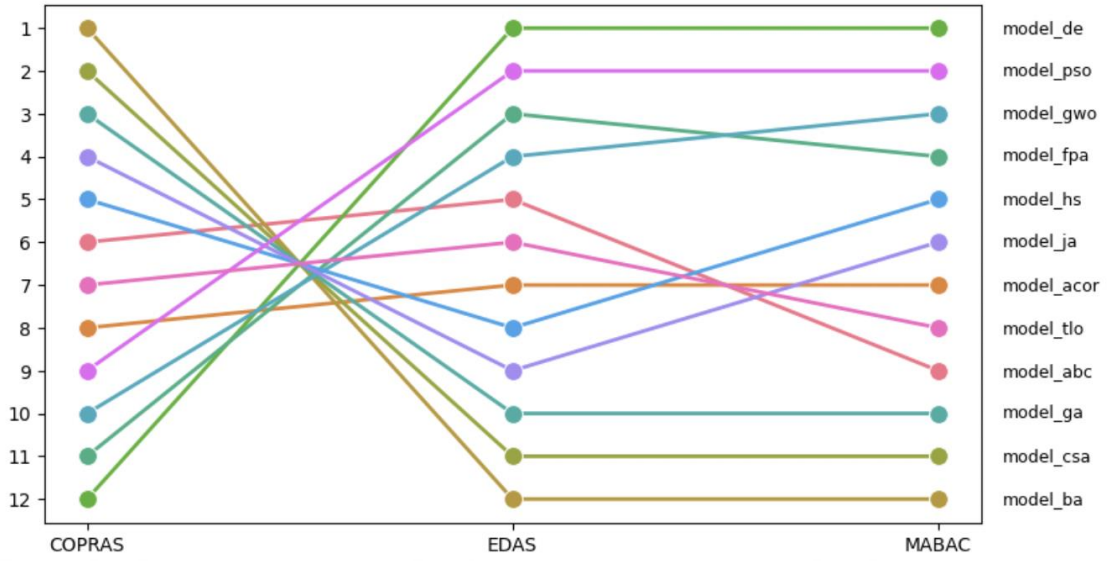
<i>AI Optimizasyon Algoritmaları Performans Metrikleri</i>									<i>Model Parametreleri</i>	
<i>Model</i>	<i>İterasyon</i>	<i>İterasyon Süresi</i>	<i>Toplam Çalışma Süresi</i>	<i>EKK</i>	<i>RMSE</i>	<i>MAE</i>	<i>MAPE</i>	<i>R2</i>	β_1	β_2
ABC	97	0.012005	1.222145	25.990267	2.081276	1.760106	0.118997	0.757584	19.142575	0.531091
ACOR	99	0.008068	0.876528	25.990267	2.081276	1.760106	0.118997	0.757584	19.142575	0.531091
BA	21	0.004178	0.462945	25.990328	2.081279	1.760195	0.119031	0.757583	19.150244	0.530543
CSA	84	0.005943	0.646842	25.990332	2.081279	1.760155	0.119029	0.757583	19.150176	0.530468
DE	83	0.005306	0.542913	25.990267	2.081276	1.760106	0.118997	0.757584	19.142575	0.531091
FPA	92	0.005791	0.649589	25.990267	2.081276	1.760106	0.118997	0.757584	19.142575	0.531091
GA	54	0.006948	0.696849	25.990287	2.081277	1.760082	0.119008	0.757583	19.145721	0.530743
GWO	99	0.005451	0.548044	25.990268	2.081276	1.760093	0.118995	0.757584	19.141987	0.531122
HS	93	0.005135	0.541161	25.990267	2.081276	1.760106	0.118998	0.757584	19.142631	0.531087
JA	11	0.004187	0.443945	25.990381	2.081281	1.760259	0.119045	0.757583	19.153039	0.530408
PSO	62	0.004961	0.506731	25.990267	2.081276	1.760106	0.118997	0.757584	19.142575	0.531091
TLO	98	0.011618	1.159260	25.990267	2.081276	1.760106	0.118997	0.757584	19.142575	0.531091

Şekil 5.4, kullanılan ağırlıklandırma yönteminin belirli kriterlere göre nasıl dağıldığını göstermektedir. Yatay eksen ağırlıklandırma yöntemi IDOCRIW yer alırken, dikey eksen kriterlere verilen ağırlıkların toplamı 1 olacak biçimde yığılmış dağılımı bulunmaktadır. Şekil 5.4'te yer alan grafikte, yöntemin farklı kriterlere nasıl ağırlık verdiği gözlemlenebilir ve ağırlıklandırma stratejisine göre değişen dağılımlar görülebilir. Şekil 5.4'te sunulan sütun grafiğinde, her bir metriğe karşılık gelen ağırlık değerleri etiketler üzerinde belirtilmiştir.



Şekil 5.3 Kriterlerin IDOCRIW yöntemine göre ağırlıkları

Şekil 5.5, ağırlıklandırma yöntemine göre AI optimizasyon algoritmalarının ÇKKV yöntemleri üzerinden sıralamalarını göstermektedir. Yatay eksen ÇKKV yöntemleri yer alırken, dikey eksen AI optimizasyon algoritmalarının sıralamaları bulunmaktadır. Uygulanan yöntemlere göre PSO ve DE algoritmaları belirlenen kriterlere göre iyi performans sergilemiştir.



Şekil 5.4 ÇKKV yöntemlerine göre algoritmaların sıralaması

Şekil 5.6, modellere uygulanan ÇKKV yöntemlerinden elde edilen sıralama değerlerinin Copeland yöntemi ile objektif bir biçimde birleştirilerek nihai sıralamanın oluşturulduğunu göstermektedir. Yapılan değerlendirme sonucunda, PSO algoritmasının üstün performans sergilediği, buna karşın BA algoritmasının en düşük performansı gösterdiği tespit edilmiştir.



Şekil 5.5 AI optimizasyon algoritmalarının Copeland yöntemine göre sıralamaları

5.2 Uygulama 2 (Meyer-1 Modeli)

Bu çalışmada, Meyer-1 modeli kullanılarak doğrusal olmayan bir regresyon analizi gerçekleştirilmiştir. Bu model, belirli bağımsız değişkenler (x_1 ve x_2) ve bilinmeyen parametreler (β_1 , β_2 , β_3) arasındaki ilişkiyi matematiksel olarak açıklamak için geliştirilmiştir. Model, aşağıdaki fonksiyon ile tanımlanmaktadır.

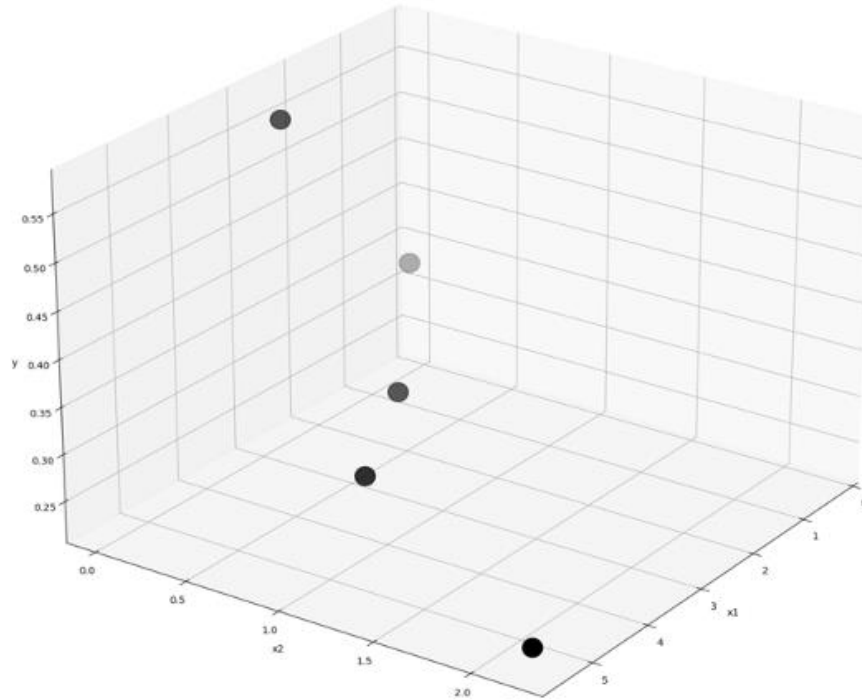
$$Y = \frac{\beta_1 \beta_3 x_1}{(1 + \beta_1 x_1 + \beta_2 x_2)} + \varepsilon \quad (5.2)$$

Eşitlik (5.2)'de verilen modelin amacı, belirli deneysel veriler kullanılarak parametrelerin en küçük kareler yöntemiyle tahmin edilmesi, veri setine dayalı olarak bir dağılım grafiği oluşturulmuş, parametrelerin sınır aralıkları belirlenmiş, fonksiyon modellenmiş ve kod çıktıları analiz edilmiştir.

Çizelge 5.6 Modelde kullanılan değişkenler ve veri seti

x_1	0.28	2.90	3.20	4.70	5.50
x_2	0	0.10	0.82	1.10	2.20
Y	0.33	0.57	0.33	0.31	0.23

Modelin doğruluğunu değerlendirmek için Çizelge 5.7'deki veri seti kullanılarak görselleştirme ve model uyumu testleri yapılmıştır.



Şekil 5.6 Veri setinin saçılım grafiği

Çalışma kapsamında, veri setine dayalı olarak Şekil 5.8'de görülen bir dağılım grafiği oluşturulmuş, Çizelge 5.7'deki sınır aralıkları belirlenmiş, fonksiyon modellenmiş ve kod çıktıları analiz edilmiştir.

Çizelge 5.7 Model parametrelerinin arandığı alt ve üst sınırlar

	β_1	β_2	β_3
Alt Sınır	0.14	3.94	0.49
Üst Sınır	5.55	26.45	0.99

Performans metrikleri ile farklı optimizasyon algoritmalarının belirli bir modele uygulanması sonucunda elde edilen sonuçları karşılaştırmaktadır. Her bir satır, belirli bir algoritmanın performansını gösterirken, farklı sütunlar bu algoritmaların çalışma süresi, öğrenme süreci, elde edilen parametre değerleri ve performans metriklerini içermektedir. Çizelge 5.9’da farklı algoritmaların aynı model üzerinde nasıl çalıştığını ve birbirlerine göre nasıl bir performans sergilediğini görmek mümkündür. Bu karşılaştırma, en verimli ve en uygun optimizasyon yönteminin belirlenmesine yardımcı olmaktadır.

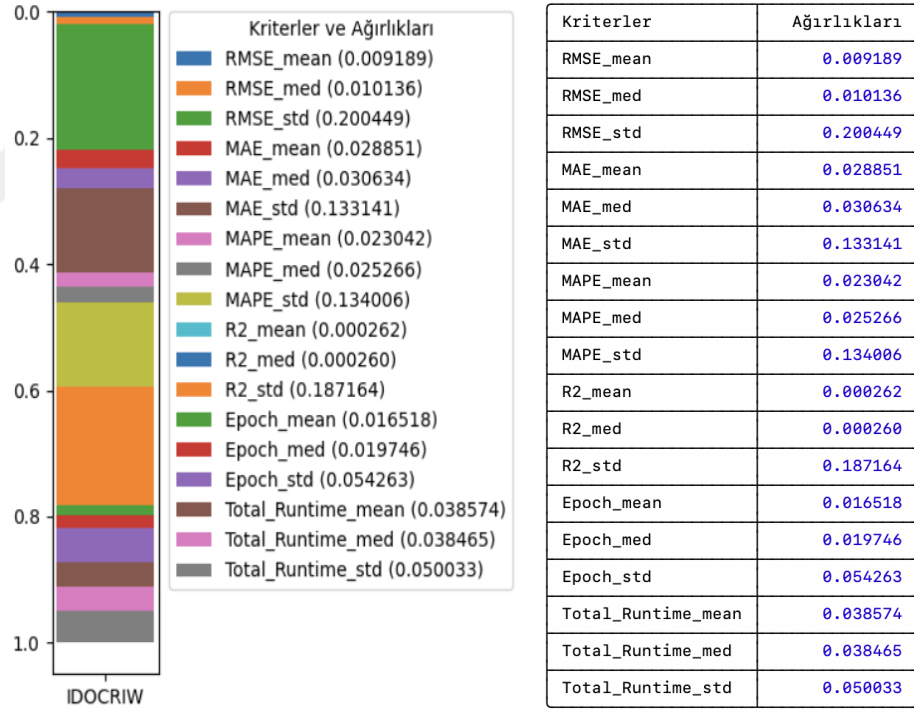
Çizelge 5.8 AI optimizasyon algoritmalarının en iyi performans gösterdiği ayarlanabilir parametre değerleri

<i>Algoritmalar</i>	<i>Ayarlanabilir Parametreler</i>
ABC	{'epoch': 100, 'pop_size': 100, 'n_limits': 45}
ACOR	{'epoch': 100, 'pop_size': 100, 'sample_count': 50, 'intent_factor': 0.3, 'zeta': 1}
BA	{'epoch': 100, 'pop_size': 100, 'loudness': 1.4, 'pulse_rate': 0.7, 'pf_min': 0, 'pf_max': 10}
CSA	{'epoch': 100, 'pop_size': 100, 'p_a': 0.1}
DE	{'epoch': 100, 'pop_size': 100, 'wf': 0.4, 'cr': 0.9, 'strategy': 4}
FPA	{'epoch': 100, 'pop_size': 100, 'p_s': 0.9, 'levy_multiplier': 5}
GA	{'epoch': 100, 'pop_size': 100, 'pc': 0.9, 'pm': 0.1}
GWO	{'epoch': 100, 'pop_size': 100}
HS	{'epoch': 100, 'pop_size': 100, 'c_r': 0.45, 'pa_r': 0.5}
JA	{'epoch': 100, 'pop_size': 100}
PSO	{'epoch': 100, 'pop_size': 100, 'c1': 4.2, 'c2': 0.7, 'w': 0.4}
TLO	{'epoch': 100, 'pop_size': 100}

Üzerinde çalışılan algoritmalar, kendine özgü ayarlanabilir parametreler ile tasarlanmış olup, her biri belirli stratejilere dayanarak optimizasyon sürecini yürütmektedir. Bu algoritmalar, belirlenen ayarlanabilir parametreler doğrultusunda modellenerek en uygun

çözümüne ulaşmayı hedeflemektedir. En iyi algoritma ayarlanabilir parametrelerinin performansına göre aldığı değerler Çizelge 5.8’de yer almaktadır. İncelenen metriklerde en düşük değerler koyu renk ile belirtilmiştir, sadece R^2 metriğinde ise en yüksek olanlar koyu renk ile belirtilmiştir.

Şekil 5.10, kullanılan ağırlıklandırma yönteminin belirli kriterlere göre nasıl dağıldığını göstermektedir. Yatay ekseninde ağırlıklandırma yöntem IDOCRIW yer alırken, dikey ekseninde kriterlere verilen ağırlıkların toplamı 1 olacak biçimde yığılmış dağılımı bulunmaktadır. Şekil 5.10’da yer alan grafikte, yöntemin farklı kriterlere nasıl ağırlık verdiği gözlemlenebilir ve ağırlıklandırma stratejisine göre değişen dağılımlar görülebilir. Şekil 5.10’da sunulan sütun grafiğinde, her bir metriğe karşılık gelen ağırlık değerleri etiketler üzerinde belirtilmiştir

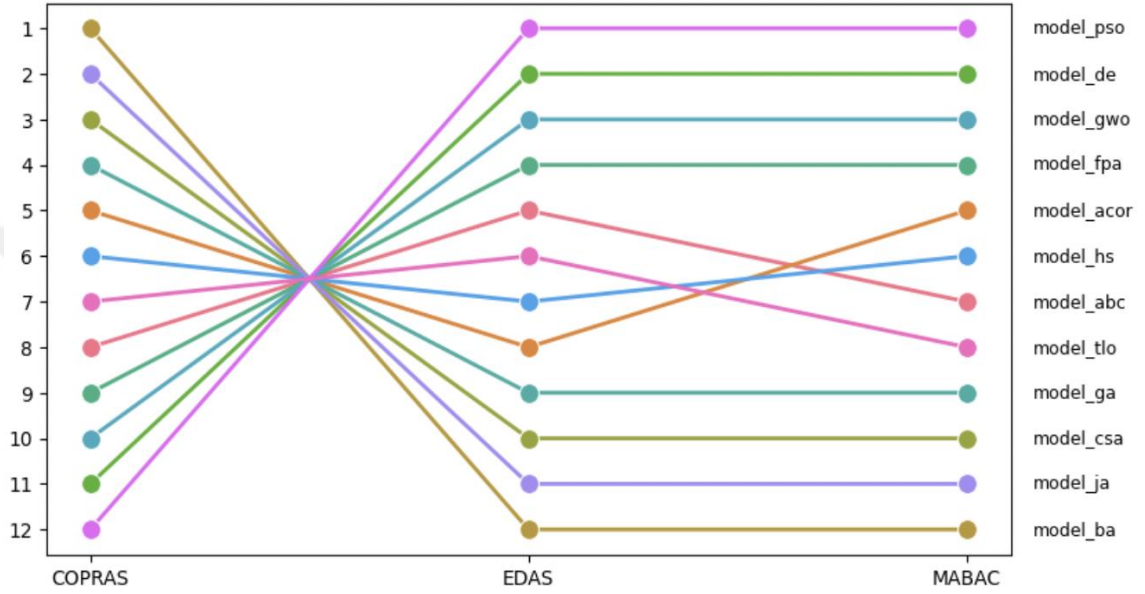


Şekil 5.7 Kriterlerin IDOCRIW yöntemine göre ağırlıkları

Çizelge 5.9 AI optimizasyon algoritmalarının performans çıktıları ve model parametre tahminleri

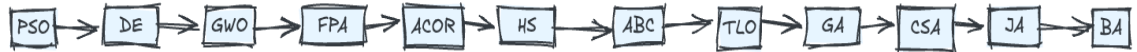
Model	İterasyon	İterasyon Süresi	AI Optimizasyon Algoritmaları Performans Metrikleri						Model Parametreleri		
			Toplam Çalışma Süresi	EKK	RMSE	MAE	MAPE	R2	β_1	β_2	β_3
ACOR	93	0.005210	0.525834	0.000897	0.013397	0.009823	0.032362	0.986220	2.896767	14.788302	0.738734
BA	4	0.004259	0.469621	0.000921	0.013568	0.010775	0.035291	0.985864	2.960722	14.597635	0.729295
CSA	33	0.008250	0.683115	0.000913	0.013509	0.010419	0.034394	0.985987	2.933115	14.567268	0.731968
DE	59	0.005312	0.532176	0.000897	0.013395	0.009760	0.032288	0.986223	2.897681	14.759801	0.738144
FPA	89	0.006716	0.694250	0.000897	0.013395	0.009759	0.032287	0.986223	2.897690	14.759727	0.738140
GA	78	0.008069	0.786943	0.000898	0.013401	0.009702	0.032063	0.986210	2.878085	14.686825	0.738950
GWO	100	0.005889	0.604212	0.000897	0.013395	0.009762	0.032308	0.986223	2.897834	14.755902	0.738095
HS	86	0.006452	0.638400	0.000897	0.013396	0.009702	0.032093	0.986221	2.898706	14.769992	0.737882
JA	1	0.004694	0.470264	0.000903	0.013436	0.010004	0.033359	0.986140	2.875292	14.539585	0.737773
PSO	1	0.005154	0.504818	0.000897	0.013395	0.009760	0.032288	0.986223	2.897681	14.759801	0.738144
TLO	78	0.011682	1.163178	0.000897	0.013395	0.009760	0.032288	0.986223	2.897681	14.759801	0.738144

Şekil 5.11, ağırlıklandırma yöntemine göre AI optimizasyon algoritmalarının ÇKKV yöntemleri üzerinden sıralamalarını göstermektedir. Yatay ekseninde ÇKKV yöntemleri yer alırken, dikey ekseninde AI optimizasyon algoritmalarının sıralama bulunmaktadır. Uygulanan yöntemlere göre PSO ve DE algoritmaları belirlenen kriterlere göre iyi performans sergilemiştir.



Şekil 5.8 ÇKKV yöntemlerine göre algoritmaların sıralaması

Şekil 5.12, modellere uygulanan ÇKKV yöntemlerinden elde edilen sıralama değerlerinin Copeland yöntemi ile objektif bir biçimde birleştirilerek nihai sıralamanın oluşturulduğunu göstermektedir. Yapılan değerlendirme sonucunda, PSO algoritmasının üstün performans sergilediği, buna karşın BA algoritmasının en düşük performansı gösterdiği tespit edilmiştir.



Şekil 5.9 AI optimizasyon algoritmalarının Copeland yöntemine göre sıralamaları

5.3 Uygulama 3 (İkili Kompartman Modeli)

İntravenöz (IV) enjeksiyon yoluyla vücuda alınan bir ilacın plazma konsantrasyon-zaman profili analiz edilmiştir. İlaç farmakokinetiğinin modellenmesi amacıyla ikili kompartman modeli kullanılmıştır. Bu model, ilacın vücutta iki farklı bölme (merkezi ve periferik) dağıldığını ve bu bölmeler arasındaki değişimlerin üstel azalma fonksiyonları ($\beta_1, \beta_2, \beta_3, \beta_4$) ile ifade edilebileceğini varsaymaktadır. Kullanılan model

$$Y = \beta_1 e^{(-\beta_2 x)} + \beta_3 e^{(-\beta_4 x)} + \varepsilon \quad (5.3)$$

biçiminde tanımlanmıştır. Eşitlik (5.3)'te verilen model, ilaç enjeksiyonunun ardından plazma konsantrasyonunun (Y) zamana (x) bağlı değişimini iki üstel azalma terimi ile açıklamaktadır.

Çizelge 5.10 Vücuda alınan bir ilacın zamana göre plazma konsantrasyonu

x	0	0.25	0.50	0.75	1	2	3	4	6	8	12	16	24
Y	16.4	14.2	12.53	11.17	10.09	7.56	6.44	5.85	5.16	4.65	3.18	3.12	2.09

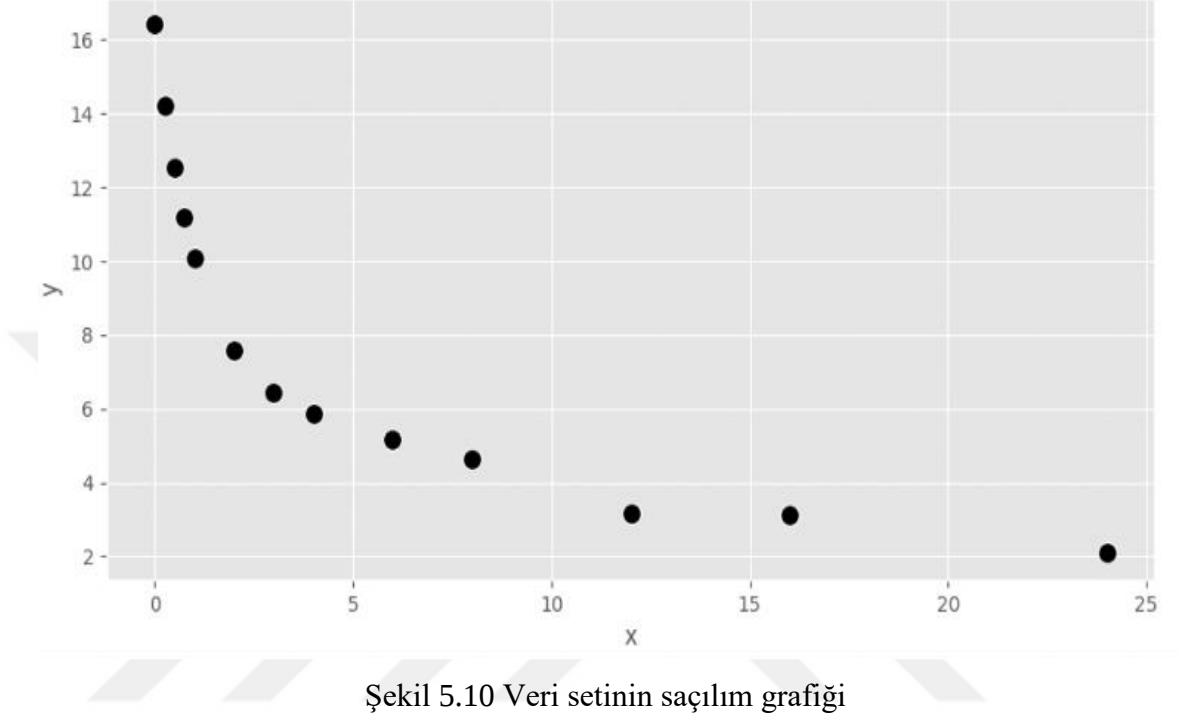
Çalışmada, deneysel olarak elde edilen Çizelge 5.10'da yer alan plazma konsantrasyon verileri kullanılarak dağılım grafiği oluşturulmuş, model parametrelerinin çizelge 5.11'de gösterilen sınır aralıkları belirlenmiştir. Modelleme süreci, farmakokinetik analizlerde yaygın olarak kullanılan doğrusal olmayan regresyon teknikleriyle gerçekleştirilmiştir.

Çizelge 5.11 Model parametrelerinin arandığı alt ve üst sınırlar

	β_1	β_2	β_3	β_4
Alt Sınır	9.2	0.9	6.9	0
Üst Sınır	9.8	1.2	7.2	0.1

Bu analiz, ilacın vücuttaki dağılım ve eliminasyon süreçlerini anlamaya yönelik önemli bilgiler sağlamaktadır. Modelin doğruluğunu değerlendirmek için veri seti kullanılarak

görselleştirme ve model uyumu testleri yapılmıştır. Çalışma kapsamında, veri setine dayalı olarak Şekil 5.14'teki bir dağılım grafiği oluşturulmuş, sınır aralıkları belirlenmiş, fonksiyon modellenmiş ve kod çıktıları analiz edilmiştir.



Performans metrikleri ile farklı optimizasyon algoritmalarının belirli bir modele uygulanması sonucunda elde edilen sonuçları karşılaştırmaktadır. Her bir satır, belirli bir algoritmanın performansını gösterirken, farklı sütunlar bu algoritmaların çalışma süresi, öğrenme süreci, elde edilen parametre değerleri ve performans metriklerini içermektedir. Çizelge 5.13'te farklı algoritmaların aynı model üzerinde nasıl çalıştığını ve birbirlerine göre nasıl bir performans sergilediğini görmek mümkündür. Bu karşılaştırma, en verimli ve en uygun optimizasyon yönteminin belirlenmesine yardımcı olmaktadır.

Çizelge 5.12 AI optimizasyon algoritmalarının en iyi performans gösterdiği ayarlanabilir parametre değerleri

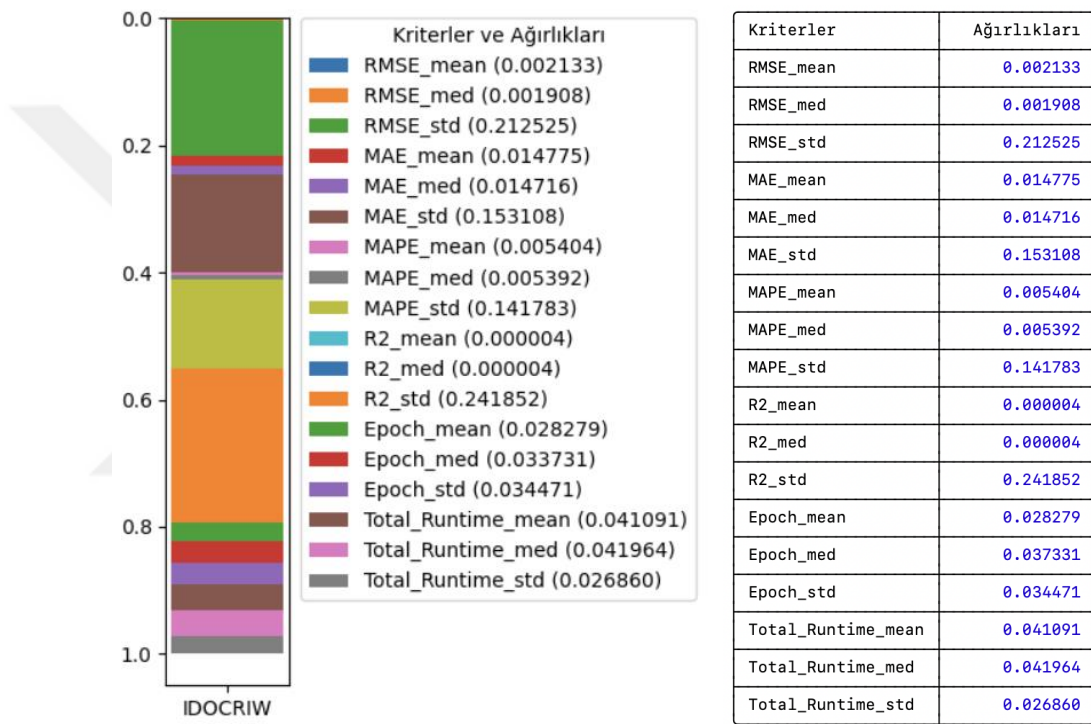
<i>Algoritmalar</i>	<i>Ayarlanabilir Parametreler</i>
ABC	{'epoch': 100, 'pop_size': 100, 'n_limits': 45}
ACOR	{'epoch': 100, 'pop_size': 100, 'sample_count': 100, 'intent_factor': 0.5, 'zeta': 1}
BA	{'epoch': 100, 'pop_size': 100, 'loudness': 0.8, 'pulse_rate': 0.6, 'pf_min': 0, 'pf_max': 10}
CSA	{'epoch': 100, 'pop_size': 100, 'p_a': 0.5}
DE	{'epoch': 100, 'pop_size': 100, 'wf': 0.4, 'cr': 0.9, 'strategy': 4}
FPA	{'epoch': 100, 'pop_size': 100, 'p_s': 0.7, 'levy_multiplier': 5}
GA	{'epoch': 100, 'pop_size': 100, 'pc': 0.9, 'pm': 0.1}
GWO	{'epoch': 100, 'pop_size': 100}
HS	{'epoch': 100, 'pop_size': 100, 'c_r': 0.45, 'pa_r': 0.5}
JA	{'epoch': 100, 'pop_size': 100}
PSO	{'epoch': 100, 'pop_size': 100, 'c1': 2.1, 'c2': 2.1, 'w': 0.4}
TLO	{'epoch': 100, 'pop_size': 100}

Üzerinde çalışılan algoritmalar, kendine özgü ayarlanabilir parametreler ile tasarlanmış olup, her biri belirli stratejilere dayanarak optimizasyon sürecini yürütmektedir. Bu algoritmalar, belirlenen ayarlanabilir parametreler doğrultusunda modellenerek en uygun çözüme ulaşmayı hedeflemektedir. En iyi algoritma ayarlanabilir parametrelerinin performansına göre aldığı değerler Çizelge 5.12’de yer almaktadır. İncelenen metriklerde en düşük değerler koyu renk ile belirtilmiştir, sadece R^2 metriğinde ise en yüksek olanlar koyu renk ile belirtilmiştir.

Çizelge 5.13 AI optimizasyon algoritmalarının performans çıktıları ve model parametre tahminleri

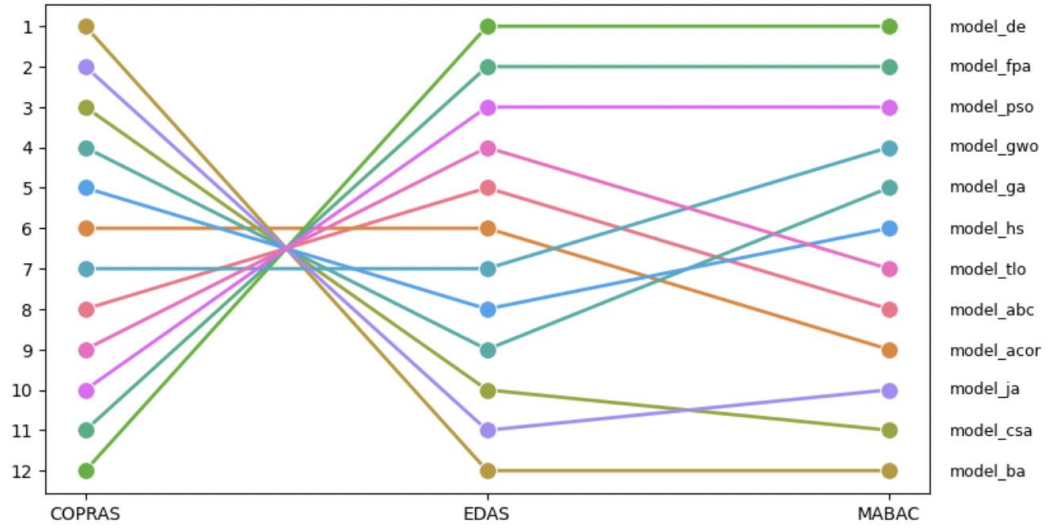
<i>AI Optimizasyon Algoritmaları Performans Metrikleri</i>									<i>Model Parametreleri</i>			
<i>Model</i>	<i>İterasyon</i>	<i>İterasyon Süresi</i>	<i>Toplam Çalışma Süresi</i>	<i>EKK</i>	<i>RMSE</i>	<i>MAE</i>	<i>MAPE</i>	<i>R2</i>	β_1	β_2	β_3	β_4
ABC	86	0.011886	1.225778	0.309132	0.154206	0.084258	0.025422	0.998775	9.399273	0.993751	6.985475	0.053616
ACOR	99	0.012120	1.227061	0.309132	0.154206	0.084254	0.025425	0.998775	9.398977	0.993795	6.985671	0.053619
BA	1	0.004457	0.466096	0.316744	0.156093	0.092218	0.027034	0.998744	9.354298	0.995441	7.048614	0.054648
CSA	54	0.006111	0.628315	0.309858	0.154387	0.083035	0.024987	0.998772	9.407638	0.993136	6.987465	0.053485
DE	99	0.005399	0.587094	0.309132	0.154206	0.084248	0.025417	0.998775	9.399413	0.993731	6.985288	0.053612
FPA	99	0.006030	0.617798	0.309132	0.154206	0.084247	0.025416	0.998775	9.399473	0.993726	6.985275	0.053611
GA	100	0.006938	0.700704	0.309141	0.154208	0.084499	0.025404	0.998775	9.400638	0.992863	6.982698	0.053569
GWO	100	0.005535	0.556835	0.309138	0.154207	0.084321	0.025439	0.998775	9.400247	0.993722	6.983980	0.053604
HS	89	0.006459	0.645597	0.309162	0.154213	0.083916	0.025391	0.998774	9.398393	0.995100	6.987699	0.053637
JA	3	0.004355	0.451040	0.311759	0.154860	0.081264	0.024096	0.998764	9.401764	0.989856	6.980489	0.053098
PSO	34	0.005319	0.524040	0.309132	0.154206	0.084248	0.025417	0.998775	9.399413	0.993731	6.985288	0.053612
TLO	99	0.012346	1.229931	0.309132	0.154206	0.084248	0.025417	0.998775	9.399413	0.993731	6.985288	0.053612

Şekil 5.16, kullanılan ağırlıklandırma yönteminin belirli kriterlere göre nasıl dağıldığını göstermektedir. Yatay eksen ağırlıklandırma yöntemi IDOCRIW yer alırken, dikey eksen kriterlere verilen ağırlıkların toplamı 1 olacak biçimde yığılmış dağılımı bulunmaktadır. Şekil 5.16’da yer alan grafikte, yöntemin farklı kriterlere nasıl ağırlık verdiği gözlemlenebilir ve ağırlıklandırma stratejisine göre değişen dağılımlar görülebilir. Şekil 5.16’te sunulan sütun grafiğinde, her bir metriğe karşılık gelen ağırlık değerleri etiketler üzerinde belirtilmiştir



Şekil 5.11 Kriterlerin IDOCRIW yöntemine göre ağırlıkları

Şekil 5.17, ağırlıklandırma yöntemine göre AI optimizasyon algoritmalarının ÇKKV yöntemleri üzerinden sıralamalarını göstermektedir. Yatay eksen ÇKKV yöntemleri yer alırken, dikey eksen AI optimizasyon algoritmalarının sıralama bulunmaktadır. Uygulanan yöntemlere göre DE ve FPA algoritmaları belirlenen kriterlere göre iyi performans sergilemiştir.



Şekil 5.12 ÇKKV yöntemlerine göre algoritmaların sıralaması

Şekil 5.18, modellere uygulanan ÇKKV yöntemlerinden elde edilen sıralama değerlerinin Copeland yöntemi ile objektif bir biçimde birleştirilerek nihai sıralamanın oluşturulduğunu göstermektedir. Yapılan değerlendirme sonucunda, DE algoritmasının üstün performans sergilediği, buna karşın BA algoritmasının en düşük performansı gösterdiği tespit edilmiştir.



Şekil 5.13 AI optimizasyon algoritmalarının Copeland yöntemine göre sıralamaları

6. SONUÇ

Bu tez çalışmasında, doğrusal olmayan regresyon modellerinde parametre tahmini sürecinde AI optimizasyon algoritmaları performansının kapsamlı bir biçimde değerlendirilmesi hedeflenmiştir. Çalışmada, evrim tabanlı algoritmalar arasında Genetik Algoritma (GA), Diferansiyel Evrim Algoritması (DE) ve Çiçek Tozlaşması Algoritması (FPA) yer alırken, sürü tabanlı algoritmalar arasında Yapay Arı Koloni Algoritması (ABC), Karınca Koloni Optimizasyonu (ACOR), Yarasa Algoritması (BA), Guguk Kuşu Algoritması (CSA), Gri Kurt Optimizasyonu (GWO), Jaya Algoritması (JA) ve Parçacık Sürü Optimizasyonu (PSO) bulunmaktadır. Ayrıca müzik esinli algoritmalarından Armoni Arama (HS) Algoritması, insan davranışından ilham alan Öğretme-Öğrenme Tabanlı Optimizasyonu (TLO) olmak üzere on iki farklı AI optimizasyon algoritması kullanılarak, farklı parametre sayılarına ve özgün seviyelerine sahip üç doğrusal olmayan regresyon model (BoxBOD, Meyer-1 ve İkili Kompartman) üzerinde test edilmiştir. Bu doğrusal olmayan modeller, biyokimyasal süreçlerden farmakokinetiğe kadar geniş bir uygulama yelpazesini temsil etmekte olup, algoritmaların hem basit hem de zor problemlerdeki davranışlarını analiz etmek için kritik bir çerçeve sunmuştur. Elde edilen sonuçlar, AI tabanlı optimizasyon algoritmalarının geleneksel yaklaşımlara kıyasla sağladığı avantajları ve sınırlılıkları sistematik bir biçimde ortaya koymuştur.

Çalışmanın temel bulguları, popülasyon tabanlı algoritmaların doğrusal olmayan regresyon problemlerinde global optimuma ulaşma yeteneklerinin belirgin biçimde üstün olduğunu göstermektedir. Özellikle, parametre sayısı arttıkça (BoxBOD'un 2 parametresinden İkili Kompartman'ın 4 parametresine) algoritmaların arama uzayını tarama ve yerel minimumlardan kaçınma stratejileri, performans farklılıklarını belirleyici bir rol oynamıştır. BoxBOD modelinde, basit yapısı nedeniyle tüm algoritmalar nispeten benzer sonuçlar üretirken, Meyer-1 ve İkili Kompartman modellerinde popülasyon çeşitliliğini koruyan ve adaptif parametre ayarlamaları yapan algoritmaların daha düşük hata metrikleri (*RMSE*, *MAE*, *MAPE*) elde ettiği gözlemlenmiştir. Örneğin, PSO ve DE algoritmaları, yüksek boyutlu uzaylarda dengeli bir keşif ve sömürü dengesi sağlayarak, özellikle İkili Kompartman modelinde en tutarlı sonuçları sunmuştur. Buna karşılık, BA ve JA algoritmaları, özgün modellerde yerel minimumlara takılma eğilimi göstermiştir.

Çalışma kapsamında elde edilen bulgular, BA ve JA algoritmalarının yerel minimumlara sıkışma eğilimi nedeniyle hesap sürelerinin oldukça düşük çıktığını ve bazı sıralamalarda ön plana çıktığını göstermektedir. Bu durum, optimizasyon sürecinde erken sonlanmaya neden olarak modelin gerçek performansının tam olarak değerlendirilememesine yol açmaktadır. Bu nedenle, zaman değerlerinin daha az ağırlıklandırıldığı metrikler olan ortalama, ortanca ve standart sapma gibi istatistiksel ölçütler de değerlendirmeye alınmıştır. Bu ek metriklerin kullanımıyla, algoritmaların hesap sürelerindeki dengesizlikler giderilmiş, yerel minimum problemleri azaltılmış ve genel optimizasyon sürecinin daha sağlıklı bir biçimde analiz edilmesi sağlanmıştır. Böylece, yalnızca en düşük hata metriklerine değil, aynı zamanda algoritmaların tutarlılığına da odaklanılarak daha kapsamlı bir değerlendirme yapılmıştır.

Performans değerlendirmesi, sadece hata metrikleriyle sınırlı kalmamıştır. Hesaplama süreleri, iterasyon başına maliyet ve algoritmaların kararlılığı faktörleri de ÇKKV yöntemleriyle entegre edilerek analiz edilmiştir. IDOCRIW objektif ağırlıklandırma yöntemi, kriterlerin önemini vurgularken, COPRAS, EDAS ve MABAC yöntemleri aracılığıyla algoritmalar alternatiflerin çok boyutlu sıralamasını sağlamıştır. Nihai sıralaması ise Copeland yöntemi ile bütünleştirilerek elde edilmiştir. Sonuçlar, belirli algoritmaların belirli problem türlerinde ön plana çıktığını göstermiştir. ÇKKV yöntemlerinin farklı kriter ağırlıklandırmalarına duyarlı olması, bazı optimizasyon algoritmalarının farklı senaryolarda avantajlı veya dezavantajlı hale gelmesine neden olmuştur. Örneğin, hata oranı kriterine ağırlık verildiğinde DE ve PSO algoritmaları üst sıralarda yer alırken, hesaplama süresinin öncelikli olduğu senaryolarda daha hızlı çalışan ABC ve GWO algoritmalarının avantaj sağladığı görülmüştür. Bu bulgular, algoritma seçiminin problemin doğasına ve öncelikli kriterlere (hata minimizasyonu, hesaplama hızı vb.) bağlı olduğunu vurgulamaktadır.

Çalışmanın teorik katkısı, AI optimizasyon algoritmalarının doğrusal olmayan regresyon problemlerindeki etkinliğini sistematik bir biçimde belgelemektedir. Geleneksel yöntemlerin (Gradient Descent, Levenberg-Marquardt) aksine, bu algoritmalar türev bilgisine ihtiyaç duymadan, başlangıç değerlerinden bağımsız ve geniş çözüm uzaylarında etkili sonuçlar üretmiştir. Özellikle, algoritma ayarlanabilir parametreleri

bulunması sürecinde 50 tekrarlı denemelerle elde edilen istatistiksel güvenilirlik, algoritmaların pratik uygulamalarda kullanımını destekleyen önemli bir kanıt sunmaktadır. Ayrıca, *MEALPY* Python kütüphanesi kullanımı, algoritmaların esnek ve ölçeklenebilir bir biçimde uygulanabilmesine olanak tanımıştır.

Çalışmada kullanılan algoritmaların performans değerlendirmesinde, Veri Zarflama Analizi gibi etkinlik analiz yöntemleri de kullanılabilir. Bu yöntemler, algoritmaların göreceli etkinliğini girdi ve çıktı değişkenleri açısından değerlendirmede yaygın olarak tercih edilmektedir. Ayrıca verilerin dağılımı hakkında belirli varsayımlarda bulunmadan iki değişken arasındaki ilişkinin gücünü ve yönünü incelemek amacıyla doğrusal olmayan korelasyon analizleri yaygın bir biçimde kullanılmaktadır. Bu yöntemler, parametrik testlerde olduğu gibi ortalama ve standart sapma gibi parametrelerin kullanılmasına gerek duymadan, verilerin sıralı hali üzerinden ilişki analizi yaparlar. Böylece verinin dağılımının normalliği ya da lineerlik varsayımı gereksizdir, daha esnek ve yaygın durumlarda kullanılabilirler.

Pratik açıdan, bu çalışma, araştırmacılara ve uygulayıcılara doğrusal olmayan regresyon problemlerinde algoritma seçimi ve parametre optimizasyonu konusunda yardımcı olabilecek bilgiler sunmaktadır. Biyokimyasal veri analizinde BoxBOD modeli kullanılacaksa, düşük hesaplama maliyetli algoritmalar tercih edilebilirken, yüksek özgünlükteki problemlerde ise kararlılığı, tutarlılığı kanıtlanmış algoritmalar öne çıkmaktadır. Ayrıca, çok kriterli değerlendirme yaklaşımları, karar vericilere farklı senaryolar altında en uygun çözümü seçme esnekliği sağlamaktadır.

Bu çalışma, doğrusal olmayan modellerin optimizasyonu konusunda AI optimizasyon algoritmalarının kullanımının problemlerin çözümünde ne ölçüde verimlilik sağlayabileceğini ortaya koyarak karşılaştırmalı analizlerde ÇKKV yöntemlerini kullanarak karar vermenin önemini vurgulamaktadır. Çalışmanın sınırlılıkları arasında, test edilen modellerin sayısının ve veri setlerinin boyutunun nispeten sınırlı olması yer almaktadır. Gelecek çalışmalarda, daha fazla parametre içeren modeller ve gerçek dünya veri setleri üzerinde testler yapılması, algoritmaların genellenebilirliğini artıracaktır. Ayrıca, hibrit algoritmaların performansının incelenmesi, optimizasyon süreçlerinde daha

verimli sonuçlar elde edilmesine katkı sağlayabilir. Ek olarak, yapay zeka tabanlı optimizasyon yöntemlerinin doğrusal olmayan regresyon problemlerine entegrasyonu, bu alandaki araştırmalar için yeni bir perspektif sunabilir. Bu tez çalışması, AI optimizasyon algoritmalarının doğrusal olmayan regresyon modellerinde parametre tahmini için çözüm sağlayan etkin bir alternatif yöntem olduğunu sunmuştur. Elde edilen bulgular hem akademik literatüre hem de endüstriyel uygulamalara metodolojik ve pratik katkılar sunarken bulguların, gelecekteki araştırmalar için algoritma geliştirme konusunda katkı sağlayacağı düşünülmektedir.



KAYNAKLAR

- Abdelwanis, M. I. (2022). Linear induction motor parameter estimation based on Gray Wolves Optimization Algorithm. *2022 23rd International Middle East Power Systems Conference (MEPCON)* (pp. 1-6), Kafrelshiekh University, Egypt. IEEE.
- Al-Betar, M. A. (2021). Island-based harmony search algorithm for non-convex economic load dispatch problems. *Journal of Electrical Engineering and Technology*, 16, 1985–2015.
- Alinezhad, A. and Khalili, J. (2019). New methods and applications in Multiple Attribute Decision Making (MADM). *Springer Nature Switzerland AG*. <https://doi.org/10.1007/978-3-030-15009-9>
- Amavasai, A. and Dijkstra, J. (2024). Particle filter based on Jaya optimisation for Bayesian updating of nonlinear models. *Applied Soft Computing*, 158, 111429.
- Anggreini, D., Sukiyanto, S. and Saputra, B. D. (2023). Population projection with the application of the differential equation of the logistic and exponential model (Case study: Yogyakarta Special Region Province). *Mathline: Jurnal Matematika dan Pendidikan Matematika*, 8(3), 795–804.
- Asokan, A., Popescu, D. E., Anitha, J. and Hemanth, D. J. (2020). Bat algorithm-based non-linear contrast stretching for satellite image enhancement. *Geosciences*, 10(2), Article 78.
- Ayan, B., Abacıoğlu, S. and Basilio, M. P. (2023). A Comprehensive Review of the Novel Weighting Methods for Multi-Criteria Decision Making. *Information*, 14, 285. <https://doi.org/10.3390/info14050285>
- Bala, A., Ismail, I. and Ibrahim, R. (2018). Cuckoo search-based optimization of echo state network for time series prediction. *2018 International Conference on Intelligent and Advanced Systems (ICIAS)*, IEEE, 104–110.
- Chen, B., Ouyang, H., Li, S., Gao, L. and Ding, W. (2023). *Photovoltaic parameter extraction through an adaptive differential evolution algorithm with multiple linear regression*. SSRN Preprints.
- Cheng, G. and Liu, Y. (2024). Hybrid optimization algorithm for solving path planning problems based on Grey Wolf Optimization Algorithm. *Springer Open*. <https://doi.org/10.21203/rs.3.rs-4691285/v1>
- Çakır, E. (2017). Kriter Ağırlıklarının SWARA – Copeland Yöntemi ile Belirlenmesi: Bir Üretim İşletmesinde Uygulama. *Adnan Menderes Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, 4(1), 42–56.

- Çiftaslan, M. E. ve Rençber, Ö. F. (2022). IDOCRIW ve CoCoSo Yöntemleri ile Sistemik Önemli Bankaların Performans Analizi: Türkiye Örneği. *Kahramanmaraş Sütçü İmam Üniversitesi Sosyal Bilimler Dergisi*, 19(özel sayı), 54–72.
- Çilek, A. (2023). Piyasa Yapıcı Bankaların Performanslarının Sıralanması: CILOS, MARCOS ve COPELAND Çok Kıstaslı Karar Verme Analizi. *Pamukkale Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, 54, 1–24. <https://doi.org/10.30794/pausbed.1092801>
- Dudek, A. and Baranowski, J. (2022). Gaussian Processes for Signal Processing and Representation in Control Engineering. *Applied Sciences*, 12(4946). <https://doi.org/10.3390/app12104946>
- Dym, C. L. (2004). Principles of Mathematical Modeling (2nd ed., pp. 117-150). Academic Press. <https://doi.org/10.1016/B978-012226551-8/50006-5>
- Erdoğan, P. ve Ekiz, S. (2017). Doğrusal Olmayan Regresyon Parametrelerinin Sezgisel Yöntemlerle Tahmini. *Turkish Studies: International Periodical for the Languages, Literature and History of Turkish or Turkic*, 12(23), 121–132. <http://dx.doi.org/10.7827/TurkishStudies.12000>
- Favardin, P., Lepelley, D. and Serais, J. (2002). Borda Rule, Copeland Method and Strategic Manipulation. *Review of Economic Design*, 7(3), 213–228.
- Feng, W., Li, M., Zhang, W. and Zhang, H. (2023). Parameter estimation based on novel enhanced self-learning particle swarm optimization algorithm with Levy flight for PMSG. *IET Generation, Transmission & Distribution*, 17(5), 1111–1122.
- Fu, Z., Wang, Z. and Chen, G. (2024). Enhanced parameter estimation with improved particle swarm optimization algorithm for cell culture process modeling. *AIChE Journal*, 70(4), e18388.
- Geem, Z. W., Kim, J. H. and Loganathan, G. V. (2001). A new heuristic optimization algorithm: Harmony search. *Simulation*, 76(2), 60–68.
- Girdzijauskas, S. and Streimikiene, D. (2008). Logistic Growth Models for Analysis of Stock Markets Bubbles. *Proceedings of the World Congress on Engineering 2008 Vol II, WCE 2008*, July 2-4, London, U.K., ISBN: 978-988-17012-3-7.
- Gopal, K., Lee, L. S. and Seow, H.-V. (2021). Parameter estimation of compartmental epidemiological model using Harmony Search algorithm and its variants. *Applied Sciences*, 11(3), 1138.
- Grossman-Ponemon, B. E., Heimisson, E. R., Lew, A. J. and Segall, P. (2020). Logarithmic Growth of Dikes From a Depressurizing Magma Chamber. *Geophysical Research Letters*, 47, e2019GL086230. <https://doi.org/10.1029/2019GL086230>

- Gupta, N., Mai, A., Abouzied, A. and Shasha, D. (2023). On the calibration of compartmental epidemiological models. *arXiv preprint* arXiv:2312.05456. <https://arxiv.org/abs/2312.05456>
- Holland, J. H. (1992). Genetic algorithms. *Scientific American*, 267(1), 66–73.
- Işıldak, B., Keleş, M. K. ve Özdağoğlu, A. (2023). Tekirdağ Çorlu Atatürk Havalimanı'nın Performansının IDOCRIW ve WEDBA Yöntemleriyle Yıllara Göre Değerlendirilmesi. *Humanitas – Uluslararası Sosyal Bilimler Dergisi*, 11(INCSOS VIII Özel Sayısı), 169–182.
- Jabbar, A. K. and Al-saedi, H. M. (2022). Mathematical Modeling of Tumors Growth: Competition based on Gompertz model in Two Dimensions. *Baghdad Science Journal*, 21(8), 2681-2690.
- Kalinina, I. O., Gozhyj, O. P., Nechahin, V. V. and Shiyan, S. I. (2023). Synthesis of the parameters of a nonlinear predictive model using a genetic algorithm. *System Technologies*, 2 (145), 66–75.
- Kang, F., Wu, Y., Li, J. and Li, H. (2021). Dynamic parameter inverse analysis of concrete dams based on Jaya algorithm with Gaussian processes surrogate model. *Advanced Engineering Informatics*, 49, 101348.
- Karaboğa, D. (2005). *An idea based on honey bee swarm for numerical optimization* (Vol. 200, pp. 1-10). Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department.
- Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. *Proceedings of IEEE International Conference on Neural Networks*, 1942–1948.
- Keshavarz Ghorabae, M., Zavadskas, E. K., Olfat, L. and Turskis, Z. (2015). Multi-criteria inventory classification using a new method of evaluation based on distance from average solution (EDAS). *Informatica*, 26(3), 435–451.
- Khuri, A. I. and Cornell, J. A. (1996). *Response surfaces: Designs and analyses (2nd ed.)*. CRC Press. <https://doi.org/10.1201/9780203740774>
- Kumar, P., Sonkar, S. K., George, R. C., Philip, D. and Ghosh, A. K. (2023). *Data-driven approach for estimating longitudinal aerodynamic parameters using neural artificial bee colony fusion algorithm*. *Asian Journal of Control*.
- Lai, X., Wang, X., Nie, Y. and He, X. (2020). An efficient parameter estimation method for nonlinear high-order systems via surrogate modeling and cuckoo search. *Soft Computing*. <https://doi.org/10.1007/s00500-020-04997-3>
- Lara, C. A. and Guerra, C. (2023). Programming an evolutionary algorithm for the estimation of non-linear damping vibration parameters. In *Genetic Algorithms – Theory, Design and Programming*. IntechOpen.

- Lartey, F. M. (2020). Predicting product uptake using bass, gompertz, and logistic diffusion models: Application to a broadband product. *Journal of Business Administration Research*, 9(2), 5-12.
- Lestari, S., Adji, T. B. and Permanasari, A. E. (2018). *Performance Comparison of Rank Aggregation Using Borda and Copeland in Recommender System*. 2018 International Workshop on Big Data and Information Security (IWBIS), 1–6. IEEE. <https://doi.org/10.1109/IWBIS.2018.8471722>
- Maciel, L., Gomide, F. and Ballini, R. (2016). A differential evolution algorithm for yield curve estimation. *Mathematics and Computers in Simulation*, 129, 10–30.
- Martinelli, C., Coraddu, A. and Cammarano, A. (2024). Experimental Parameter Identification of Nonlinear Mechanical Systems via Meta-heuristic Optimisation Methods. In M. R. W. Brake, L. Renson, R. J. Kuether, & P. Tiso (Eds.), *Nonlinear Structures and Systems: Proceedings of the 41st IMAC - A Conference and Exposition on Structural Dynamics 2023 (Vol. 1, pp. 215–223)*. Springer. https://doi.org/10.1007/978-3-031-36999-5_28
- Mirjalili, S., Mirjalili, S. M. and Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software*, 69, 46–61.
- Miyoshi, S., Luc, K.M., Kuchida, K. and Mitsumoto, T. (1996). Application of non-linear models to egg production curves in chickens. *Japanese Poultry Science*, 33(3), 178–184.
- Murugan, R. (2024). Theory on the rate equations of Michaelis-Menten type enzyme kinetics with competitive inhibition. *PLOS ONE*, 19(7), e0302679. <https://doi.org/10.1371/journal.pone.0302679>
- Nguyen V.T. and Mirjalili, S. (2023). MEALPY: An open-source library for latest meta-heuristic algorithms in Python. *Journal of Systems Architecture*, 139, 102871. <https://doi.org/10.1016/j.sysarc.2023.102871>
- Nobahari, H. and Sharifi, A. (2019). A hybridization of extended Kalman filter and Ant Colony Optimization for state estimation of nonlinear systems. *Applied Soft Computing Journal*, 74, 411–423.
- Özçalıcı, M. (2021). *MATLAB ile Çok Kriterli Karar Verme Teknikleri (Güncellenmiş 2. Basım)*. Nobel Akademik Yayıncılık. ISBN: 978-605-320-753-5.
- Paltanea, M., Tabirca, S., Scheiber, E. and Tangney, M. (2010). Logarithmic growth in biological processes. In *2010 12th International Conference on Computer Modelling and Simulation*. IEEE. <https://doi.org/10.1109/UKSIM.2010.29>
- Pamučar, D. and Ćirović, G. (2015). The selection of transport and handling resources in logistics centers using Multi-Attributive Border Approximation Area Comparison (MABAC). *Expert Systems with Applications*, 42(6), 3016–3028.

- Ramsundram, N., Keerthana, M., Arun Kumaar, M. and Shreevarshini, S. (2023). Parameter estimation for non-linear Muskingum flood routing model: Natural inspired algorithm. *International Journal of Scientific Research in Engineering and Management (IJSREM)*, 7(10), 1–8.
- Rao, R. V. (2016). Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. *International Journal of Industrial Engineering Computations*, 7(1), 19–34.
- Rao, R. V., Savsani and V. J., Vakharia, D. P. (2011). Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems. *Computer-Aided Design*, 43(3), 303–315.
- Ray, A. K. (2023). Logistic modelling of economic dynamics. *arXiv preprint arXiv:2309.01139*. <https://arxiv.org/abs/2309.01139>
- Roeva, O. and Trenkova, T. (2012). Genetic Algorithms and Firefly Algorithms for non-linear bioprocess model parameters identification. In *Proceedings of the 4th International Joint Conference on Computational Intelligence (ECTA-2012)*, 164–169.
- Saari, D. G. and Merlin, V. R. (1996). The Copeland method. *Economic Theory*, 8(1), 51–76. Springer-Verlag.
- Sabastine, E., Okoye, I., Ezenweke, C. P., Shobanke, D. A. and Adeniyi, I. A. (2022). estimating nonlinear regression parameters using particle swarm optimization and genetic algorithm. *FUDMA Journal of Sciences (FJS)*, 6(6), 202–213. <https://doi.org/10.33003/fjs-2022-0606-1114>
- Saganuwan, A. S. (2021). Application of modified Michaelis-Menten equations for determination of enzyme inducing and inhibiting drugs. *BMC Pharmacology and Toxicology*, 22(57).
- Sel, A. ve Demir, G. (2020). Çok Kriterli Karar Verme Problemlerinde Kriter Ağırlıklandırma Yöntemleri (H. Bircan, Ed.). Nobel Akademik Yayıncılık.
- Sepulchro, W. d. N. and Encarnaç o, L. F. (2024). Harmonic state estimation in power systems using the Jaya algorithm. *Electronics*, 13(17), 3559.
- Socha, K. and Dorigo, M. (2008). Ant colony optimization for continuous domains. *European Journal of Operational Research*, 185(3), 1155–1173.
- Storn, R. and Price, K. (1997). Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4), 341–359.
- Tefek, M. F. (2021). Artificial bee colony algorithm based on a new local search approach for parameter estimation of photovoltaic systems. *Journal of Computational Electronics*, 20, 2530–2562.

- Turan-Karaoğlu, S. and Göktürkler, G. (2021). Cuckoo search algorithm for model parameter estimation from self-potential data. *Journal of Applied Geophysics*, 194, Article 104461.
- Türkşen, Ö. (2023). *Optimizasyon Yöntemleri ve Matlab, Python, R Uygulamaları*. Nobel Akademi Yayıncılık.
- Türkşen, Ö. and Tez, M. (2016). An Application of Nelder-Mead Heuristic-based Hybrid Algorithms: Estimation of Compartment Model Parameters. *International Journal of Artificial Intelligence*, 14(1), 112–129.
- Türkşen, Ö. ve Akgün, F. (2018). Genetik-Simpleks hibrit algoritması ile doğrusal olmayan regresyon model parametrelerinin nokta tahmini. *İstatistikçiler Dergisi: İstatistik ve Aktüerya*, 11(2), 81-92. <http://www.istatistikciler.org>
- Ulu Metin, G. (2024). *Ar-Ge ve Tasarım Merkezlerinin İstatistiksel Olarak Değerlendirilmesi: Veri Madenciliği Yöntemleri ile Hibrit Karar Verme* (Doktora tezi). Ankara Üniversitesi, Fen Bilimleri Enstitüsü, İstatistik Ana Bilim Dalı.
- Welagedara, W. A. D. M., Nawarathna, L. S. and Nawarathna, R. D. (2019). Forecasting the Sri Lankan population with the Gompertz and Verhulst Logistic Growth Models. *Sri Lanka Journal of Economic Research*, 07(1), 1-12.
- Widodo, W., Raflesia, F., Alawiyah, S., Setianingsih, S., Santoso, D., Parnadi, W. W. and Fatkhan. (2022). *Flower pollination and elitism algorithms for inversion of TDEM data*. *J. Math. Fund. Sci*, 54(1), 121–137.
- Yakut, E. (2020). OECD Ülkelerinin Bilgi ve İletişim Teknolojileri Gelişmişliklerinin MOORA ve WASPAS Yöntemiyle Değerlendirilerek Kullanılan Yöntemlerin Copeland Yöntemiyle Karşılaştırılması. *Atatürk Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, 24(3), 1275–1294
- Yang, B., Zhao, S. and Zhi, H. (2023). Parameter estimation of photovoltaic cell model based on improved particle swarm optimization. *2023 IEEE 3rd International Conference on Data Science and Computer Application (ICDSCA)*, Dalian, China. IEEE.
- Yang, X.-S. (2010). A new metaheuristic bat-inspired algorithm. In J. R. Gonzalez et al. (Eds.), *Nature Inspired Cooperative Strategies for Optimization (NISCO 2010)* (Vol. 284, pp. 65–74). Springer.
- Yang, X.-S. (2012). *Flower pollination algorithm for global optimization*. *LNCS 7445: Unconventional Computation and Natural Computation*, 240–249. Springer.
- Yang, X.-S. and Deb, S. (2009). *Cuckoo search via Lévy flights*. *2009 World Congress on Nature and Biologically Inspired Computing (NaBIC 2009)*, IEEE, 210–214.
- Yonar, A. and Yonar, H. (2022). An Efficient hybrid algorithm with particle swarm optimization and nelder-mead algorithm for parameter estimation of nonlinear

regression modeling. *Gazi University Journal of Science*, 35(2), 716–729.
<https://doi.org/10.35378/gujs.864980>

Zavadskas, E. K. and Podvezko, V. (2016). Integrated determination of objective criteria weights in MCDM. *International Journal of Information Technology and Decision Making*, 15(02), 267–283.

Zavadskas, E. K., Kaklauskas, A., Peldschus, F. and Turskis, Z. (2007). Multi-attribute assessment of road design solutions by using the COPRAS method. *The Baltic Journal of Road and Bridge Engineering*, 2(4), 195–203.

Zhang, Z., Lin, J. and Shi, Y. (2013). Joint angle-frequency estimation based on WSF using Artificial Bee Colony algorithm. *In Proceedings of the Third International Conference on Information Science and Technology (ICIST)*, 1312–1315.
<https://doi.org/10.1109/ICIST.2013.6747643>

