



**T.C.
AKSARAY ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

MATEMATİK ANABİLİM DALI

**GRİ KURT VE KAMBUR BALINA OPTİMİZASYONLARININ
MATEMATİKSEL MODELLEMELER ÜZERİNE
UYGULAMALARI**

**YÜKSEK LİSANS TEZİ
Abdulsamed ÖZKEPİR**

**DANIŞMAN
Prof. Dr. Halis BİLGİL**

**İKİNCİ DANIŞMAN
Dr. Öğr. Üyesi Ebutalib ÇELİK**

AKSARAY, 2024

Aksaray Üniversitesi Fen Bilimleri Enstitüsü'nün 222342706 numaralı Yüksek Lisans öğrencisi Abdulsamed ÖZKEPİR tarafından hazırlanan “**GRİ KURT VE KAMBUR BALINA OPTİMİZASYONLARININ MATEMATİKSEL MODELLEMELER ÜZERİNE UYGULAMALARI**” adlı tez çalışması aşağıdaki jüri tarafından OY BİRLİĞİ ile Matematik Anabilim Dalında YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Danışman: Prof. Dr. Halis BİLGİL

Kayseri Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum/onaylamıyorum.....

Üye: Dr. Öğr. Üyesi Hasan ARSLAN

Erciyes Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum/onaylamıyorum.....

Üye: Doç. Dr. Tunçar ŞAHAN

Aksaray Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum/onaylamıyorum.....

Tez Savunma Tarihi: 28/06/2024

Jüri tarafından kabul edilen bu tezin Yüksek Lisans Tezi olması için gerekli şartları yerine getirdiğini onaylıyorum.

.....

Prof. Dr. Mehmet Ali HINIS

Fen Bilimleri Enstitüsü Müdürü

DOĞRULUK BEYANI

Yüksek lisans tezi olarak sunduğum bu çalışmayı, akademik kurallara ve bilimsel etik, ahlak ve geleneklere aykırı düşecek bir yol ve yardıma başvurmaksızın yazdığımı, yararlandığım eserlerin kaynakçada gösterilenlerden oluştuğunu, çalışmamda kullandığım verilerin orijinalliğini ve her türlü intihalden uzak olduğunu beyan ederim.

Enstitü tarafından belli bir zamana bağlı olmaksızın, tezimle ilgili yaptığım bu beyana aykırı bir durumun saptanması durumunda, ortaya çıkacak tüm ahlaki ve hukuki sonuçlara katlanacağımı bildiririm.

Abdulsamed ÖZKEPİR

TEŞEKKÜR

Aksaray Üniversitesi, Fen Edebiyat Fakültesi, Matematik Anabilim Dalı'nda yapmış olduğum bu yüksek lisans tez çalışmam süresince bilgi ve deneyimleri ile beni yönlendirerek destekleyen ve tezin her aşamasında yapıcı eleştirileriyle katkı sağlayan danışman hocam sayın Prof. Dr. Halis BİLGİL'e ve çalışmaların her aşamasında her türlü yardım ve desteklerini eksik etmeyen Dr. Öğr. Üyesi Ebutalib ÇELİK'e teşekkür ederim. Bu dönemde her zaman yanımda olan sevgilerini esirgemeyen tüm aile fertlerime ve son olarak hayatta benim için her şeyden önemli olan ve her zaman yanımda olan sevgili eşime ve oğluma teşekkür etmeyi bir borç bilirim.

Abdulsamed ÖZKEPİR

AKSARAY, 2024

İÇİNDEKİLER

TEŞEKKÜR	i
İÇİNDEKİLER	ii
ÖZET.....	iii
ABSTRACT	iv
ŞEKİLLER DİZİNİ	v
ÇİZELGELER DİZİNİ	vi
SİMGELER VE KISALTMALAR	vii
1. TEMEL KAVRAMLAR	1
1.1 Giriş	1
1.2 Literatür Taraması	1
1.3 Optimizasyon.....	3
1.4 Optimizasyon Algoritmalarının Sınıflandırılması.....	3
1.5 Meta-Sezgisel Optimizasyon Algoritması.....	4
2. GRİ KURT OPTİMİZASYONU (GREY WOLF OPTİMİZER-GWO).....	5
2.1 Avın Kuşatılması.....	6
2.2 Avlanma	7
2.3 Ava Saldırma	8
2.4 Av Arama	10
3. BALİNA OPTİMİZASYON ALGORİTMASI (BOA).....	25
3.1 Algoritmanın İlham Kaynağı.....	25
3.2 Matematiksel Model.....	27
3.2.1 Avı çevreleme	27
3.2.2 Kabarcık ağı yöntemi (Saldırı aşaması).....	29
3.2.2.1 Daralan kuşatma mekanizması	30
3.2.2.2 Spiral hareket.....	30
3.2.3 Av Arama.....	31
4. GRİ KURT VE KAMBUR BALİNA OPTİMİZASYON YÖNTEMLERİNİN BİR UYGULAMASI	45
4.1 Uyumlu Kesir Hesabına Dair Bazı Tanımlar ve Özellikler	45
4.1.1 Uyumlu kesir türevi	45
4.2 Kesirli Gri Model Formülasyonu	46
5. SONUÇ.....	54
KAYNAKÇA	55
ÖZGEÇMİŞ.....	57

YÜKSEK LİSANS TEZİ

GRİ KURT VE KAMBUR BALINA OPTİMİZASYONLARININ MATEMATİKSEL MODELLEMELER ÜZERİNE UYGULAMALARI

Abdulsamed ÖZKEPİR

**Aksaray Üniversitesi
Fen Bilimleri Enstitüsü
Matematik Anabilim Dalı**

**Danışman: Prof. Dr. Halis BİLGİL
İkinci Danışman: Dr. Öğr. Üyesi Ebutalib ÇELİK**

ÖZET

Gri Kurt ve Kambur Balina Optimizasyonlarının Matematiksel Modellemeler Üzerine Uygulamaları başlıklı bu çalışma, Gri Kurt (Grey Wolf Optimizer) ve Kambur Balina Optimizasyonu (Humpback Whale Optimization) algoritmalarının matematiksel modellemeler üzerindeki uygulamalarını incelemektedir. İlk olarak, Gri Kurt Optimizasyonu, kurt sürülerinin liderlik hiyerarşisinden ilham alarak, doğal bir şekilde doğadaki avlanma sürecini taklit eder. İkinci olarak, Kambur Balina Optimizasyonu, kambur balinaların avlanma stratejisinden esinlenerek geliştirilmiştir. Bu algoritma, balinaların sosyal davranışlarını ve iletişimlerini temel alarak, optimizasyon problemlerine çözüm üretir. Bahsi geçen algoritmaların matematiksel modeller üzerindeki başarılı uygulamalarını sunulmakta ve bu algoritmaların gerçek hayattan esinlenen optimizasyon problemlerindeki performanslarını değerlendirmektedir. Sonuçlar, Gri Kurt ve Kambur Balina Optimizasyonlarının çeşitli matematiksel modellemelerde etkili ve verimli çözümler üretebildiğini göstermektedir. Bu iki algoritmanın bilimsel ve endüstriyel alanlarda çeşitli uygulamalara sahip olabileceği vurgulanmaktadır.

Anahtar Kelimeler: Gri Kurt Optimizasyonu, Kambur Balina Optimizasyonu, Matematiksel Modelleme, Algoritma, Simülasyon

Haziran, 2024; 57 sayfa

M.Sc. THESIS

**APPLICATIONS OF GRAY WOLF AND HUMPTY WHALE
OPTIMIZATIONS ON MATHEMATICAL MODELING**

Abdulsamed ÖZKEPİR

**Aksaray University
Institute of Science and Technology
Department of Mathematics**

**Supervisor: Prof. Dr. Halis BİLGİL
Co-Supervisor: Ass. Prof. Dr. Ebutalib ÇELİK**

ABSTRACT

This paper, titled Applications of Grey Wolf and Humpback Whale Optimization on Mathematical Modeling, investigates the applications of the Grey Wolf Optimizer and Humpback Whale Optimization algorithms on mathematical modeling. First, the Grey Wolf Optimizer naturally mimics the hunting process in the wild, inspired by the leadership hierarchy of wolf packs. Second, Humpback Whale Optimization is inspired by the hunting strategy of humpback whales. This algorithm generates solutions to optimization problems based on the social behavior and communication of whales. It presents successful implementations of these algorithms on mathematical models and evaluates their performance on real-life inspired optimization problems. The results show that Gray Wolf and Humpback Whale Optimization can produce effective and efficient solutions in various mathematical models. It is emphasized that these two algorithms can have various applications in scientific and industrial fields.

Keywords: Gray Wolf Optimization, Humpback Whale Optimization, Mathematical Modeling, Algorithm, Simulation

June, 2024; 57 pages

ŞEKİLLER DİZİNİ

Şekil 1.1. Optimizasyon algoritmalarının sınıflandırılması	1
Şekil 1.2. Meta-sezgisel optimizasyon algoritmalarının sınıflandırılması.....	2
Şekil 2.1. Gri kurtların algoritma liderlik hiyerarşisi.....	6
Şekil 2.2. Gri kurtların avlanma stratejisi	8
Şekil 2.3. Avını ararken ve avına saldırırken gri kurtlar	9
Şekil 2.4. 2 ve 3 boyutlu vektörel ve bir sonraki adımda olası tüm pozisyonlar.	10
Şekil 2.5. Optimizasyon algoritmasının kaba kodu	11
Şekil 2.6. Kurtların(ajanların) pozisyonları ve konumları.	18
Şekil 3.1. Kambur balinaların kabarcık ağı şeklindeki avlanma taktiği	26
Şekil 3.2. Arama ajanlarının a) İki boyutlu alanda b) Üç boyutlu alanda konum vektörleri ve olası konumları.	28
Şekil 3.3. BOA' da uygulanan keşif uygulaması (X * rastgele seçilen bir arama ajanı).....	29
Şekil 3.4. Spiral şeklindeki pozisyon güncelleme aşaması.....	29
Şekil 3.5. Kambur balina optimizasyon algoritması adımları ve akış diyagramı	32
Şekil 3.6. Balinaların (Ajanların) avlanma sırasındaki hareketleri.	39
Şekil 4.1. α hesabında kullanılan algortimanın kaba kodu.....	51
Şekil 4.2. Gri kurt yöntemi.	52
Şekil 4.3. Kambur balina yöntemi.	52

ÇİZELGELER DİZİNİ

Çizelge 2.1. Örnek 2.1 için arama uzayındaki rastgele değerler.	12
Çizelge 2.2. Örnek 2.1 için başlangıç arama ajanı (α, β, δ).	12
Çizelge 2.3. Örnek 2.1 için arama uzayındaki ajanların yeni konumları.....	17
Çizelge 2.4. Örnek 2.1 için arama uzayındaki ajanların başlangıç, birinci ve ikinci adımdaki konumları.	17
Çizelge 2.5. Örnek 2.1 için arama uzayındaki ajanların başlangıçtan otuzuncu adıma kadar bazı konumları.....	17
Çizelge 2.6. Örnek 2.2 için arama uzayındaki rastgele değerler.	18
Çizelge 2.7. Örnek 2.2 için başlangıç arama ajanı (α, β, δ).	19
Çizelge 2.8. Örnek 2.2 için arama uzayındaki ajanların yeni konumları.....	22
Çizelge 2.9. Örnek 2.2 için ajanların daha büyük adımdaki konumları.....	22
Çizelge 2.10. Örnek 2.3 için başlangıç arama ajanı (α, β, δ).	23
Çizelge 3.1. Örnek 3.1 için arama uzayındaki rastgele değerler.	33
Çizelge 3.2. Örnek 3.1 arama uzayındaki ajanlarımızın yeni konumları.....	35
Çizelge 3.3. Örnek 3.1 ajanların daha büyük adımdaki konumları.....	35
Çizelge 3.4. Örnek 3.2 arama uzayındaki rastgele değerler.	36
Çizelge 3.5. Örnek 3.2 için arama uzayındaki ajanların yeni konumları.....	38
Çizelge 3.6. Örnek 3.2 için ajanların daha büyük adımdaki konumları.....	38
Çizelge 3.7. Örnek 3.3 için arama uzayındaki rastgele değerler.	39
Çizelge 3.8. Örnek 3.3 için av arama için gerekli olan tüm parametreler.	40
Çizelge 3.9. Örnek 3.3 için arama uzayındaki ajanların yeni konumları.....	40
Çizelge 3.10. Örnek 3.3 için ajanların daha büyük adımdaki konumları.....	41
Çizelge 3.11. Örnek 3.4 için arama uzayındaki rastgele değerler.	41
Çizelge 3.12. Örnek 3.4 için arama uzayındaki ajanların yeni konumları.....	43
Çizelge 3.13. Örnek 3.4 için ajanların daha büyük adımdaki konumları.....	44
Çizelge 4.1. Çin'in yıllık rüzgar enerjisi tüketim verileri.	50
Çizelge 4.2. Çin'in yıllık rüzgar enerjisi tüketim verileri tahminleri.	53

SİMGELER VE KISALTMALAR

GWO	Gri Kurt Optimizasyonu
BOA	Kanbur Balina optimizasyonu



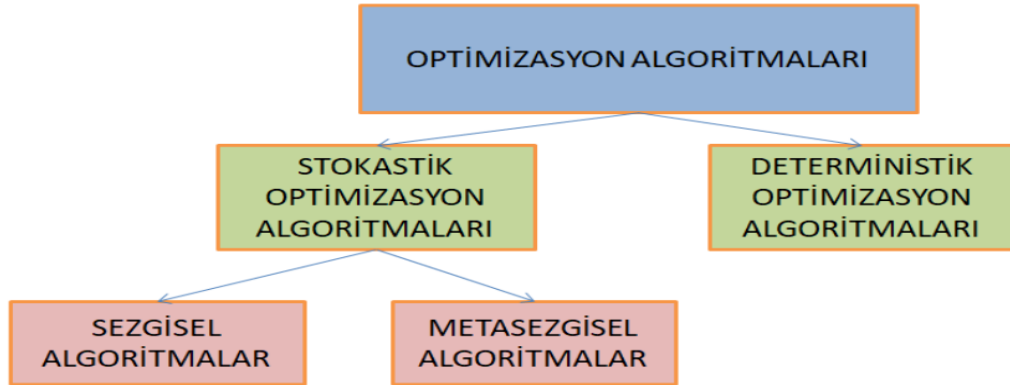
1. TEMEL KAVRAMLAR

1.1 Giriş

Optimizasyon mevcut verileri en iyi şekilde kullanıp hedefe ulaşmak ve sorunları gidermek anlamına gelir. Optimize etmek mevcut problemleri çözüme ulaştırmak için hamle yapmak demektir. Matematik, mühendislik gibi birçok alanda optimizasyon problemleriyle karşılaşmaktadır. Optimizasyon problemlerinde belirli koşullarda bir fonksiyonun maksimum veya minimum noktalarının tespiti ya da optimal çözümlerinin elde edilmesi amaçlanmaktadır [1].

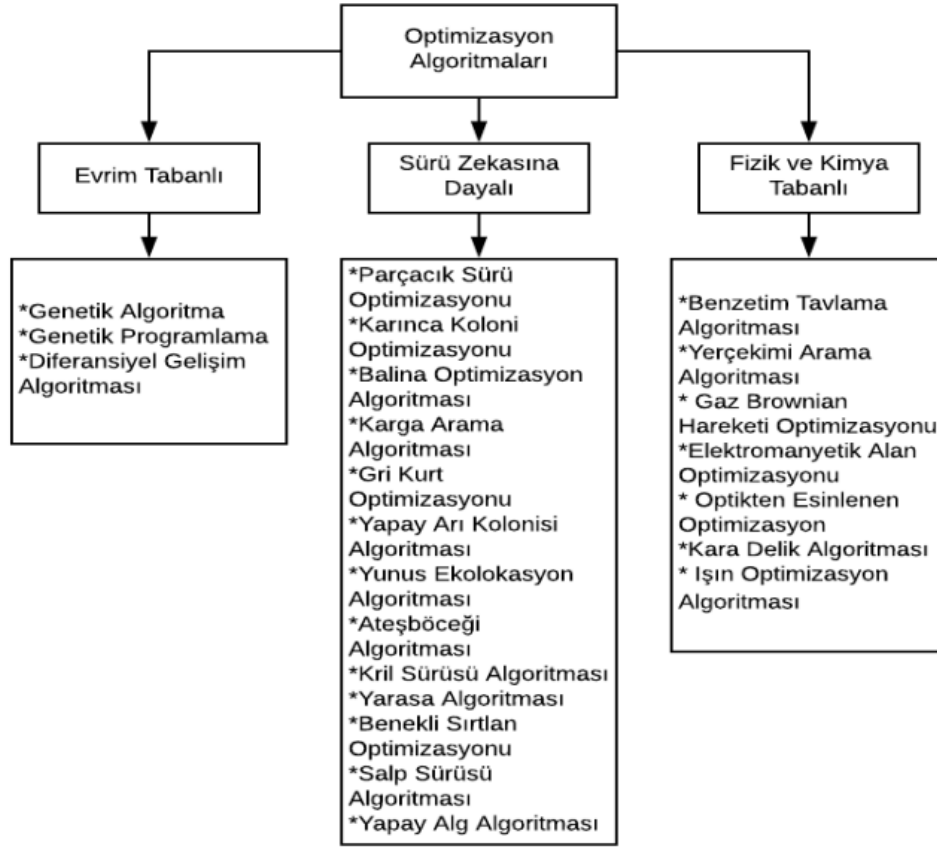
1.2 Literatür Taraması

Optimizasyon problemleri Şekil 1.1’de görüldüğü gibi deterministik optimizasyon algoritmaları ve deterministik optimizasyon algoritmalarına rastgelelik eklenerek üretilen stokastik optimizasyon algoritmaları olarak ikiye ayrılır. Stokastik optimizasyon algoritmaları ise sezgisel algoritmalar ve meta-sezgisel algoritmalar olarak ikiye ayrılır [2].



Şekil 1.1. Optimizasyon algoritmalarının sınıflandırılması.

Meta-sezgisel optimizasyon algoritmaları genellikle Şekil 1.2’de görüldüğü gibi evrim tabanlı algoritmalar, sürü zekasına dayalı algoritmalar, fizik ve kimya tabanlı algoritmalar olmak üzere üç ana kategoriye ayrılır.



Şekil 1.2. Meta-sezgisel optimizasyon algoritmalarının sınıflandırılması.

2016 yılında Seyedali Mirjalili ve Andrew Lewis tarafından önerilen Balina Optimizasyon Algoritması [3], kambur balinaların sosyal davranışını modeller ve kabarcık-ağ avlama stratejisinden esinlenmiştir. 2014 yılında Seyedali Mirjalili, Seyed Mohammad Mirjalili ve Andrew Lewis tarafından önerilen Gri Kurt Optimizasyonu [4], doğadaki gri kurtların liderlik hiyerarşisini ve avlanma mekanizmasını taklit eder.

Gri kurt ve Kambur balina optimizasyonları hayvanların avlanma davranışları esinlenerek oluşturulmuş sürü temelli bir meta-sezgisel algoritmadır. Meta-sezgisel optimizasyon algoritmaları, son yıllarda mühendislik uygulamalarında sıklıkla kullanılmaya başlanmıştır. Bu algoritmalar, çözümlerinin kolay olması, türev gerektirmeyen yapıları ve yerel minimumu aşabilmeleri gibi çeşitli avantajlara sahiptir [3]. Ayrıca, farklı alanlardaki problemlere çözüm üretebildikleri için geniş kullanım alanları bulunmaktadır. Sürü zekasına dayalı algoritmalar; genellikle biyolojik olaylardan ilham alan, belirli bir grup canlı türünün yaşam mücadelesi içerisinde hedeflerini elde etmek için gerçekleştirdikleri davranışların matematiksel olarak modellenip geliştirilmiş algoritmalarıdır.

1.3 Optimizasyon

Optimizasyon (en iyileme), belirli kısıtlar dâhilinde, karmaşık yapıdaki sistemlerin bilinmeyenlerinin (parametrelerin) çözümünde en iyi sonucun elde edilmesi yöntemi olarak tanımlanmaktadır. Optimizasyon yöntemlerinin genel amacı; üzerinde çalışılan problemler için hesaplama maliyetlerinin düşürülmesi ve yüksek oranda verimli sonuçlar elde edilmesidir. Bu nedenle, optimizasyonun kullanımı, hemen hemen mühendisliğin her dalında ve günlük hayatta karşılaşılabilecek problemlerde son yıllarda oldukça yaygınlaşmış ve geliştirilmiştir [6].

Optimizasyon problemlerinde karar değişkenlerinin sayısı problemin boyutunu verir. Eğer karar değişkeni sadece bir tane ise tek boyutlu optimizasyon problemi olarak adlandırılırken karar değişkeninin sayısının birden fazla olmasında ise çok boyutlu optimizasyon problemi olarak adlandırılır [2].

1.4 Optimizasyon Algoritmalarının Sınıflandırılması

Optimizasyon problemlerinin çözümü için geliştirilmiş algoritmalar optimizasyon algoritmaları olarak isimlendirilir. Genel olarak optimizasyon algoritmaları deterministik optimizasyon algoritmaları ve stokastik optimizasyon algoritmaları olarak iki ana kategoriye ayrılır. Deterministik algoritmalar aynı başlangıç değerleri kullanılarak gerçekleştirilen her çalışmada aynı sonuçları üretir ve problemin kesin çözümünü garanti ederler. Stokastik algoritmalar ise rastgelelik içerdiğinden dolayı aynı başlangıç değerleri kullanılarak koşturulsalar bile genellikle birbirlerine çok yakın fakat farklı sonuçlar üretirler ancak kesin çözümü garanti etmeyip kesin çözüm yakınındaki çözümü garanti ederler [7].

Deterministik algoritmalar kolay problemlerin çözümünde ve lokal optimumu bulma konusunda oldukça başarılıdır fakat bu algoritmaların lokal optimuma yakalanıp global optimuma ulaşamama riskleri mevcuttur. Deterministik algoritmaların lokal optimuma yakalanmasını önlemek amacıyla algoritmaya rastgelelik eklenerek stokastik yetenekler kazandırılır [8].

1.5 Meta-Sezgisel Optimizasyon Algoritması

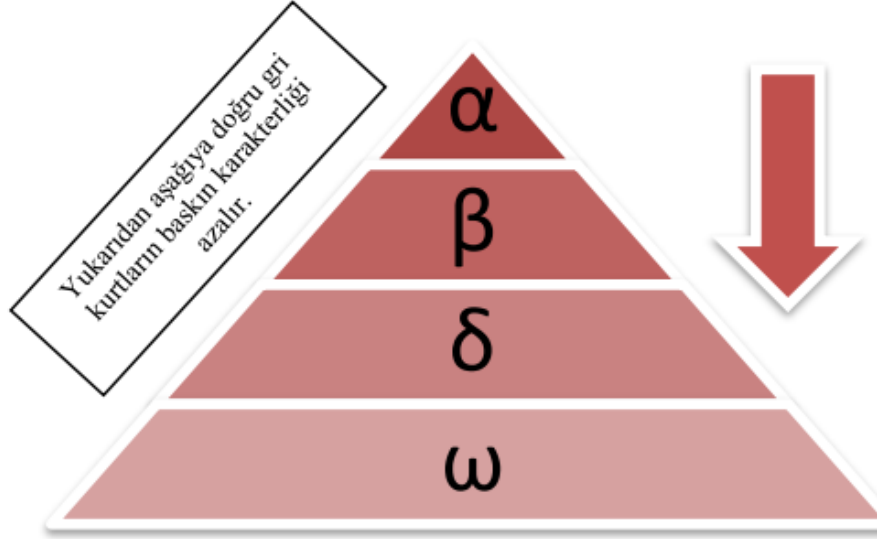
Meta sezgisel terimi “üst seviye sezgisel” şeklinde ifade edilmektedir. Üst seviye sezgisel yaklaşım, çözüm uzayında olasılık temelli ancak bilinçli bir mantıkla arama gerçekleştiren yöntemleri içermektedir. Bu yöntemler her aşamada gerçekleştirilen çözüm kümesinden hareketle yeni çözümler üretmektedirler. Böylece arama uzayının en uygununa yakın olan noktalarında aramalar yapılarak, yerel en iyi nokta seçiminden de kurtularak en uygun çözüme ulaşmaya çalışır. Geleneksel optimizasyon algoritmalarının dezavantajları arasında optimum çözüme ulaşmaya çalışırken ıraksama ve global minimum veya maksimumdan çıkıp daha dar bir çözüm bölgesi içerisinde kalmaları sayılabilir. Problem çözümlerinde bu dezavantajları ortadan kaldırmak için yeni bir yaklaşım olan sezgisel yöntemler geliştirilmiştir. Sezgisel yöntemlerin geliştirilmesinde amaç kısaca karmaşık yapıdaki problemleri veya sistemleri daha basit hale getirerek en iyi çözüme ulaşmak denilebilir. Sezgisel algoritmalar en iyi çözümü garanti edemezler. Bu tür algoritmalar yakınsama özelliğinden dolayı çözüm arama bölgesi içerisinde, en kısa sürede bulunan optimum çözümün civarındaki bir çözümü seçerler. Sezgisel algoritmalarda çözümün doğruluğunu kanıtlama yetkinliği güdülmez.

Sezgisel algoritmaların başlıca kullanım amaçları arasında, araştırmacılar için karmaşık yapıdaki problemleri basitleştirip anlaşılabilirliğini artırmak, optimizasyon problemlerinde kesin çözümün tanımlanamadığı bir yapıda veya sistemde olması, kesin çözümü bulmayı amaçlayan işlemlerde bir ön öğreniminin gerçekleşmesi olarak sıralanabilir [7].

2. GRİ KURT OPTİMİZASYONU (GREY WOLF OPTİMİZER-GWO)

Gri kurt optimizasyon algoritması (GWO), 2014 yılında Mirjalili tarafından üretilen gri kurtların avlanma stratejisini ve sosyal liderliğini taklit eden bir optimizasyon algoritmasıdır. Gri kurtlar genellikle grup olarak yaşamayı tercih ederler. Grup büyüklüğü ortalama 5–12 birey arasındadır. Lider kurtlar alfa kurdu olarak adlandırılırlar. Gri kurtların hiyerarşisi, alfa, beta, delta ve omega kurtları olmak üzere dört grup şeklindedir (Şekil 2.1). Lider ya da baskın kurda alfa kurdu denir ve alfa kurt gruptaki diğer kurtları yönetmek açısından en iyi kurttur. Bunlar bir dişi bir de erkek kurttan oluşur ve genellikle avlanma, uyuma yeri, uyanma zamanı ve benzeri konularda karar vermekle sorumludurlar. Kurt grubunun sosyal hiyerarşisindeki ikincisi ise beta kurttur. Beta birçok etkinlikte lider kurdun (alpha'nın) yardımcısı konumundadır. Beta, alfanın ölmesi veya çok yaşlanması gibi durumlarda alfanın yerine geçecek en muhtemel adaydır. Beta, alfadan aldığı emirleri uygulamanın yanı sıra hiyerarşi piramidindeki düşük seviyedeki kurtlara ise emirler verir. Ayrıca alfanın komutlarının sürüye aktarılmasına yardımcı olur ve alfaya farklı geri bildirimler verir. Delta kurdu alfa ve beta kurtlarına uymakla zorunlu üçüncü kurttur ve sadece omega kurtlarına hükmedebilirler. İzçiler, gözcüler, yaşlılar, avcılar ve bekçiler deltayı oluşturur. İzçiler, bölgenin sınırlarını gözetlemek ve herhangi bir tehlike durumunda sürüyü uyarmakla sorumludur. Gözcüler, sürünün güvenliğini sağlar. Yaşlılar, daha önceden alfa ya da betaya ait olan tecrübeli kurtlardır. Avcılar, avlanma sırasında alfa ve betaya yardım eder ve sürüye yiyecek sağlar. Bekçiler ise sürüdeki hasta, yaralı ve zayıf kurtlara bakmakla sorumludur. Yani omega kurdu hiyerarşide en alt seviyedeki gri kurttur.

En alt seviyedeki omega kurtlar (ω) sürünün büyük bir kısmını kaplar ve esas olarak nüfusun iç ilişkilerini dengelemekten ve genç kurt nüfusunu korumak ve gözetlemekten sorumludur. Gri Kurt Optimizasyon algoritması, gri kurtların yukarıda anlatıldığı gibi avcılık davranışını ve toplumsal davranışlarını esinlenerek önerilmiştir. GWO'nun ana bölümleri avı çevreleme, avlama ve avına saldırma aşamaları oluşturmaktadır [4].



Şekil 2.1. Gri kurtların algoritma liderlik hiyerarşisi [5].

2.1 Avın Kuşatılması

Gri Kurt Optimizasyonun oluşturulmasında sosyal hiyerarşinin dışında gri kurtlara özgü avlanma davranışı da bir diğer önemli sosyal davranıştır. Gri kurtların avlanma davranışını; takip etme, avın çevrenmesi ve ava doğru saldırı olarak açıklamıştır. Avlama sırasında gri kurtların avını kuşatması matematiksel olarak denklem (2.1) ve (2.2) olduğu gibi modellenmiştir.

$$\vec{D}_p = |\vec{C}\vec{X}_p(t) - \vec{X}(t)| \quad (2.1)$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D}_p \quad (2.2)$$

Burada t iterasyon sayısı, $\vec{A}$ ve $\vec{C}$ sırasıyla Denklem (2.3) ve (2.4) ile elde edilen katsayı vektörleri, $\vec{X}_p$ avın konumu ve $\vec{X}$ popülasyondaki herhangi bir bireyin mevcut konumudur.

$$\vec{A} = 2 \cdot \vec{a} \cdot \vec{r}_1 - \vec{a} \quad (2.3)$$

$$\vec{C} = 2 \cdot \vec{r}_2 \quad (2.4)$$

a 'nın değeri iterasyon boyunca doğrusal olarak 2'den 0'a azalır. $\vec{r}_1$ ve $\vec{r}_2$, $[0,1]$ arasında rastgele bir vektördür. $\vec{r}_1$ ve $\vec{r}_2$ rastgele vektörler, kurtların arama uzayındaki noktalardan herhangi bir yere ulaşmasına olanak sağlar. Böylelikle, gri kurt, rastgele

konumda yer alan avın etrafında bulunan boşluk içerisindeki konumunu Denklem (2.1)-(2.2)'ye göre günceller. Aynı şekilde, iki ve üç boyutlu uzayı, n boyutlu bir arama uzayına genişletilebilir ve böylelikle gri kurtların, şimdiye kadar elde ettikleri en iyi çözüm etrafında hareket etmesi sağlanacaktır. Böylece bir gri kurt kendi konumunu (2.1) ve (2.2) denklemlerini kullanarak alan dahilinde avının etrafındaki boş tüm konumlara güncelleyebilir [4]. (2.1) ve (2.2) denkleminin yarattığı etkileri görebilmek için Şekil 2.4 (a)'ya bakmak gereklidir. Şekil 2.4(a)'da iki boyutlu pozisyon vektörleri ve birbirilerine yakın olan bazı komşu kurtlar gösterilmektedir. Bu şekilde de görüldüğü gibi (X,Y) pozisyonunda olan bir kurt kendi konumunu (X^*,Y^*) pozisyonunda olan bir ava göre güncelleyebilir. Kurt kendi mevcut konumunu en iyi ajanın etrafındaki herhangi bir yere A ve C vektörlerinin değerlerini ayarlayarak güncelleyebilir. Örneğin (X^*-X,Y^*-Y) pozisyonuna (X,Y) pozisyonundaki kurdun $\vec{A}$ ve $\vec{C}$ değerlerini $\vec{A} = (1,0)$ ve $\vec{C} = (1,1)$ şeklinde ayarlayarak ulaşılabilir. Bir kurdun 3 boyutlu alandaki ulaşabileceği tüm mümkün pozisyonlar Şekil 2.4(b)'de gösterilmektedir. Yine $\vec{r}_1$ ve $\vec{r}_2$ değerleri sayesinde bir kurdun ulaşabileceği tüm konumlar da Şekil 2.2 görselinden kolayca gözükmemektedir.

2.2 Avlanma

Gri kurtlar avını tanıma ve kuşatma yeteneğine sahiptir. Av genellikle alfa (α) tarafından yönlendirilir. Beta (β) ve delta (δ) ara sıra avlanmaya katılabilir. Soyut bir arama uzayında optimum konum hakkında hiçbir bilgi yoktur. Gri kurtların avlanma davranışını matematiksel olarak modellemek için alfa (α) (en iyi aday çözüm), beta (β) ve deltanın (δ) avın potansiyel konumu hakkında daha iyi bilgiye sahiptirler. Bundan dolayı, elde edilen çözümlerden en iyi ilk üç çözüm kaydedilir. Diğer kurtların en iyi arama ajanlarının pozisyonları da bu kaydedilen çözüme göre güncellenir. Bu bağlamda Denklem (2.5) ve Denklem 2.11 kullanılır [9].

$$\vec{D}_\alpha = |\vec{C}_1\vec{X}_\alpha - \vec{X}| \quad (2.5)$$

$$\vec{D}_\beta = |\vec{C}_2\vec{X}_\beta - \vec{X}| \quad (2.6)$$

$$\vec{D}_\delta = |\vec{C}_3\vec{X}_\delta - \vec{X}| \quad (2.7)$$

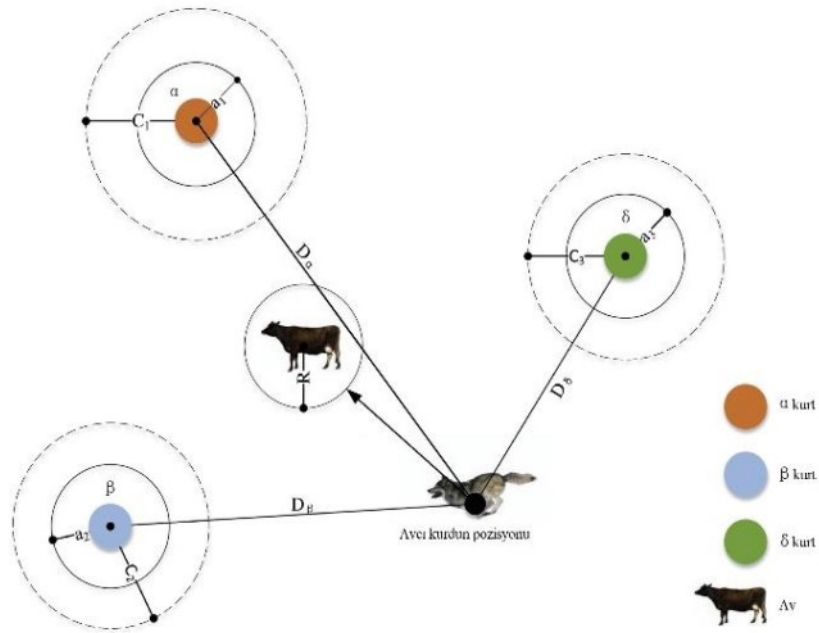
$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 \cdot \vec{D}_\alpha \quad (2.8)$$

$$\vec{X}_2 = \vec{X}_\beta - \vec{A}_2 \cdot \vec{D}_\beta \quad (2.9)$$

$$\vec{X}_3 = \vec{X}_\delta - \vec{A}_3 \cdot \vec{D}_\delta \quad (2.10)$$

$$\vec{X}_{(t+1)} = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad (2.11)$$

Eşitlik (2.5) – (2.11) ‘e göre $\vec{D}_\alpha, \vec{D}_\beta, \vec{D}_\delta$ sırasıyla alfa, beta, delta kurtlarının av ile arasında olan mesafeleri, $\vec{X}_\alpha, \vec{X}_\beta, \vec{X}_\delta$, alfa, beta ve delta kurtları için avın konumunu, $\vec{X}$ gri kurdun t. iterasyondaki konumunu, $\vec{C}_1, \vec{C}_2, \vec{C}_3, \vec{A}_1, \vec{A}_2, \vec{A}_3$ alfa, beta ve delta kurtlarının katsayı vektörlerini, $\vec{X}_1, \vec{X}_2, \vec{X}_3$ alfa, beta ve delta kurtları için deneme vektörlerini göstermektedir. Şekil 2.2’de gri kurtların avlanma stratejisi gösterilmiştir.



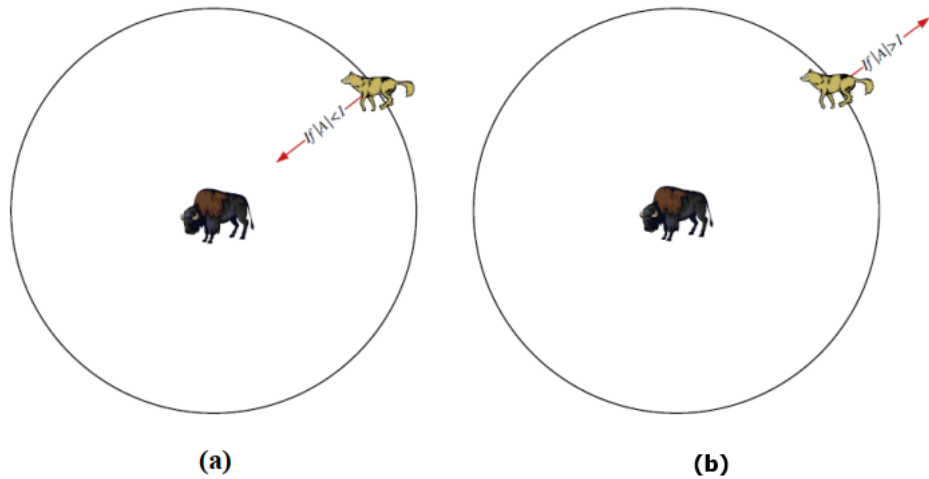
Şekil 2.2. Gri kurtların avlanma stratejisi [5].

2.3 Ava Saldırma

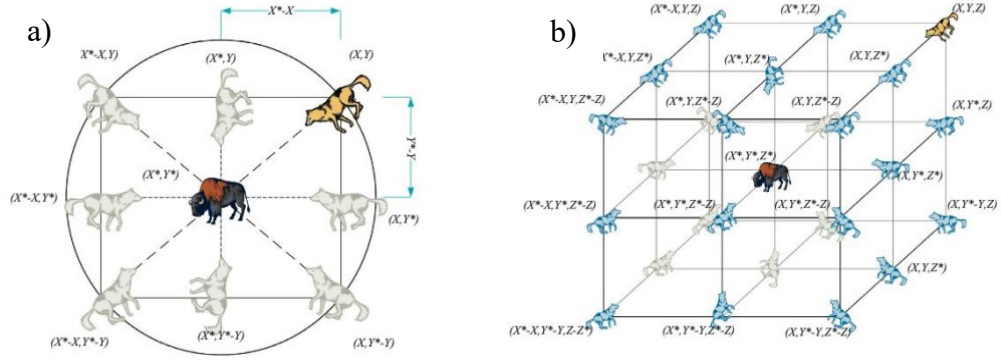
Gri kurtlar avları sabit durduklarında onlara saldırarak avlanmalarını bitirirler. Avlarına yavaşça ve sessizce yaklaşmalarını matematiksel olarak modellemek için ise

α parametresi kullanılır. Bu parametrenin değerinin gittikçe düşmesi ajanın avına yaklaşması manasına gelmektedir. Matematiksel modellerdeki kurtların av arama veya buldukları avlara saldırma aşamalarını belirleyen A parametresinin değeri de α 'nın değeri göre olarak iterasyonlar ilerledikçe giderek küçülür. Burada A 'nın değeri $[-2\vec{a}, 2\vec{a}]$ aralıklarında değişmektedir, α ise 2' den 0' a doğru giderek doğrusal olarak küçülür. A 'nın değeri $[-1, 1]$ aralıklarında olması ise arama ajanının arama alanındaki bir sonraki pozisyonunun önceki pozisyonu ile avın pozisyonu arasındaki herhangi bir yerde olacağı anlamına gelmektedir [10]. Şekil 2.3 (a) $|A| < 1$ şartı sağlandığında A parametresinin arama ajanını avına saldırmasına nasıl ittiğini göstermektedir.

Şimdiye kadar bahsedilen tüm işlemler sayesinde Gri Kurt Optimizasyon algoritmasındaki tüm arama ajanları arama alanındaki kendi pozisyonlarını α, β ve δ ajanlarının konumlarına göre güncelleye bilmektedirler. Ama bir optimizasyon algoritmasında sadece bu işlemlerin kullanmak arama ajanlarının lokal sonuçlarda takılıp kalmalarına ve global çözümleri hiçbir zaman bulamamalarına sebep olabilir. Bu sebepten dolayı da Gri Kurt Algoritmasının av arama aşamasını daha iyi yerine getirebilmesi için bu işlemlerin dışında da bazı işlemlere ihtiyacı vardır.



Şekil 2.3. Avını ararken ve avına saldırırken gri kurtlar [4].



Şekil 2.4. 2 ve 3 boyutlu vektörel ve bir sonraki adımda olası tüm pozisyonlar.

2.4 Av Arama

Gri kurt sürüsü çoğunlukla α , β ve δ 'nin konumlarına göre arama yaparlar. Av aramak için birbirlerinden ayrılırlar ve avlarına saldırmak ise için birleşirler. Birbirlerinden ayrılma mekanizmasını matematiksel olarak modellemek için A parametresinden yararlanılır. A parametresinin rastgele olan değeri 1'den büyükse veya -1'den küçükse bu parametre arama ajanını alanda diğer ajanlardan ayırır ve arama yapması için zorlar. Bu işlem keşif yapma anlamına gelir ve Gri Kurt Optimizasyon algoritmasının global olan optimum fonksiyon değerini bulmasını sağlar. Yine Şekil 2.3 (b) $|A| > 1$ şartının arama ajanını nasıl diğer ajanlardan ayırmasını ve daha iyi sonuçlar bulması için keşif yapmasına zorladığını yansıtmaktadır. GWO algoritmasının daha iyi keşif yapmasını sağlayan bir diğer bileşeni de C 'dir. Denklem 2.4'den de kolayca görülebileceği gibi C parametresi $[0, 2]$ aralığında rastgele herhangi bir değeri alabilmektedir. Bunun sayesinde de Gri Kurt Optimizasyon algoritması daha rastgele kararlar alabilmektedir ki bu da onun daha iyi keşif yapması anlamına gelmektedir. C parametresinin sadece iterasyonların başında değil en son iterasyona kadar rastgele oluşturulması sayesinde algoritma son ana kadar lokal sonuçlara takılıp kalmaktan daha başarılı bir şekilde kaçabiliyor.

$\vec{C}$ Vektörü doğada kurtların avlarına yaklaşmaya çalışırken karşılarına çıkabilecek engelleri de temsil etmektedirler. Genel olarak doğada avlanmaya çalışan yırtıcı bir hayvanın av arama veya avına saldırma aşamasında karşılarına çıkabilecek engeller olur ve bu engeller yırtıcıları caydırır. Aslında $\vec{C}$ vektörü yaptığı şey de tam olarak budur. Toparlamak gerekirse Gri Kurt Optimizasyon Algoritması keşif yapmaya uygun değerleri rastgele oluşturulmuş bir kurt sürüsü ile başlar İterasyonlar devam ettikçe ise

alfa, beta ve delta kurtları olası bir avın pozisyonunu tahmin eder. Tüm arama ajanları kendi pozisyonlarını buldukları avların pozisyonlarına uygun bir şekilde günceller. C Parametresinin değeri iterasyonlar devam ettikçe 2'den 0'a doğrusal olarak iner ki bu da kurt sürüsünün yeteri kadar keşif yaptıktan sonra saldırma aşamasına geçtiğinin göstergesidir. En sonda da Gri Kurt Optimizasyon algoritması iterasyonlar bittiği zaman çalışmayı durudur ve elde edilen en iyi uygun değerine sahip olan arama ajanının sonucu algoritmanın çıktısı olarak kabul edilir [5]. Optimizasyon algoritmasının kaba kodu Şekil 2.5'de verilmiştir.

```

Kurt sürüsünün popülasyonu başlatın  $X_i$  ( $i = 1, 2, \dots, n$ )
 $a$ ,  $A$  ve  $C$  parametrelerini belirleyin
Her bir arama ajanının fitness değerini hesaplayın
 $X_\alpha$  = en iyi arama ajanı
 $X_\beta$  = en iyi ikinci arama ajanı
 $X_\delta$  = en iyi üçüncü arama ajanı
while ( $t < \text{en fazla iterasyon sayısı}$ )
  for her bir arama ajanı
    Mevcut arama ajanının konumunu Denklem (2.11)'e göre
    güncelleyin
  end for
  Güncelle  $a$ ,  $A$  ve  $C$ 
  Herhangi bir arama ajanının arama alanının sınırlarını geçip geçmediğini
  kontrol edin
  Her bir arama ajanının uygunluk değerini hesaplayın
   $X_\alpha$ ,  $X_\beta$  ve  $X_\delta$  'yı güncelle
   $t = t + 1$ 
end while
Geri dön  $X_\alpha$ 

```

Şekil 2.5. Optimizasyon algoritmasının kaba kodu

Örnek 2.1:

$-2 \leq x_1, x_2 \leq 2$ aralığında $F(x) = x_1^2 + x_1 \cdot x_2 + x_2^2$ fonksiyonunu minimum yapan x_1, x_2 değerlerini bulmak için optimizasyonu 3 adım ilerletelim.

Çözüm: İlk olarak $[-2, 2]$ aralığında 4 tane rastgele x_1, x_2 değeri alınır. Bu değerler Çizelge 2.1’de verilmiştir:

Çizelge 2.1. Örnek 2.1 için arama uzayındaki rastgele değerler.

	X_1	X_2	$f_{min}(x)$
1. Ajan	-1,5	1	1,75
2. Ajan	-1	1,6	1,96
3. Ajan	0,5	-0,5	1
4. Ajan	1,5	1,5	6,75

Çizelge 2.2’de görüldüğü gibi 4 adaydan verilen fonksiyonu en küçük yapan 3 değer α, β, δ ajanları aşağıda listelendi:

Çizelge 2.2. Örnek 2.1 için başlangıç arama ajanı (α, β, δ) .

	X_1	X_2
x_α	0,5	-0,5
x_β	-1,5	1
x_δ	-1	1,6

Başlangıç arama ajanları belirlendiğine göre iterasyona başlanabilir.

1. Adım: $\vec{a}$, iterasyon boyunca 2’den 0’ a azalmaktadır. Bunun için

$$a = 2 \left(1 - \frac{\text{iterasyon}}{\max \text{iterasyon}} \right)$$

formülü kullanılır [11]. Burada maksimum iterasyon sayısı 3 olduğuna göre $a = 2 - 2 \cdot \frac{1}{3} \cong 1,33$ elde edilir. 1. Ajanı α kurduna göre konumunu güncelleyelim: (2.3)-(2.4) denklemlerinden $r_1 = 0,3$ ve $r_2 = 0,41$ olmak üzere

$$A_1 = 2 \cdot a \cdot r_1 - a = 2 \cdot 1,33 \cdot 0,30 - 1,33 = -0,532$$

$$C_1 = 2r_2 = 2 \cdot 0,41 = 0,82$$

elde edilir. (2.5) eşitliğine göre

$$\vec{D}_\alpha = |C_1 \vec{X}_\alpha - \vec{X}| = |0.82 \cdot (0.5, -0.5) - (-1.5, 1)| = (1.91, 1.41)$$

‘dir. 1. Ajanın α kurduna göre yeni konumu (2.8)’den

$$\vec{X}_1 = \vec{X}_\alpha - A_1 \cdot \vec{D}_\alpha = (0.5, -0.5) - (-0.532) \cdot (1.91, 1.41) = (1.51612, 0.25012)$$

Benzer şekilde 1. Ajanın sırasıyla β ve δ kurtlarına göre konumlarını bulacağız.

$r_1 = 0,20$ ve $r_2 = 0,70$ için $A_2 = 2.1,33.0,20 - 1,33 = 0,789$, $C_2 = 2.0,70 = 1,4$ olduğundan

$$\vec{D}_\beta = |1.4(-1.5, 1) - (-1.5, 1)| = (0.6, 0.4)$$

$$\vec{X}_2 = (-1.5, 1) - (-0.798) \cdot (0.6, 0.4) = (1.0212, 1.3192)$$

Bulunur. $r_1 = 0.35$ ve $r_2 = 0.6$ için $A_3 = 2.1,33.0,35 - 1,33 = -0,399$, $C_3 = 2.0,60 = 1,20$ olduğundan

$$\vec{D}_\delta = |1.20(-1, 1.6) - (-1.5, 1)| = (0.3, 0.92)$$

$$\vec{X}_3 = (-1, 1.6) - (-0.399) \cdot (0.3, 0.92) = (-0.8803, 1.96708)$$

elde edilir. Son olarak 1. Ajanın yeni konumu (2.11)’den

$$\begin{aligned} \vec{X}_1 &= \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \\ &= \frac{(1.51612, 0.25012) + (1.0212, 1.3192) + (-0.8803, 1.96708)}{3} \\ &= (-0.12846, 1.1788) \end{aligned}$$

Bulunur.

2. Ajanı α kurduna göre konumunu güncelleyelim: (2.3)-(2.4) denklemlerinden $r_1 = 0,2$ ve $r_2 = 0,34$ olmak üzere

$$A_1 = 2. a. r_1 - a = 2.1,33.0,20 - 1,33 = 0,798$$

$$\vec{C}_1 = 2\vec{r}_2 = 2.0,34 = 0,68$$

elde edilir. (2.5) eşitliğine göre

$$\vec{D}_\alpha = |C_1 \vec{X}_\alpha - \vec{X}| = |0.68(0.5, -0.5) - (-1, 1.61)| = (1.34, 1.94)$$

‘dür. 2. Ajanın α kurduna göre yeni konumu (2.8)’den

$$\vec{X}_1 = (0.5, -0.5) - (-0.798) \cdot (1.34, 1.94) = (1.56932, 1.048)$$

Benzer şekilde 2. Ajanın sırasıyla β ve δ kurtlarına göre konumlarını bulunacaktır.

$$r_1 = 0,70 \text{ ve } r_2 = 0,80 \text{ için } A_2 = 2.1,33.0,70 - 1,33 = 0,53212, C_2 = 2.0,80 = 1,6$$

olduğundan

$$\vec{D}_\beta = |1.6(-1.5, 1) - (-1, 1.6)| = (1.4, 0)$$

$$\vec{X}_2 = (-1.5, 1) - 0.532(1.4, 0) = (-2.2448, 1)$$

bulunur. $r_1 = 0,65$ ve $r_2 = 0,47$ için

$$A_3 = 2.1,33.0,65 - 1,33 = 0,399, C_3 = 2.0,47 = 0,94$$

olduğundan

$$\vec{D}_\delta = |0.94 \cdot (-1, 1.6) - (-1, 1.6)| = (0.06, 0.096)$$

$$\vec{X}_3 = (-1, 1.6) - 0.399 \cdot (0.06, 0.096) = (-1.02394, 1.561696)$$

bulunur. Son olarak 2. Ajanın yeni konumu (2.11)’den

$$\begin{aligned} \vec{X}_1 &= \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} = \frac{(1.56932, 1.048) + (-2.2448, 1) + (-1.02394, 1.561696)}{3} \\ &= (-0.5664, 1.2032) \end{aligned}$$

olarak elde edilir.

3. Ajanı α kurduna göre konumunu güncelleyelim: (2.3)-(2.4) denklemlerinden $r_1 = 0,40$ ve $r_2 = 0,62$ olmak üzere $A_1 = 2. a. r_1 - a = 2.1,33.0,40 - 1,33 = -0,266,$

$C_1 = 2. r_2 = 2. 0,62 = 1,24$ elde edilir. (2.5) eşitliğine göre

$$\vec{D}_\alpha = |1,24. (0,5, -0,5) - (0,5, -0,5)| = (0,12,0,12)$$

dir. 3. Ajanın α kurduna göre yeni konumu (2.8)'den

$$\vec{X}_1 = (0,5, -0,5) - (-0,266). (0,12,0,12) = (0,53192, -0,46808)$$

olur. Benzer şekilde 3. Ajanın sırasıyla β ve δ kurtlarına göre konumlarını bulacağız.

$r_1 = 0,45$ ve $r_2 = 0,20$ için $A_2 = 2. 1,33. 0,45 - 1,33 = -0,133$ $C_2 = 2. 0,20 = 0,40$

olduğundan

$$\vec{D}_\beta = |0,40 (-1,5,1) - (0,5, -0,5)| = (1,1,0,9)$$

$$\vec{X}_2 = (-1,5,1) - (-0,133). (1,1,0,9) = (-1,3537,1,1197)$$

$r_1 = 0,50$ ve $r_2 = 0,70$ için $A_3 = 2. 1,33. 0,50 - 1,33 = 0$ $C_3 = 2. 0,70 = 1,40$

olduğundan

$$\vec{D}_\delta = |1,40. (-1,1,6) - (0,5, -0,5)| = (1,9,2,74)$$

$\vec{X}_3 = (-1,1,6) - 0(0,19,0,74) = (-1,1,6)$ bulunur. Son olarak 3. Ajanın yeni konumu (2.11)'den

$$\begin{aligned} \vec{X}_1 &= \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} = \frac{(0,53192, -0,46808) + (-1,3537,1,1197) + (-1,1,6)}{3} \\ &= (-0,60726,0,75054) \end{aligned}$$

olarak hesaplanır.

4. Ajanı α kurduna göre konumunu güncelleyelim: (2.3)-(2.4) denklemlerinden $r_1 = 0,10$ ve $r_2 = 0,50$ olmak üzere

$A_1 = 2 \cdot a \cdot r_1 - a = 2 \cdot 1,33 \cdot 0,10 - 1,33 = 1$, $C_1 = 2r_2 = 2 \cdot 0,50 = 1$ elde edilir.
(2.5) eşitliğine göre

$$\vec{D}_\alpha = |1(0,5, -0,5) - (1,5,1,5)| = (1,2)$$

4. Ajanın α kurduna göre yeni konumu (2.8)'den

$$\vec{X}_1 = (0,5, -0,5) - (-1,064) (1,2) = (1,564,1,628)$$

bulunur.

Benzer şekilde 4. Ajanın sırasıyla β ve δ kurtlarına göre konumlarını bulacağız.

$r_1 = 0,25$ ve $r_2 = 0,55$ için $A_2 = 2 \cdot 1,33 \cdot 0,25 - 1,33 = -0,665$, $C_2 = 2 \cdot 0,55 = 1,1$ olduğundan

$$\vec{D}_\beta = |1,1 (-1,5,1) - (1,5,1,5)| = (3,15,0,4)$$

$$\vec{X}_2 = (-1,5,1) - (-0,665) \cdot (3,15,0,4) = (0,59475,1,266)$$

bulunur.

$r_1 = 0,70$ ve $r_2 = 0,80$ için $A_3 = 2 \cdot 1,33 \cdot 0,70 - 1,33 = 0,532$, $C_3 = 2 \cdot 0,80 = 1,6$ olduğundan

$$\vec{D}_\delta = |1,6 (-1,1,6)) - (1,5,1,5)| = (3,1,1,06)$$

$$\vec{X}_3 = (-1,1,6) - 0,532 (0,31,1,06) = (-2,6492,1,03608)$$

bulunur. Son olarak 4. Ajanın yeni konumu (2.11)'den

$$\begin{aligned} \vec{X}_1 &= \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} = \frac{(1,564,1,628) + (0,59475,1,266) + (-2,6492,1,03608)}{3} = \\ &= (-0,1634,1,3100) \end{aligned}$$

bulunur. Çizelge 2.3'te ajanlarımızın yeni konumu verilmiştir.

Çizelge 2.3. Örnek 2.1 için arama uzayındaki ajanların yeni konumları.

	X_1	X_2	$f_{min}(x)$
1. Ajan	-0,1284	1,1788	1,2546
2. Ajan	-0,5664	1,2032	1,0870
3. Ajan	-0,6072	0,7505	0,4763
4. Ajan	-0,1634	1,3100	1,5287

Benzer şekilde ikinci iterasyon yapılırsa ajanların yeni konumları Çizelge 2.4’ de verilmiştir.

Çizelge 2.4. Örnek 2.1 için arama uzayındaki ajanların başlangıç, birinci ve ikinci adımdaki konumları.

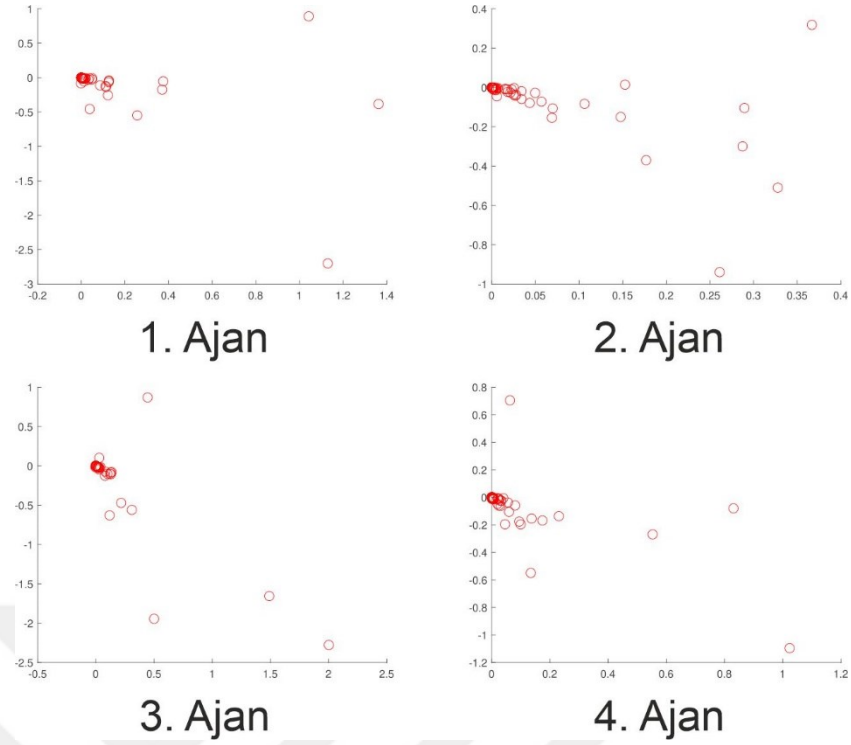
	X_1	X_2	$f_{min}(x)$	X_1	X_2	$f_{min}(x)$	X_1	X_2	$f_{min}(x)$
1	-1,5	1	1,75	-0,1284	1,1788	1,2546	-0,3997	1,0992	0,9286
2	-1	1,6	1,96	-0,5664	1,2032	1,0870	-0,3833	0,8440	0,5357
3	0,5	-0,5	0,25	-0,6072	0,7505	0,4763	-0,2424	1,2354	1,2855
4	1,5	1,5	6,75	-0,1634	1,3100	1,5287	-0,4519	0,7013	0,3791

Bu problem bilgisayar ortamında kodlanarak 100 adımda çözdürülmüş olup 4 ajanın yeni konumları aşağıdaki Çizelge 2.5’te verilmiştir.

Çizelge 2.5. Örnek 2.1 için arama uzayındaki ajanların başlangıçtan otuzuncu adıma kadar bazı konumları.

	1. Ajan		2. Ajan		3. Ajan		4. Ajan	
0	1.2589	1.6232	-1.4921	1.6535	0.5294	-1.6098	-0.8860	0.1875
1	1.4505	-0,1527	0.3273	-1.6191	3.5676	-1.2427	1.4845	-0.2854
5	0.4913	-0.4563	0.5303	0.0682	0.7304	-0.1604	0.8250	-0.7604
10	0.1818	-0.1581	0.1898	-0.0639	0.2574	-0.0676	0.1096	-0.0551
15	0.0117	-0.0489	0.0519	-0.0150	0.0725	-0.0715	0.0627	-0.0121
20	0.0171	-0.0103	0.0175	-0.0198	0.0176	-0.0181	0.0002	-0.0087
25	-0.0008	-0.0053	-0.0021	-0.0049	-0.0003	-0.0035	0.0004	-0.0035
30	0.0005	-0.0015	0.0004	-0.0017	0.0004	-0.0017	0.0003	-0.0011

Kurtların hareketlerini ve sıfır noktası etrafında toplanmasını Şekil 2.6’te gösterildi.



Şekil 2.6. Kurtların(ajanların) pozisyonları ve konumları.

Örnek 2.2:

$-2 \leq X_1, X_2 \leq 2$ aralığında $f(x) = X^2$ fonksiyonunu minimum yapan X değerlerini araştıralım.

Çözüm: İlk olarak $[-2, 2]$ aralığında 4 tane rastgele değerler alalım. Bu değerler Çizelge 2.6’da verilmiştir. İterasyon 0 için yeni konumlarını bulalım.

Çizelge 2.6. Örnek 2.2 için arama uzayındaki rastgele değerler.

	X	$f_{min}(x) = X^2$
1. ajan	-1,2	1,44
2. ajan	-0,4	0,16
3. ajan	1,5	2,25
4. ajan	0,8	0,64

Çizelge 2.7’de görüldüğü gibi 4 adaydan verilen fonksiyonu en küçük yapan 3 değer α, β, δ ajanları aşağıda listelenmiştir:

Çizelge 2.7. Örnek 2.2 için başlangıç arama ajanı (α, β, δ) .

	X
X_α	$-0,4$
X_β	$0,8$
X_δ	$-1,2$

Başlangıç arama ajanları belirlendiğine göre iterasyona başlayabiliriz:

1. Adım: $\vec{a}$, iterasyon boyunca 2’den 0’ a azalmakta. Bunun için

$a = 2(1 - \frac{\text{iterasyon}}{\text{maxiterasyon}})$ Formülünü kullanacağız [11]. Burada maksimum iterasyon sayısı 3 ve İterasyon 0 için yeni konumlarını bulalım. olduğuna göre $\alpha = 2 - 2 \cdot \frac{0}{3} \cong 2$ elde edilir. 1. Ajanı α kurduna göre konumunu güncelleyelim: (2.3)-(2.4) denklemlerinden $r_1 = 0,2$ ve $r_2 = 0,4$ olmak üzere

$$A = 2ar_1 - a = A_1 = 2 \cdot 2 \cdot 0,2, A_1 = -1,2$$

$$C = 2r_2 = C_1 = 2 \cdot 0,4, C_1 = 0,8$$

elde edilir. (2.5) eşitliğine göre

$$\vec{D}_\alpha = |\vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}| = |0,8 \cdot (-0,4) - (-1,2)| = 0,88$$

dir. 1. Ajanın α kurduna göre yeni konumu (2.8)’den

$$\vec{X}_1 = \vec{X}_\alpha - A_1 \cdot \vec{D}_\alpha = X_1 = -0,4 - (-1,2) \cdot (0,88) = 0,656$$

Benzer şekilde 1. Ajanın sırasıyla β ve δ kurtlarına göre konumları hesaplanacaktır.

$$D_\beta = |0,8 \cdot 0,8 - (-1,2)| = 1,84, X_2 = 0,8 - (-1,2) \cdot (1,84) = 3,008$$

$$D_\delta = |0,8 \cdot (-1,2) - (-1,2)| = 0,24, X_3 = -1,2 - (-1,2) \cdot (0,24) = -0,912$$

Elde edilir. Son olarak 1. Ajanın yeni konumu (2.11)’den

$$x_{(t+1)} = \frac{X_1 + X_2 + X_3}{3} = \frac{0,656 + 3,008 - 0,912}{3} = 0,9173$$

bulunur.

2. Ajanı α kurduna göre konumunu güncelleyelim: (2.3)-(2.4) denklemlerinden $r_1 = 0,3$ ve $r_2 = 0,3$ olmak üzere

$A_1 = 2 \cdot 2 \cdot 0,3 - 2 = -0,8$, $C_1 = 2 \cdot 0,3 = 0,6$ elde edilir. (2.5) eşitliğine göre

$$D_\alpha = |0,6 \cdot (-0,4) - (-0,4)| = 0,16$$

dır. 2. Ajanın α kurduna göre yeni konumu (2.8)'den

$$X_1 = -0,4 - (-0,8) \cdot (0,16) = -0,272$$

olur. Benzer şekilde 2. Ajanın sırasıyla β ve δ kurtlarına göre konumları bulunacaktır.

$$D_\beta = |(0,6) \cdot (0,8) - (-0,4)| = 0,88$$

$$X_2 = 0,8 - (-0,8) \cdot (0,88) = 1,504$$

$$D_\delta = |(0,6) \cdot (-1,2) - (-0,4)| = 0,32$$

$$X_3 = (-1,2) - (-0,8) \cdot (0,32) = -0,944$$

bulunur. Son olarak 2. Ajanın yeni konumu (2.11)'den

$$X_{(t+1)} = \frac{X_1 + X_2 + X_3}{3} = \frac{0,272 + 1,504 - 0,944}{3} = 0,096$$

Bulunur.

3. Ajanı α kurduna göre konumu güncellernirse: (2.3)-(2.4) denklemlerinden $r_1 = 0,4$ ve $r_2 = 0,2$ olmak üzere

$$A_1 = 2 \cdot 2 \cdot 0,4 - 2 = -0,4, C_1 = 2 \cdot 0,2 = 0,4$$

elde edilir. (2.5) eşitliğine göre

$$D_\alpha = |(0,4) \cdot (0,4) - 1,5| = 1,66$$

3. Ajanın α kurduna göre yeni konumu (2.8)'den

$$X_1 = -0,4 - (-0,4) \cdot (1,66) = 0,264$$

olur.

Benzer şekilde 3. Ajanın sırasıyla β ve δ kurtlarına göre konumlarını bulacağız.

$$D_\beta = |(0,4) \cdot (0,8) - 1,5| = 1,18$$

$$X_2 = 0,8 - (-0,4) \cdot (1,18) = 1,272$$

$$D_\delta = |(0,4) \cdot (-1,2) - 1,5| = 1,98$$

$$X_3 = -1,2 - (-0,4) \cdot (1,98) = -0,408$$

bulunur.

Son olarak 3. Ajanın yeni konumu (2.11)'den

$$X_{(t+1)} = \frac{X_1 + X_2 + X_3}{3} = \frac{0,264 + 1,272 - 0,408}{3} = 0,376$$

olarak bulunur.

4. Ajanı α kurduna göre konumunu güncellersek; (2.3)-(2.4) denklemlerinden $r_1 = 0,1$ ve $r_2 = 0,7$ olmak üzere

$$A_1 = 2 \cdot 2 \cdot 0,1 - 2 = -1,6, C_1 = 2 \cdot 0,7 = 1,4$$

elde edilir. (2.5) eşitliğine göre

$$D_\alpha = |(1,4) \cdot (-0,4) - 0,8| = 1,36$$

bulunur. 4. Ajanın α kurduna göre yeni konumu (2.8)'den

$$X_1 = -0,4 - (-1,6) \cdot (1,36) = 1,776$$

bulunur. Benzer şekilde 4. Ajanın sırasıyla β ve δ kurtlarına göre konumları aşağıdaki gibi hesaplanır.

$$D_{\beta} = |(1,4) \cdot (0,8) - 0,8| = 0,32$$

$$X_2 = 0,8 - (-1,6) \cdot (0,32) = 1,312$$

$$D_{\delta} = |(1,4) \cdot (-1,2) - 0,8| = 2,48$$

$$X_3 = -1,2 - (-1,6) \cdot (2,48) = 2,768$$

bulunur. Son olarak 4. Ajanın yeni konumu (2.11)'den

$$X_{(t+1)} = \frac{X_1 + X_2 + X_3}{3} = \frac{1,776 + 1,312 + 2,768}{3} = 1,952$$

Ajanlarımızın yeni konumları Çizelge 2.8'de gösterilmiştir.

Çizelge 2.8. Örnek 2.2 için arama uzayındaki ajanların yeni konumları.

X	$f_{min}(x) = X^2$	X	$f_{min}(x) = X^2$
-1,2	1,44	0,9173	0,8414
-0,4	0,16	0,096	0,0092
1,5	2,25	0,376	0,1413
0,8	0,64	1,952	3,8103

Bu problem bilgisayar ortamında kodlanarak 100 adımda çözdürülmüş olup 4 ajanın yeni konumları Çizelge 2.9'da elde edilmiştir.

Çizelge 2.9. Örnek 2.2 için ajanların daha büyük adımdaki konumları.

	1. Ajan	2. Ajan	3. Ajan	4. Ajan
0	1.2589	1.6232	-1.4921	1.6535
1	-0.0301	-1.4863	0.0010	-0.9000
5	0.1542	-0.0119	0.0110	0.1105
10	-0.0025	0.0020	0.0003	0.0009
15	$-0.4642 \cdot 10^{-4}$	$0.6629 \cdot 10^{-4}$	$0.1117 \cdot 10^{-4}$	$-0.7833 \cdot 10^{-4}$
20	$0.0768 \cdot 10^{-4}$	$0.1032 \cdot 10^{-4}$	$0.1031 \cdot 10^{-4}$	$0.0148 \cdot 10^{-4}$
100	$-0.1236 \cdot 10^{-14}$	$-0.1234 \cdot 10^{-14}$	$-0.1237 \cdot 10^{-14}$	$-0.1236 \cdot 10^{-14}$

Örnek 2.3:

Ava en yakın konumlardan bir tanesi $X = 1$ dir. bu konuma en yakın üç kurt α, β, δ olsun. Konumları ise $X_\alpha, X_\beta, X_\delta$ şeklinde gösterilmiş olup Çizelge 2.10'da verilmiştir. α, β, δ kurtlarının konumları en yakında kurda göre güncellenecektir.

Çizelge 2.10. Örnek 2.3 için başlangıç arama ajanı (α, β, δ) .

	X
X_α	5
X_β	10
X_δ	14

Başlangıç arama ajanları belirlendiğine göre iterasyona başlanabilir. a , iterasyon boyunca 2'den 0' a azalmakta ve $a = 2(1 - \frac{iterasyon}{maxiterasyon})$ formülünden $t = 0$ alınırsa $a = 2$ bulunur. Ajanlarımızın konumlarını güncellemek için (2.3)-(2.4) denklemlerinden

A_1, C_1 için

$$r_1 = 0,75, r_2 = 1,$$

A_2, C_2 için

$$r_1 = 1, r_2 = 0,2,$$

A_3, C_3 için

$$r_1 = 0,2 \quad r_2 = 0,75$$

olmak üzere $A_1 = 1, A_2 = 2, A_3 = -1,2$ $C_1 = 2, C_2 = 0,4, C_3 = 1,5$ bulunur.

Ava en yakın noktalardan biri olan $X = 1$ göre güncелейelim. Denklemler (2.5)-(2.8)

$$D_\alpha = |C_1 \cdot X_\alpha - X_t|$$

$$D_\alpha = |2,5 - 1| = 1,5$$

$$X_1 = 5 - 1.9 = -4$$

1. ajanımızın yeni konumu bu şekilde bulunur. Denklemler (2.6)-(2.9)

$$D_\beta = |C_1 \cdot X_\beta - X_t|$$

$$D_\beta = |2.10 - 1| = 19$$

$$X_2 = 10 - 2.19 = -28$$

2. ajanımızın yeni konumu bu şekilde bulunur. Denklemler (2.7)-(2.10)

$$D_\delta = |C_1 \cdot X_\delta - X_t|$$

$$D_\delta = |1.14 - 1| = 13$$

$$X_3 = 14 - (-1,2) \cdot 13 = 29,6$$

3. ajanımızın yeni konumu bu şekilde bulunur. Denklem

$$X_{(t+1)} = \frac{(x_1 + x_2 + x_3)}{3}$$

$$X_{(t+1)} = \frac{(-4 - 28 + 29,6)}{3} = -0,8$$

-0,8 olarak ava dahada yaklaşmış olundu burada eğer bu konumda bir kurt olsaydı bu kurt ava dahada yaklaşmış olurdu.

3. BALİNA OPTİMİZASYON ALGORİTMASI (BOA)

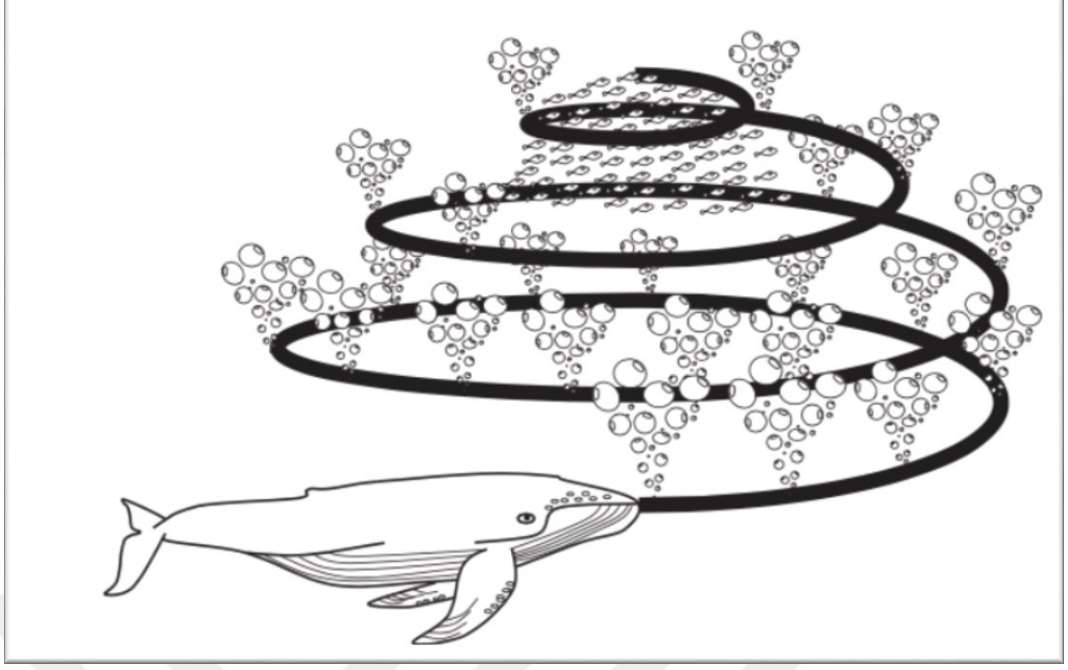
Bu kısımda Balina Optimizasyon algoritmasının detayları verilerek, örnekler üzerinde detaylı uygulaması gösterildi.

3.1 Algoritmanın İlham Kaynağı

Balinalar, dünya üzerinde literatürde geçen en büyük memeli canlılar olarak bilinmektedir. Yetişkin bir balinanın boyu 30 m uzunluğa ve ağırlığı ise 180 tona varabilir. Bu dev memelinin Katil, Minke, Sei, Kambur, Sağ, Finback ve Mavi gibi 7 farklı türü vardır. Balinalar genellikle yırtıcı hayvan olarak kabul edilir. Okyanusların yüzeyinden nefes almak zorunda oldukları için asla tam olarak uyuyamazlar. Aslında beynin sadece yarısı uyur. Balinalar hakkında ilginç olan başka bir şey ise çok zeki hayvanlar olarak bilinmeleridir.

Hof ve Van Der Gucht'e [12] göre, balinaların beyinlerinin belirli bölgelerinde iğsi hücreler olarak adlandırılan aslında ise insanların hücrelerine benzer hücreler vardır. Bu hücreler insanlarda duygulardan ve sosyal davranışlardan sorumludur. Başka bir deyişle, iğsi hücreleri bizi diğer tüm canlılardan ayırır. Balinalar akıllı olmalarını borçlu oldukları bu hücrelerden yetişkin bir insana kıyasla 2 kat daha fazlasına sahiptirler. Balinanın bir insan gibi düşünebildiği, öğrenebildiği, yargılayabildiği, iletişim kurabildiği ve hatta duygusal olabileceği kanıtlanmıştır, ancak açıkçası çok daha düşük bir potansiyelle tüm bunları yapabilirler.

Bir başka ilginç nokta da balinaların sosyal davranışlarıdır. Yalnız ya da gruplar halinde yaşarlar. Ancak daha çok gruplar halinde görülürler. Türlerinden bazıları (örneğin katil balinalar) tüm hayatları boyunca bir aile içinde yaşayabilir. En büyük balinalardan biri kambur balinalardır. Yetişkin bir kambur balina, neredeyse bir kamyon kadardır. En sevdikleri avlar ise kril ve küçük balık sürüleridir. Şekil 3.1 bu memeliyi göstermektedir.



Şekil 3.1. Kambur balinaların kabarcık ağı şeklindeki avlanma taktiği [13].

Kambur balinalarla ilgili ilginç olan bir şey ise özel avlanma taktikleridir. Bu av arama ve buldukları avlara saldırma şekline kabarcık ağ besleme yöntemi adı verilmiştir [14]. Kambur balinalar, yüzeye yakın kril sürüsü veya küçük balıkları avlamayı tercih ederler. Kambur balinaların bu avlanma taktiğini Şekil 3.1'de gösterildiği gibi bir daire veya "9" şeklindeki bir yörünge ile baloncuklar oluşturarak hayata geçirdikleri kayıtlara alınmıştır. 2011'den önce, bu davranış yalnızca yüzeyden yapılan gözlemlere dayanılarak araştırılıyordu. Ancak Goldbogen ve ark. etiket sensörlerini kullanarak bu davranışı araştırdı. 9 ayrı kambur balinanın 300 etiketten türetilmiş kabarcık-ağ besleme olayını yakaladılar. Kabarcıkla bağlantılı iki manevra buldular ve bunlara "yukarı doğru spiraller" ve "çift döngüler" adını verdiler. Önceki manevrada, kambur balinalar yaklaşık 12 m aşağıya dalar ve ardından avın etrafında spiral şeklinde kabarcık oluşturmaya başlar ve yüzeye doğru yüzer. Sonraki manevra üç farklı aşamadan oluşmaktadır: mercan döngüsü, lob kuyruğu ve yakalama döngüsü. Bu davranışlar hakkında ayrıntılı bilgi [15]'de bulunmaktadır.

Kabarcık ağıyla beslenmenin yalnızca kambur balinalarda gözlemlenebilen benzersiz bir davranış olduğunu burada belirtmekte fayda var. Bu çalışmada spiral kabarcık-ağ besleme manevrası, optimizasyonu gerçekleştirmek için matematiksel olarak modellenmiştir [5].

3.2 Matematiksel Model

Bu bölümde avı çevreleme, spiral şeklindeki kabarcık ağı manevrası ve av aramanın matematiksel modelleri detaylı bir şekilde incelenmiştir. Daha sonra ise BOA algoritması önerilmiştir.

3.2.1 Avı çevreleme

Kambur balinalar yetenekleri sayesinde avlarının konumunu belirleyebilir ve bu sayede avlarının etrafında kabarcıklar çıkararak onları kuşatırlar. BOA'da çözülmek istenen problemin optimum değeri av olarak nitelenir. Optimizasyon problemlerinin optimum değeri genel olarak bilinmediğinden, bu algoritmalar optimum değeri bulabildikleri en iyi değer veya ona yakın bir değer olarak kabul edebilirler. Bu en iyi değere sahip arama ajanı bulunur ve her bir adımda en iyi değere sahip arama ajanı güncellenir. Avın etrafını sarmanın matematiksel modeli Denklem (3.1) ve Denklem (3.2)'de gösterilmiştir [3]

$$\vec{D} = | \vec{C} \cdot \vec{X}^*(t) - \vec{X}(t) | \quad (3.1)$$

$$\vec{X}(t+1) = | \vec{X}^*(t) - \vec{A} \cdot \vec{D} | \quad (3.2)$$

Burada t , mevcut iterasyonu gösterir, A ve C katsayı vektörleridir, X^* ise şimdiye kadar elde edilen en iyi çözümün konum vektörüdür, X konum vektörüdür, $| \cdot |$ mutlak değerdir ve $\cdot$ elemanlar arası çarpımı gösterir. Burada, daha iyi bir çözüm varsa, X^* değeri güncellenir. Buradaki katsayı vektörleri, A ve C şu şekilde hesaplanır:

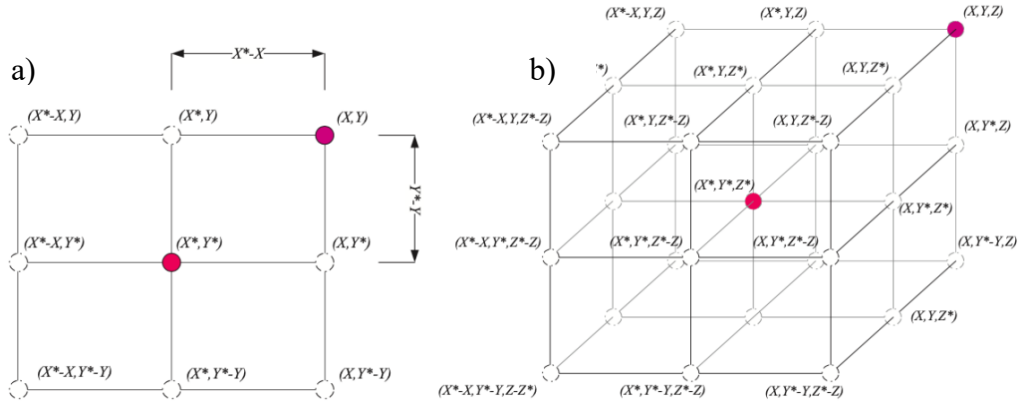
$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a} \quad (3.3)$$

$$\vec{C} = 2 \cdot \vec{r}_2 \quad (3.4)$$

Denklem (3,3)' de ve denklem (3,4)' de bulunan $\vec{r}_1$ ve $\vec{r}_2$ $[0,1]$ aralığında rastgele değer alır. Denklem (3,3)' de yer alan a ise iterasyon boyunca 2' den 0'a kadar lineer olarak azalır:

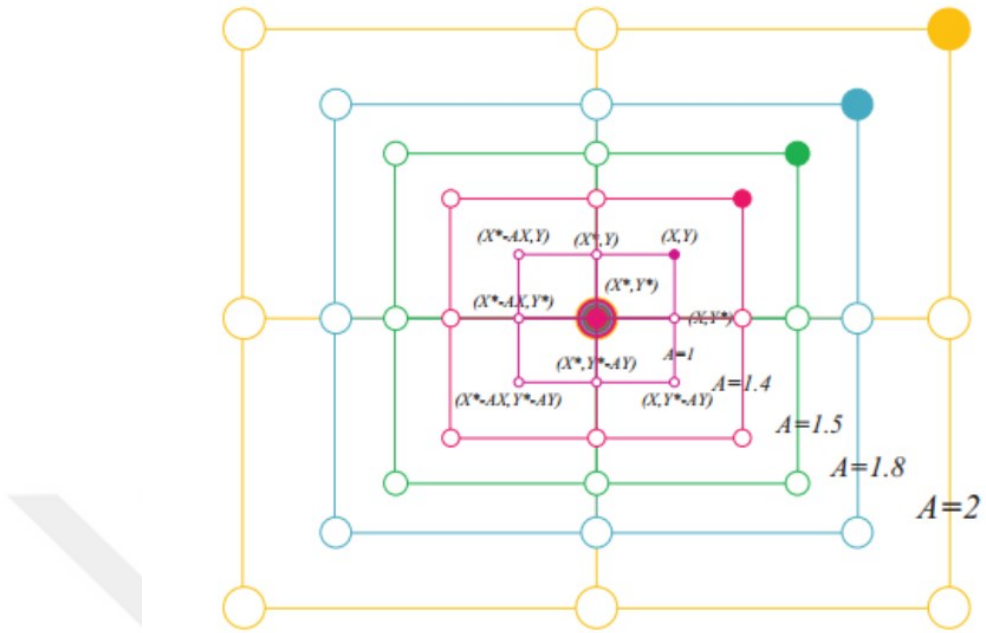
$$a_1 = 2 - t \cdot \frac{2}{\text{maksimum iterasyon sayısı}} \quad (3,5)$$

Şekil 3.2 (a) (3,2)'deki denklemde ve 2 boyutlu problem arasındaki mantığı göstermektedir. Bir arama ajanının konumu (X, Y) mevcut en iyi ajana göre (X^*, Y^*) güncellenir. Bilinen en iyi ajanın etrafındaki noktalara ise $\vec{A}$ ve $\vec{C}$ sabitleri değiştirilerek erişilebilir. Bir arama ajanının 3 boyutlu alanda pozisyonunu güncellemesi ise Şekil 3.2(b)' de gösterilmiştir. $\vec{r}_1$ ve $\vec{r}_2$ rastgele vektörlerini belirleyerek arama alanında Şekil 3.2 (a) ve (b)'de gösterilen anahtar noktalar arasında yerleşen herhangi bir pozisyona ulaşmak mümkündür ve denklem (3,2)'den de açıkça görüldüğü gibi arama ajanları kendi pozisyonlarını komşu en iyi ajanın pozisyonuna göre günceller. Bu metot avı çevreleme aşamasını matematiksel olarak modellenmesidir. Aynı şekilde n boyutlu bir arama alanına genişletilebilir. Arama ajanları şimdiye kadar elde edilen en iyi çözümün etrafında hiper küpler içinde hareket eder. Daha önce de bahsedildiği gibi, kambur balinalar da kabacık ağı stratejisiyle avına saldırır. Bu yöntem matematiksel olarak bir sonraki bölümde detaylı olarak anlatılmıştır.

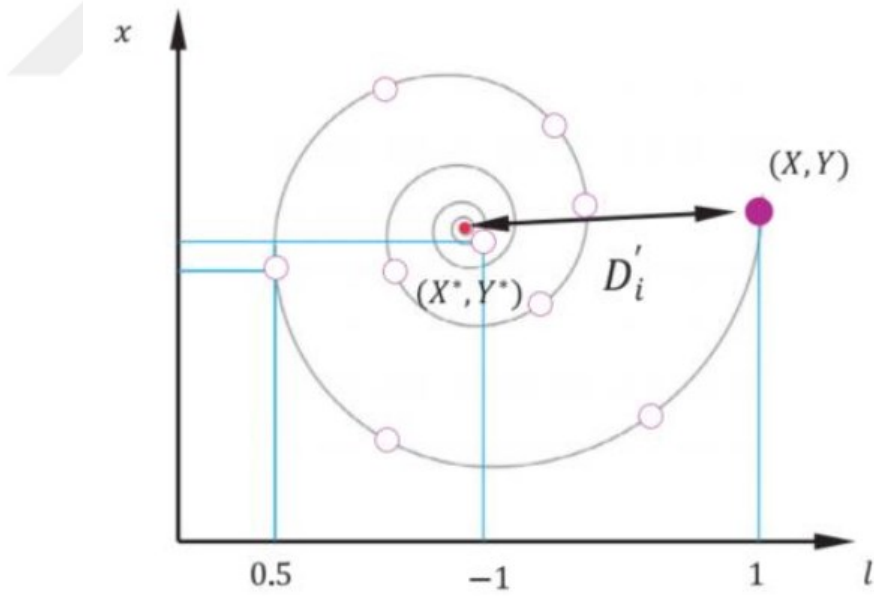


Şekil 3.2. Arama ajanlarının a) İki boyutlu alanda b) Üç boyutlu alanda konum vektörleri ve olası konumları.

3.2.2 Kabarcık ağı yöntemi (Saldırı aşaması)



Şekil 3.3. BOA' da uygulanan keşif uygulaması (X^* rastgele seçilen bir arama ajanı) [3].



Şekil 3.4. Spiral şeklindeki pozisyon güncelleme aşaması.

Kambur balinaların kabarcık-ağ davranışı aşağıda yer alan iki yaklaşımla matematiksel olarak modellenmiştir.

3.2.2.1 Daralan kuşatma mekanizması

Gittikçe Küçülen Avın Etrafını Sarma Mekanizma davranışı (3,3) denklemindeki a değerinin giderek doğrusal olarak azaltılmasıyla elde edilir. A Değerinin de a 'nın azalmasıyla beraber azalmasını bilmekte fayda vardır. Başka bir ifadeyle $\vec{A} [-a, a]$ aralığında rastgele bir değerdir ve burada a daha önce de belirtildiği gibi iterasyonlar boyunca 2'den 0'a giderek doğrusal olarak azaltılır. $\vec{A}$ Parametresine $[-1, 1]$ aralığından rastgele değerler atayarak ajanın kendi konumu ile o anki en iyi ajanın konumu arasında herhangi bir yerde pozisyon ajana atanır. Şekil (3.3)'deki iki boyutlu bir alanda (X, Y) ile (X^*, Y^*) aralığında ajanın konumlanabileceği değerlerindeki tüm pozisyonlar gösterilmektedir. Burada A parametresinin $0 \leq A \leq 1$ aralığında değerleri dikkate alınmıştır.

3.2.2.2 Spiral hareket

Ajanların Pozisyonlarını Spiral Şeklinde Güncellemek; Şekil 3.4'den de görülebileceği gibi bu yöntemde ilk önce (X, Y) konumunda bulunan balina ile (X^*, Y^*) konumunda bulunan av arasındaki mesafe hesaplanır. Sonrasında kumbur balinaların sarmal şeklindeki hareketini modelleyebilmek için Denklem (3.5) kullanılır.

$$\vec{X}(t+1) = D' \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}^*(t) \quad (3.6)$$

$$D' = |X^*(t) - X(t)| \quad (3.7)$$

D' üzerinde hesaplamalar yapılan balina ile av arasındaki mesafeni hesaplamak için kullanılır. b Logaritmik spirali şeklini tanımlamak için gereken bir sabittir.

$$a_2 = -1 + t \cdot \frac{-1}{\text{maksimum iterasyon}} = -1 \quad (3.8)$$

l , ise $[-1, 1]$ aralığında bulunan herhangi bir rastgele bir değerdir ve $.$ elemanlar arası çarpımı göstermektedir ve [11]'den

$$l = (a_2 - 1) \cdot r_{rand} + 1 \quad (3.9)$$

dir. Kumbur balinalar avının etrafında eş zamanlı olarak hem daralan kuşatma mekanizmasına göre hem de spiral harekete göre yüzerler. Bu kompleks davranışı modellemek için optimizasyon sırasında balinaların konumunu güncellemek için

gitgide küçülen avın etrafını saran mekanizma veya spiral model arasında seçim yapma olasılığının %50 olduğunu varsayıyoruz. Sonuç olarak (3.6)'daki matematiksel bir model ortaya çıkıyor.

$$\vec{X}(t+1) = \begin{cases} \vec{X}^*(t) - \vec{A} \cdot \vec{D} & p < 0.5 \\ De^{bl} \cdot \cos(2\pi l) + \vec{X}^*(t) & p \geq 0.5 \end{cases} \quad (3.10)$$

Burada p , $[0, 1]$ aralığında olan rastgele bir sayıdır. Kabarcık ağı yöntemine ek olarak kambur balinalar rastgele olarak av arama metotları da vardır.

3.2.3 Av Arama

$\vec{A}$ vektörünün varyasyonuna dayalı olan aynı yaklaşım av arama(keşif) için de kullanılmaktadır. Kambur balinalar birbirlerinin konumuna göre rastgele arama yaparlar. Bu nedenle de -1 ve 1 aralıklarındaki rastgele $\vec{A}$ parametresini kullanırız ki arama ajanı referans bir balinadan uzaklaşmak zorunda kalsın. Ava saldırma aşamasındaki mekanizmanın tam tersine keşif aşamasında arama ajanının bir sonraki pozisyonu belirlenirken sürüdeki o andaki en iyi ajanın pozisyonu değil de onun yerine sürüdeki rastgele olarak seçilmiş bir ajanın pozisyonu baz alınır. Bu mekanizmada $|A| > 1$ şartı uyuyorsa ajanın o aşamada keşif yaptığı anlamına gelmektedir ve Balina Optimizasyon Algoritmasının global bir arama yapması demektir. Bunun matematiksel modeli (3,7) ve (3,8) de gösterilmiştir.

$$\vec{D} = |\vec{C} \cdot \overrightarrow{Xrand} - \vec{X}| \quad (3.11)$$

$$\vec{X}(t+1) = \overrightarrow{Xrand} - \vec{A} \cdot \vec{D} \quad (3.12)$$

Burada $\vec{X}_{rand}$ mevcut popülasyondan rastgele seçilen bir konum vektörüdür (rastgele bir balina). $A > 1$ Aralığında önceden belli olan bir global en iyi değerin etrafında olabilecek konumlardan bazıları Şekil 3.3'te gösterilmektedir. Kambur balina optimizasyon algoritması adımları ve akış diyagramı Şekil 3.5'de verilmiştir.

Popülasyonu başlatın X_i ($i = 1, 2, \dots, n$)
 Her bir arama ajanının uygunluk değerini hesaplayın
 $X^* = \text{en iyi arama ajanı}$
while ($t < \text{en fazla iterasyon sayısı}$)
for her bir arama ajanı
 Güncelle a , A , C , l ve p
if ($p < 0.5$)
if ($|A| < 1$)
 Mevcut arama ajanının konumunu Denklem (3.1)'e göre güncelleyin
else if ($|A| \geq 1$)
 Rastgele bir arama ajanı seçin (X_{rand})
 Mevcut arama ajanının konumunu Denklem (3.12)'ye göre güncelleyin
end if
if ($p \geq 0.5$)
 Mevcut arama ajanının konumunu Denklem (3.6)'ya göre güncelleyin
end if
end for
 Herhangi bir arama ajanının arama alanının sınırlarını geçip geçmediğini kontrol edin
 Her bir arama ajanının uygunluk değerini hesaplayın
 X^* 'e güncelle
 $t = t + 1$
end while
 Geri dön X^*

Şekil 3.5. Kambur balina optimizasyon algoritması adımları ve akış diyagramı.

Örnek 3.1:

$$f(x) = 2,47.x_1 + 7,35.x_2, 0 \leq x_1, x_2 \leq 1 \text{ aralığında}$$

Maksimum iterasyon Sayısı: 10

Ajan sayısı: 4 olsun.

Kambur balina optimizasyonunu kullanarak minumum $f(x)$ değerini bulunuz

Çözüm:

Çizelge 3.1. Örnek 3.1 için arama uzayındaki rastgele değerler.

Ajan	x_1	x_2	$f_{min}(x)$
(1)	0,54	0,23	3,0243
(2)	0,34	1,70	2,0893
(3)	0,84	0,64	6,7788
(4)	0,51	0,26	3,1707

Çizelge 3.1’de görüldüğü gibi en iyi çözüm 2,0893 dir. Gerekli olan parametreler

$$a_1 = 2 - t \cdot \frac{2}{\text{maksimum iterasyon sayısı}}, a_2 = -1 + t \cdot \frac{-1}{\text{maksimum iterasyon sayısı}}$$

$$t = 0, C = 2r_2, A = 2 \cdot a \cdot r_1 - a, P < 0,5 \text{ ise}$$

$|A| < 1$ olduğunda denklem (3.1)-(3.2) göre en iyi ajanına göre güncellenir.

$$\vec{D} = |\vec{C} \cdot \vec{X}^*(t) - \vec{X}^*(t)|, \vec{X}(t+1) = |\vec{X}^*(t) - \vec{A} \cdot \vec{D}|$$

Eğer $|A| \geq 1$ ise denklem (3.12)-(3.13) göre en iyi ajanına göre güncellenir.

$$\vec{D} = |\vec{C} \cdot \vec{Xrand} - \vec{X}|, \vec{X}(t+1) = \vec{Xrand} - \vec{A} \cdot \vec{D}$$

$P \geq 0,5$ ise denklem (3.6)-(3.7) göre en iyi ajanına göre güncellenir.

$$D' = |\vec{X}^*(t) - X(t)|, \vec{X}(t+1) = D' \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}^*(t)$$

$$A = 2 \cdot 2 \cdot 0,6 - 2 = 0,4, C = 2 \cdot 0,31 = 0,62$$

$p = 0,12 < 0,5$ olsun $A = 0,4 < 1$ olduğundan en iyi çözüm ajanlarımız $X_1 =$

0,34 $X_2 = 0,17$ dir. Bu ajanlarımıza göre 1. ajanlarımızın konumun güncellenmesi

$$\vec{D} = |\vec{C}.\vec{X}^*(t) - \vec{X}^*(t)| = |0.62(0.34,0.17) - (0.54,0.23)| = (0.3292,0.1246)$$

$$\begin{aligned}\vec{X}(t+1) &= |\vec{X}^*(t) - \vec{A}.\vec{D}| = (0.34,0.17) - 0.4(0.3292,0.1246) \\ &= (0.20832,0.12016)\end{aligned}$$

bulduğumuz değerleri $f(x)$ fonksiyonunda yerine yazarsak 1,3978 bulunur. $p = 0,42 < 0,5$ olsun $A = 0,4 < 1$ olduğundan 2. ajanımızın en iyi ajana göre konumunun güncellenmesi

$$\vec{D} = |\vec{C}.\vec{X}^*(t) - \vec{X}^*(t)| = |0.62(0.34,0.17) - (0.34,0.17)| = (0.1292,0.0646)$$

$$\begin{aligned}\vec{X}(t+1) &= |\vec{X}^*(t) - \vec{A}.\vec{D}| = (0.34,0.17) - 0.4(0.1292,0.0646) = \\ &= (0.29832,0.14416)\end{aligned}$$

bulduğumuz değerleri $f(x)$ fonksiyonunda yerine yazarsak 1,796 bulunur. $P = 0,75 > 0,5$ olduğundan 3. ajanımızın en iyi ajana göre konumunun güncellenmesi

$$\vec{D} = |\vec{C}.\vec{X}^*(t) - \vec{X}^*(t)| = |0.62(0.34,0.17) - (0.84,0.64)| = (0.6292,0.5346)$$

$$\begin{aligned}\vec{X}(t+1) &= |\vec{X}^*(t) - \vec{A}.\vec{D}| = (0.34,0.17) - 0.4(0.6292,0.5346) = \\ &= (0.08832, -0.04384) \cong (0.08832, 0)\end{aligned}$$

Burada X' in sınırları $0 \leq x_1, x_2 \leq 1$ olduğundan $-0.04384 \cong 0$ alınır. Bulduğumuz değerleri $f(x)$ fonksiyonunda yerine yazarsak 0.218 bulunur. 4. ajanımızın en iyi ajana göre konumunun güncellenmesi

$$\vec{D} = |\vec{C}.\vec{X}^*(t) - \vec{X}^*(t)| = |0.62(0.34,0.17) - (0.51,0.26)| = (0.2992,0.1546)$$

$$\begin{aligned}\vec{X}(t+1) &= |\vec{X}^*(t) - \vec{A}.\vec{D}| = (0.34,0.17) - 0.4(0.2992,0.1546) \\ &= (0.22032,0.10816)\end{aligned}$$

olup bulduğumuz değerleri $f(x)$ fonksiyonunda yerine yazarsak 1.34 değeri elde edilir. ajanlarımızın yeni konumları Çizelge 3.2’de gösterilmiştir.

Çizelge 3.2. Örnek 3.1 arama uzayındaki ajanlarımızın yeni konumları.

Ajan	x_1	x_2	$f_{\min}(x)$
(1)	0,20832	0,12016	1,3978
(2)	0,29832	0,14416	1,7960
(3)	0,08832	0	0,2180
(4)	0,22032	0,10816	1,3400

Bu problem bilgisayar ortamında kodlanarak 100 adımda çözdürülmüş olup 4 ajanın yeni konumları aşağıdaki çizelge 3.3’deki gibi elde edilmiştir.

Çizelge 3.3. Örnek 3.1 ajanların daha büyük adımdaki konumları.

	1. Ajan		2. Ajan		3. Ajan		4. Ajan	
0	0.8147	0.9058	0.1270	0.9134	0.6324	0.0975	0.2785	0.5469
1	-0.4165	-0.0011	0.8573	-0.2456	-0.9869	-0.3542	-0.3884	0.1893
2	0	0	0.6468	0	0	0	0	-0.1072
3	0	0	-0.9415	0	0	0	0	0
4	0	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0

Örnek 3.2 :

$F(x) = x_1^2 + x_1 \cdot x_2 + x_2^2$ fonksiyonuna kumbur Balina optimizasyon algoritması uygulayarak minimum $f(x)$ değerini bulalım.

Gerekli olan tüm parametreler:

Alt sınır: -2

Üst sınır: +2

Maksimum iterasyon Sayısı: 3

Ajan sayısı: 4

Arama uzayındaki ajanlarımıza rastgele değerler Çizelge 3.4’de gösterilmiştir.

Çizelge 3.4. Örnek 3.2 arama uzayındaki rastgele değerler.

Ajan	x_1	x_2	$f(x)$
(1)	-1,5	1	1,75
(2)	-1	1,6	1,96
(3)	0,5	-0,5	1
(4)	1,5	1,5	6,75

$$C = 2r_2 \quad a = 2 - t \cdot \frac{2}{\text{maksimum iterasyon sayısı}}$$

$$D = |C \cdot x^*(t) - x(t)|, X_{(t+1)} = X^*(t) - A \cdot D, A = 2 \cdot a \cdot r_1 - a$$

$$t = 1 \text{ için, } a = 2 - 2 \cdot \frac{1}{3} = 1,33$$

$$r_1 = 0,02 \quad r_2 = 0,03$$

$$A = 2 \cdot 1,33 \cdot 0,02 - 1,33 = -1,2768$$

$$C = 2 \cdot 0,03 = 0,06$$

En iyi arama ajanı $X_1^* = 0,5$

En iyi arama ajanı $X_2^* = -0,5$ olsun

Birinci arama ajanı için $x_1 = -1,5 \quad x_2 = 1$

$P = 0,2441 < 0,5 \quad A \geq 1, \quad | -1,2768 | \geq 1$ Olduğundan denklem (3.11)-(3.12) göre güncellenirse

$$\vec{D} = | \vec{C} \cdot \vec{X}^*(t) - \vec{X}(t) |, X_{(t+1)} = \vec{X}_{rand} - A \cdot D$$

$$\vec{D} = | 0,06 \cdot 0,5 - (-1,5, -1) | = (1,53, 1,03)$$

$$\vec{X} = (0,5, -0,5) - (-1,2768) \cdot (1,53, 1,03) = (2,4535, 0,8151)$$

İkinci arama ajanı için $x_1 = 1, x_2 = 1,6$

$$P = 0,2741 < 0,5$$

$$\vec{D} = |(0,06 \cdot (0,5,0,5) - (-1,1,6))| = (1,03,1,57)$$

$$\vec{X} = (0,5, -0,5) - (-1,2768) \cdot (1,03,1,57) = (1,8151,1,5045)$$

Üçüncü arama ajanı için $x_1 = 0,5, x_2 = -0,5, P = 0,0244 < 0,5$

$$\vec{D} = |0,06 (0,5, -0,5) - (0,5, -0,5)| = (0,47,0,47)$$

$$\vec{X} = (0,5, -0,5) - (-1,2768)(0,47,0,47) = (1,1,0,1)$$

Dördüncü arama ajanı için $x_1 = 1,5, x_2 = 1,5, P = 0,7417 > 0,5$

$$\vec{D} = |\vec{X}^*(t) - \vec{X}(t)|, \quad \vec{X}_{(t+1)} = \vec{D} \cdot e^{bl} \cdot \cos(2\pi L) + X^*(t)$$

$$D = |0,5 - 1,5| = 1, \quad a_2 = -1 + t \cdot \frac{-1}{\text{maksimum iterasyon sayısı}}$$

$t = 0, a_2 = -1$ için $l = (a_2 - 1)\vec{r}_{rand} + 1$ ve $r = 0,2412$ ' den

$$l = (-1 - 1) \cdot (0,2412) + 1 = 0,5176, b = 1$$

$$X_1 = 1 \cdot e^{0,5176} \cdot \cos(2\pi \cdot 0,5176) + 0,5 = -1,1677$$

$x_2 = 1,5$ için

$$D = |-0,5 - 1,5| = 2, X_2 = 2 \cdot e^{0,5176} \cdot \cos(2\pi \cdot 0,5176) - 0,5 = -3,8354$$

ajanların yeni konumları Çizelge 3.5. de verilmiştir.

Çizelge 3.5. Örnek 3.2 için arama uzayındaki ajanların yeni konumları.

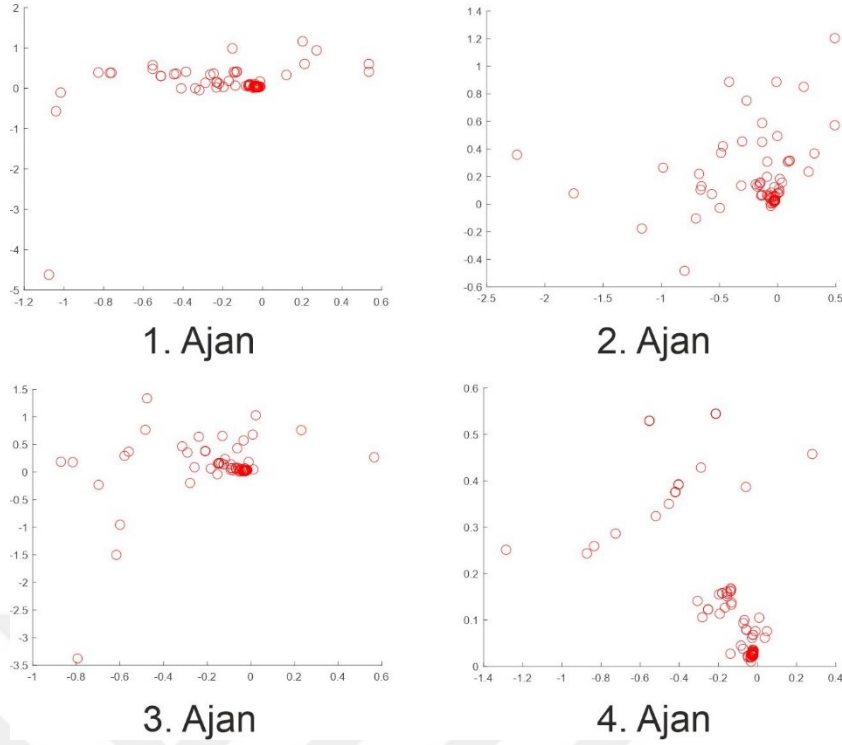
X_1	X_2	$F_{min}(x)$
2,4535	0,8151	8,8638
1,8151	1,5045	8,2889
1,1000	0,1000	1,3300
0,5176	-3,8354	12,9929

Bu problem Bilgisaray ortamında kodlanarak 100 adımda çözdürülmüş olup 4 ajanın yeni konumları aşağıdaki gibi çizelge 3.6. de verilmiştir.

Çizelge 3.6. Örnek 3.2 için ajanların daha büyük adımdaki konumları.

	1. Ajan		2. Ajan		3. Ajan		4. Ajan	
0	0.9938	-0.5876	-0.2142	-1.8643	0.9121	0.4855	-1.4968	-1.1382
1	0.2020	1.1616	-0.2671	0.7504	-0.5996	-0.9547	-0.5531	0.5295
5	-0.1512	0.9870	-0.8000	-0.4838	-0.4835	0.7660	-1.2851	0.2512
10	-0.4461	0.3501	-1.1675	-0.1769	-0.8707	0.1873	-0.4203	0.3757
20	-0.1450	0.4073	-0.0025	0.4936	-0.2091	0.3805	-0.7257	0.2864
30	0.1200	0.3313	-0.1488	0.1565	-0.1463	0.1629	-0.1372	0.1683
50	-0.0403	0.0516	-0.0597	0.0393	-0.0432	0.0548	-0.0288	0.0613
100	-0.0231	0.0230	-0.0230	0.0230	-0.0231	0.0229	-0.0231	0.0229

Ajanlarımızın şekil 3.6'da görüldüğü gibi sıfır noktası ve etrafında konumlanmaktadır.



Şekil 3.6. Balinaların (Ajanların) avlanma sırasındaki hareketleri.

Örnek 3.3:

$f(x) = x_1 + x_2^2$, $0 \leq x_1, x_2 \leq 1$ fonksiyonun $f_{max}(x)$ değerini kumbur balina optimizasyon algoritması kullanarak bulunuz.

Çözüm: İlk olarak $[0,1]$ aralığında 4 tane rastgele x_1, x_2 değeri alalım. Bu değerler Çizelge 3.7’de verilmiştir:

Çizelge 3.7. Örnek 3.3 için arama uzayındaki rastgele değerler.

X_1	X_2	$f(x)$
0,5499	0,6021	0,9124
0,0682	0,3507	0,1911
0,3135	0,9746	1,2633
0,8803	0,2761	0,9565

$$C = 2r_2, A = 2 \cdot a \cdot r_1 - a, P < 0,5 D = |C \cdot X^*(t) - X(t)|$$

$$P \geq 0,5, D = |X^*(t) - X(t)|$$

Burada X^* $x_1 = 0,3135, x_2 = 0,9746, f(x) = 1,2633$ alınır.

$$\vec{X}(t+1) = \begin{cases} \vec{X}^*(t) - \vec{A} \cdot \vec{D} & p < 0.5 \\ De^{bl} \cdot \cos(2\pi l) + \vec{X}^*(t) & p \geq 0.5 \end{cases}$$

b , logaritmik spirali şeklini tanımlamak için gereken bir sabittir. l ise $[-1, 1]$ aralığında bulunan herhangi bir rastgele bir değerdir.

$b = 1, l = 0,2845$ alalım.

Çizelge 3.8. Örnek 3.3 için av arama için gerekli olan tüm parametreler.

a [2,0]	r [0,1]	A [-1,1]	$C = 2r$	P [0,1]	D_1	D_2
0,9781	0,2288	-0,5304	0,4576	0,3511	0,4064	0,1561
0,8550	0,3507	-0,2553	0,7014	0,8964	0,3134	0,97463
0,8318	0,4989	-0,0018	0,9978	0,7561	0,3135	0,9746
0,7740	0,5586	0,0907	1,1172	0,3677	0,5300	0,8127

Çizelge 3.8 da verilen parametreler kullanılarak arama uzayındaki ajanların yeni konumları bulunur sonuçlar çizelge 3.9’da verilmiştir.

Çizelge 3.9. Örnek 3.3 için arama uzayındaki ajanların yeni konumları.

X_1	X_2	$f_{max}(x)$
0,5290	1,0573	1,6468
0,1481	0,4307	0,1911
0,3140	0,9751	1,2633
0,2654	0,9008	1,0768

Bu problem bilgisayar ortamında kodlanarak 100 adımda çözdürülmüş olup 4 ajanın yeni konumları aşağıda çizelge 3.10’da verilmiştir.

Çizelge 3.10. Örnek 3.3 için ajanların daha büyük adımdaki konumları.

	1. Ajan		2. Ajan		3. Ajan		4. Ajan	
0	0.6877	0.5684	0.3808	0.6346	0.3632	0.4076	0.3687	0.4684
1	0.6282	0.4617	0.5293	0.3718	-0.1252	-0.1212	0.2714	0.6841
5	1.0126	1.0920	1.0000	1.0000	1.0000	1.0000	0.9979	0.9967
10	-1.1261	-1.1261	1.0965	1.0965	1.0000	1.0000	1.0348	1.0348
20	1.0000	1.0000	0.3177	0.3177	1.6288	1.6288	-0.2199	0.2927
30	1.3091	1.2747	1.0000	1.0000	0.8134	0.8134	1.0000	1.0000
50	1.0055	1.0055	1.0000	1.0000	1.2072	1.2072	0.9596	0.9596
100	1.0146	1.0146	0.9972	0.9972	1.0000	1.0000	1.0000	1.0000

Örnek 3.4:

$$F(x) = \sum_{i=1}^n X_i^2$$

fonksiyonun $n = 1$ için kibur Balina optimizasyon algoritması uygulayarak minimum $f(x)$ değerini bulalım.

Çözüm:

$$F(X_i) = \sum_{i=1}^1 X_i^2 = X_1^2$$

Çözüme geçmeden ajanları belirleyelim. Bunun için alt sınır -10, üst sınır 10, maksimum iterasyon sayısı 10, ajan sayısı: 4 olsun. Bu durumda arama ajanları Çizelge 3.11'deki gibi olsun.

Çizelge 3.11. Örnek 3.4 için arama uzayındaki rastgele değerler.

	X	X^2
1. ajan	1,3	1,69
2.ajan	2	4
3. ajan	0,6	0,36
4.ajan	4,1	16,81

Burada konum güncellememizde kullanacağımız parametlerden A ve C denklem (3.3)-(3.4) den hesaplanır. $A = 2 \cdot a \cdot r_1 - a$ $C = 2r_2$ Denklem (3,3)' de yer alan a ise iterasyon boyunca 2' den 0'a kadar lineer olarak azalır:

$$a_1 = 2 - t \cdot \frac{2}{\text{maksimum iterasyon sayısı}}$$

$$t = 0 \text{ için } a = 2, r_1 = 0,2356, r_2 = 0,6542$$

$$A = 2 \cdot 2 \cdot 0,2356 - 2 = -1,0576$$

$$C = 2 \cdot 0,6542 = 1,3084 \text{ bulunur.}$$

En iyi arama ajanı $X^* = 0,6$ dır. Bu ajana göre diğer arama ajanlarının konumlarını güncelleyelim

1. arama ajanı için $P = 0,2691 < 0,5$ olsun ve $A \geq 1, | -1,0576 | \geq 1$ olduğundan bu durumda denklem (3.10) belirtilen $X_{(t+1)} = X^*(t) - A \cdot D$ kullanılır. Burada D bulunurken (3.1) denklemini kullanılır.

$$D = | C \cdot X^*(t) - x(t) | = D = | 1,3084 \cdot 0,6 - 1,3 | = 0,51496$$

bulunur.

Ajanımızın yeni konu son olarak

$$X_{(t+1)} = \vec{X}_{rand} - A \cdot D = 0,6 - (-1,0576) \cdot 0,51496 = 1,1495$$

bulunur. Benzer şekilde diğer ajanlarımız için de aynı yöntem uygulanılır ve yeni konumları bulunur.

2. arama ajanı için

$$P = 0,3525 < 0,5, \quad | -1,0576 | \geq 1, A \geq 1$$

$$D = | 1,3084 \cdot 0,6 - 2 | = 1,2149$$

$$x_2 = 0,6 + 1,0576.1,2149 = 1,8848$$

3. arama ajanı için;

$$P = 0,3196 < 0,5$$

$$A = |-1,0576| \geq 1 \text{ Olduğu için}$$

$$D = |1,3084.0,6 - 0,6| = 0,1850$$

$$x_3 = 0,6 + 1,0576.0,1850 = 0,7956$$

4. arama ajanı için $P = 0,6444 > 0,5$, $D = |0,6 - 4,1| = 3,5$ ve $t = 0$ için

$$a_2 = -1 + t. \frac{-1}{\text{maksimum iterasyon}} = -1, l = (a_2 - 1).r_{rand} + 1$$

$$r = 0,3245, l = (-1 - 1).0,3245 + 1 = 0,351, b = 1$$

$$x_4 = 3,5. e^{0,351}. \cos(2\pi.0,351) + 0,6 = -2,3475$$

Ajanlarımızın yeni konumları Çizelge 3.12’de verilmiştir.

Çizelge 3.12. Örnek 3.4 için arama uzayındaki ajanların yeni konumları.

X	$F(x) = X^2$	X	$F(x) = X^2$
1,3	1,69	1,1495	1,3213
2	4	1,8848	3,5524
0,6	0,36	0,7956	0,6329
4,1	16,81	-2,3475	5,5107

Bu problem bilgisayar ortamında kodlanarak 100 adımda çözdürülmüş olup 4 ajanın yeni konumları aşağıdaki çizelge 3.13 deki gibi elde edilmiştir.

Çizelge 3.13. Örnek 3.4 için ajanların daha büyük adımdaki konumları.

	1. Ajan	2. Ajan	3. Ajan	4. Ajan
0	-2.6923	1.7020	-6.333	-8.4616
1	4.4466	-20.8154	0.6662	2.1806
2	5.2369	-1.4860	0.6898	1.1818
10	0.0339	-1.3964	-1.8366	0.0001
30	1.0221	0.0000	-0.2004	0.0000
50	$0.0111 \cdot 10^{-5}$	$0.0011 \cdot 10^{-5}$	$0.6116 \cdot 10^{-5}$	$0.0014 \cdot 10^{-5}$
100	$-0.1912 \cdot 10^{-17}$	$-0.1915 \cdot 10^{-17}$	$-0.1910 \cdot 10^{-17}$	$-0.1907 \cdot 10^{-17}$

4. GRİ KURT VE KAMBUR BALINA OPTİMİZASYON YÖNTEMLERİNİN BİR UYGULAMASI

Gri sistem, belirsizlik sorununu inceleyen ve ilk kez 1982 yılında Deng tarafından önerilen bir disiplindir ve gri sistem teorisinde önemli bir rol oynar . Küçük örneklem ve sınırlı veri setine sahip belirsiz veya sapkın sistemleri işlemek için, makine öğrenme modelleri, hibrit modeller ve ampirik modellerden farklı olarak, yararlı bir araç olmuştur. Gri tahmin modelleri, bilim ve teknoloji, enerji, çevre sorunları, ekonomi, sağlık ve diğer alanlar gibi çeşitli alanlarda yaygın ve başarılı bir şekilde uygulanmıştır . Diğer tahmin yöntemlerine göre daha az sayıda bilgiye ihtiyaç duyan Gri modelleme yöntemi pratikliği açısından da son yılların popüler konuları arasına girmiştir.

Gri modelleme üzerine yapılan son çalışmalar, iki ana amaca odaklanmaktadır: pratiklik ve tahmin doğruluğu. Bu amaçlar doğrultusunda, son zamanlarda önemli çalışmalar gerçekleştirilmiştir. Son yıllarda kesirli mertebeden türev operatörlerini içeren matematiksel modellerin daha kesin sonuçlar verdiği değerlendirilmektedir. Buna rağmen, bir modeldeki kesir mertebenin tam olarak kaç olacağı ise en büyük problemlerin başında gelmektedir. Bu kısımda yaptığımız çalışmada, elimizde var olan ham verilere göre en iyi tahmin üretecek şekilde bir gri modelin mertebesinin optimizasyonu yapıldı. Kesir mertebeden operatör olarak uyumlu türev operatörü kullanıldı.

4.1 Uyumlu Kesir Hesabına Dair Bazı Tanımlar ve Özellikler

Bu bölümde, uyumlu kesir türevine dair bazı faydalı tanımlar ve özellikler özetlenmiştir.

4.1.1 Uyumlu kesir türevi

Tanım 4.1.

Eğer $f: [0, \infty) \rightarrow \mathbb{R}$ diferensiyellenebilir bir fonksiyon ise, $\alpha \in (n, n + 1]$ olan uyumlu kesir türevi şu şekilde tanımlanır.

$$T_{\alpha}(f)(t) = \lim_{\varepsilon \rightarrow 0} \frac{f(t + \varepsilon t^{[\alpha]-\alpha}) - f(t)}{\varepsilon} = t^{[\alpha]-\alpha} \frac{df(t)}{dt} \quad (4.1)$$

Burada $[\cdot]$ üstel fonksiyondur, yani $[\alpha]$, α dan büyük veya eşit en küçük tamsayıdır. Açıktır ki $[\alpha] = 1$ dir. eğer $\alpha \in (0,1]$ ise dolayısıyla, denklem (4.1) şu şekilde yazılabilir.

$$T_{\alpha}(f)(t) = \lim_{\varepsilon \rightarrow 0} \frac{f(t + \varepsilon t^{1-\alpha}) - f(t)}{\varepsilon} = t^{1-\alpha} \frac{df(t)}{dt} \quad (4.2)$$

$t > 0$ için aşağıdaki teorem tanımın özelliklerini verir.

Teorem 4.1. [16] Eğer f ve g fonksiyonları deferensiyellenebilir ve $\alpha \in (0,1]$ ise, her $c, p, m, n \in \mathbb{R}$ için

1. $T_{\alpha}(f)(t) = t^{1-\alpha} \frac{df(t)}{dt}$
2. $T_{\alpha}(mf + ng) = mT_{\alpha}(f) + nT_{\alpha}(g)$
3. $T_{\alpha}(f \cdot g) = f \cdot T_{\alpha}(g) + gT_{\alpha}(f)$
4. $T_{\alpha}\left(\frac{f}{g}\right) = \frac{g \cdot T_{\alpha}(f) - fT_{\alpha}(g)}{g^2}$
5. $T_{\alpha}(c) = 0$
6. $T_{\alpha}(t^p) = p t^{p-\alpha}$
7. $T_{\alpha}(e^{cx}) = c x^{1-\alpha} e^{cx}$

4.2 Kesirli Gri Model Formulasyonu

Orijinal serisi $X^{(0)} = (x^{(0)}(1), x^{(0)}(2), \dots, x^{(0)}(n))$ verilmiştir. Birikimli toplamlar dizisi aşağıdaki gibidir:

$$X^{(\alpha)} = (x^{(\alpha)}(1), x^{(\alpha)}(2), \dots, x^{(\alpha)}(n))$$

$$X^{(\alpha)}(k) = \nabla^{\alpha} x^{(0)}(k) = \sum_{i=1}^k \left[\begin{matrix} [\alpha] \\ k-i \end{matrix} \right] \frac{x^{(0)}(i)}{i^{[\alpha]-\alpha}}, \alpha \in \mathbb{R}^+ \quad (4.3)$$

$$\begin{bmatrix} [\alpha] \\ k-i \end{bmatrix} = \frac{\Gamma(k-i+[\alpha])}{\Gamma(k-i+1)\Gamma([\alpha])} = \frac{(k-i+[\alpha]-1)!}{(k-i)! + ([\alpha]-1)!}$$

4.2. Tanım

ECFGM (1,1) modeli için birinci dereceden beyazlatma diferansiyel denklemi aşağıdaki şekilde tanımlanır.

$$\frac{dx^{(\alpha)}(t)}{dt} + \alpha x^{(\alpha)}(t) = b + ce^{-t}. \quad (4.4)$$

Burada α büyüme katsayısıdır. b ye hareket katsayısı denir ve ce^{-t} üstel gri aksiyon

miktardır. Böylece monoton azalan terim ce^{-t} tahmin hatasının büyümesini engeller.

Eğer $\alpha = 1$ ise ECFGM(1,1) modeli EXGM(1,1) modeline dönüşür. Ayrıca, önerilen model $c = 0$ olduğunda geleneksel CFGM(1,1) modeline çevrilebilir.

4.3. Parametrelerin Belirlenmesi

Teoerem 4.1.

ECFGM(1, 1) modelinin sistem parametreleri a , b ve c aşağıdaki denklemi sağlar. Aslında bu ifade en küçük kareler yönteminin bir sonucudur.

$$[a, b, c]^T = (B^T B)^{-1} B^T Y, \quad (4.5)$$

Burada B ve Y Matris aşağıdaki şekilde bulunur.

$$B = \begin{bmatrix} -0.5(x^{(\alpha)}(2) + x^{(\alpha)}(1)) & 1 & (e-1)e^{-2} \\ -0.5(x^{(\alpha)}(3) + x^{(\alpha)}(2)) & 1 & (e-1)e^{-3} \\ \vdots & \vdots & \vdots \\ -0.5(x^{(\alpha)}(n) + x^{(\alpha)}(n-1)) & 1 & (e-1)e^{-n} \end{bmatrix} \quad (4.6)$$

$$Y = \begin{bmatrix} x^{(\alpha)}(2) - x^{(\alpha)}(1) \\ x^{(\alpha)}(3) - x^{(\alpha)}(2) \\ \vdots \\ x^{(\alpha)}(n) - x^{(\alpha)}(n-1) \end{bmatrix} \quad (4.7)$$

4.3.1. Yanıt Fonksiyonu ve Geri Yüklenen Değerler

Teorem 4.2.

ECFGM(1,1) Fonksiyonun ayrık formunun modeli şu şekilde verilmiştir.

$$\hat{x}^{(\alpha)}(k) = \left(x^{(0)}(1) - \frac{b}{a} - \frac{c}{a-1} e^{-1} \right) e^{a(1-k)} + \frac{b}{a} + \frac{c}{a-1} e^{-k} \quad (4.8)$$

Burada $k = 2, 3, \dots, n$.

Teorem 4.3. [16]

$\hat{x}^{(\alpha)}(k)$ verilerinden $\hat{x}^{(0)}(k)$ veri tahminlerine dönüş aşağıdaki gibidir:

$$\hat{x}^{(0)}(k) = \nabla^\alpha \hat{x}^{(\alpha)}(k) = k^{[\alpha]-\alpha} \nabla^{n+1} \hat{x}^{(\alpha)}(k), \alpha \in (n, n+1] \quad (4.9)$$

burada $k = 2, 3, \dots, n$.

İspat : denklemden (4.3) de görüldüğü gibi $\hat{x}^{(\alpha)}(k) = \nabla^\alpha \hat{x}^{(0)}(k)$ ters operatör ∇^α uygulanırsa şu şekilde elde edilir.

$\hat{x}^{(0)}(k) = \nabla^\alpha \hat{x}^{(\alpha)}(k)$ elde edilir e şu şekilde yazılır.

$\hat{x}^{(0)}(k) = \nabla^\alpha \hat{x}^{(\alpha)}(k) = k^{[\alpha]-\alpha} \nabla^{n+1} \hat{x}^{(\alpha)}(k), \alpha \in (n, n+1]$ bu kanıtı tamamlar.

Görüldüğü gibi, $\alpha \in (0,1]$ için geri kazanılan değerler şu şekilde yazılabilir:

$$\hat{x}^{(0)}(k) = k^{1-\alpha} (\hat{x}^{(\alpha)}(k) - \hat{x}^{(\alpha)}(k-1)) \quad (4.10)$$

4.4. Tahmin Modelinin Doğruluk Analizi

ECFGM(1,1) 'in uyum ve tahmin performansını değerlendirmek için göreceli yüzde hata (RPE) ve ortalama mutlak yüzde hata (MAPE) kullanılır. En düşük MAPE değeri en iyi tahmin modelini gösterir. Bunlar şu şekilde tanımlanır.

$$RPE(k) = \left| \frac{\hat{x}^{(0)}(k) - x^{(0)}(k)}{x^{(0)}(k)} \right| \times 100\% \quad (4.11)$$

$$MAPE = \frac{1}{n} \sum_{k=1}^n RPE(k). \quad (4.12)$$

Burada $x^{(0)}(k)$ orijinal seriyi temsil eder $\hat{x}^{(0)}(k)$ tahmin edilen seriyi temsil eder. Burada n örnekle bir dizi için uyum performans metriği $MAPE_{fit}$ ve $MAPE_{pre}$ ise tahmin performansı metriği olarak tanımlanır. Matematiksel olarak şu şekilde formüle edilebilirler:

$$MAPE_{fit} = \frac{1}{p} \sum_{k=1}^p RPE(k) \quad (4.13)$$

$$MAPE_{pre} = \frac{1}{n-p} \sum_{k=p+1}^n RPE(k). \quad (4.14)$$

Burada p , bir modeli uyumlamak için kullanılan örneklerin sayısını temsil ederken, ham dizinin kalan kısmı modelin tahmin doğruluğunu incelemek için kullanılır. Toplam MAPE, denklem (4.10) dan bulunur.

Yenilenebilir temiz enerji kaynaklarından biri rüzgardır. Genellikle diğer birçok enerji kaynağıyla karşılaştırıldığında, rüzgar kullanarak enerji üretmenin çevre üzerindeki etkileri daha azdır. Havayı veya suyu kirletebilecek herhangi bir emisyon salınımı yoktur ve soğutma için suya ihtiyaç duymazlar. Rüzgar türbinleri ayrıca fosil yakıtlardan enerji üretimini azaltma avantajına da sahip olabilir, bu da toplam hava kirliliği ve karbon dioksit emisyonlarının azalmasıyla sonuçlanır ve enerji kıtlığı sorununu çözmeye yardımcı olabilir. Tarih öncesinden bugüne kadar, insanlar rüzgar enerjisini yelken açmak, yel değirmenleri ve rüzgar türbinleri için kullanmışlardır. Elektrik jeneratörleri, rüzgar enerjisini elektrik enerjisine dönüştürür

Büyük kara kütlesi ve kıyı şeridinde sahip en büyük ülkelerden biri olan Çin, zengin rüzgar kaynaklarına sahiptir. Çin Meteoroloji İdaresi tarafından yapılan değerlendirmelere göre, yerden nispeten düşük bir yükseklik olan 10 m üzerinde, Çin'in toplam teorik rüzgar enerjisi rezervleri 4350 GW, teknik olarak kullanılabilir rüzgar kaynakları ise 297 GW olarak tahmin edilmektedir

Bu kısımda, Çin'deki rüzgar enerjisi tüketimini tahmin etmek için ECFGM (1,1) modeli kullanıldı. Modelin yönetici diferensiyel denklemi olan (4.4) denklemindeki kesir merteye olan α nın değerinin belirlenmesi en önemli problemlerden biridir. Bir çok uygulamada optimum α değeri aranmadan α nın farklı değerleri için sonuçlar verilir. Ancak bu yaklaşım bilimsel olarak ve problemin çözümü için büyük bir eksiklik olarak görülür. Bu kısımda Çin'deki rüzgar enerjisi tüketimini tahmin etmek için kullandığımız ECFGM (1,1) modelindeki optimum α değerini (0,1) aralığında gri kurt vekambur balina optimizasyonları ile araştırıldı. Buradaki amaç en düşük MAPE değerlerini verecek optimum kesir mertebeye ulaşmaktır.

Erdoğan ve arkadaşları [16] tarafından yapılan çalışmada 2009-2020 yılları arasındaki, Çin'in rüzgar enerjisi tüketimini için ham veriler aşağıdaki gibi verilmiştir.

Çizelge 4.1. Çin'in yıllık rüzgar enerjisi tüketim verileri.

YIL	Ham Veri
2009	6.25
2010	10.10
2011	15.91
2012	21.72
2013	31.95
2014	35.32
2015	42.03
2016	53.64
2017	66.75
2018	82.82
2019	93.31
2020	107.30

Bu verilere göre Erdoğan ve arkadaşları [16] tarafından yapılan çalışmada optimal α değeri $\alpha = 0.3319$ ve buna bağlı parametreler $a = -0.10331$, $b = 6.39111$ ve $c = -4.39633$ olarak hesaplandı. Bu çalışmada araştırmacılar kaba kuvvet algoritması kullandılar.

α değerinin tezde bahsedilen Gri kurt ve Kambur balina algoritmaları ile hesaplanması için kullanılan algoritmanın kaba kodu aşağıda verilmiştir. Gri kurt için maksimum iterasyon sayısı 10, kambur balina için ise 50 olarak alınmıştır. Arama ajanlarının sayısı gri kurt algoritmasında 10, kambur balinada ise 5'tir. Kesirli mertebeden türev α ise [0,1] aralığında aranmış olup bilinmeyen sayısı bir olduğu için

parametre uzay boyutu birdir. Şekil 4.1 de α hesabında kullanılan algortimanın kaba kodu verilmiştir. Optimizasyon yöntemlerinin parametre ve hata uzay grafikleri Şekil 4.2 ve Şekil 4.3’de verilmiştir.

Giriş: Arama ajanlarının (gri kurt yada kambur balina) sayısı, Maksimum iterasyon sayısı (Ite_{maks}), Arama uzayının alt ve üst sınırı, Parametre uzay boyutu

Çıkış: Kesirli mertebeden türevin (α) optimum değeri

*Arama ajanlarının başlangıç değerlerini ata

while ($t \leq Ite_{maks}$)

for her bir arama ajanı

* Ham veriler kullanılarak (4.3) ile birikimli toplamalar dizisi oluşturulur.

*(4.6)-(4.7) ile B ve Y matrisleri oluşturulur.

*(4.5) ile a, b, c katsayıları hesap edilir.

*(4.8) ile tahmini $\hat{x}^{\alpha}(k)$ elde edilir.

*(4.9) ile $\hat{x}^0(k)$ dizisi elde edilir.

*(4.12) ile fitness değeri hesaplanır.

end for

İlgili optimizasyon yöntemine göre (WHO yada GWO) arama ajanlarının konumu güncellenir.

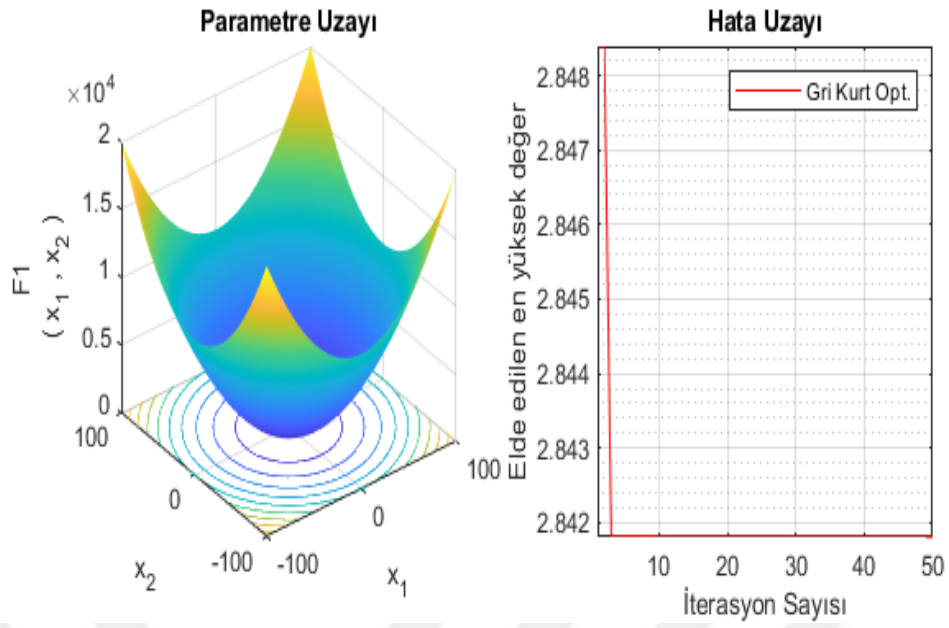
$t = t + 1$

end while

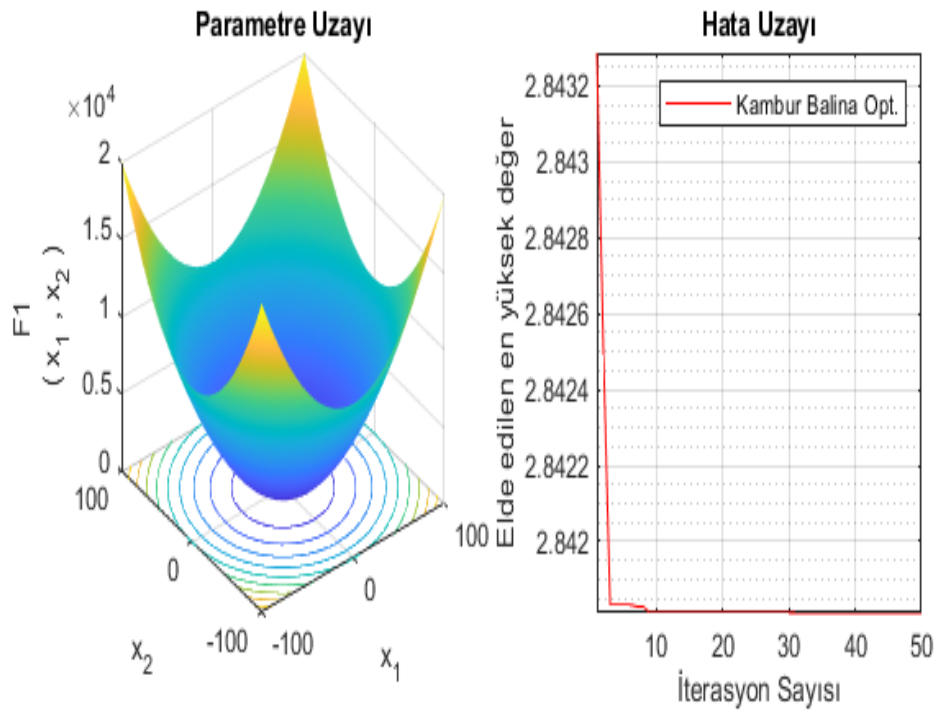
En iyi (α)’yı geri döndür.

Şekil 4.1. α hesabında kullanılan algortimanın kaba kodu.

Çizelge 4.1’deki verilere göre gri kurt ve kambur balina optimizasyonları ile optimal α değeri sırasıyla $\alpha = 0.3327$ ve $\alpha = 0.3328$ olarak elde edildi. Elde edilen optimal parametrelere göre tahmin sonuçları elde edildi ve Tablo 4.2 de gösterildi. Her iki optimal α değeri için de MAPE değerleri yaklaşık olarak 2.8418 elde edildi.



Şekil 4.2. Gri kurt yöntemi.



Şekil 4.3. Kambur balina yöntemi.

Çizelge 4.2. Çin'in yıllık rüzgar enerjisi tüketim verileri tahminleri.

YIL	Tahmin Verileri (Gri Kurt Opt.)	Tahmin Verileri (Kambur Balina Opt.)
2009	6.25	6.25
2010	10.0531	10.0533
2011	16.0458	16.0462
2012	22.2011	22.2009
2013	28.8473	28.8474
2014	36.2439	36.2435
2015	44.5940	44.5937
2016	54.0792	54.0799
2017	64.8827	64.8824
2018	77.1936	77.1931
2019	91.2229	91.2235
2020	107.2061	107.2066

Çizelge 4.2 incelendiğinde ham veriler ile tahmini verilerin birbiriyle oldukça uyumlu olduğu görülmektedir. α değeri her iki yöntem için birbirine oldukça yakındır. Her iki yöntemle elde edilen tahmini değerlerin virgülden sonraki 2. basamakta aynı oldukları görülmektedir. Ayrıca her iki yöntemde elde edilen α değeri Erdinç ve arkadaşları [16] tarafından kaba kuvvet algoritması ile elde edilen değer ile yakındır.

5. SONUÇ

Optimizasyon son zamanlarda oldukça popüler olan zor bir çalışma alanıdır. Çok çeşitli mühendislik ve matematik problemlerinin çözümü için bir çok optimizasyon yöntemi geliştirilmiştir. Bu yöntemler karmaşık gerçek yaşam problemlerinin ve birden fazla yerel optimuma sahip problemlerin çözümü için yetersiz kalmaktadır. Türev bilgisine dayanan optimizasyon yöntemlerinin çeşitli dezavantajlarından dolayı araştırmacılar daha kullanışlı yöntemleri geliştirmek için çalışmalar yapmıştır ve bu çalışmalar sonucunda ortaya çıkan metasezgisel algoritmalar mühendislik problemlerinin, matematik problemlerinin ve gerçek hayat problemlerinin optimize edilmesinde oldukça başarılı sonuçlar üretmiştir.

Tez çalışmasında literatürde yer alan ve sıklıkla kullanılan metasezgisel optimizasyon algoritmaların matematiksel modellerinden Gri Kurt ve Kambur Balina Optimizasyonlarını kullanarak , örnek çalışması yapılmış ve sonunda popülasyondaki en iyi bireyi kullanarak kambur balinaların avlama davranışını modelleyen Balina Optimizasyon Algoritması, popülasyondaki en iyi üç bireyi kullanarak gri kurtların sosyal hiyerarşi ve avlamadavranışını modelleyen Gri Kurt Optimizasyon Algoritması üzerine basit matematiksel maksimum ve minimum problemleri çözülmüş ve yöntemlerin etkinliği izlenmiştir. Her iki yöntemdeki arama ajanlarının çözüm uzayında dağılımları incelenmiş ve Gri kurt yönteminin kambur balina yöntemine göre daha fazla yayılım gösterdiği belirlenmiştir. Ardından Gri kurt ve Kambur balina optimizasyonları yardımıyla literatürdeki popüler tahmin yöntemlerinden biri olan NFGM (1,1)'deki kesirli mertebeden türev α değeri hesap edilmiştir. Sonuçların birkaç adımda elde edildiği ve birbiriyle uyumlu olduğu görülmüştür. Sonuçların Erdiñç ve arkadaşları [16] ile yakın olması ve düşük iterasyonda elde edilmesi her iki yöntemin de etkinliğini göstermektedir.

KAYNAKÇA

1. Dalmaz, H., 2023. GWO ve MFO algoritmalarının hibritlenmesiyle sürü zekası tabanlı optimizasyon algoritması geliştirilmesi ve yapay sinir ağı yaklaşımıyla ağ saldırılarının tespitinde kullanılması, Yüksek Lisans Tezi, Kırıkkale Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Ana Bilim Dalı, Kırıkkale.
2. Doğan, C., 2019. Balina Optimizasyon Algoritması ve Gri kurt optimizasyonu algoritmaları kullanılarak yeni hibrit optimizasyon algoritmalarının geliştirilmesi, Yüksek Lisans Tezi, Erciyes Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Ana Bilim Dalı, Kayseri.
3. Mirjalili S., Lewis A., 2016. The Whale Optimization Algorithm, *Advances in Engineering Software*, 95, 51, 67.
4. Mirjalili, S., Mirjalili S.M. ve Lewis A., 2014. Grey wolf Optimizer, *Advances in Engineering, Software*, 69, 46-61.
5. Toren, M. ve Mollahasanoglu, H., 2023. Gri Kurt Optimizasyon Algoritması ile güç ve dağıtım türü transformatörlerin ağırlık optimizasyonu, *Ömer Halisdemir Üniversitesi Mühendislik Bilimleri Dergisi*, 12, 1, 1-9.
6. Yetkin, M., 2016. Metasezgisel Algoritmaların Jeodezi’de Kullanımı, *Geomatik Dergisi*, 1, 1.
7. Karaboğa, D., 2014. Yapay Zeka Optimizasyon Algoritmaları, Nobel Akademik Yayıncılık, Ankara.
8. Yang, X.S., 2010. Firefly Algorithm, Stochastic Test Functions and Design optimisation, *International Journal of Bio-Inspired Computation*, 2, 2, 78-84.
9. Jayakumar, N., Subramanian, S., Ganesan, S. ve Elanchezhian, E.B. 2016. Grey Wolf Optim "ization for Combined Heat and Power Dispatch with Cogeneration Systems. *International Journal Electronic, Power Energy systems*, 74, 252-264.
10. Koc, I., Baykan, O.K., Babaoglu, I., 2018. Gri kurt optimizasyon algoritmasına dayanan çok seviyeli imge eşik seçimi, *J. Polytech.*, 21, 4, 841-847.
11. URL-1 <<https://www.mathworks.com/matlabcentral/fileexchange/55667-the-whale-optimization-algorithm>>, Erişim Tarihi 15.05.2024
12. Hof, P.R, Van Der Gucht E., 2007. Structure of the cerebral cortex of the humpback whale, *Megaptera novaeangliae* (Cetacea, Mysticeti, Balaenopteridae), *Anatomical Record*, 290, 1–31.
13. Watkins, W.A., 1979. Schevill WE. Aerial observation of feeding behavior in four baleen whales: *Eubalaena glacialis*, *Balaenoptera borealis*, *Megaptera novaeangliae*, and *Balaenoptera physalus*, *J Mammal*, 155–63.

14. Goldbogen J.A., Friedlaender, A.S., Calambokidis, J., Mckenna, M.F., Simon, M., Nowacek, D.P., 2013. Integrative approaches to the study of baleen whale diving behavior, feeding performance, and foraging ecology, *BioScience* 63, 90–100.
15. Büyüközer, A., 2019. Benzetilmiş tavlamaalı hibrit balina optimizasyon algoritmasının karesel atama problemine uygulanması, Yüksek Lisans Tezi, Karabük Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı, Karabük.
16. Erdiñç, Ü., Bilgil, H. ve Öztürk, Z., 2022. A novel fractional forecasting model for time dependent real world cases, *Revstat Statistical Journal*, 33, 1-21.



ÖZGEÇMİŞ

Adı ve Soyadı : Abdulsamed ÖZKEPİR

EĞİTİM BİLGİLERİ (Kurum ve Yıl)

Lisans : Aksaray Üniversitesi, Fen Edebiyat Fakültesi, 2011-2016

MESLEKİ DENEYİM

1. Aksaray Özel Ak İrfan Koleji, 2018-2022
2. Özel Üç Dört Beş Özel Ders Kişisel Gelişim Kurs Merkezi, 2022-2024

TEZDEN ÜRETİLEN YAYINLAR, SUNUMLAR VE PATENTLER

1. Özkepir, A., Bilgil, H. ve Çelik, E. 2024. Gri Kurt ve Humpty Whale optimizasyonlarının matematiksel modellemede uygulamaları, GAP 12. Uluslararası Zirve Bilimsel Araştırma Kongresi, 29-31 Mayıs Gaziantep, Türkiye.