



**BALON BALIĞI DAVRANIŞI TABANLI YENİ BİR METASEZGİSEL
OPTİMİZASYON ALGORİTMASI**

Mehmet Cem ÇATALBAŞ

Doktora Tezi

Elektrik Elektronik Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Arif GÜLTEN

EKİM-2018

T.C
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**BALON BALIĞI DAVRANIŞI TABANLI YENİ BİR METASEZGİSEL
OPTİMİZASYON ALGORİTMASI**

DOKTORA TEZİ

Mehmet Cem ÇATALBAŞ

(13211301)

Tezin Enstitüye Verildiği Tarih : 5 Eylül 2018
Tezin Savunulduğu Tarih : 5 Ekim 2018

Tez Danışmanı : Doç. Dr. Arif GÜLTEN (F.Ü.)
Diğer Jüri Üyeleri : Doç. Dr. Vedat ÇELİK (F.Ü.)
Doç. Dr. Burhan ERGEN (F.Ü.)
Dr. Öğr. Üyesi Ali ARSERİM (D.Ü.)
Dr. Öğr. Üyesi Mehmet ÜSTÜNDAĞ (B.Ü.)

EKİM-2018

ÖNSÖZ

Tez çalışmam ve diğer akademik çalışmalarım boyunca değerli katkı ve eleştirileri ile bana yardımcı olan kıymetli danışman hocam Doç. Dr. Arif GÜLTEN'e teşekkür ederim.

Hayatım boyunca benden hiçbir konuda desteklerini esirgemeyen kıymetli ve güzel insanlardan oluşan aileme şükranlarımı sunar ve teşekkür ederim.

Gerek akademik hayatımdaki gerek günlük hayatımdaki problemlerin üstesinden gelmemde her konuda bana sabırla yardımcı olan değerli eşim ve gönlü güzel meslektaşım Arş. Gör. Nurbanu ÇATALBAŞ'a teşekkür ve minnettarlığımı sunarım.

Akademik bilgi birikimleri ile bana yol gösteren kıymetli hocalarım Doç. Dr. Vedat ÇELİK ve Dr. Öğr. Üye. Ahmet ORHAN'a teşekkürlerimi sunarım.

Mehmet Cem ÇATALBAŞ
ELAZIĞ – 2018

İÇİNDEKİLER

Sayfa No

ÖNSÖZ	I
İÇİNDEKİLER.....	II
ÖZET	IV
SUMMARY	V
ŞEKİLLER LİSTESİ	VI
TABLolar LİSTESİ	IX
KISALTMALAR.....	X
GİRİŞ	1
1.1.Tezin Amacı	2
1.2. Tezin Organizasyonu.....	2
2.OPTİMİZASYON	4
2.1.Optimizasyon Kavramı	4
2.2. Optimizasyon ve Mühendislik.....	11
2.3. Sürekli ve Ayrık Zamanlı Optimizasyon.....	18
2.4. Optimizasyon Problemlerinin Türleri.....	19
2.5. Optimizasyon için Temel Araçlar.....	20
2.6. Kısıtlı ve Kısıtsız Optimizasyon.....	26
2.7. Yerel ve Genel Optimizasyon.....	28
2.8. Stokastik ve Deterministik Optimizasyon	29
3. NÜMERİK OPTİMİZASYON.....	31
3.1. Doğrusal Denklemlerin Nümerik Çözümlerine ilişkin yöntemler	32
3.1.1. Gauss Eleme Yöntemi	35
3.1.2. Gauss-Jordan Eleme Yöntemi	36
3.1.3. LU Ayrıştırma Yöntemi	37
3.1.4. Jacobi İterasyon Yöntemi	39
3.1.5. Gauss-Seidel İterasyon Yöntemi	40
3.1.6. Rahatlatma Yöntemi.....	41
3.2. Doğrusal Olmayan Denklemlerin Nümerik Çözümlerine İlişkin Yöntemler.....	42
3.2.1. Lin Yöntemi.....	43
3.2.2. Bairstow Yöntemi.....	44

3.2.3. İkiye Bölme Yöntemi	45
3.2.4. Sabit Nokta İterasyon Yöntemi	46
3.2.5. Newton-Rapson Yöntemi	47
3.2.6. Kiriş Yöntemi	47
4. RASTGELE ARAMA VE OPTİMİZASYON.....	49
4.1. Rassal Süreçler ve Dağılımlar	49
4.2. Rastgele Araştırma Yöntemleri	51
4.3. Rastgele Yürüyüş veya Arama Yöntemleri	52
4.3.1. Tekrarsız Rastgele Arama	52
4.3.2. Sıralı Rastgele Arama	52
4.3.3. İki Aşamalı Yöntemler	53
4.4. Rastgele Sayı Üreteçleri	54
5. METASEZGİSEL OPTİMİZASYON YÖNTEMLERİ.....	57
5.1. Genetik Algoritması	57
5.2. Karınca Kolonisi Algoritması.....	63
5.3. Yapay Arı Kolonisi.....	70
5.4. Parçacık Sürü Optimizasyonu	76
6. BALON BALIĞI VE OPTİMİZASYON	82
6.1. Balon Balığı ve Dairesel Yapılar	83
6.2. Balon Balığının Oluşturduğu Dairesel Yapıların İncelenmesi	92
7. BALON BALIĞININ DAİRESEL YAPILARI YÖNTEMİ	97
7.1. Balon Balığının Dairesel Yapıları Yöntemi ve Performans Analizi	98
8. BALON BALIĞININ DAİRESEL YAPILARI İLE KONTROLÖR TASARIMI	115
9. SONUÇLAR ve ÖNERİLER.....	123
KAYNAKLAR.....	125
ÖZGEÇMİŞ	132

ÖZET

Bu tez çalışması kapsamında tamamen yenilikçi bir metasezgisel optimizasyon algoritması önerilmiştir. Bu yeni yöntem, Torquigener albomaculosus isimli balon balığı türünün deniz tabanında oluşturduğu dairesel yapılardan esinlenerek oluşturulmuştur. Oluşturulan yenilikçi yöntemin temel parametreleri balon balığının dairesel yapıyı oluşturduğu süreçte gözlemlenen parametreler ile belirlenmiştir. Bu amaçla bu doğal fenomenin gerçekleştirilmesi sürecine detaylı bir şekilde yer verilmiştir. Ayrıca çalışmada optimizasyon kavramı, gelişim süreci ve etkinlik alanları hakkında detaylı bilgilere yer verilmiştir. Bu tezde metasezgisel optimizasyon kavramını oluşturan genel terimler hakkında bilgi verilmiş ve literatürde yaygın olarak kullanılan metasezgisel optimizasyon yöntemleri açıklanmıştır.

Tez çalışmasına konu olan balık davranışı, imgeler ve videolar üzerinden detaylı bir şekilde analiz edilmiş ve bu davranışın matematiksel modeli gerçekleştirilmiştir. Oluşturulan yöntemin başarımını artıracak çeşitli yaklaşımlara da değinilmiştir. Balon balığına ait dairesel yapıların detaylı bir şekilde irdelenmesi sonucu, doğada sıklıkla karşılaşılan altın orana yakınsayan değerler ilk olarak yine bu tez kapsamında belirlenmiştir. Geliştirilen yöntemin performans analizi literatürde sıklıkla kullanılan çok sayıda test fonksiyonu üzerinden gerçekleştirilmiştir. Ayrıca bir elektrik motorunun birim basamak cevabının iyileştirilmesi amacıyla kullanılan kontrolör yapısının parametreleri de bu önerilen yöntem ile ayarlanmıştır.

Metasezgisel optimizasyon algoritmalarının popülaritesi ve kullanım alanları gelişen hesaplama kapasitesine bağlı olarak her geçen gün artmaktadır. Bilim insanlarının doğada yaşayan canlıları etkili ve başarılı bir şekilde izleyebilmelerini sağlayan cihazların yeteneklerine bağlı olarak gelecekte bu eğilimin daha da artacağı tahmin edilmektedir.

Anahtar Kelimeler: Optimizasyon, Metasezgisel Optimizasyon, Nümerik Optimizasyon, Levy Uçuşu, Stokastik Optimizasyon, Balon Balığı, Balığına ait Dairesel Yapılar.

SUMMARY

A NEW METAHEURISTIC OPTIMIZATION ALGORITHM BASED ON PUFFER FISH BEHAVIOR

In this thesis study, a completely innovative meta-heuristic optimization algorithm is proposed. This new metaheuristic optimization method was inspired by the circular structures on the sea floor which are created by one of the puffer fish species which is called as *Torquigener albomaculosus*. The basic parameters of the inventive method were determined by the parameters observed during the process of the circular structure of the puffer fish. The process of performing this natural phenomenon has been detailed in this work. In addition, detailed information about the concept of optimization, development process and activity areas is given in the study. Information about the general terms that constitute the concept of metaheuristic optimization are given in this thesis and the metaheuristic optimization methods which are widely used in the literature are explained.

The behavior of the puffer fish which is subject to this thesis study was analyzed in detail through the images and videos and a mathematical model of this behavior was realized. Various approaches that will enhance the performance of the proposed method have also been discussed. As a result of the detailed analysis of the circular structures of the puffer fish, the values converging to the golden ratio which is frequently observed in nature were first determined within the scope of this thesis. Performance analysis of the developed method has been performed through a large number of test functions which are frequently used in the literature. In addition, the parameters of the controller structure used to improve the unit step response of an electric motor are also set by this proposed method.

The popularity and usage of metaheuristic based optimization algorithms are increasing day by day depending on the evolving computation capacity. It is anticipated that this popularity will grow even further in the future, depending on the ability of the devices that enable scientists to monitor the living creatures in nature effectively and successfully.

Keywords: Optimization, Metaheuristic Optimization, Numerical Optimization, Levy Flight, Stochastic Optimization, Puffer Fish, Circular Structures of Puffer Fish.

ŞEKİLLER LİSTESİ

Sayfa No

Şekil 2.1. 3 Boyutlu Öklid Uzayı.....	4
Şekil 2.2. (a)Fonksiyonun x ve y değerlerine göre değişimi (b)Fonksiyonun gradyenti.....	5
Şekil 2.3. Gezgin satıcı problemi ve ilgili eşitlikl.....	6
Şekil 2.4. Bellman Denklemi.....	7
Şekil 2.5.Optimizasyon problemlerinin türleri ve arasındaki ilişkiler.....	8
Şekil 2.6. Farklı normlara göre x ve y parametrelerin değişimi.....	9
Şekil 2.7. Ortogonalite Kavramı.....	10
Şekil 2.8. Kanat tasarım problemi.....	11
Şekil 2.9. Rosenbrock fonksiyonunun dağılımı.....	13
Şekil 2.10.Konveks ve Konkav fonksiyonlar.....	14
Şekil 2.11. Konveks ve Konkav şekillerden bazıları.....	15
Şekil 2.12. İp veya zincir probleminin (catenary) temsili gösterimi.....	15
Şekil 2.13. Devrenin bağlantı şeması.....	17
Şekil 2.14. Ayırık ve sürekli zaman optimizasyon problem uzaylarının gösterimi.....	19
Şekil 2.15. $f(x,y) = x^2 + y^2$ fonksiyona ait gradyentin değişiminin gösterimi.....	21
Şekil 2.16. Orijinal fonksiyon ile Taylor serisi açılımının karşılaştırılması.....	24
Şekil 2.17. İki zarın toplamalarının değerlerinin gerçekleşme olasılığına oranı.....	30
Şekil 3.1. Nümerik optimizasyon türlerinin sınıflandırılması.....	31
Şekil 3.2. Doğruluk ve hassasiyet kavramlarının gösterimi.....	32
Şekil 4.1. Levy dağılımı.....	51
Şekil 4.2. (a) Gerçek rastgele sayı üretici (b) Sözde rastgele sayı üretici.....	56
Şekil 5.1. Genetik Algoritmasına ait temel süreçlerin gösterimi.....	59
Şekil 5.2. GA’nda kodlama ve kod çözme.....	60
Şekil 5.3. Egg holder fonksiyonun üç boyutlu değişimi.....	61
Şekil 5.4. Egg holder fonksiyonuna ait parametrelerin iterasyon sayısına bağlı değişimi.....	62
Şekil 5.5. GA’sı sonucu bulunan optimal değerlerin fonksiyon üzerinde gösterimi.....	63
Şekil 5.6. Feromon yapısı ve karıncaların besin bulması süreci.....	65
Şekil 5.7. Michalewicz fonksiyonun üç boyutlu değişimi.....	69
Şekil 5.8. Michalewicz fonksiyona ait parametrelerin iterasyon sayısına bağlı değişimi.....	69

Şekil 5.9. KKA'sının iterasyon sayısına bağlı değişimi.....	70
Şekil 5.10. Arı kolonisi gösterimi.....	71
Şekil 5.11. Bal arılarının nektar toplama süreci.....	73
Şekil 5.12. YAK algoritmasına ait akış şeması.....	74
Şekil 5.13. Levy 13 Fonksiyonunun üç boyutlu değişimi.....	75
Şekil 5.14. Levy 13 Fonksiyonuna ait parametrelerin iterasyon sayısına bağlı değişimi.....	75
Şekil 5.15. PSO algoritması ile arama noktasını güncelleme süreci.....	77
Şekil 5.16. Styblinski-Tang fonksiyonun üç boyutlu değişimi.....	79
Şekil 5.17. Styblinski-Tang fonksiyona ait parametrelerin iterasyon sayısına bağlı değişimi.....	80
Şekil 5.18. PSO algoritmasının iterasyon sayısına bağlı değişimi.....	80
Şekil 6.1. Torquigener albomaculosus.....	82
Şekil 6.2. a)Holotipin lateral görünümü, erkek, 87.8 mm b)Holotipin sırta ait görünümü c)Holotipin paratip görünümü, dişi, 90.5mm.....	83
Şekil 6.3. Gözlem yapılan konumların detaylı gösterimi.....	84
Şekil 6.4. Erkek balon balığı tarafından deniz tabanında oluşturulmuş dairesel yapı.....	85
Şekil 6.5. Oluşturulan dairesel yapının detaylı gösterimi.....	86
Şekil 6.6. Farklı balıkların oluşturduğu dairesel yapılara ait sayısal özellikler.....	86
Şekil 6.7. Yarıçap oranlarının karşılaştırılması.....	87
Şekil 6.8. Balon balığı tarafından yapılan dairesel yapının temsili gösterimi.....	88
Şekil 6.9. Dairesel yapı üzerinden su akış dinamiğinin incelenmesi.....	89
Şekil 6.10. Su akış yönünün temsili gösterimi.....	90
Şekil 6.11. a)Erken Aşama b) Orta Aşama c)Son Aşama.....	91
Şekil 6.12. a)Yumurtlamadan bir gün önce toplanan(Beyaz bar 5 mm)b)Yumurtlama sonrası.....	92
Şekil 6.13. Erkek balon balığı tarafından gerçekleştirilen dairesel yapı(kırmızı bar=20 cm).....	93
Şekil 6.14.Farklı bölgelere ait yarıçap değerlerinin gösterilmesi.....	94
Şekil 6.15. Balon Balığına ait dairesel yapıya ilişkin çizim.....	96
Şekil 7.1. Ackley Fonksiyonu.....	98
Şekil 7.2. Maliyet fonksiyonunun (Ackley) iterasyon sayısına bağlı değişimi.....	99
Şekil 7.3. Ackley Fonksiyonu için Dairesel Yapıların Değişimi.....	100
Şekil 7.4. Beale Fonksiyonu.....	101
Şekil 7.5. Maliyet fonksiyonunun (Beale) iterasyon sayısına bağlı değişimi.....	101
Şekil 7.6. Beale Fonksiyonu için Dairesel Yapıların Değişimi.....	102
Şekil 7.7. Booth Fonksiyonu.....	103

Şekil 7.8. Maliyet fonksiyonunun (Booth) iterasyon sayısına bağlı değişimi.....	103
Şekil 7.9. Booth Fonksiyonu için Dairesel Yapıların Değişimi.....	104
Şekil 7.10. Chichinadze Fonksiyonu.....	105
Şekil 7.11. Maliyet fonksiyonunun (Chichinadze) iterasyon sayısına bağlı değişimi.....	105
Şekil 7.12. Chichinadze Fonksiyonu için Dairesel Yapıların Değişimi.....	106
Şekil 7.13. Crownedcross Fonksiyonu.....	107
Şekil 7.14. Maliyet fonksiyonunun (Crownedcross) iterasyon sayısına bağlı değişimi.....	107
Şekil 7.15. Crownedcross Fonksiyonu için Dairesel Yapıların Değişimi.....	108
Şekil 7.16. Easom Fonksiyonu.....	109
Şekil 7.17. Maliyet fonksiyonunun (Easom) iterasyon sayısına bağlı değişimi.....	109
Şekil 7.18. Easom Fonksiyonu için Dairesel Yapıların Değişimi.....	110
Şekil 7.19. Giunta Fonksiyonu.....	111
Şekil 7.20. Maliyet fonksiyonunun (Giunta) iterasyon sayısına bağlı değişimi.....	111
Şekil 7.21. Giunta Fonksiyonu için Dairesel Yapıların Değişimi.....	112
Şekil 7.22. Goldstein-Price Fonksiyonu.....	113
Şekil 7.23. Maliyet fonksiyonunun (Giunta) iterasyon sayısına bağlı.....	113
Şekil 7.24. Giunta Fonksiyonu için Dairesel Yapıların Değişimi.....	114
Şekil 8.1. Doğru akım motorunun temsili gösterimi.....	115
Şekil 8.2. Doğru akım motora ait transfer fonksiyonu.....	117
Şekil 8.3. Kapalı çevrim hız kontrolüne ilişkin blok diyagramı.....	117
Şekil 8.4. Birim basamak cevabı ve analiz parametreleri.....	118
Şekil 8.5. PI kontrolör maliyet uzayı.....	119
Şekil 8.6. Maliyet fonksiyonunun (PI) iterasyon sayısına bağlı değişimi.....	119
Şekil 8.7. PI kontrolör maliyet fonksiyonu için daireseel yapıların değişimi.....	120
Şekil 8.8. PI kontrolör tasarlanmış sisteme ait birim basamak cevabı.....	121

TABLÖLAR LİSTESİ

Sayfa No

Tablo 2.1. Optimizasyon Problemlerinin türleri.....	20
Tablo 2.2. Kısıtlı Optimizasyon Yöntemleri.....	28
Tablo 4.1. Sözde ve Gerçek Rastgele Sayı Üreteçlerinin Karşılaştırılması.....	56
Tablo 4.2. Farklı rastgele sayı üreteçlerinin kullanım alanlarının karşılaştırılması.....	56
Tablo 5.1. Gerçek ve yapay karıncalar arasındaki benzerlikler.....	66
Tablo 6.1. Balon Balığının Dairesel Yapıları.....	95
Tablo 7.1. Performans Analizi Yapılan Yöntemlere İlişkin Parametreler.....	97
Tablo 7.2. Ackley Fonksiyonuna ait performans sonuçları.....	99
Tablo 7.3. Beale Fonksiyonuna ait performans sonuçları.....	102
Tablo 7.4. Booth Fonksiyonuna ait performans sonuçları.....	104
Tablo 7.5. Chichinadze Fonksiyonuna ait performans sonuçları.....	106
Tablo 7.6. Crownedcross Fonksiyonuna ait performans sonuçları.....	108
Tablo 7.7. Easom Fonksiyonuna ait performans sonuçları.....	110
Tablo 7.8. Giunta Fonksiyonuna ait performans sonuçları.....	112
Tablo 7.9. Goldstein-Price Fonksiyonuna ait performans sonuçları.....	114
Tablo 8.1. Doğru akım motoruna ait parametreler.....	116
Tablo 8.2. PI kontrolör optimizasyon sonuçları.....	120
Tablo 8.3. PI kontrolörlü ve kontrolörsüz sisteme ait birim basamak parametreleri.....	122

KISALTMALAR

KKA : Karınca Kolonisi Algoritması

GA : Genetik Algoritma

YAK : Yapay Arı Kolonisi

PSO : Parçacık Sürü Optimizasyonu

BBDY : Balon Balığına ait Dairesel Yapılar



1. GİRİŞ

Metasezgisel optimizasyon yöntemlerinin kullanım alanı bilgisayarların işlem kabiliyetlerinin gelişmesine bağlı olarak her geçen gün artmaktadır. Literatürde siyah kutu (black-box) olarak adlandırılan ve kullanıcının sistemin davranışı hakkında bilgi sahibi olmadığı veya çok az miktarda ön bilgi sahibi olduğu sistemlerin analizi sürecinde sıklıkla metasezgisel yöntemler kullanılmaktadır [1]. Optimizasyon kavramı olarak en uygun hale getirmek anlamı taşımaktadır. Bu noktada en uygunluk kavramının bir miktar daha irdelenmesi gerekmektedir. Farklı türdeki değişken problemleri en uygun şekilde getirmek eylemi son derece ucu açık ve oldukça geniş kapsamlı bir sorunsaldır [2]. Bilim adamları bu tarz bir sorunsalın çözümünün geniş çaplı bir tecrübe ile sağlanacağı konusunda hemfikirdir. Fakat bu oranda ki geniş çaplı bir tecrübe sadece nümerik hesaplamalar ile elde edilmemektedir [3]. Bu kısıt yüzünden bilim adamları belirtilen çözümün cevabını tecrübesi veya birikimi daha yüksek olan bir olgu üzerinden irdemişlerdir. Bu olgu doğa, yaşam veya canlılardır.

Uzun yıllar doğayı gözlemleyen bilim adamları doğadaki canlıların belirli problemleri çözme süreçlerindeki davranışlarının bin yıllarca süre gelen ve değişkenlik gösteren şartlardan dolayı en uygun hale geldiğini gözlemlemişlerdir [4]. Bu davranışlar ışığında metasezgisel veya doğadan esinlenmeli yöntemler ortaya çıkmıştır. Her geçen gün artan nümerik hesaplama veya detaylı canlı davranışlarının analize imkan veren sistemlere bağlı olarak bu tarz yöntemlerin kullanım alanı ve başarımı her geçen gün artmıştır.

Doğadaki bireylerin, sürülerin veya mikroorganizmaların besin arama, hayatta kalma süreçlerinin irdelenmesi sonucu elde edilen bu yöntemler ile uyarlamalı bir şekilde literatürdeki basit veya karmaşık problemlerin çözümüne imkan sağlanmıştır [5,6]. Bu tarz yenilikçi yöntemlerin mühendislik alanındaki kullanım alanı da oldukça geniştir.

Bu tez çalışması kapsamında ise tamamen yenilikçi, doğadan esinlenmiş yeni bir metasezgisel yöntem geliştirilmiştir. Balon balığı olarak adlandırılan ve karşı cinsi etkilemek amacıyla deniz tabanında eşsiz yapılar oluşturan bu canlının davranışının matematiksel modeli oluşturulmuş ve irdelenmiştir. Bu yapının oluşturulması sürecindeki çeşitli kısıtlar incelenmiş, oluşturulan tamamen yenilikçi bu yöntemin başarımı ise literatürde sıklıkla kullanılan test fonksiyonları üzerinden diğer yöntemler ile detaylı bir şekilde karşılaştırılmış, bu yöntemlere göre artıları veya eksikleri açıklanmıştır. Bunlara ek olarak yöntemin başarımı uyarlamalı kontrolör tasarımının gerçekleştirilmesini temel alan bir uygulama üzerinden test edilmiştir.

1.1. Tezin Amacı

Metasezgisel optimizasyon yöntemlerinin kullanım alanları gelişen işlemsel kabiliyetlere bağlı olarak günden güne artmaktadır. Bu tez çalışmasında ise doğadan esinlenmiş tamamen yeni bir metasezgisel optimizasyon algoritmasının gerçekleştirilmesi ve detaylı performans analizleri üzerinden geliştirilen yöntemin başarımı irdelenmiştir. Bu tez çalışmasının temelini balon balığı olarak adlandırılan ve tür adı *Torquigener albomaculosus* olan erkek balon balığının karşı cinsi etkilemesi sürecinde deniz tabanında oluşturduğu dairesel yapının matematiksel modellenmesi ve bu modelin bir metasezgisel optimizasyon yöntemi olarak literatürde yer bulmasının sağlanmasıdır.

Geliştirilecek metasezgisel yöntemin diğer yöntemlerden farklı hesaplama karmaşıklığının az olması birinci öncelik olarak düşünülmüştür. Ayrıca ilgili yöntemle ait parametrelerin mümkün olduğu kadar balığın oluşturduğu dairesel yapıdaki parametreler irdelenerek elde edilmesi amaçlanmıştır. Bu şekilde yöntemle ait dairesel yapılar ve oluşum süreci detaylı bir şekilde incelenmiştir. Bu oluşum süreci yöntemin gerçekleştirilmesi sürecine uyarlanmıştır. İlgili balık türünün deniz tabanında oluşturduğu dairesel yapının, merkez noktasındaki su akış oranının 24.2% oranında azaltılıyor olması bu tez çalışmasının gerçekleştirilmesi için önemli motivasyonlardan bir tanesidir. Ayrıca literatürdeki metasezgisel yöntemler detaylı bir şekilde incelenmiştir ve bu eksiklerin giderilmesini veya azaltılmasını sağlayan bir metasezgisel optimizasyon yöntemi ortaya koyulmuştur.

Genellikle iterasyon tabanlı olan metasezgisel yöntemlerinin en önemli performans kriterlerinden biri hesaplama süresidir. Hesaplama süresi yüksek olan yöntemlerin problem çözme süreçleri ne kadar yüksek olursa olsun gerçek zamanlı uygulamalardaki zaman kısıtlarından dolayı literatürde yer bulması oldukça zordur. Bu yüzden hesaplama süresi ile başarımlı performans arasında bir denge sağlayabilecek ve algoritma karmaşıklığı daha az olan yeni bir metasezgisel optimizasyon yönteminin gerçekleştirilmesi amaçlanmıştır.

1.2. Tezin Organizasyonu

Bu tez çalışması dokuz bölümden oluşmaktadır. Tezin birinci bölümde; metasezgisel optimizasyon yöntemlerinin gelişim süreci hakkında temel hakkında bilgiler verilmiştir.

İkinci bölümde; optimizasyon kavramı ve alt başlıklar hakkında detaylı bilgiler verilmiştir.

Üçüncü bölümde; nümerik optimizasyon kavramı ve bu kavrama ait çeşitli yöntemlerin gerçekleştirilmesi ve uygulamaları hakkında bilgiler verilmiştir.

Dördüncü bölümde; rastgele arama ve optimizasyon arasındaki ilişki hakkında bilgiler verilmiştir. Ayrıca metasezgisel optimizasyon yöntemleri için gerekli olan rastgele süreçlerin oluşturulmasını sağlayan rassal sayı üreteçleri ve türleri hakkında bilgilerde bu bölümde verilmiştir.

Beşinci bölümde; literatürdeki çeşitli metasezgisel yöntemler hakkında detaylı bilgiler verilmiştir ve bu yöntemlerin çalışmasının gözlemlenmesi için bir adet test fonksiyonu üzerinden davranış karakteristiği hakkında bilgiler verilmiştir.

Altıncı bölümde; balon balığı ve bu balığın deniz tabanında oluşturduğu dairesel yapılar hakkında detaylı bilgiler verilmiştir.

Yedinci bölümde; balon balığına ait dairesel yapılar yöntemine ait performans analiz sonuçları literatürdeki metasezgisel yöntemler ile karşılaştırılmıştır.

Sekizinci bölümde; balon balığına ait dairesel yapılar yöntemi ile bir doğru akım motoruna *PI* kontrolör yapısı tasarlanmıştır ve tasarım sonucuna çıktılar detaylı bir şekilde verilmiştir.

Dokuzuncu bölümde; tez çalışmasına ait sonuçlar hakkında bilgiler verilmiştir.

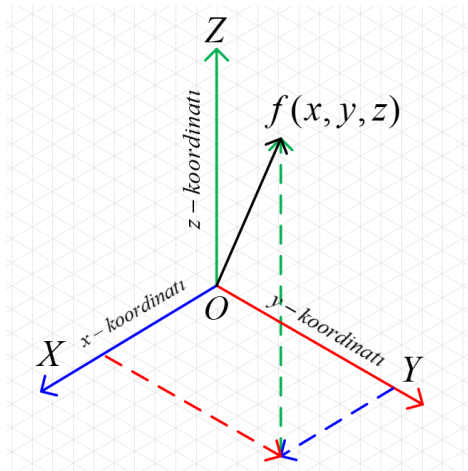
2. OPTİMİZASYON

Tez çalışmasının bu bölümünde, optimizasyon kavramı tarihsel gelişim süreci ışığında detaylı bir şekilde irdelenecektir. Optimizasyon kavramının ortaya çıkışı ve uygulama alanları hakkında bilgiler verilecektir. Optimizasyon konusu ile alakalı matematiksel terimler detaylı bir şekilde açıklanacak ve bu terimlerin gerçek hayattaki yansımaları hakkında bilgiler verilecektir.

2.1. Optimizasyon Kavramı

Doğa, bir takım en iyi veya en uygun hale getirme süreçleri üzerinden döngüsünü tamamlamaktadır. Optimizasyon kavramı bu cümle üzerine şekillenmiştir ve bu kavram yaşamın temelini oluşturmaktadır. Optimizasyon hayatın her alanında geçerli olan fakat yazılı olmayan bir kural olarak karşımıza çıkar. Daha açık bir ifade ile mühendislik ile ilintili tasarımlarda veya insanların günlük yaşamlarında optimizasyon kavramı, adı konulmamış bir kanun olarak hayatımızda yer almaktadır [7]. Optimizasyon kavramı genel hatları ile ilk olarak ünlü matematikçi Öklid ile antik Yunan'da kendine yer bulmuştur. Daha sonraki süreçte yine bir diğer antik yunan bilimcisi olan Heron, ışığın iki nokta arasındaki en kısa yol üzerinden gittiğine ait çalışma ile optimizasyon kavramı ile ilgili antik Yunanda kendine yer bulmuştur [8,9].

Öklid geometrisinin temel oluşturduğu Öklid uzayı hala günümüzde birçok optimizasyon probleminin çözümünde önemli rol oynamaktadır [10]. Şekil 2.1'de 3 boyutlu bir Öklid uzayının gösterimi bulunmaktadır.

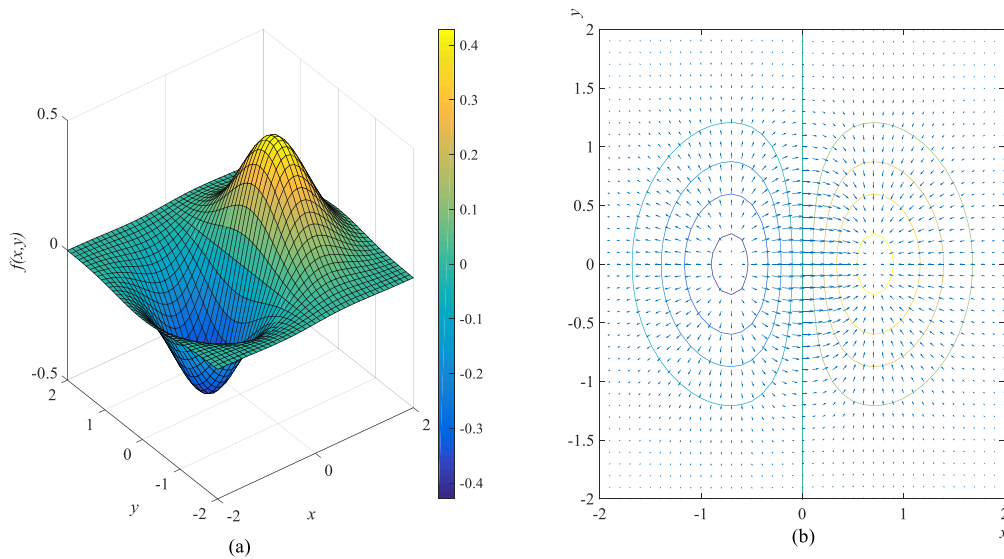


Şekil 2.1. 3 Boyutlu Öklid Uzayı

Optimizasyon kavramının tarihsel sürecine devam edecek olursak, 1615 yılında Kepler sıvı alan şişelerin optimal boyutları hakkında çalışmalarda bulunacaktır [11]. 1636 yılında ise P.de Fermat, en uç noktada bir fonksiyonun türevinin kaybolduğunu gözlemlemiştir [12].

1638 yılında Galileo asılı bir zincirin boyutunun belirlenmesine ilişkin çalışmalarda bulunmuştur. 1657 yılında ise P. De Fermat ışığın minimum sürede iki nokta arasında dolaştığını göstermiştir. Tüm bu gözlemler Öklid uzayı ışığında şekillenmiş ve optimizasyon kavramı yıllar boyunca insanlar tarafından anlamlandırılmaya çalışılmıştır. Devam eden süreçte, 1660 yıllarında Isaac Newton ve 1670 yıllarında Gottfried Leibniz varyasyon hesabının temelini oluşturan matematiksel analizlerde bulunmuşlardır [13]. Ayrıca bu çalışmalar kapsamında bazı ayırık sonlu optimizasyon problemleri de irdelenmiştir. İlk optimizasyon algoritmaları 19. yüzyılda Karl Weierstrass, Jakob Steiner, William Rowan Hamilton ve Carl Jacobi tarafından geliştirilmiştir [14,15]. Aynı zamanda bu bilim adamlarının çalışmaları varyasyon hesabı alanında yapılan çalışmaları daha ileriye taşımıştır.

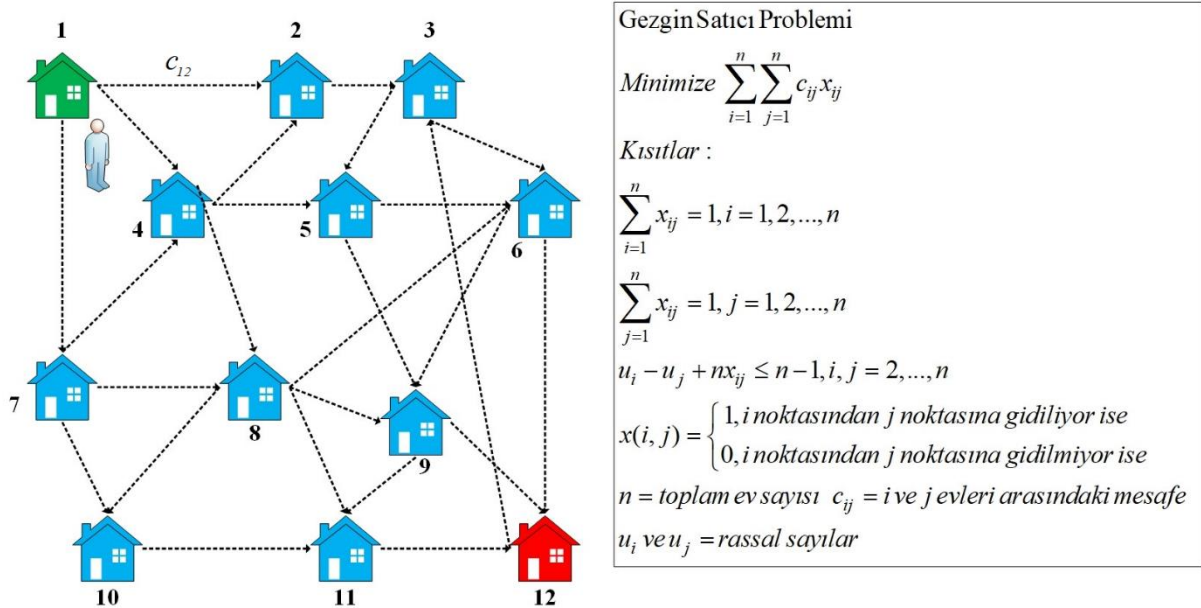
1806 yılında Adrien-Marie Legendre en küçük kareler yöntemini sunmuş ve bu alandaki çalışmalara oldukça büyük oranda ivme kazandırmıştır [16]. 1826 yılında Joseph Fourier, mekanik ve olasılık teorisinde ortaya çıkan sorunları çözmek için doğrusal programlama problemini formülize etmiştir [17]. 1847 yılında Augustin-Louis Cauchy modern optimizasyon yöntemleri açısından oldukça fazla önem arz eden gradyent(yönelim) yöntemini bulmuştur [18]. Şekil 2.2’de $f(x,y) = xe^{-x^2-y^2}$ denkleminin ilişkin 3 boyutlu gradyent çıktısının eş yükselti eğri çizgileri ve vektörler ile gösterimi bulunmaktadır.



Şekil 2.2. (a) Fonksiyonun x ve y değerlerine göre değişimi (b) Fonksiyonun gradyenti

1870 yıllarında gerçekleşen ekonomi alanındaki marjinal devrim ile birlikte kaynakların en verimli bir şekilde kullanılması süreci daha detaylı bir şekilde irdelemiştir. Bu sayede optimizasyon kavramı, ekonomi teorisinin ayrılmaz parçası haline gelmiştir. Johan Ludwig Jensen, optimizasyon için önem arz eden konveksite kavramını 1905 yılında oluşturmuştur [19]. 1917 yılında ise Harris Hancock tarafından optimizasyon alanındaki ilk kitap olma özelliği taşıyan “*Maksimum ve Minimum Teorisi*” kitabı basılmıştır [20].

Günümüze yaklaştıkça, Karl Menger 1932 yılında optimizasyon problemlerinin temelini oluşturan gezgin satıcı problemini genelleştirerek formülize etmiştir [21]. Bu temel optimizasyon probleminde, satıcının ilgili tüm şehirlere yalnızca bir kez uğramalı" ve bu gezi sonucunda toplamda en kısa mesafeyi katederek başlangıç noktasına geri dönmelidir. Çeşitli kısıtlar içeren bu optimizasyon problemi, graf teorisinin temelini oluşturmaktadır [22]. Kapalı yollardan oluşan gezgin satıcı problemi, Hamilton döngüsü ile oldukça fazla benzerlik göstermektedir. Şekil 2.3'te gezgin satıcı problemine ilişkin görsel ve formülasyon verilmektedir.

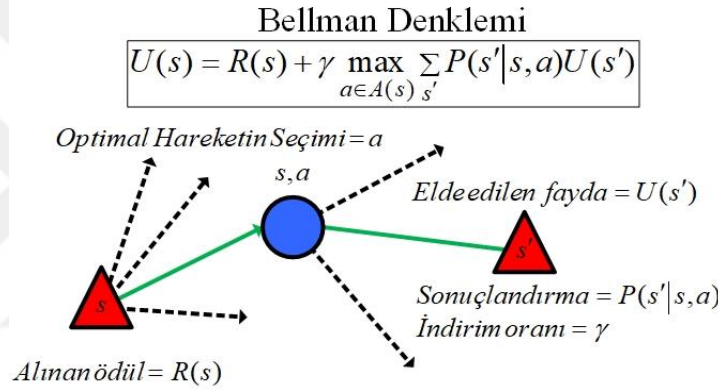


Şekil 2.3. Gezgin satıcı problemi ve ilgili eşitlikler

Optimizasyon kavramı ikinci dünya savaşından sonra elektronik alanındaki gelişmelere bağlı olarak sıçrama yapmıştır. Savaş sırasında ortaya çıkan hammadde ve insan kaynaklarındaki sıkıntılar, bilim adamlarının kaynak ve enerji optimizasyonu hakkında yeni yaklaşımlarda bulunmasına neden olmuştur. Bu bağlamda ortaya çıkan yöneylem araştırma (operational research) kavramı, belirli hedeflere ulaşmak için elde bulunan kaynakların en

uygun ve yararlı şekilde kullanılması için farklı disiplinlerin işbirliğine imkan veren bir bilimsel yaklaşımlar bütünü olarak tanımlanmaktadır. Bu alanda önemli çalışmaları bulunan John von Neumann aynı zamanda bilgisayar biliminin de öncülerinden biridir.

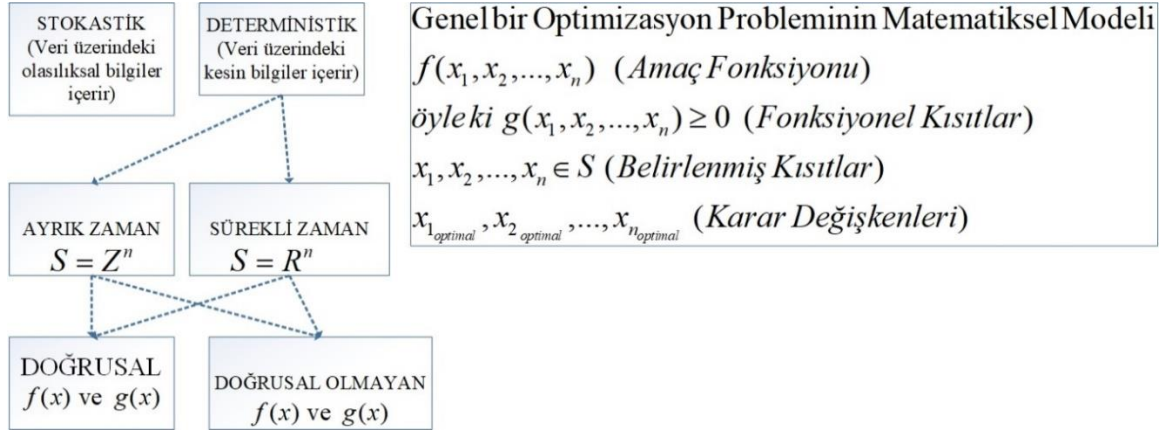
İkinci Dünya Savaşı sonrası bir trend haline gelen uzay yarışları sonucunda varyasyon hesaplarından farklı bir disiplin olarak optimal kontrol teorisi kavramı gelişmiştir ve bu kavram optimizasyon alanındaki çalışmalara önemli bir miktarda ivme kazandırmıştır [23,24]. 1957 yılında Richard Ernest Bellman optimallik ilkesi (principle of optimality) kavramını sunmuştur [25,26]. Bu bağlamda ortaya çıkan ve optimizasyon tarihi açısından oldukça önem arz eden Bellman denklemlerine ait görsel Şekil 2.4'te verilmektedir. Görseldeki denklem aynı zamanda dinamik programlama olarak da literatürde yer bulmaktadır [27].



Şekil 2.4. Bellman Denklemini

Leonid Khachiyan 1979 yılında doğrusal programlama için ilk polinom zaman algoritmasını oluşturmak amacıyla elipsoid yöntemini uygulanmıştır [28]. 1980 yıllarında bilgisayarın işlem kapasitesinin artmasına bağlı olarak global optimizasyon ve büyük ölçekli problemler için metasezgisel yöntemler önem kazanmış ve bu popülerite günümüze kadar hızlı bir şekilde artarak gelmiştir.

Optimizasyon kavramının derinlemesine irdelenmesini için ilk aşamada optimizasyon problemlerine ait türlerin doğru bir biçimde sınıflandırılması gerekmektedir. Optimizasyon kavramına ait temel başlıklar ve bunlar arasındaki ilişkiler Şekil 2.5'te gösterilmiştir.



Şekil 2.5. Optimizasyon problemlerinin türleri ve arasındaki ilişkiler

Optimizasyonun daha detaylı bir şekilde anlamlandırılması için çeşitli nümerik yaklaşımların ön bilgi olarak verilmesi gerekmektedir. Bunlardan ilki optimizasyon için önem arz eden vektör kavramıdır [29]. Vektör kavramı, Denklem 2.1’de verilen eşitlikler ile açıklamıştır. Gerçek n-boyutlu bir vektör, n sayıda $\{x_1, x_2, \dots, x_n\}$ gerçek sayıdan oluşur.

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \text{ veya } x = [x_1, x_2, \dots, x_n] \text{ veya } x = [x_i]_{i=1}^n \quad (2.1)$$

Bir diğer önemli ifade ise vektör normlarıdır [30]. Girdi verisinin boyutunu temsil etmektedir. x değişkeninin normu $\|x\|$ ile gösterilir. Ayrıca iki değişken olan x ve y arasındaki mesafe, Eşitlik 2.2’de gösterildiği gibi norm vasıtasıyla gösterilebilir. Bir diğer anlamda, x değişkeninin normu, $y = 0$ olarak alınan orijine uzaklığı temsil etmektedir.

$$\|x - y\| \quad (2.2)$$

x vektörünün normu, $x \in V$ ve V gerçek ve negatif olmayan boyut, uzunluk ve genlik değerlerini sezgisel olarak temsil etmektedir. $x \in V$ normu Eşitlik 2.3 de gösterilen üç şartı sağlamalıdır.

Pozitiflik : $\|x\| \geq 0$, $\|x\| = 0$ eğer $x = 0$

Homojenlik : $\|ax\| = |a|\|x\|$, $a \in R$

(2.3)

Üçgen eşitsizliği : $\|x + y\| \leq \|x\| + \|y\|$

Genelleştirecek olursak, bir vektöre ait norm ifadesinin genel tanımı Eşitlik 2.4'te gösterilmiştir. İlgili eşitlikte, n vektörün boyutunu temsil etmektedir.

$$\|x\|_p = \left(\sum_{i=1}^n |x_i|^p \right)^{1/p} \quad p \in R \text{ ve } p \geq 1 \quad (2.4)$$

Literatürde genellikle kullanılan normlar Eşitlik 2.5'te gösterilmiştir.

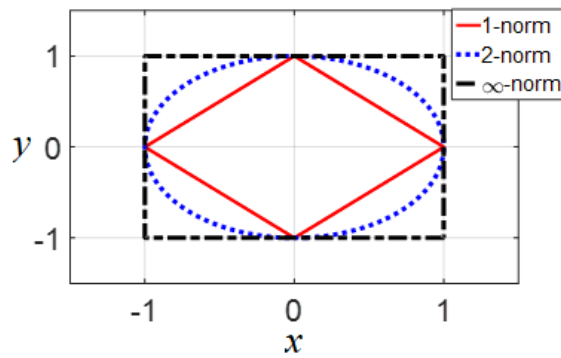
• 1-norm : $\|x\|_1 = \sum_{i=1}^n |x_i|$

• 2-norm : $\|x\|_2 = \sqrt{\sum_{i=1}^n |x_i|^2}$

(2.5)

• ∞ -norm : $\|x\|_\infty = \max_{1 \leq i \leq n} |x_i|$

Şekil 2.6'da farklı norm değerlerine göre x ve y değerlerinin değişimi gösterilmiştir. Norm seviyesinin artmasına bağlı olarak dağılımlardaki değişim açıkça gözlemlenmektedir.



Şekil 2.6. Farklı normlara göre x ve y parametrelerinin değişimi

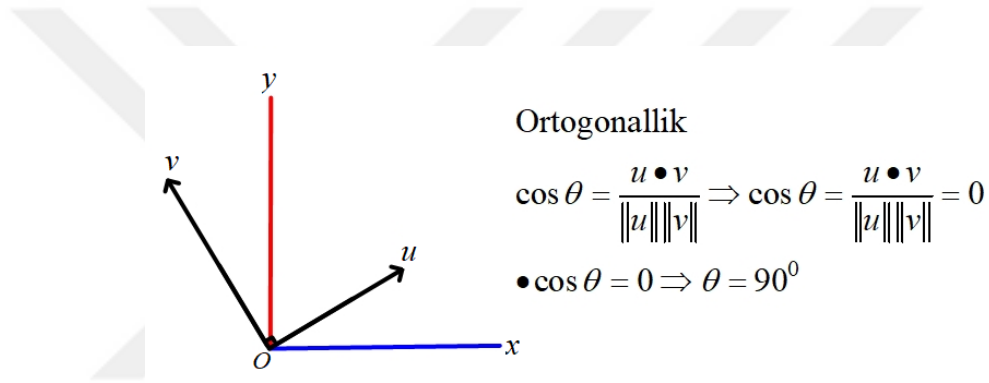
Optimizasyon problemleri için diğer önemli kavramlar ise sırasıyla, iç çarpım, ortogonalite ve ortonormalite kavramlarıdır. x ve y vektörlerinin iç çarpımı, Eşitlik 2.6'da gösterilmiştir.

$$x^T y = \sum_{i=1}^n x_i y_i \quad (2.6)$$

Ortogonalite ise iki vektörün çarpımının sıfır olması durumudur. . Bu kavram optimizasyon problemlerinde sıklıkla kullanılmaktadır. $\{v^{(1)}, v^{(2)}, \dots, v^{(k)}\}$ vektör kümesi, Eşitlik 2.7’de gösterilen şartı sağlıyorsa ortogondur [31].

$$v^{(i)T} v^{(j)} = 0, i \neq j, i, j = 1, \dots, k \quad (2.7)$$

Şekil 2.7’de ortogonalite kavramının örnek görseli bulunmaktadır.



Şekil 2.7. Ortogonalite Kavramı

Ortonormalite kavramından ise vektör kümesinin birim vektör olması ve Eşitlik 2.8’in gerçekleştiği durumlarda bahsedilebilmektedir.

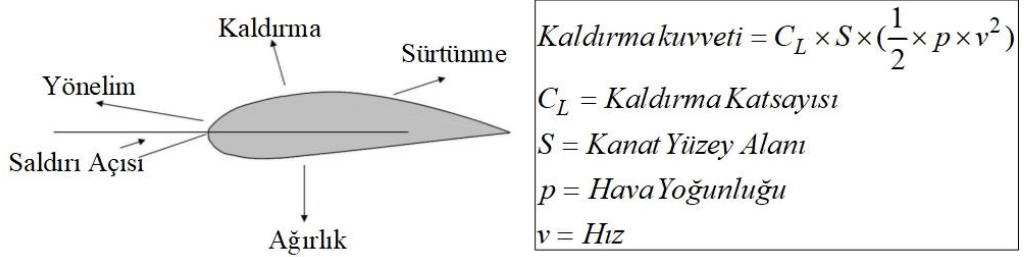
$$\left\{ \|v^{(1)}\| = \|v^{(2)}\| = \dots = \|v^{(k)}\| = 1 \right\} \quad v^{(i)T} v^{(j)} = \begin{cases} 0, & \text{eğer } i \neq j \\ 1, & \text{eğer } i = j \end{cases} \quad (2.8)$$

2.2. Optimizasyon ve Mühendislik

Optimizasyon kavramının geniş kullanım alanları arasında mühendislik problemleri de bulunmaktadır. Bu alanlardan bazıları aşağıda verilmiştir [32-35].

- Tasarım Alanında (Otomotiv, Havacılık, Biyomekanik, Devre)
- Kontrol Alanında (Optimal Kontrol Tasarımı)
- Sinyal İşleme Alanında (Ses Konumlandırma)
- Haberleşme (Anten Tasarımı)
- Kablosuz Sensör Ağları (Güç Tüketimi)

Bu alanlardan havacılık alanındaki uygulamaya ait gösterim Şekil 2.8’de gösterilmiştir.



Şekil 2.8. Kanat tasarım problemi

Şekilde 2.8’de görüldüğü gibi bir hava taşıtının havalanması için çeşitli şartların yerine gelmesi gerekmektedir. Bu bağlamda, aracın havalanması için hayati önem arz eden kanat yapısı belirli kısıtlar ışığında gerçekleştirilmesi gereken bir tasarım problemi olarak uzmanların karşısına çıkmaktadır. Ayrıca ilgili kanat tasarımı gerçekleştirilirken her türlü değişken hava şartlarına göre tasarlanmalı ve hava taşıtının ağırlığına en az oranda etki etmelidir [36]. Böylece optimizasyon problemleri genel olarak aşağıda belirtilen matematiksel formda tanımlanabilmektedir.

$$\begin{aligned} \min_{x \in R^n} \quad & f_i(x), \quad (i=1, 2, \dots, M) \\ \text{öyle ki} \quad & \phi_j(x) = 0, \quad (j=1, 2, \dots, J) \\ & \psi_k(x) = 0, \quad (k=1, 2, \dots, K) \end{aligned} \tag{2.9}$$

$f_i(x)$, $\phi_j(x)$ ve $\psi_k(x)$ tasarım vektörünün fonksiyonlarını temsil eder.

$$x = (x_1, x_2, \dots, x_n)^T \quad (2.10)$$

Eşitlik 2.9’da x_i parametresi karar veya tasarım değişkeni olarak tanımlanır ve bu parametre sürekli zamanlı, ayrık zamanlı veya bu ikisinin karışımı şeklinde olabilir. $f_i(x)$ ise amaç fonksiyonları (objective functions) olarak tanımlanır ve optimizasyon probleminin çekirdeğini temsil eder. Amaç fonksiyonu yerine literatürde bazen *maliyet fonksiyonu* (cost function) veya *enerji fonksiyonu* (energy function) olarak da kullanılmaktadır. Karar değişkenlerinden oluşan uzay ise arama uzayı olarak (R^n) tanımlanır [37].

Amaç fonksiyonu tarafından oluşturulan uzay ise çözüm uzayı (solution space) olarak adlandırılır. Amaç fonksiyonu, doğrusal veya doğrusal olmayan şekilde olabilir. $\phi_j(x)$ için eşitlikler ve $\psi_k(x)$ için eşitsizlikler ise kısıtlar olarak tanımlanır.

Amaç fonksiyonu bir maksimizasyon problemi olarak da tanımlanabilir. Bunun nedeni amaç fonksiyonun $f(x)$ maksimizasyonu ile $-f(x)$ amaç fonksiyonun minimizasyonu birbirine eşittir. Kısıtlar için ise en temel durum açısından karar değişkeninin Eşitlik 2.11’de ki gösterimi sınırlar(bounds) olarak tanımlanmaktadır.

$$x_i, \underbrace{x_{i,\text{minimum}} \leq x_i \leq x_{i,\text{maksimum}}}_{\text{Sınırlar}} \quad (2.11)$$

Optimizasyon problemlerinin temelini oluşturan bu parametreler, Rosenbrock fonksiyonu üzerinden detaylandırılmıştır [38]. Eşitlik 2.12’de Rosenbrock fonksiyonu gösterilmiştir.

$$f(x, y) = (1 - x)^2 + 100(y - x^2)^2 \quad (2.12)$$

İlgili fonksiyonun minimum değerini bulmak için ilk aşamada kısmi türevi alınarak sıfıra eşitlenir. Rosenbrock fonksiyonun x ve y parametrelerine göre türevleri sırasıyla, Eşitlik 2.13 ve 2.14’de gösterilmiştir.

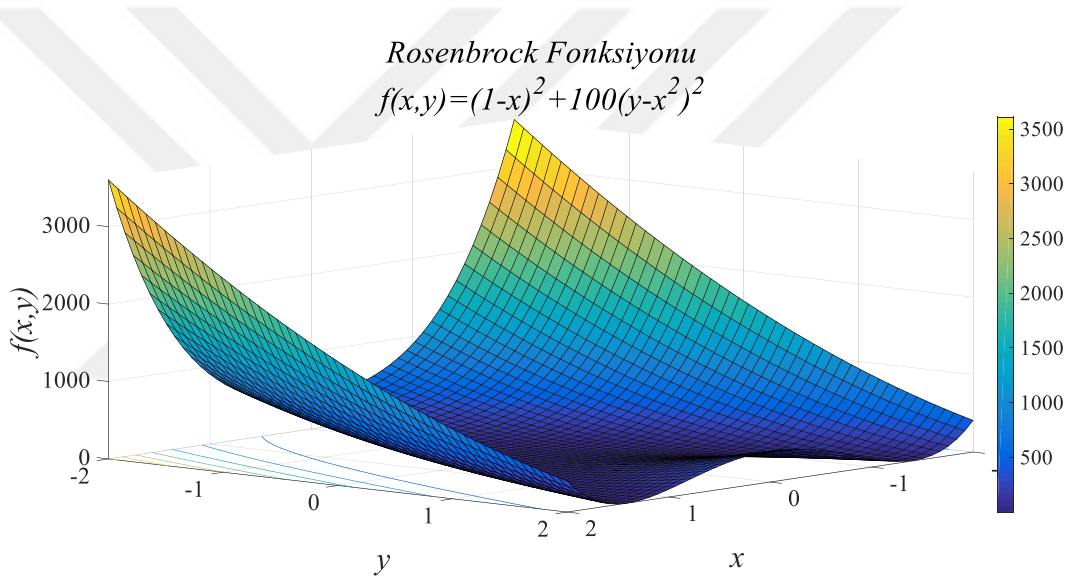
$$\frac{df(x, y)}{dx} = 2(1 - x) - 400(y - x^2)x = 0 \quad (2.13)$$

$$\frac{df(x, y)}{dy} = 200(y - x^2) = 0 \quad (2.14)$$

Eşitlik 2.14'ün irdelenmesi sonucunda $y = x^2$ eşitliği elde edilir ve bu elde edilen eşitliğin Eşitlik 2.13'de yerine yazılması sonucunda, Eşitlik 2.15 elde edilir.

$$1 - x - 200(x^2 - x^2) = 1 - x = 0 \Rightarrow x = 1 \quad (2.15)$$

Bu bağlamda Rosenbrock fonksiyonu için $\min f(x, y) = 0$ değeri $x = y = 1$ 'de elde edilir. İlgili amaç fonksiyonun minimum değeri elde edilirken uygulanan yaklaşım birinci dereceden türev ya da gradyan hesabıdır. Şekil 2.9'da Rosenbrock fonksiyonunun, $-2 \leq x_i \leq 2$ ve $-2 \leq y_i \leq 2$ sınırları içindeki değişimi gösterilmiştir.



Şekil 2.9. Rosenbrock fonksiyonunun dağılımı

Optimizasyon kavramları irdelenirken problemlerin tanım sürecinin merkezini oluşturan birkaç önemli kavram daha bulunmaktadır. Yerel minimum veya maksimum olarak bilinen bu kavramlar, amaç fonksiyonun tanımlı olduğu aralıktaki *yerel* maksimum veya minimum noktayı temsil ederler. Bu noktalara yerel ekstremum noktalar denir ve bu belirli sınırlar içerisindeki aykırılığı temsil ederler.

Bu kavramlardan sonra mutlak(global) maksimum veya minimum kavramları gelmektedir. İlgili amaç fonksiyonun tanımlı olduğu aralıktaki maksimum veya minimum değerleri aldığı noktalar olarak tanımlanır ve mutlak ekstremum olarak da bilinirler [39].

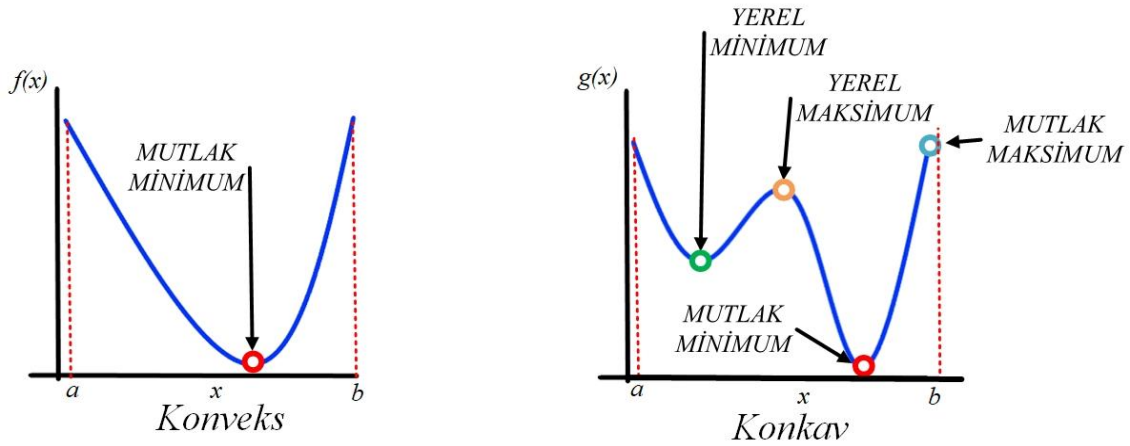
Bu kavramlara ek olarak konveks ve konkav fonksiyonlar bulunmaktadır. Konveks fonksiyonlarda yerel minimum değeri ile mutlak minimum değerleri birbirine eşittir. Konkav fonksiyonlarda ise bu noktalar farklılık gösterir [40].

Bir fonksiyonun konveks veya konkav olması Jensen eşitsizliği vasıtasıyla Eşitlik 2.16'da gösterilmiştir.

$$f(p_1x_1 + p_2x_2 + \dots + p_nx_n) \geq p_1f(x_1) + p_2f(x_2) + \dots + p_nf(x_n) \quad (2.16)$$

$f(x_i)$ amaç fonksiyonu konveks ise $p_i \geq 0, (i=1,2,\dots,n)$ ve $p_1 + p_2 + \dots + p_n = 1$ eşitliği sağlanırsa, konvekslik aralığından alınan her bir $x_1, x_2, \dots, x_n$ değeri için aşağıdaki eşitsizlik geçerlidir.

Şekil 2.10'da konvekslik ve konkavlık kavramları bir boyutlu fonksiyonlar yardımıyla tanımlanmıştır. Ayrıca ilgili şekilde yerel ve mutlak ekstremum noktaları da gösterilmiştir.

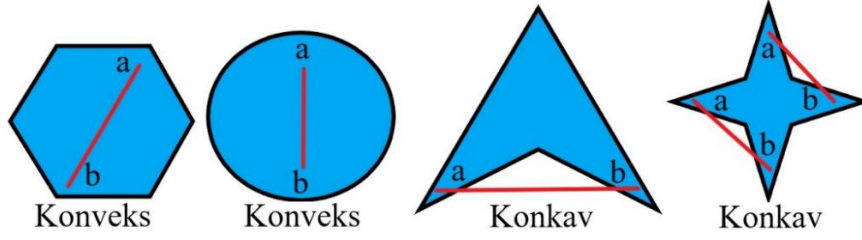


Şekil 2.10. Konveks ve Konkav fonksiyonlar

Konveks fonksiyonda görüldüğü gibi $f : [a,b] \rightarrow R$ aralığında tanımlı fonksiyonda, yerel minimum noktası aynı zamanda mutlak minimum noktasıdır.

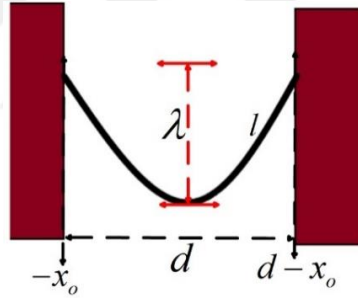
Konkav fonksiyonda ise yerel minimum ve mutlak maksimum noktaları farklılık göstermektedir. Bu farklılık hesaplama sürecinde kullanıcı karmaşıklığına ve hatalı hesaplamalara neden olabilmektedir.

Şekil 2.11’de konveks ve konkav özellikleri gösteren bazı şekil örnekleri verilmiştir.



Şekil 2.11. Konveks ve Konkav şekillerden bazıları

Konveks fonksiyonlara ait ekstremum noktaların hesaplanması kolayken konkav fonksiyonlarda yerel ekstremum noktalar yüzünden bu işlem biraz zorlaşmaktadır. Bir diğer örnek mühendislik problemi ise iki nokta arasında asılı ipin toplam potansiyel enerjisini minimize edecek eğiminin ya da şeklinin ($y(x)$) bulunmasını sağlayan değerlerin belirlenmesidir [41]. İlgili problemin görseli Şekil 2.12’de verilmiştir.



Şekil 2.12. İp veya zincir probleminin (catenary) temsili gösterimi

Verilen probleme ait amaç fonksiyonu veya ipe ait potansiyel enerji denklemi, Eşitlik 2.17’de gösterilmiştir. Amaç fonksiyonunu gerçekleştiren $y(x)$ değeri ile potansiyel enerjiyi minimize edecek şekil elde edilecektir.

Eşitlik 2.17’de m kütle değerini, l ipin uzunluğunu, g yerçekimi ivmesini, λ ise ipin sarkma miktarını temsil etmektedir.

$$\min_{y(x)} E_p = \frac{mg}{l} \int_{x_0}^{d-x_0} y(x) \sqrt{1+y'^2} dx \quad (2.17)$$

Lagrange denklemleri ($L = T - V$) vasıtasıyla ilgili denklem yeniden ifade edilir. Denklemdaki ifadelerin statik olması nedeniyle T parametresi sıfıra eşittir ve eşitlik aşağıda gösterildiği şekilde yazılabilir.

$$L = -V = -\left(\frac{mg}{l} \int_{x_0}^{d-x_0} y(x) \sqrt{1+y'^2} dx\right) \quad (2.18)$$

Lagrange yoğunluğu ile Lagrange çarpanı arasında aşağıda gösterildiği gibi bir ilişki vardır.

$$L = \int \ell dx \quad (2.19)$$

Böylece:

$$\ell = -\left(\frac{mg}{l} y(x) \sqrt{1+y'^2}\right) \quad (2.20)$$

Elde edilen eşitlik ve Euler-Lagrange denklemi ile aşağıda gösterilen denklem grupları meydana gelir.

$$\begin{aligned} \frac{d}{dx} \left[\frac{yy'}{\sqrt{1+(y')^2}} \right] - \sqrt{1+(y')^2} &= 0 \\ \frac{(y')^2}{\sqrt{1+(y')^2}} + \frac{yy''}{\sqrt{1+(y')^2}} - \frac{y(y')^2 y''}{\sqrt{1+(y')^2}^3} - \sqrt{1+(y')^2} &= 0 \end{aligned} \quad (2.21)$$

Eşitlik 2.21'deki elde ifadelerde bazı sadeleştirmeler sonucunda Eşitlik 2.22 elde edilir.

$$\frac{yy''}{(1+(y')^2)^{3/2}} - \frac{1}{\sqrt{1+(y')^2}} = 0 \quad (2.22)$$

Eşitlik 2.22, $y(x)$ parametresinin hesabı için çözülürse:

$$\frac{yy''}{(1+(y')^2)^{3/2}} - \frac{1}{\sqrt{1+(y')^2}} = 0 = \frac{d}{dx} \left[\frac{y}{\sqrt{1+(y')^2}} \right] = 0 \quad (2.23)$$

x parametresine göre integralinin alınması sonucunda eşitlik aşağıdaki basit hale getirilir.

$$\frac{y}{\sqrt{1+(y')^2}} = C_1, C_1 \in R \quad (2.24)$$

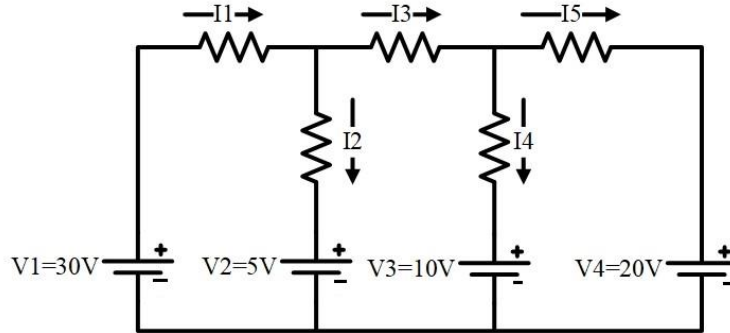
Birinci dereceden diferansiyel denklem haline getirilen problem için çözüm aşağıdaki gibi olur.

$$y = C_1 \cosh\left(\frac{x+C_2}{C_1}\right), C_2 \in R \quad (2.25)$$

Burada C_1 ve C_2 değerleri $y(x)$ fonksiyonun sınır koşulları ile kolayca hesaplanır.

Bir diğer optimizasyon problemi ise elektrik mühendisliği ile alakalı olacak şekilde bir pilden maksimum güç çekilmesini sağlayacak topolojinin belirlenmesini sağlayacak doğrusal programlama tabanlı bir problemidir [42].

Şekil 2.13'te gösterildiği gibi 30V değerindeki kaynak vasıtasıyla diğer üç pil grubunun beslenmesi hedeflenmektedir.



Şekil 2.13. Devrenin bağlantı şeması

Fakat kollardan geçecek akımların değerleri farklı kısıtlara sahiptir. Bu bağlamda maksimize edilmek istenen amaç fonksiyonu, Eşitlik 2.26'da gösterilmiştir. Optimizasyon problemi haline getirilen problem kısıtları ile beraber Eşitlik 2.27'de verilmiştir.

$$P_{toplam}(I_2, I_4, I_5) = 5 \times I_2 + 10 \times I_4 + 20 \times I_5 \quad (2.26)$$

$$\begin{aligned}
& \text{maksimize} \{ P_{\text{toplam}}(I_2, I_4, I_5) = 5 \times I_2 + 10 \times I_4 + 20 \times I_5 \\
& \text{öyle ki} \begin{cases} I_1 = I_2 + I_3 \\ I_3 = I_4 + I_5 \end{cases} \\
& \text{kısıtlar} \{ I_1 \leq 4, I_2 \leq 3, I_3 \leq 3, I_4 \leq 2, I_5 \leq 2, \\
& I \geq 0,
\end{aligned} \tag{2.27}$$

Bu kısıtlara göre, uygun direnç değerleri belirlenerek ilgili devre için maksimum güç transferini sağlayacak parametreler elde edilebilir.

2.3. Sürekli ve Ayırık Zamanlı Optimizasyon

Sürekli ve ayırık zamanlı modellemeler arasındaki temel farklılıklar, optimizasyon kavramı ana başlığı altında da benzerlik göstermektedir. Sürekli zamanlı optimizasyon, geometri tabanlı, konveks analiz gibi bir takım matematiksel hesaplamalara dayanmaktadır ve optimizasyon kavramının temelini oluşturmaktadır. Ayırık zamanlı optimizasyon ise bilgisayarların işlem kapasitesine bağlı olarak nispeten yenilikçi bir kavram şeklinde karşımıza çıkmaktadır [43].

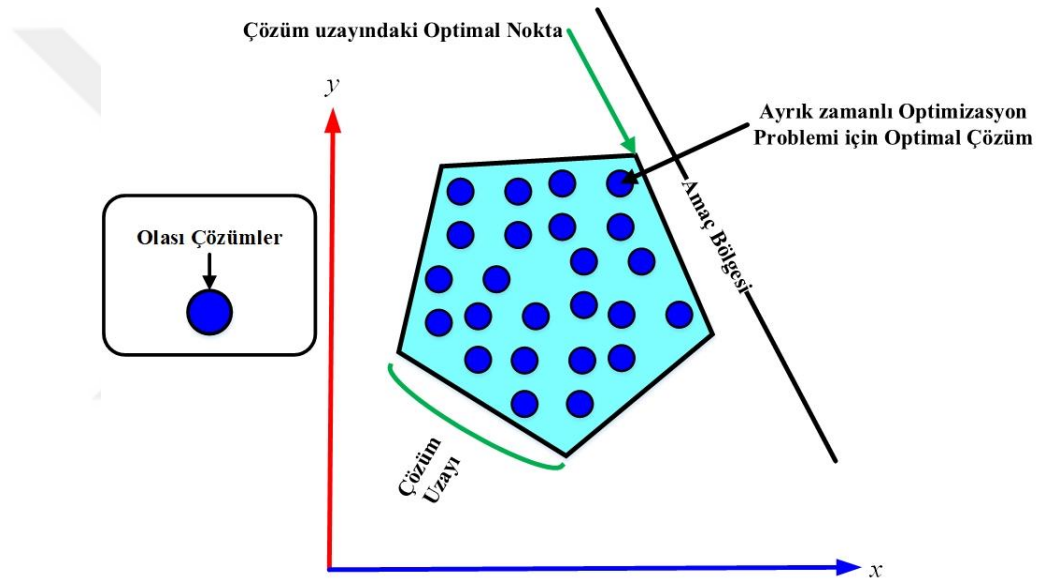
Sürekli zamanlı optimizasyonda, modeldeki değişkenler gerçek sayılar kümesinden rastgele bir değeri alabilmektedir. Bu özelliği ile ayırık zamanlı optimizasyonda bulunan ikilik taban (0 & 1) değerleri ile problem uzayının sınırlaması durumu sürekli zamanlı optimizasyon için geçerli değildir. Bazı modeller sadece ayırık zamanlı değerler alması suretiyle mantıklı bir yapı oluştururken bazı modeller, sürekli zamanlı herhangi bir gerçek değer alabilmektedir. Bu ayrım, problemin yapısı, çözüm tekniği veya süreci ile doğrudan ilişkilidir.

Ayırık zamanlı değişken içeren modeller, ayırık optimizasyon problemi olarak adlandırılırken, sürekli zamanlı değişken içeren modeller ise sürekli optimizasyon problemi olarak adlandırılır. Genel hatları ile sürekli zamanlı optimizasyon problemleri daha kolay çözülebilme eğilimindedirler.

Bilgisayar tabanlı işlem kapasitesindeki gelişmeler, ayırık zamanlı olarak karmaşıklıkları artan problemlerin çözümüne imkan sağlamaktadır [44]. Fakat sürekli zamanlı optimizasyon algoritmaları, ayırık zamanlı optimizasyon algoritmaları için hayati önemdedir. Çünkü ayırık zamanlı optimizasyon problemlerinin çözüm sürecinde, arka planda sürekli zamanlı problemler kümesi çözümlenmektedir ve bu çözümün kalitesi veya başarımı ayırık zamanlı çözümün başarımını doğrudan etkilemektedir.

Sürekli zamanlı optimizasyon yöntemlerinin arka planında, konveks analiz ve iyi tanımlanmış geometri tabanlı yaklaşımlar bulunmaktadır. Bu yüzden, operasyonel araştırma, kontrol teorisi ve makine öğrenmesi alanlarında, sürekli zamanlı optimizasyon yöntemlerinden esinlenilmiştir. Sürekli zamanlı optimizasyon, doğal fenomenleri tanımlamak, anlamlandırmak ve yorumlanmak amacıyla önemli bir araç olarak karşımıza çıkmaktadır.

Ayrık zamanlı optimizasyon ise ayrık zamanlı bileşenler ile sonlu bir uzaydaki optimal düzenlemeleri, gruplamaları, sıralamaları veya seçimleri bulmayı sağlamaktadır. Şekil 2.14'te ayrık zamanlı bir optimizasyon probleminin çözüm uzayının temsili gösterimi verilmiştir.



Şekil 2.14. Ayrık ve sürekli zaman optimizasyon problem uzaylarının gösterimi

2.4. Optimizasyon Problemlerinin Türleri

Optimizasyon başlığı altında yapılan çalışmalar günden güne arttığından, optimizasyon kavramı ile ilintili genel bir terminoloji oluşturulmuştur. Farklı türdeki optimizasyon problemlerini tanımlamak için çeşitli ana başlıklar ortaya çıkmıştır. Bu terimler Tablo 2.1'de gösterildiği gibidir [45].

Tablo 2.1. Optimizasyon Problemlerinin türleri

<i>Problem Türü</i>	<i>Açıklaması</i>
Kombinasyonel problemler	Sınırlı ama genellikle çok büyük çözümler grubu
Kısıtlanmamış problemler	Kısıtlama bulunmaz ve $x \in R^n$ değerlerinin hepsine izin verilir
Doğrusal olarak kısıtlanmış problemler	Sadece doğrusal kısıtlar (basit sınırlar ve genel doğrusal kısıtlar)
Doğrusal olmayan kısıtlanmış problemler	En az bir kısıt doğrusal değildir.
Doğrusal programlama	Amaç fonksiyonu ve tüm kısıtlar doğrusaldır. Ayrıca sürekli zamanlı değişkenler ile tanımlıdır.
Doğrusal olmayan programlama	Amaç fonksiyonu veya kısıtlar doğrusal değildir. Ayrıca sürekli zamanlı değişkenler ile tanımlıdır.
Tamsayı Programlama	Değişkenler tamsayı olacak şekilde sınırlandırılmıştır
Karışık Tamsayı Programlama	Bazı değişkenler tamsayı bazıları sürekli.
Stokastik Optimizasyon	Problem verilerinden bazıları deterministik değildir.

2.5. Optimizasyon için Temel Araçlar

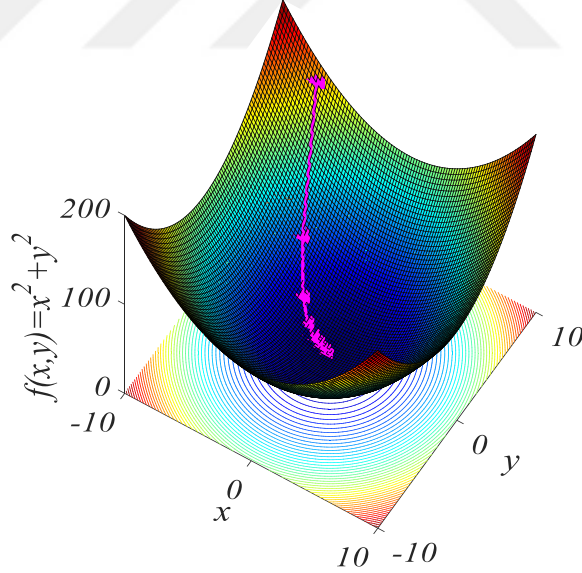
Optimizasyon sürecinin gerçekleştirilmesinde çeşitli temel kavramlar bulunmaktadır. Bu kavramlardan bazıları bu tez kapsamında açıklanacaktır. Optimizasyon fenomeni için temel oluşturan; gradyent, Jacobian, Hessien fonksiyonları ile Taylor serisi bunlardan bazılarıdır. Bu kavramlar, sürekli zamanlı optimizasyon problemleri için geliştirilen geometri tabanlı çözüm yöntemlerinin temelini oluşturmaktadır [29].

Gradyent (yönelim) hesabı, bir ($n = 1$) veya daha fazla değişken için hesaplanabilir ($n > 1$). Bir fonksiyonun gradyenti o fonksiyonun yönelimi veya değişimi hakkında bilgi vermektedir. 1-boyutlu bir fonksiyon için ilgili gradyent, fonksiyonun türevine eşittir.

Eşitlik 2.28’de $f(x)$ fonksiyonuna ait gradyent vektörünün hesabı gösterilmektedir. İlgili vektör, fonksiyon uzayının değişimi hakkında tasarımcıya fikir vermektedir [46].

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_{n-1}} \\ \frac{\partial f}{\partial x_n} \end{bmatrix} \quad (2.28)$$

Şekil 2.15'te $f(x,y) = x^2 + y^2$ şeklinde tanımlanan fonksiyona ilişkin gradyent tabanlı meyilli azalım (gradient descent) yöntemi kullanılarak oluşturulan optimizasyon probleminin çözüm süreci gösterilmiştir. İlgili fonksiyonda ilk aşamada başlangıç noktasına ilişkin fonksiyonun değeri hesaplanmıştır. Sonraki aşamada ise gradyent bilgisi kullanarak fonksiyonun yeni değeri hesaplanır. Bu sayede sürekli zamanda tanımlı konveks bir fonksiyon için global minimum değeri bulunabilir.



Şekil 2.15. $f(x,y) = x^2 + y^2$ fonksiyona ait gradyentin değişiminin gösterimi

Gradyent hesabı genel olarak ilgili fonksiyonun birinci dereceden kısmi türevlerinden oluşan bir vektör iken birinci dereceden kısmi türevlerden oluşan matris ise Jacobian matrisi olarak tanımlanmıştır. Bunlara ek olarak ikinci derecen kısmi türevlerden oluşan matris ise Hessian matrisi olarak tanımlanmıştır. Gradyent fonksiyonu ∇f , Jacobien fonksiyonu Jf ve

Hessian fonksiyonu $\nabla^2 f$ veya Hf şeklinde gösterilmektedir. $f : R^n \rightarrow R^m$ ve $f_1(x_1, \dots, x_n), \dots, f_m(x_1, \dots, x_n)$ fonksiyonlar kümesine ait Jacobien fonksiyonu, Eşitlik 2.28'de gösterilmektedir. Eşitlik 2.29'da ise $f(x)$ fonksiyonuna ait Hessian fonksiyonu ifadesi gösterilmektedir.

$$Jf = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \dots & \frac{\partial f_m}{\partial x_n} \end{bmatrix} \quad (2.28)$$

$$\nabla^2 f = H_f = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \dots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \dots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix} \quad (2.29)$$

Bu fonksiyonlar ile ilgili çeşitli özellikler bulunmaktadır. Jacobian fonksiyonun gradyenti, Hessian fonksiyonuna eşittir ve Eşitlik 2.30'da gösterilmiştir.

$$\nabla f(J_f) = H_f = \nabla^2 f \quad (2.30)$$

Bunlara ek olarak Hessian matrisinin çeşitli özellikleri vardır. Bu özellikler, Hessian matrisinin simetrik olması ve karesel olmasıdır [47].

Bir diğer önemli kavram ise Taylor serisidir. Taylor serisi, bir f fonksiyonun belirli bir x_0 noktasında yakınsamasını sağlar. Bu yakınsama polinom tabanlı gerçekleştirilir ve elde edilen fonksiyon kolayca manipüle edilebilir. Türevi olan tüm fonksiyonlara uygulanabilir ve aşağıda maddeler halinde gösterilen çeşitli avantajlara sahiptir [48].

- Doğrudan değerlendirilmesi zor olan fonksiyonların verilen bir nokta civarında tahmin edilebilmesi sağlanır.
- Orijinal fonksiyona ait türev veya integral değerlerinin yaklaşık olarak kestirilmesini mümkün olur.
- Fonksiyonların sıfır noktalarının bulunmasını sayesinde fonksiyonları minimize eden çok sayıda yöntemin türetilmesine imkan sağlar.

Çoğu problemin doğrudan hesaplanması oldukça zordur. Bu bağlamda bazı problemlere ait yaklaşık çözümler yeterli olmaktadır ve Taylor serileri bu çözüm sürecinde teorik ve pratik olarak sıklıkla kullanılmaktadır. Verilerin kesin olduğu durumda bile yaklaşık çözüm yeterli olmaktadır ve işlem sürecine ait maliyeti ise oldukça azaltmaktadır [49]. Bu süreci sağlayan ve literatürde sıklıkla kullanılan Taylor serisine ait n 'nci dereceden genel tanım bağıntısı Eşitlik 2.31'de gösterilmiştir. p değeri burada adım boyutunu temsil etmektedir.

$$f(x_0 + h) \approx f(x_0) + pf'(x_0) + \frac{1}{2}p^2 f''(x_0) + \dots + \frac{p^n}{n!} f^{(n)}(x_0) \quad (2.31)$$

İlgili eşitlikte, $f^{(n)}(x_0)$ ifadesi f fonksiyonunun belirli bir x_0 noktasında n 'nci dereceden türevinin değerini temsil etmektedir ve $n! = n(n-1)(n-2)\dots 3 \times 2 \times 1$ şeklinde ifade edilir. h değeri ise bir değişken olup yaklaşık çözüm sadece küçük h değerlerinde doğru olmaktadır. Eşitlik 2.32'de örnek bir fonksiyona ait Taylor serisi açılımı 3'ncü dereceye kadar gösterilmiştir.

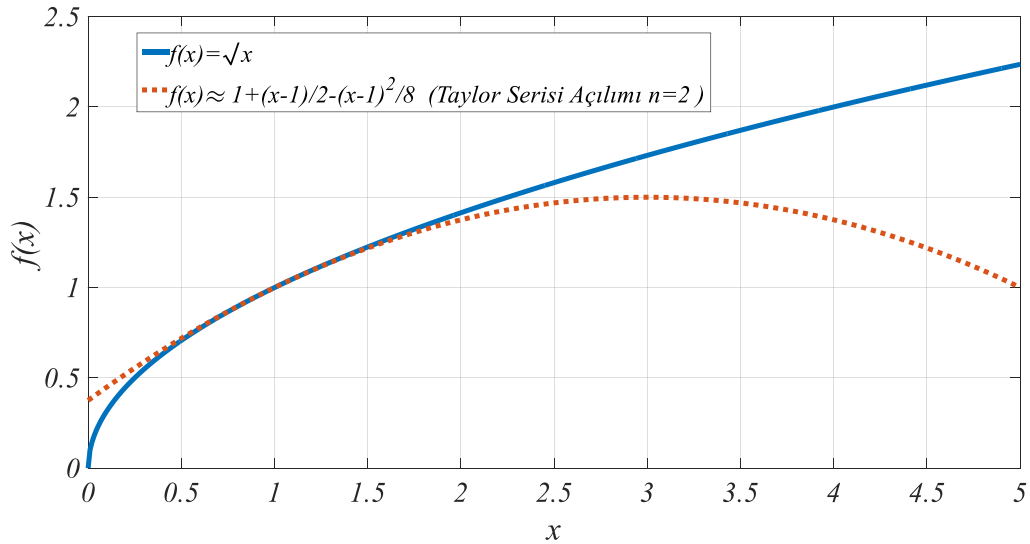
$$\begin{aligned} f(x) &= \sqrt{x} \quad \& \quad x_0 = 1 \\ f(x_0) &= \sqrt{x_0} = \sqrt{1} = 1 \\ f'(x_0) &= \frac{1}{2} x_0^{-\frac{1}{2}} = \frac{1}{2} 1^{-\frac{1}{2}} = \frac{1}{2} \\ f''(x_0) &= -\frac{1}{4} x_0^{-\frac{3}{2}} = -\frac{1}{4} 1^{-\frac{3}{2}} = -\frac{1}{4} \\ f'''(x_0) &= \frac{3}{8} x_0^{-\frac{5}{2}} = \frac{3}{8} 1^{-\frac{5}{2}} = \frac{3}{8} \\ &\vdots \end{aligned} \quad (2.32)$$

İlgili eşitlikler kullanılarak $x_0 = 1$ civarında bir değer olarak $f(1.4)$ değeri kolayca yüksek başarı ile hesaplanabilecektir. Eşitlik 2.33'de $f(1.4)$ değerinin hesaplanmasına ilişkin Taylor

serisi açılımı gösterilmiştir. Belirlenen başlangıç koşuluna yakın seçilen $p = 0.4$ değerine göre $f(1.4)$ fonksiyonun değeri yüksek bir başarımla kestirilmiştir ve ilgili eşitlikte yüzde hata hesabı da gösterilmiştir.

$$\begin{aligned}
 \sqrt{1+p} &= f(x_0 + p) \\
 f(x_0 + p) &\approx f(x_0) + pf'(x_0) + \frac{1}{2}p^2 f''(x_0) + \frac{p^3}{6} f'''(x_0) \\
 f(x_0 + p) &\approx 1 + p\left(\frac{1}{2}\right) + \frac{1}{2}p^2\left(-\frac{1}{4}\right) + \frac{1}{6}p^3\left(\frac{3}{8}\right) \\
 f(x_0 + p) &\approx 1 + p\left(\frac{1}{2}\right) - \frac{1}{8}p^2 + \left(\frac{1}{16}\right)p^3 \tag{2.33} \\
 p = 0.4 &\rightarrow f(1+0.4) \rightarrow f(1.4) = \sqrt{1.4} = 1.183 (\text{Gerçek Değer}) \\
 \text{Taylor serisi yardımı ile} &\rightarrow f(1.4) \approx 1 + 0.4\left(\frac{1}{2}\right) - \frac{1}{8}(0.4)^2 + \left(\frac{1}{16}\right)(0.4)^3 \rightarrow f(1.4) \approx 1.184 \\
 \text{Yüzde Hata} &= \frac{|\text{Hesaplanan Değer} - \text{Gerçek Değer}|}{\text{Gerçek Değer}} \times 100 = \frac{10^{-3}}{1.183} \times 100 = \%0.0845
 \end{aligned}$$

Taylor serisi açılımındaki ilk iki terim, x_0 noktasındaki f fonksiyonuna ait formülü vermektedir. İlgili benzetim Şekil 2.16'da gösterilmiştir.



Şekil 2.16. Orijinal fonksiyon ile Taylor serisi açılımının karşılaştırılması

Şekil 2.16'da görüldüğü gibi $x_0 = 1$ noktasında fonksiyon başarıyla yakınsarken belirtilen noktadan uzaklaştıkça gerçek fonksiyona ait dağılım ile benzetim sonuçları arasındaki fark artmaktadır.

Literatürdeki optimizasyon problemlilerinin çoğu çok boyutludur. Çok boyutlu bir fonksiyonun belirli bir noktadaki değerinin belirlenmesini sağlayan bir Taylor serisi açılımına ait bir örneğin bu tez kapsamında verilmesi uygun görülmüştür. Taylor serisinin çok boyutlu uzaylara ilişkin genelleştirilmiş hali Eşitlik 2.34’de gösterilmiştir.

$$f(x_0 + p) = f(x_0) + \sum_{i=1}^n p_i \left. \frac{\partial f(x)}{\partial x_i} \right|_{x=x_0} + \sum_{i=1}^n \sum_{j=1}^n p_i p_j \left. \frac{\partial^2 f(x)}{\partial x_i \partial x_j} \right|_{x=x_0} + \dots \quad (2.34)$$

$f(x, y) = x^3 + 5xy + 7xy^2 + 2y^3$ fonksiyonunun $x_0 = (-2, 3)^T$ noktasındaki yakınında ki bir değerinin belirlenmesi için ilk aşamada ilgili fonksiyona ait gradyent değeri Eşitlik 2.35’te verilmiştir.

$$\nabla f(x) = \begin{pmatrix} 3x^2 + 10xy + 7y^2 \\ 5x^2 + 14xy + 6y^2 \end{pmatrix} \quad (2.35)$$

Ayrıca aynı fonksiyona ait Hessian matrisi ise Eşitlik 2.36’da gösterilmiştir.

$$\nabla^2 f(x) = \begin{pmatrix} 6x + 10y & 10x + 14y \\ 10x + 14y & 14x + 12y \end{pmatrix} \quad (2.36)$$

Belirtilen $x_0 = (-2, 3)^T$ noktasında gradyent ve Hessian matrisi Eşitlik 2.37’de gösterilen verilen değerleri almaktadır.

$$\nabla f(x) = \begin{pmatrix} 15 \\ -10 \end{pmatrix} \quad \nabla^2 f(x) = \begin{pmatrix} 18 & 22 \\ 22 & 8 \end{pmatrix} \quad (2.37)$$

İlgili problem için $p = (p_1, p_2)^T = (0.1, 0.2)^T$ olarak seçilmiştir. Bu bağlamda yaklaşık olarak hesaplanmak istenen değer ise $f(x_0 + p) = f(-2 + 0.1, 3 + 0.2) = f(-1.9, 3.2)$ şeklinde olur ve bu noktadaki değeri Eşitlik 2.38’de gösterilen Taylor serisi yardımı ile kestirilmiştir.

$$\begin{aligned}
f(-1.9, 3.2) &= f(x_0 + p) \\
f(-1.9, 3.2) &\approx f(x_0) + p^T \nabla f(x_0) + \frac{1}{2} p^T \nabla^2 f(x_0) p \\
f(-1.9, 3.2) &\approx -20 + (0.1 \quad 0.2) \begin{pmatrix} 15 \\ -10 \end{pmatrix} + \frac{1}{2} (0.1 \quad 0.2) \begin{pmatrix} 18 & 22 \\ 22 & 8 \end{pmatrix} \begin{pmatrix} 0.1 \\ 0.2 \end{pmatrix} \\
f(-1.9, 3.2) &\approx -20 - 0.5 + 0.69 \\
f(-1.9, 3.2) &\approx -19.81
\end{aligned} \tag{2.38}$$

İlgili çok boyutlu fonksiyonun $f(-1.9, 3.2)$ noktasındaki değeri Taylor serisi yardımı ile $f(-1.9, 3.2) \approx -19.81$ olarak bulunmuştur. Ayrıca, fonksiyonun belirtilen noktadaki gerçek değeri ise $f(-1.9, 3.2) = -19.755$ olarak hesaplanır.

2.6. Kısıtlı ve Kısıtsız Optimizasyon

Buraya kadar optimizasyon problemlerinin sürekli zamandaki çözümlerine çeşitli yaklaşımlar ve uygulamalara değinilmiş ve çözüm uzayının reel sayılar kümesinden oluşması ve herhangi bir kısıtın bulunmadığı durumlar açıklanmıştır. Fakat gerçek hayattaki optimizasyon problemlerinin çok büyük kısmında çeşitli nedenlerden dolayı kısıtlar ve sınırlar bulunmaktadır. Bu yüzden optimal koşullar kavramı ortaya çıkmıştır ve bu kavram optimizasyon fenomeni için oldukça önem arz etmektedir.

Optimal koşullar çeşitli sınırlar altında gerçekleştirilebileceği gibi bazı durumlarda sınırlamalara tabidir. Bu tarz sınırlamalar hayatın her aşamasında bulunmaktadır [50].

Bir aracın belirli bir karakteristikteki bir virajdan başarı ile geçebilmesi için aracın çeşitli koşulları sağlaması gerekmektedir. Bunlardan bazıları hız, ağırlık merkezi v.b şeklindedir.

Gerçek hayatta optimizasyon problemleri asıl olarak bu kısıtlar altında şekillenir. Bir fonksiyona ait optimal koşullar araştırılırken de birçok kısıt bulunur. Bu kısıtlar kullanıcının içinde bulunduğu koşullar ile doğrudan ilişkilidir.

Kısıtlar bazen toplam iterasyon sayısı iken bazen tolerans ya da minimum hata olarak adlandırılan ve fonksiyonun farklı değerleri arasındaki farkın azalmasına istinaden tanımlanan parametreler olarak karşımıza çıkmaktadır.

Genellikle de fonksiyona ait değişken veya değişkenlerin alabileceği değerler uzayının boyutunun sınırlandırılması şeklindedir. Bu bağlamda kısıtlı ve kısıtsız optimizasyon teknikleri arasındaki ilişkiler belirlenmelidir. Bu ilişkiler aşağıda sıralanmıştır.

- Kısıtlamaların bazı tasarım problemlerinde başarıma doğrudan etkisi olmamaktadır.
- Kısıtsız optimizasyon yöntemleri alanındaki çalışmalar, kısıt içeren optimizasyon problemlerinin çözüm süreci için temel oluşturmuşlardır.
- Belirli bir karmaşıklığa sahip mühendislik problemlerinin çözümünde kısıtsız optimizasyon yöntemleri kullanılabilir. Doğrusal ve doğrusal olmayan bir sistemdeki yer değiştirme etkisinin irdelenmesi sürecinde potansiyel enerjinin minimizasyonu kullanılabilir. Benzer şekilde, ayrık zamanlı bir sistemin öz değerleri veya öz vektörleri hesaplanırken Rayleigh katsayısının en aza indirgenmesinden faydalanabilir.

Doğrusal olmayan ve kısıtsız bir optimizasyon probleminde Eşitlik 2.39'de gösterilen koşulun sağlanması istenir. Bu bağlamda x vektörü kısıt olmayacak şekilde parametre uzayındaki herhangi bir değer olabilir.

$$\begin{aligned} x^* &= \arg \min_x f(x) \\ x &\in R^n \end{aligned} \quad (2.39)$$

Kısıtlı bir optimizasyon probleminde ise çözüm uzayı Eşitlik 2.40'de gösterildiği gibi bazı koşullara bağlı olarak sınırlandırılmıştır.

$$\begin{aligned} x^* &= \arg \min_x f(x) \\ \text{öyle ki } \begin{cases} g_j(x) \leq 0, & j = 1, 2, \dots, m \\ h_k(x) = 0, & k = 1, 2, \dots, p \end{cases} \end{aligned} \quad (2.40)$$

Kısıtlı optimizasyon problemlerindeki optimal nokta ile kısıtsız optimizasyon problemdeki minimum nokta arasında benzerlikler vardır. Bu benzerlikler Eşitlik 2.41'de gösterilmiştir.

$$\begin{aligned} \nabla f(x^*) &= 0 \\ Jf &= \left[\frac{\partial^2 f}{\partial x_i \partial x_j} \right]_{x^*} = \text{pozitif tanımlıdır} \end{aligned} \quad (2.41)$$

Tablo 2.2’de doğrudan ve doğrudan olmayan kısıtlı optimizasyon yöntemlerinin gösterimi bulunmaktadır [46].

Tablo 2.2. Kısıtlı Optimizasyon Yöntemleri

<i>Doğrudan Yöntemler</i>	<i>Dolaylı Yöntemler</i>
Rassal araştırma yöntemleri	Değişkenlerin transformasyonu tekniği
Sezgisel arama yöntemleri	Ardışık kısıtlamasız küçültme tekniği
Kompleks yöntem	Dahili ceza fonksiyonu yöntemi
Amaç ve kısıt yaklaştırma yöntemleri	Harici ceza fonksiyonu yöntemi
Sıralı doğrusal programlama yöntemleri	Artırılmış Lagrange çarpanı yöntemi
Sıralı ikinci dereceden programlama yöntemleri	
Uygulanabilir yönelim tarifleri	
Zoutendijk’in yöntemi	
Rosen’in gradyan projeksiyon yöntemi	
Genelleştirilmiş indirgenmiş gradyan yöntemi	

2.7. Yerel ve Genel Optimizasyon

Çok sayıda doğrusal olmayan optimizasyon yöntemi optimizasyon problemlerinin çözüm sürecinde genellikle yerel sonuçlar ile ilgilenir. İlgili doğrusal olmayan fonksiyona ait genel minimum veya maksimum noktaların belirlenmesi süreci bazen hesaplama maliyetini oldukça artırmaktadır. Ayrıca bazı problemlerde hesaplanan noktanın tam olarak genel minimum noktası olduğunun belirlenebilmesi ve tanımlanabilmesi, ilgili fonksiyona ait genel hesaplamaları yapmaktan çok daha zordur. Bu yüzden kullanıcıların hesaplama sürecine ait bazı kısıtlar koyması gerekmektedir [51].

Konveks karakteristiğe sahip problemlerde ise yerel minimum noktası aynı zamanda genel minimum noktasını temsil etmektedir. Bunlara ek olarak gürbüz ve isabetli sonuçlar genel optimizasyon yöntemleri çoğunlukla farklı karakteristik gösteren yöntemlerin başarılı birer kombinasyonu sonucunda karşımıza çıkmaktadır.

2.8. Stokastik ve Deterministik Optimizasyon

Stokastik kavramı rassal veya belirsizlik içeren sistemleri tanımlamak için kullanılan genel bir terimdir. Bu rassallık ilgili sisteme ait tüm parametrelerde söz konusu olabilir. Bu parametreler girdi ve sistemin genel karakteristiği olabilir [52].

Deterministik modeller ise sisteme ait çıktıların geçmiş durumlara ait karakteristikler veya bilgiler yardımı ile her koşulda hesaplanabildiği ve herhangi bir rassallık veya belirsizlik içermeyen modellerdir. Deterministik modellerde, sisteme ait çıktılar kesin ve net bir şekilde belirlenebilirken, stokastik modellerde çıktı karakteristiği belirli bir dağılıma benzetilir ve olasılıksal bir yapıdadır. Bundan dolayı stokastik sistemlerde herhangi bir kesinlik söz konusu değildir ve sadece tahminler veya kestirimler söz konusudur. Böylece stokastik bir sistemin başarımı, sisteme ait modellemenin ve farklı durumlara ait olasılıkların modelleme başarımı ile doğru orantılıdır.

Örnek olarak iki farklı zarın atılması sonucu üst yüzeylerinde elde edilen sayıların toplamına ait olasılık dağılımına ait çıktı gösterilmektedir [53]. İlgili sürece ait dağılım Eşitlik 2.42’de, elde edilen sonuçların grafik gösterimi Şekil 2.17’de verilmiştir.

Pozitif Değerler Kümesi : $X = \{2, 3, \dots, 11, 12\}$

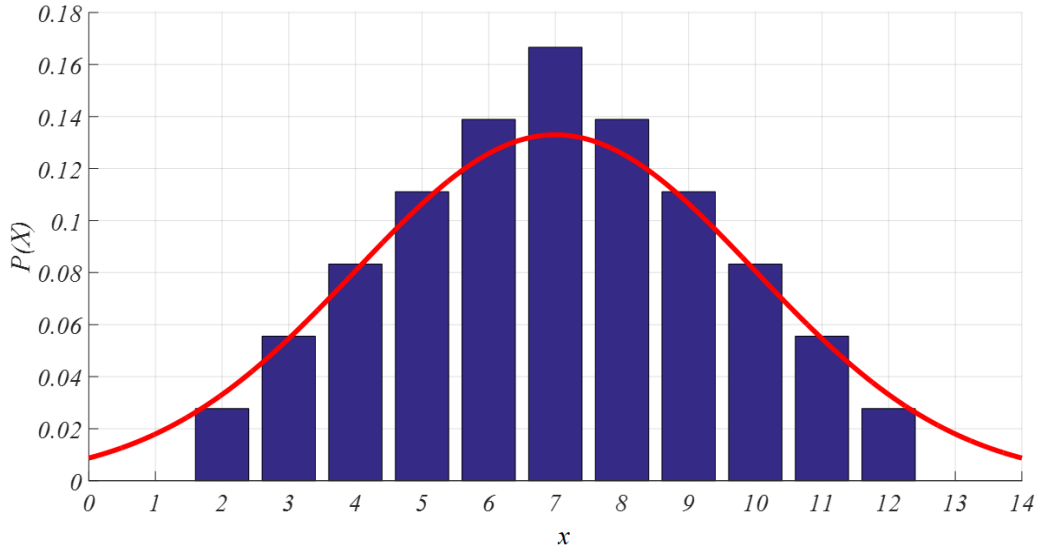
Rassal değişken spesifik değeri : $x \in X$

x değerinin olasılığı : $p(x)$

(2.42)

x	2	3	4	5	6	7	8	9	10	11	12
$P(x)$	$\frac{1}{36}$	$\frac{2}{36}$	$\frac{3}{36}$	$\frac{4}{36}$	$\frac{5}{36}$	$\frac{6}{36}$	$\frac{5}{36}$	$\frac{4}{36}$	$\frac{3}{36}$	$\frac{2}{36}$	$\frac{1}{36}$

Şekil 2.17 irdelendiğinde, iki zarın toplamı ve belirtilen değerin bulunma olasılığının dağılımı bir Gauss dağılımı ya da normal dağılım özelliği gösterilmektedir. Bu bağlamda zarların toplamının hangi değerleri yakınsayacağı bilinmemekle beraber belirli bir dağılımı oluşturacağı gözlenmektedir. Bu tarz sistemlere stokastik sistemler denmektedir.



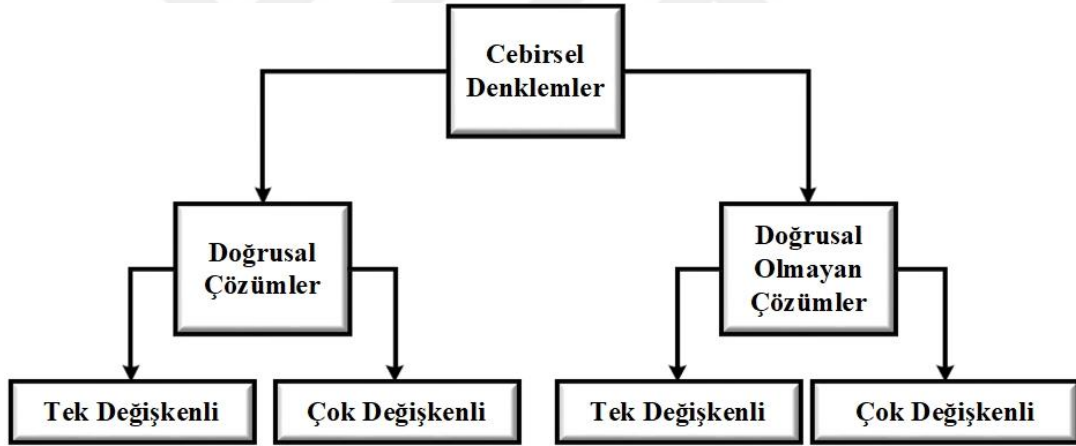
Şekil 2.17. İki zarın toplamalarının değerlerinin gerçekleşme olasılığına oranı

Açıkça bilinmektedir ki doğadaki problemler stokastik karakteristik taşımaktadırlar ve optimizasyon problemi olarak gerçekleştirilmesi zor veya önem arz eden problemlerin büyük bir kısmı stokastik bir karakteristiğe sahiptir.

Sadece en iyi tahmini kullanmak yerine, modelleyiciler bu bilgilere ek olarak ilgili modele ait yeni bilgiler elde ederek daha kullanışlı çözümler bulabilirler. Belirsiz bir durumu oluşturan bir dizi senaryonun farklı olasılıklarının birleşimi başlangıçta belirsiz olarak betimlenen durum hakkında bir model oluşturabilir. Stokastik optimizasyon yöntemleri bu belirsizlik değerlerini belirleyerek, modelin beklenen performansını oluşturacak çözümleri optimize etmektedir.

3. NÜMERİK OPTİMİZASYON

Nümerik optimizasyon kavramı makine öğrenmesi yapay zeka yöntemleri gibi yenilikçi ve yükselen trendlerin merkezinde bulunmaktadır. Çoğu zaman bir probleme ait en iyi çözümün bulunması oldukça zor veya uzun hesaplama süreleri isteyen bir süreçtir. Fakat bunun yerine oluşturan bir maliyet fonksiyonu ile çözümün başarısını ölçen veya başarıyı hakkında kullanıcıya bir ön fikir veren bir kayıp fonksiyonu oluşturulması çok daha kolaydır. Bu bağlamda nümerik optimizasyon yöntemleri çözüm için kesin olarak optimal sonuçları bulmak yerine optimal sonucu bulmaya yönelik karakteristikler içeren yöntemleri irdeleyen bir kavramdır [54]. Bu kavram çok çeşitli başlıklar içermektedir. Bunlardan en temel iki başlık doğrusal olan ve doğrusal olmayan denklemlerin nümerik optimizasyonudur. Şekil 3.1’de nümerik optimizasyon yöntemlerinin temel sınıflandırılması gösterilmektedir.

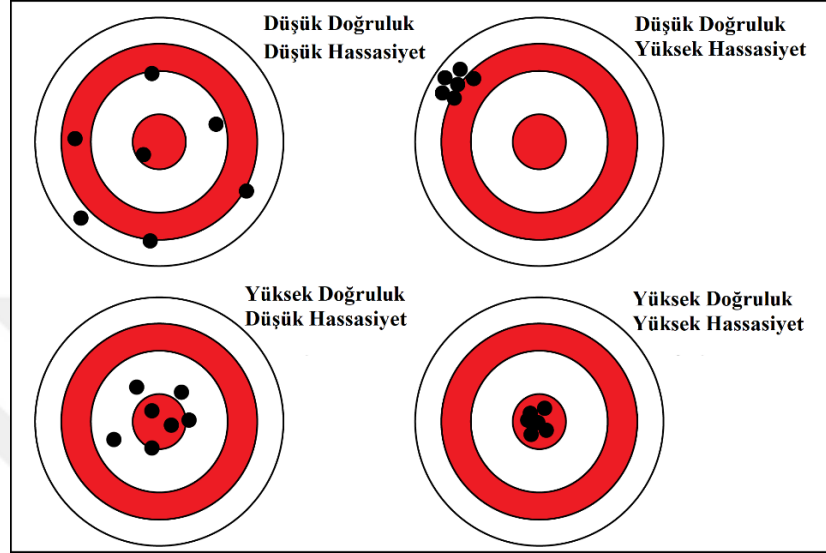


Şekil 3.1. Nümerik optimizasyon türlerinin sınıflandırılması

Mühendislik alanında karşılaşılan çoğu problem daha önce belirtildiği gibi doğrusal olmayan ve stokastik olarak tanımlanmış problemlerdir. Bu bilgiler ışığında bir optimizasyon probleminin başarımı farklılık gösteren bu koşullara rağmen başarılı veya kabul edilebilir çıktılar verebilmesidir. Bir nümerik optimizasyon yönteminin başarımı genel hatları ile iki parametreye bağlıdır. Bunlardan birincisi, doğruluk (accuracy) kavramıdır. Bu kavram önerilen yöntemin başarılı sonuçlar vermesidir. Bu sayede farklı karakteristikler ihtiva eden optimizasyon problemlerinde bile kabul edilebilir oranda başarılı sonuçlar elde edilebilir.

Bir diğeri ise kesinlik (precision) kavramıdır. Bu kavram önerilen optimizasyon yönteminin farklı başlangıç koşullarında veya karakteristikte olan problemlerde bile benzer

veya rassal olmayan sonuçlar vermesidir. Bu özellik gürbüzlük(robustness) olarak da literatürde yer almaktadır [55]. Bu iki kavramın başarılı bir birleşimi sonucunda, kabul edilebilir oranda sonuçlar üreten nümerik optimizasyon yöntemleri ortaya çıkmıştır. Bu karakteristik Şekil 3.2’de gösterilen bir hedef tahtası üzerinden temsil edilmiştir.



Şekil 3.2. Doğruluk ve hassasiyet kavramlarının gösterimi

3.1. Doğrusal Denklemlerin Nümerik Çözümlerine İlişkin Yöntemler

Bu bölümünde optimizasyon problemlerinin çözümü için temel oluşturan doğrusal denklem sistemlerinin çözümünü sağlayan yöntemler ve kavramlar irdelenecektir. Bu bağlamda katsayı matrisinin tersine dayanmayan çözüm yöntemleri ile ilgili bilgiler verilecektir.

Günümüzde katsayı matrisinin tersinin (A^{-1}) alınması sürecinde gerekli olan determinant ve eşlenik bilgilerinin hesaplanması için hantal hesaplamalar yerine daha verimli ve hızlı yöntemler kullanılmaktadır. Eşitlik 3.1’de n bilinmeyenli m adet denklemden oluşan bir doğrusal sisteme ait denklem takımı gösterilmiştir [56].

$$\begin{aligned} a_{11}u_1 + a_{12}u_2 + \dots + a_{1n}u_n &= b_1 \\ a_{21}u_1 + a_{22}u_2 + \dots + a_{2n}u_n &= b_2 \\ &\vdots \\ a_{m1}u_1 + a_{m2}u_2 + \dots + a_{mn}u_n &= b_n \end{aligned} \tag{3.1}$$

Eşitlik 3.1’de verilen doğrusal denklem takımı Eşitlik 3.2’de gösterildiği gibi matrissel bir formda yazılabilir.

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} \quad (3.2)$$

Ayrıca ilgili denklem takımları daha basit bir form olan $Au = b$ şeklinde de ifade edilebilir. Denklem takımının tek bir çözüme sahip olması için çeşitli şartların gerçekleşmesi gerekmektedir. Eğer sistem $m < n$ şartını sağlayacak şekilde ise ilgili doğrusal denklem takımı için tek bir eşsiz çözüm elde edilemez. Diğer taraftan $m > n$ ise çok fazla durum bulunmaktadır ve tek bir eşsiz çözüm yine elde edilemez. İlgili denklem takımında tek bir çözümünün elde edilebilmesi için ilgili denklem takımının parametrelerinin $m = n$ şartını sağlaması gerekmektedir. Bu matris denkleminin çözümü birçok problem için gereklidir. Bu problemlere veri enterpolasyonu, sonlu eleman problemlerine çözüm bulma, doğrusal matematiksel programlama, doğrusal denklemlerin analizi gibi örnekler verilebilir.

Katsayı matrisi A 'nın tersi sadece $m = n$ koşulunda mümkündür. A^{-1} in bulunmadığı durumlarda, doğrusal sistem tek bir çözüme veya herhangi bir çözüme sahip değildir. İlgili çözümün bulunması için Eşitlik 3.3'de gösterildiği gibi A^{-1} ifadesi ile çarpılır ve sonuç elde edilir.

$$Au = b \Rightarrow \boxed{u = A^{-1}b} \quad (3.3)$$

Eşitlik 3.3'de gösterilen problemin özel bir durumu ise $b = \lambda u$ şartları için geçerlidir. Bu sayede ilgili problem özdeğer problemine dönüşür ve özdeğerler (λ) ve karşılık gelen özvektörler (v) kare matris A arasındaki ilişkiyi sağlar. İlgili ilişki Eşitlik 3.4'de gösterilmiştir.

$$Av = \lambda v \quad \text{veya} \quad (A - \lambda I)v = 0 \quad (3.4)$$

Bir matrise ait öz vektörler ve öz değerlerin hesaplanması veri analizi, sinyal işleme gibi alanlar için önem arz etmektedir. Temel bileşenler analizi, doğrusal diskriminant analizi gibi veri analizi ve boyut indirgeme alanındaki önemli yöntemlere temel oluşturmaktadır [57]. Bu bağlamda öz değer ve öz vektörlerin hesaplanması süresi ve doğru belirlenmesi günümüzde artan veri boyutları da göz önüne alındığında daha da önemli bir konu olmuştur.

Öz değerler ile katsayı matrisi A arasındaki ilişki Eşitlik 3.5’de gösterilen ilişkiler bulunmaktadır.

$$tr(A) = \sum_{i=1}^n a_{ii} = \lambda_1 + \lambda_2 + \dots + \lambda_n \quad \text{ve} \quad \det(A) = \prod_{i=1}^n \lambda_i \quad (3.5)$$

Simetrik ve karesel bir matrisin iki farklı özdeğer, λ_i ve λ_j ye karşılık gelen özvektörleri ortogondur $v^T v = 0$. Genel olarak doğrusal sistemler Cramer kuralı yardımı ile çözülebilir. Eşitlik 3.6’da Cramer kuralı ve uygulama süreci gösterilmiştir.

$$u_i = \frac{\det A_i}{\det A}, \quad i = 1, 2, \dots, n \quad (3.6)$$

A_i matrisi i ’nci sütun ile b sütun vektörünün değiştirilmesi ile elde edilir. Üç adet denklem takımına ait denklem takımları Eşitlik 3.7’de gösterilmiştir. Üç adet doğrusal denklem için u_1 , u_2 ve u_3 şeklinde üç adet bilinmeyen bulunmaktadır.

$$\begin{aligned} a_{11}u_1 + a_{12}u_2 + a_{13}u_3 &= b_1 \\ a_{21}u_1 + a_{22}u_2 + a_{23}u_3 &= b_2 \\ a_{31}u_1 + a_{32}u_2 + a_{33}u_3 &= b_3 \end{aligned} \Rightarrow \Delta = \det(A) = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} \quad (3.7)$$

$$u_1 = \frac{1}{\Delta} \begin{vmatrix} b_1 & a_{12} & a_{13} \\ b_2 & a_{22} & a_{23} \\ b_3 & a_{32} & a_{33} \end{vmatrix}, \quad u_2 = \frac{1}{\Delta} \begin{vmatrix} a_{11} & b_1 & a_{13} \\ a_{21} & b_2 & a_{23} \\ a_{31} & b_3 & a_{33} \end{vmatrix}, \quad u_3 = \frac{1}{\Delta} \begin{vmatrix} a_{11} & a_{12} & b_1 \\ a_{21} & a_{22} & b_2 \\ a_{31} & a_{32} & b_3 \end{vmatrix}$$

Cramer kuralı teoriye dayanak için başarılı olsa da gerçek uygulamalar açısından çok kullanışlı değildir. Büyük boyutlardaki bir matrisin determinantının hesaplanması kolay bir süreç değildir. Bir matrisin tersini belirlemek için çok daha etkin yöntemler bulunmaktadır. Karesel bir matris olan ve $n \times n$ boyutlarındaki A matrisinin tersinin A^{-1} hesaplanması algoritmaların hesaplama karmaşıklığı olarak $O(n^3)$ şeklinde bir ilişki vardır. Büyük O notasyonu, zaman karmaşıklığını hesaplamak için tanımlanan bir ölçüttür [58]. Bir hesaplamının gerçekleştirilme süresini, tamamlanması için gereken adım sayısı ile bağlantılı olarak tanımlar.

n değerinin büyük olduğu koşullarda hesaplama süresi oldukça uzamaktadır. Bu bağlamda katsayı matrisinin tersini hesaplayan yöntemler genel hatları ile iki farklı kategoriye ayrılabilir. Bunlar doğrudan iteratif yöntemlerdir. Bu yöntemlerin seçim sürecinde kullanıcı birçok kriteri göz önüne almalıdır. Bunlardan bazıları, A matrisinin boyutu, hesaplama süresi, problemin türü ve istenen çözüm kalitesidir. Bu nümerik yöntemlerden birincisi ve en temeli Gauss Eleme Yöntemidir.

3.1.1 Gauss Eleme Yöntemi

Gauss eleme yönteminde temel mantık, karesel matrisi satır operasyonları ile üçgensel bir matrise çevirerek elde edilen basitleştirilmiş üçgensel matrisin doğrudan yerine koyma ile çözülmesini sağlamaktır [29]. Tekil olmayan $A = [a_{ij}]_{n \times n}$ matrisi ve $b \in R^n$ vektörü verilmiş olsun, Gauss eleme yöntemini doğrusal denklem takımı $Au = b$ eşitliğini $\bar{A}u = \bar{b}$ şeklindeki eşdeğer eşitliğe dönüşümünü sağlar. İlgili eşitlikte $\bar{A}$ yeni üçgensel katsayı matrisini temsil etmektedir. Sonraki aşamada $\bar{A}u = \bar{b}$ ifadesi geri yerine koyma yardımı ile çözülerek asıl hedeflenen sonuç olan $Au = b$ denkleminin sonucu elde edilir [59].

Gauss eleme yöntemi 3 temel satır operasyonundan oluşmaktadır.

- İki satırın sırası değiştirilir.
- Sıfırdan farklı bir sabit ile satır çarpılır.
- Satır bir diğer satırın katları ile değiştirilir.

Tüm bu işlemler Eşitlik 3.8’de gösterilen arttırılmış (augmented) bir matris olan $[A|b]$ üzerinden gerçekleştirilir. $A|b$ notasyonu A ve b şeklindeki iki matrisin arttırılmış şekilde birleşimini temsil etmektedir.

$$[A|b] = \left[\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & b_n \end{array} \right] \Rightarrow M^{(n)} = \underbrace{\left[\begin{array}{cccc} a'_{11} & a'_{12} & \cdots & a'_{1n} \\ 0 & a'_{22} & \cdots & a'_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a'_{nn} \end{array} \right]}_{\bar{A}} \underbrace{\left[\begin{array}{c} a'_{1(n+1)} \\ a'_{2(n+1)} \\ \vdots \\ a'_{n(n+1)} \end{array} \right]}_{\bar{b}} \quad (3.8)$$

Süreç sonunda $\bar{A}u = \bar{b}$ şeklindeki eşitlik elde edilir ve bu sayede $Au = b$ eşitliğini sağlayan u değerleri elde edilir. Bu bağlamda Gauss eleme yönteminin hesaplama karmaşıklığı ise $O(2n^3 / 3)$ şeklindedir.

3.1.2 Gauss-Jordan Eleme Yöntemi

Gauss-Jordan eleme yöntemi, Gauss eleme yönteminin geliştirilmiş halidir. Doğrusal denklem takımını çözerken aynı zamanda bir kare matrisin tersini hesaplar [60]. Gauss-Jordan yöntemi temel satır işlemlerini arttırılmış matris üzerinden gerçekleştirilirken matrisin katsayıları birim matrise dönüştürülür [29]. İlk aşamada arttırılmış matrisin tanımlanması gerekmektedir. Ayrıca A katsayı matrisi ile aynı boyutlardaki birim (I) matris tanımlanır. Bu işlemler sonucunda Eşitlik 3.9’da gösterilen arttırılmış B matris elde edilir.

$$B = [A|b|I] = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} & |b_1| & 1 & 0 & \cdots & 0 \\ a_{21} & a_{22} & \cdots & a_{2n} & |b_2| & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & |b_n| & 0 & 0 & \cdots & 1 \end{bmatrix} \quad (3.9)$$

Gauss-Jordan eleme yönteminde satır indirgeme vasıtasıyla B matrisini Eşitlik 3.10’da gösterilen forma dönüştürebilmektedir. Bu sayede ilgili u çözüm vektörü elde edilebilir. u vektörünün elde edilmesi sürecinde gerçekleştirilen işlemler Gauss eleme yöntemi ile benzerlik gösterir.

$$[I|u|A^{-1}] = \begin{bmatrix} 1 & 0 & \cdots & 0 & |u_1| & a'_{11} & a'_{12} & \cdots & a'_{1n} \\ 0 & 1 & \cdots & 0 & |u_2| & a'_{21} & a'_{22} & \cdots & a'_{2n} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & |u_n| & a'_{n1} & a'_{n2} & \cdots & a'_{nn} \end{bmatrix} \quad (3.10)$$

Gauss-Jordan eleme yöntemi nümerik olarak kararlı bir yöntem değildir ve bu bölümün başında bahsedilen hassasiyet ve doğruluk kavramlarını tam olarak karşılamaz. Daha iyi ve istikrarlı sonuçlar elde etmek için pivotlama tekniğini kullanmak yaygın bir uygulamadır. Genel olarak pivotlama satırdaki tüm değerlerin en büyük değer veya bir norm ile bölünmesi sonucu elde edilen bir ölçekleme yöntemidir. Gerekli durumlarda satırlar değiştirilebilir ve bu sayede en büyük eleman hareket ettirilebilir. Özellikle köşegen konumunda önde gelen katsayı olacak

şekilde tanınabilir. Bu sayede tüm ölçeklenmiş elemanlar $[-1,1]$ aralığında olacaktır. Bu sayede bazı istisnai karakteristiğe sahip büyük değerler katsayı matrisinden ayrıştırılmış olur. Bu sayede katsayı matrisinin çözüm süreci için daha kararlı bir çözüm elde edilir.

Gauss eleme ve Gauss-Jordan eleme yöntemleri için önem arz eden başka bir unsur ise başlıca elemanın değerinin sıfırdan farklı ($a_{11} \neq 0$) olması gerekmektedir. Bu durumdan kaçınmak için denklemleri yeniden düzenlemek mümkündür. Fakat diğer katsayı olan $\left(a_{22} - \frac{a_{21}a_{12}}{a_{11}}\right) \neq 0$ da bu istenmeyen durum belirtildiği gibi gerçekleşebilir. Bu problemten kaçınmak için pivot ve LU ayrıştırma yöntemi kullanılmaktadır.

3.1.3 LU Ayrıştırma Yöntemi

Herhangi bir karesel matris iki farklı matrisin çarpımı olacak şekilde Eşitlik 3.11’de gösterildiği gibi ifade edilebilir.

$$A = LU \quad (3.11)$$

İlgili eşitlikte, L ve U matrisleri sırasıyla alt ve üst üçgen matrisleri temsil etmektedir. İlgili üçgen matrislerden alt üçgen matris diyagonallerin alt kısmında sıfırdan farklı eleman içerirken, üst üçgen matris de ise durum tam tersidir [61]. Bu matrislere ve katsayı matrisine ait gösterim Eşitlik 3.12’de gösterilmiştir.

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \Rightarrow A = LU \Rightarrow L = \begin{bmatrix} \beta_{11} & 0 & \cdots & 0 \\ \beta_{21} & \beta_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{n1} & \beta_{n2} & \cdots & \beta_{nn} \end{bmatrix}, U = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \cdots & \alpha_{1n} \\ 0 & \alpha_{22} & \cdots & \alpha_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \alpha_{nn} \end{bmatrix} \quad (3.12)$$

İlgili eşitlikler kullanılarak, $Au = b$ genel eşitliği $LUu = b$ dönüşümü sağlanır. Sonraki aşamada ise $Uu = v$ ve $Lv = b$ temsil edilebilir. Bu dönüşüm sayesinde iki farklı üçgen matrise sahip doğrusal sistemler elde edilir ve bu sistemler geri ve ileri yönlü yer değiştirme ile çözülebilir. Eşitlik 3.13’te v_i parametresinin çözümü verilmiştir.

$$v_1 = \frac{b_1}{\beta_{11}}, \quad v_i = \frac{1}{\beta_{11}}(b_i - \sum_{j=1}^{i-1} \beta_{ij} v_j), \quad i = 2, 3, \dots, n \quad (3.13)$$

Eşitlik 3.14’de ise u_i parametresinin çözümü verilmiştir.

$$u_n = \frac{v_n}{\alpha_{nn}}, \quad u_i = \frac{1}{\alpha_{11}}(v_i - \sum_{j=i+1}^n \alpha_{ij} u_j), \quad i = n-1, \dots, 1 \quad (3.14)$$

Ayrıca LU ayrıştırma süreci sonucu elde edilen üçgen matrislerin çeşitli özellikleri vardır. Alt üçgen matris olan L’nin determinantı diyagonal elemanlarının çarpımına eşittir. Bu özellik Eşitlik 3.15’de gösterilmiştir.

$$\det(L) = |L| = \prod_{i=1}^n \beta_{ii} = \beta_{11} \beta_{22} \dots \beta_{nn} \quad (3.15)$$

Bir diğer özellik ise üçgen matrisin özdeğerleri diyagonal elemanlardır $(\beta_{11}, \beta_{22}, \dots, \beta_{nn})$. LU ayrıştırma yönteminde yine Gauss eleme ve Gauss-Jordan eleme süreçlerinde olduğu gibi sakıncalar bulunmaktadır. Bunlar başlıca elemanların değerinin sıfıra eşit olmaması gibi bazı hesaplama problemleridir. Seyrek matris gibi büyük oranda sıfırlardan oluşan matrisler bu tarz istenmeyen durumlara neden olurlar. Bu yüzden iterasyon tabanlı nümerik analiz yöntemlerinin bu tarz parametre hesabı için kullanılması daha uygundur.

Doğrusal sistemler için $Au = b$, çözümün belirlenmesini sağlayan $u = A^{-1}b$ değerinin belirlenmesi için genellikle büyük boyutlardaki A katsayı matrisinin tersinin alınması gerekmektedir. Daha önce de belirtildiği gibi doğrudan tersinin alınmasını sağlayan sistemler büyük boyutlu ($n > 100000$) matrisler için kullanışlı değildir. Bu tarz büyük boyutlu matrisler imge işleme alanında sıklıkla kullanılmaktadır. Bu yüzden daha verimli ve hesaplama karmaşıklığı daha az olan yöntemler için çok sayıda yöntem önerilmiştir. Bu tarz iterasyon tabanlı yöntemler genel olarak optimizasyon problemlerinin çözüm sürecinin irdelenmesi veya iyi bir şekilde anlaşılabilmesi için önemlidir.

Doğrusal denklem sistemlerinin iterasyon tabanlı çözümü temel birkaç maddeye dayanır. Denklemlerin her biri yeniden ifade edilecek şekilde yeniden tanımlanır ve bir başlangıç değeri belirlenir.

Başlangıç değerleri ve fonksiyon takımı kullanılarak elde edilen değerler yeni iterasyon için başlangıç şartlarını oluşturur. Sonuç olarak kullanıcı tarafından tanımlanan bazı kısıtların gerçekleşmesi sonucu iterasyon tabanlı doğrusal optimizasyon süreci gerçekleştirilmiş olur. Fakat iterasyon tabanlı çözümler bazı sakıncalarda içermektedir. Bunlardan en önemlisi bazı durumlarda gerçek çözümün yakınsayamamasıdır. Ayrıca bazı denklem takımları için başlangıç şartları oldukça önem arz etmektedir. Son olarak, iteratif yöntemler genellikle problem ile ilgili gerçek sonuçları bulmak yerine gerçek sonuca yakın değerlere yakınsarlar.

3.1.4 Jacobi İterasyon Yöntemi

Jacobi yönteminin arkasında yatan temel fikir, katsayı matrisi A 'nın Eşitlik 3.16'da gösterildiği gibi üç farklı matris olarak ifade edilmesi ile ayrıştırılmasıdır [62].

$$A_{n \times n} = D + L + U \quad (3.16)$$

D matrisi diyagonal matrisi temsil ederken, L ve U matrisleri sırasıyla alt ve üst üçgen matrisleri temsil etmektedir. Bu bağlamda L ve U matrisleri diyagonal elemanları içermemektedir ve bu matrislere ait diyagonal elemanlar sıfıra eşittir.

Jacobi yönteminde bir matris daha gerekmektedir. Bu sayede ilgili denklem Eşitlik 2.17'de gösterildiği gibi olur.

$$Au = b \Rightarrow A = D + L + U \Rightarrow \underbrace{(D + L + U)}_{Jacobi} u = b \quad (3.17)$$

Doğrusal denklem takımının çözümü için gerekli iteratif süreç ise Eşitlik 3.18'de gösterildiği gibidir.

$$Du^{(n+1)} = b - (L + U)u^{(n)} \quad (3.18)$$

Bu eşitlik ile yeni kestirim sonucu olan $u^{(n+1)}$ Eşitlik 3.19'da gösterildiği gibi kolayca hesaplanabilir.

$$u^{(n+1)} = D^{-1}[b - (L + U)u^{(n)}], \quad n = 0, 1, \dots \quad (3.19)$$

Jacobi yönteminin başarı ile uygulanması için çeşitli koşullar gereklidir. Bunlardan birincisi Eşitlik 3.20’de gösterilen ve bazı matris normları için belirtilen koşulun sağlanmasıdır. Bu koşulun sağlanması ile Jacobi yönteminin çözüme yakınsayacağı kesinleşebilir.

$$I_n - D^{-1}A = \begin{bmatrix} 0 & \frac{a_{12}}{a_{11}} & \dots & \frac{a_{1n}}{a_{11}} \\ \frac{a_{21}}{a_{22}} & 0 & \dots & \frac{a_{2n}}{a_{22}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{a_{n1}}{a_{nn}} & \frac{a_{n2}}{a_{nn}} & \dots & 0 \end{bmatrix} \Rightarrow \|I_n - D^{-1}A\|_M < 1 \quad (3.20)$$

A ($A \in R^{n \times n}$) matrisi Eşitlik 3.21 ve 3.22’de gösterilen şartları sağlaması koşulunda sırasıyla tam olarak satır ve sütun diyagonal baskın olarak tanımlanır.

$$A = D + L + U \Rightarrow |a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}|, \quad i = 1, 2, \dots, n \quad (3.21)$$

$$A = D + L + U \Rightarrow |a_{jj}| > \sum_{\substack{i=1 \\ i \neq j}}^n |a_{ij}|, \quad j = 1, 2, \dots, n \quad (3.22)$$

3.1.5. Gauss-Seidel İterasyon Yöntemi

Jacobi yönteminin hafıza alanı bakımından bazı verimsiz noktaları vardır. Bu noktada hafıza alanın önemli olduğu büyük boyutlu problemler için Gauss-Seidel iterasyon yöntemi önerilmiştir [63]. Jacobi yönteminde $u^{(n+1)}$ ve $u^{(n)}$ değerlerinin hafıza alanında saklanması ve $u_j^{(n)}$ vektörlerinin, yeni çıktı değeri için kullanılması gerekmektedir. Büyük boyutlardaki u değeri kısıtlı miktardaki hafıza alanları veya büyük boyutlardaki problemler için istenmeyen bir durumdur.

Jacobi yönteminden farklı olarak $A_{n \times n} = D + L + U$ matrisinin elde edilmesi sürecinde sadece diyagonal (D) matrisinin tersini almak yerine diyagonal matris ile beraber alt üçgen

matrisin de tersi alınır. Bu sayede aktif bir güncelleme gerçekleştirilebilir. Eşitlik 3.23'te Gauss-Seidel yönteminin gerçekleştirilmesine ilişkin genel ifade bulunmaktadır.

$$Au = b \Rightarrow A = L + U + D \Rightarrow (L + D)u^{(n+1)} = b - Uu^{(n)}$$

$$\boxed{u^{(n+1)} = (L + D)^{-1}[b - Uu^{(n)}]}$$
(2.23)

Gauss-Seidel iteratif yönteminin başarı ile çözüm vektörünü yakınsayabilmesi için Jacobi yönteminde olduğu gibi bazı kriterleri sağlamalıdır. Katsayı matrisinin tersinin bulunması ve en büyük spektral yarıçapın değerinin Jacobi yönteminde olduğu gibi 1'den küçük olması gerekmektedir.

3.1.6 Rahatlatma Yöntemi

Doğrusal denklem sistemlerinin çözümlenmesi sürecinde kullanılan Gauss-Seidel yöntemi günümüz uygulamaları için tüm kazanımlarına rağmen yavaş bir karakteristik sergilemektedir [64]. Bu bağlamda, rahatlatma (relaxation) olarak adlandırılan iteratif yöntem önerilmiştir. İlgili yöntem iki farklı aşamadan oluşmaktadır. Bu aşamalar Eşitlik 3.24'de gösterilmiştir.

$$v^{(n)} = (L + D + U)u^{(n)} - b$$

$$\boxed{u^{(n+1)} = u^{(n)} - \omega(L + D)^{-1}v^{(n)}}$$
(3.24)

Bu yöntemde diğer iteratif yöntemlerden farklı olarak, aşırı rahatlatma veya rahatlatma katsayısı olarak adlandırılan ω katsayısı bulunmaktadır. Bu değer $0 < \omega < 2$ olarak belirtilen aralıkta değişmektedir. Bu katsayı ile beraber yeniden düzenlenen denklem takımı Eşitlik 3.25'de gösterilmiştir.

$$u^{(n+1)} = (1 - \omega)u^{(n)} + \omega\tilde{u}^{(n)}$$
(3.25)

İlgili eşitlikte $\tilde{u}^{(n)} = (L + D)^{-1}(b - Uu^{(n)})$ Gauss-Seidel yöntemi ile aynıdır. Bu yüzden rahatlatma yöntemini, ağırlık katsayısı bulunan bir Gauss-Seidel yöntemi olarak düşünebiliriz ve $\omega = 1$ olduğu durumda, rahatlatma yöntemi ile Gauss-Seidel yöntemi aynıdır. Bu yöntemde farklılık yaratan asıl kısım rahatlatma katsayısının (ω) değişimidir.

Rahatlatma katsayısının düşük değeri ($0 < \omega < 1$), eksik dinleme kavramına yani gerçek sonuçlara geç yakınsamasına neden olur iken yüksek rahatlatma katsayısı $1 < \omega < 2$ aşırı

rahatlamaya yani sonucun hızlı yakınsamasını sağlar. Bunlardan farklı olarak, rahatlama katsayısının belirtilen aralık olan ($0 < \omega < 2$) değerinden farklı olması sonucu $w < 0$ veya $w > 2$ sonucunda, önerilen yöntem teorik olarak gerçek değerlere yakınsayamayacaktır.

3.2. Doğrusal Olmayan Denklemlerin Nümerik Çözümlerine İlişkin Yöntemler

Önceki bölümde doğrusal denklem takımlarının çözümlenmesine ilişkin çeşitli yöntemler hakkında bilgi verilmiştir. Bu bölümde ise doğrusal olmayan denklemlerin nümerik çözümünü sağlayan yöntemler hakkında bilgi verilecektir. Daha önce belirtildiği gibi mühendislik problemlerinin büyük bir kısmı doğrusal değildir ve genel olarak çözüm sürecinde bazı zorluklar içermektedir. Bunlardan bazıları ilgili problemlerin konveks olmaması veya teknik eşsiz çözümünün bulunmamasıdır. Ayrıca her zaman herhangi bir fonksiyonun optimal çözümünün bulunması oldukça maliyetli bazen de imkansız bir iştir. Bu yüzden kullanıcı tarafından daha önce belirlenmiş bazı kriterlerin sağlanması çözüm için yeterli sayılmaktadır. Bu bağlamda doğrusal olmayan denklem takımları tek değişkenli veya çok (n) değişkenli olacak şekilde iki farklı başlık altında incelenebilir.

Doğrusal olmayan denklem takımlarının nümerik çözümünü ise polinom denklem ve aşkın (transandantal) denklem adı verilen iki başlık altında irdelenebilir. Polinom denklemleri, x değişkeninin kuvvetlerini içeren sınırlı sayıda terimin toplamı şeklinde ifade edilebilir.

Eşitlik 3.26'da n 'nci dereceden örnek bir polinom denklem gösterilmiştir. Polinom denklemlerinin sınıfı ayrıca denklemleri de içermektedir.

$$F(x) = W_n(x) = x^n + a_1x^{n-1} + a_2x^{n-2} + a_3x^{n-3} + \dots + a_{n-1}x + a_n = 0 \quad (3.26)$$

İyi bilinen ve en temel durum olan ikinci dereceden denklem Eşitlik 3.27'de gösterilmiştir.

$$ax^2 + bx + c = 0, \quad a \neq 0 \quad (3.27)$$

İlgili eşitliğe ait kökler Eşitlik 3.28'de gösterildiği gibi kolayca hesaplanabilmektedir.

$$\Delta = b^2 - 4ac, \quad x_1 = \frac{-b + \sqrt{\Delta}}{2a}, \quad x_2 = \frac{-b - \sqrt{\Delta}}{2a}, \quad (3.28)$$

Bu çok bilinen eşitlik Cardan'ın formülü olarak bilinir. Böylece 2.dereceden bir denklemin kökleri oldukça kolay bir şekilde hesaplanabilir. Fakat yüksek dereceli denklemlerde ($n \geq 4$) bu tarz köklerin analitik olarak bulunması oldukça zorlayıcıdır. Bundan dolayı yüksek dereceli polinomların köklerinin belirlenmesinde kullanılan nümerik yöntemler devam eden kısımlarda sunulacaktır.

3.2.1. Lin Yöntemi

Lin yöntemi, polinom tabanlı denklemlerin çözümünde sıklıkla kullanılan bir yöntemdir. Eşitlik 3.29'da, n 'nci dereceden temel polinom denklemlerinin bu yöntem için uyarlanmış hali gösterilmiştir [56].

$$(x^2 + px + q)(x^{n-2} + b_1x^{n-3} + b_2x^{n-4} + \dots + b_{n-3} + b_{n-2}) + Rx + S = 0 \quad (3.29)$$

İlgili eşitlikte, $Rx+S$ terimi sıfır olması istenilen kalan terimlerdir. Sıfır kalan olması orijinal polinomun kolayca karesel form olan $(x^2 + px + q)$ polinomuna bölünebileceğini göstermektedir. Eşitlik 3.26 göz önüne alınarak, Eşitlik 3.29'da gösterilen katsayılar ile orijinal polinom arasında Eşitlik 3.30'da gösterildiği gibi bir ilişki vardır.

$$\begin{aligned} b_1 &= a_1 - p \\ b_2 &= a_2 - pb_1 - q \\ &\vdots \\ b_i &= a_i - pb_{i-1} - qb_{i-2} \\ &\vdots \\ b_{n-2} &= a_{n-2} - pb_{n-3} - qb_{n-4} \\ R &= a_{n-1} - pb_{n-2} - qb_{n-3} \\ S &= a_n - qb_{n-2} \end{aligned} \quad (3.30)$$

Lin yönteminde amaç $R=0$ ve $S=0$ değerlerini elde etmektir. Bu durumun gerçekleşmesi için Eşitlik 3.31'de gösterilen şartların gerçekleşmesi gerekmektedir.

$$\begin{aligned} R &= a_{n-1} - pb_{n-2} - qb_{n-3} = 0 \\ S &= a_n - qb_{n-2} = 0 \end{aligned} \quad (3.31)$$

Eşitlik 3.32’de gösterilen güncelleme ile daha başarılı katsayılar elde edilebilir.

$$q' = \frac{a_n}{b_{n-2}}, \quad b_{n-2} \neq 0, \quad p' = \frac{a_{n-1} - q'b_{n-3}}{b_{n-2}} \quad (3.32)$$

Eşitlik 3.30’da gösterilen denklem takımları ile başlangıç değerleri kullanılarak her bir iterasyonda yeni değerler hesaplanır. $R=0$ ve $S=0$ şartlarının sağlanması veya yakınsaması durumuna kadar bu süreç tekrarlanır. İlgili şartlar sağlandığında polinom Eşitlik 3.33’de gösterilen hale gelir.

$$(x^2 + px + q)(x^{n-2} + b_1x^{n-3} + b_2x^{n-4} + \dots + b_{n-3} + b_{n-2}) = (x^2 + px + q)W_{n-2}(x) = 0 \quad (3.33)$$

Karesel denklemin kökleri, $x_1 = c + jd$ ve $x_2 = x_1^*$ olarak belirlenir ve $W_{n-2}(x) = 0$ eşitliğini sağlayan ardışık kökler belirlenir. Lin yönteminde $R=0$ ve $S=0$ olması şartı nedeniyle yavaş bir yakınsama söz konusudur. Ayrıca başlangıç değerlerinin seçimi ile yöntemin başarımı arasında oldukça kuvvetli bir ilişki vardır. Bu tarz olumsuz özellikler Bairstow yöntemiyle ortadan kaldırılmıştır.

3.2.2. Bairstow Yöntemi

Bairstow yöntemi, yakınsama katsayıları olan p ve q değerleri iteratif olarak Eşitlik 3.34 ve 3.35’de gösterildiği gibi Newton yöntemi yardımı ile hesaplanmaktadır. İteratif hesaplamalar sonucunda Lin yönteminde olduğu gibi polinoma ait kökler belirlenebilir [65].

$$J = \frac{\partial R}{\partial p} \frac{\partial S}{\partial q} - \frac{\partial S}{\partial p} \frac{\partial R}{\partial q} \neq 0 \quad (3.34)$$

$$\begin{aligned}
p^{(n+1)} &= p^{(n)} - \frac{1}{J} \left(R \frac{\partial S}{\partial q} - S \frac{\partial R}{\partial q} \right) \\
q^{(n+1)} &= q^{(n)} + \frac{1}{J} \left(R \frac{\partial S}{\partial q} - S \frac{\partial R}{\partial q} \right)
\end{aligned}
\tag{3.35}$$

$f(x)$ fonksiyonunun karakteristiğine göre cebirsel veya aşkın olarak tanımlanabilir. Eğer $f(x)$ fonksiyonu polinom veya rasyonel bir fonksiyona formuna sahip ise cebirsel sınıfa ait bir fonksiyon olarak tanımlanır. $f(x)$ fonksiyonu, üstel, logaritmik, trigonometrik veya bunların birleşimi şeklinde ise aşkın denklem olarak tanımlanır. Sadece bazı basit aşkın denklemler doğrudan ve kolayca hesaplanabilir. Dolayısıyla, aşkın denklemlerin çoğunluğunu oluşturan, çeşitli bilimsel veya teknik problemler doğrudan yöntemler ile çözülemez. Bu yüzden literatürde çeşitli iteratif yöntemler bulunmaktadır. Bunlardan bazıları, aralığı ikiye bölme yöntemi, secant (kiriş) yöntemi, Newton-Rapson ve sabit nokta iterasyon yöntemi şeklindedir.

3.2.3. İkiye Bölme Yöntemi

Bu yöntemlerden en basiti ve temeli ikiye bölme yöntemidir. İkiye bölme yöntemi(binary search veya bisection), belirli bir aralıkta verilen fonksiyonu, maksimum veya minimum yapan değerleri bulmaya yarayan temel bir yöntemdir [56]. Sabit nokta iterasyonundan farkı iki adet başlangıç parametresine ihtiyaç vardır ve çözüm uzayı bu iki noktanın arasında olmalıdır. Sezgisel optimizasyon yöntemleri ve kısıt koşullarının fonksiyonların çözümüne etkisini irdelemek açısından oldukça başarılıdır. İkiye bölme yönteminin gerçekleştirilmesi şu şekildedir [66].

- İlk aşamada aralık değerleri olan $f(x)$ fonksiyonu üzerinden tanımlı $[a,b]$ değerler seçilmelidir.
- Seçilen aralık değerleri $f(a_0)f(b_0) \leq 0$ şartını sağlamalıdır. Bu bağlamda bu aralıkta bir sıfır geçişinin bulunduğu belirlenir.
- Sonraki aşamada $x_1 = \frac{a+b}{2}$ olacak şekilde x_1 değeri belirlenir.
- Eşitlik 2.36'da gösterilen şartlara göre yeni değer güncellenir ve aralık daraltılır.

$$\begin{aligned} f(x_1)f(a) < 0 &\Rightarrow [a, x_1] \Rightarrow x_2 = \frac{x_1 + a}{2} \\ f(x_1)f(b) < 0 &\Rightarrow [x_1, b] \Rightarrow x_2 = \frac{x_1 + b}{2} \end{aligned} \quad (3.36)$$

- İlgili eşitlik ile aralığa ait yeni değerler belirlenir ve bu süreç $|x_{n+1} - x_n| < \xi$ kriteri sağlanana ve kullanıcı tarafından belirlenmiş maksimum iterasyon sayısına kadar devam eder.

İkiye bölme yöntemi basit bir yöntem olmasına rağmen aralığa ait değerlerin başarı ile seçilmesi sonucunda her zaman yakınsamaktadır. Ayrıca her iterasyon sonucunda hata oranı gözlemlenebilir. Ayrıca türev içermemesi nedeniyle basit hesaplamalar için oldukça avantajlıdır. Dezavantajları ise yakınsama süresinin uzun olması ve işaret geçişi olmayan ve tekillik içeren fonksiyonlarda başarısız olmasıdır.

3.2.4. Sabit Nokta İterasyon Yöntemi

İlgili yöntemde bir adet değişkene sahip $f(x)=0$ olarak tanımlanmış bir doğrusal olmayan denklem tanımlıdır. Sabit nokta iterasyon yönteminin gerçekleştirilmesi süreci aşağıda gösterildiği gibidir [67].

- $f(x)$ fonksiyonu, $x=g(x)$ şeklinde bir dönüşüm yapılır ve $g(x)$ fonksiyonu iterasyon fonksiyonu olarak tanımlanmaktadır.
- x_0 olacak şekilde herhangi bir başlangıç koşulu belirlenir. Bu noktada $x_0 \approx x_g$, x_g gerçek kökün değerini temsil etmektedir.
- İteratif bir şekilde, $g(x)$ fonksiyonunun değeri yeni başlangıç değeri olacak şekilde hesaplamaya devam edilir $x_{n+1} = g(x_n)$, $n = 0, 1, 2, \dots$
- Bu süreç $|x_{n+1} - x_n| < \varepsilon$ kriteri sağlanana ve kullanıcı tarafından belirlenmiş maksimum iterasyon sayısına kadar devam eder.

Sabit nokta iterasyonu basit bir yöntem olmakla birlikte çeşitli avantajlar da içerir. Bunlar biri, analizi ve uygulaması çok kolaydır. Bir diğeri ise iyi bir kök bulma yönteminin seçilimi

hakkında bilgi verir. Dezantaj olarak doğru bir hesaplama için fonksiyonun türevlenebilir olması ve $|g'(x_0)| < 1$ şartını sağlıyor olması gerekmektedir.

3.2.5. Newton-Rapson Yöntemi

Newton-Rapson yöntemi literatürde Newton yöntemi olarak da bilinmektedir. Kökleri belirlemek için doğrusal bir yaklaşım kullanır. Bu doğrusal yaklaşım her bir iterasyon da $f(x)$ değerinin x_n değeri üzerinden oluşturulan tanjant çizgileri ile verilir [68]. $f(x)$ fonksiyonuna Newton-Rapson yönteminin uygulanabilmesi için çeşitli şartlar vardır. Bunlar, $f(x)$ 'in sürekli ve türevlenebilir olmasıdır. İlgili yöntemde birinci dereceden Taylor serisi açılımından faydalanılır [29]. $f(x) = 0$ denklemini çözmek yerine Eşitlik 3.37'de gösterilen doğrusal denklem çözülmektedir.

$$\begin{aligned} f(x) &\approx f(x_n) + f'(x_n)(x - x_n) \\ f(x_n) + f'(x_n)(x - x_n) &= 0 \end{aligned} \quad (3.37)$$

Eşitlik 3.38'de yeni tahmin değerinin (x_n) hesaplanmasını sağlayan denklem gösterilmektedir.

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad f'(x_n) \neq 0 \quad (3.38)$$

Bu tahmin süreci, $|f(x_n)| < \varepsilon_1$ veya adım aralığının belirlenen hata kriterinden $|x_{n+1} - x_n| < \varepsilon_2$ daha az olması sürecine kadar devam eder. Newton-Rapson yöntemi kök değerlerine kısa sürede yakınsamaktadır ve bu nedenle literatürde sıklıkla kullanılmaktadır. Fakat türev ihtiva etmesi ve hassas olmaması gibi çeşitli dezavantajlara da sahiptir.

3.2.6. Kiriş Yöntemi

Kiriş (Secant) yöntemi, literatürde Secant yöntemi olarak da geçmektedir [69]. Newton-Rapson yönteminin türev hesaplama sürecinden kaçınmak için üzerinde değişiklik yapılmış bir

yöntemdir [56]. Türev ifadesi yerine fark tabanlı bir yaklaştırma kullanmaktadır. Bu yaklaştırma Eşitlik 3.39'da verilmiştir.

$$f'(x_n) \approx \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}} \quad (3.39)$$

Bu yaklaştırma kullanılarak türev ifadesinden kaçınılmış ve sonuç olarak Eşitlik 3.40'da gösterilen iteratif ifade elde edilmiştir.

$$x_{n+1} = x_n - \frac{f(x_n)}{\frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}}, \quad n = 1, 2, 3, \dots \quad (3.40)$$

Sekant yöntemi, iterasyonun sonlanması süreci yönlerinden Newton-Rapson yöntemi ile benzerlik göstermektedir. İlgili yöntemin en büyük avantajı türeve ihtiyacı olmadan fark tabanlı bir türev ifadesi ile kök değerine yakınsamasıdır.

4. RASTGELE ARAMA VE OPTİMİZASYON

Rassal süreçler, optimizasyon problemlerinin çözüm süreci için oldukça önemlidir. Bu tez çalışmasına temel oluşturan metasezgisel (metaheuristic) yöntemin oluşturulması sürecinde ise rassal süreçler hayati önem arz etmektedir. Metasezgisel optimizasyon yöntemleri genel olarak çözülmesi zor ve tam olarak tanımlanmamış problemler olarak tanımlanmaktadır [70]. Bu bağlamda kesin olarak tanımlanmamış bir problem türü içi rassal süreçlerin kullanılması kaçınılmazdır. Optimizasyon yöntemlerinin bazılarında, yöntemin gerçekleştirilmesi sürecinde bir veya bir takım rassal sayı üreticinin beraber kullanılması yöntemin başarımını doğrudan etkilemektedir [71]. Bu başarıım ayrıca rassal sayı üreticilerinin kalitesi ile de ilgilidir.

Şans unsurunun kullanıldığı problem çözümlerinde her zaman bir başarısızlık olasılığı vardır. Bu durumun aksi sadece sonsuz sayıda denemeye imkan veren sistemlerde mümkündür ve bu tarz bir karakteristik sergileyen bir problem gerçek hayatta mümkün değildir. Fakat stokastik optimizasyon kavramında, diğer türlü kesin bir çözüm olmadığı için bu riskin her zaman gerekli ve yararlı olduğu öngörülmüştür. Ama gerçek bir problemin anlamlı bir çözümünde, stokastik ve deterministik süreçler bir arada bulunmalıdır. Bu yüzden çözüm uzayında bazı kısıtlara gidilmelidir ve bu kısıtların anlamlı ve problemlere göre karakteristik değiştirebilmesi gerekmektedir. Optimizasyon yönteminin uyarlamalı olması optimizasyon problemlerinin çözüm başarımını doğrudan etkilemektedir [72]. Çoğu durumda rassal sayıların dağılımı Gauss veya tekdüze dağılım karakteristiği gösteren sayılar kullanılarak gerçekleştirilir. Fakat daha kompleks problemlerde daha karmaşık rassal teknikler olan rassal yürüyüş veya Levy uçuşu gibi teknikler kullanılır.

4.1. Rassal Süreçler ve Dağılımlar

Optimizasyon yöntemlerinin rassallığı genellikle sözde rasgele sayı üreticileri yardımı ile gerçekleştirilir. Bu süreç sonucu elde edilen rassal değişkenlerin dağılımı ise Gauss dağılım, güç yasası dağılımı veya Levy dağılımı şeklindedir. Rassal değişkenlerin değerleri sürekli zamanlı değerlerdir [73]. Fakat yoldaki yayaların sayısı veya araçların sayısı gibi belirli durumlarda ayrık zamanlı değerler alabilirler. Belirli bir noktadaki gürültü gibi rassal bir süreç belirli bir aralıktaki herhangi bir gerçek değeri alabilir ve sürekli zamanlı rassal değişkenler

olarak tanımlanır. Rassal süreçler olayları gerçek sayılara eşleyen bir süreç olarak tanımlanabilir ve bu haritalamanın etki alanı, örnek alanı(sample space) olarak adlandırılır.

Eşitlik 4.1’de normal veya Gauss dağılıma ait bir olasılık yoğunluk fonksiyonunun tanımı ve özellikleri verilmiştir.

$$F(x) = P(a \leq x \leq b) = \int_a^b f(x)dx \geq 0, \int_{-\infty}^{+\infty} f(x)dx = 1, \quad (4.1)$$

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Bir diğer dağılım türü ise Levy dağılımı olarak adlandırılan dağılım türüdür. Bu dağılım türü genellikle negatif olmayan ve kısa sürede hızlı bir şekilde değişebilen belirli süreçleri modellemekte önemlidir. Levy dağılımı, ağır kuyruk (heavy tail) olarak adlandırılan bir yapı taşımaktadır ve simetrik bir dağılıma sahip değildir. Bu iki özelliği ile Levy dağılımı Gauss dağılımından farklıdır [74].

Levy dağılımı, hisse senetinin değişimini modellemek için kullanılabilir. Herhangi bir hisse senedi negatif olarak tanımlanabilir ve kısa sürede büyük oranda değişkenlik gösterebilir [75].

Levy dağılımına ait olasılık yoğunluk fonksiyonu, Eşitlik 4.2’de verilmiştir.

$$f(x) = \sqrt{\frac{c}{2\pi}} \frac{e^{-0.5(c/x)}}{x^{3/2}}, c > 0 \quad (4.2)$$

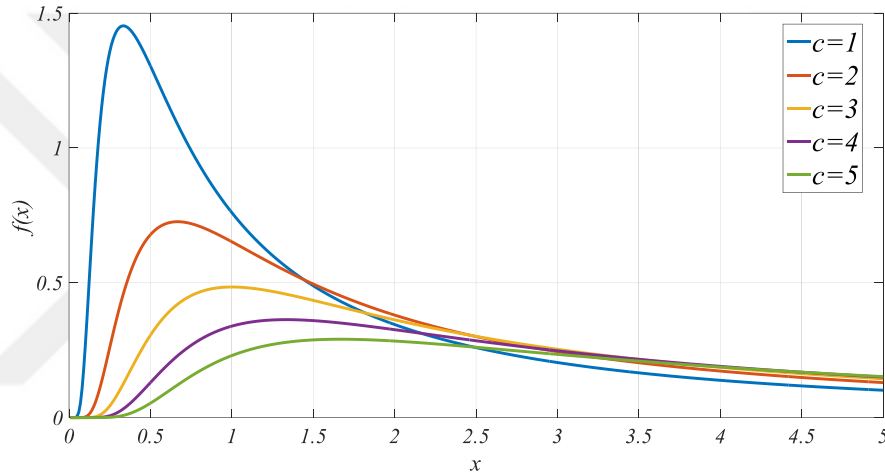
Levy dağılımı metasezgisel yöntemlerin çözüm sürecinde sıklıkla kullanılmıştır. Levy dağılımı, benzer ve bağımsız rassal değişken N in toplamı olarak tanımlanır ve Fourier dönüşümü Eşitlik 4.3’de gösterildiği gibidir.

$$F_N(k) = \exp\left[-N|k|^\beta\right] \quad (4.3)$$

Levy dağılımının doğrudan elde edilmesi kolay değildir. Eşitlik 4.4’de Levy dağılımının integral formundaki gösterimi bulunmaktadır.

$$L(s) = \frac{1}{\pi} \int_0^{\infty} \cos(\tau s) e^{-\alpha \tau^{\beta}} d\tau, (0 < \beta \leq 2) \quad (4.4)$$

$L(s)$ fonksiyonu, β indeksi ile Levy dağılımı olarak tanımlanır. Çoğu uygulamada $\alpha = 1$ olarak alınır. β değerine göre iki farklı olduğu özel durum vardır. $\beta = 1$ olduğu durumda Eşitlik 4.4’de gösterilen integral Cauchy dağılımı özelliği gösterir. $\beta = 2$ olduğu durumda ise ilgili integral normal dağılım özelliği gösterir. Şekil 4.1’de Levy dağılımının farklı c değerlerine göre değişimi gösterilmiştir [76].



Şekil 4.1. Levy dağılımı

4.2. Rastgele Araştırma Yöntemleri

Rassal süreçler kullanılarak gerçekleştirilen en temel optimizasyon yöntemleri rastgele araştırma yöntemleridir. Bu yöntemlerdeki temel motivasyon rastgele arama yöntemlerinin yaklaşık çözümleri hızlı ve kolay bir şekilde bulabilme potansiyeli olmasıdır [77]. Genel olarak, küçük boyutlarda ve yüksek oranda lokal minimum içeren bir uzay için rastgele tabanlı bir araştırma yöntemi başarılı sonuçlar verebilir.

Rastgele arama yöntemlerinin teorik analiz performansı deterministik olmayan polinomlar kadar olmasına rağmen bazı problem türlerinde çok iyi performans elde edebilirler. Bütün bu rastgele arama yöntemlerinin temelinde, uygulanabilir alandaki örneklerin gerçekleşme olasılığı üzerinden gerçekleşen hesaplamalar yatmaktadır. Bu rastgele araştırma yöntemlerinden en temeli, numaralandırma veya kapsamlı aramadır.

4.3. Rastgele Yürüyüş veya Arama Yöntemleri

Rastgele arama veya rastgele yürüyüş kara kutu (black box) olarak adlandırılan genel optimizasyon yönteminde sıklıkla kullanılmaktadır. Bu noktada kara kutu karakteristiği bilinmeyen veya kestirilmesi zor olan optimizasyon problemlerini temsil etmektedir [78]. Bu tarz problemlerin analitik analizi bazı durumlarda çok zor veya imkansızdır. Bu belirsizlik, metasezgisel yöntemlerin geliştirilmesine ve sıklıkla uygulanmasına yol açmıştır.

4.3.1. Tekrarsız Rastgele Arama

Rastgele arama yönteminin temelini oluşturan tekrarsız rastgele arama yönteminin uygulaması aşağıda gösterildiği gibi gerçekleştirilmektedir:

- Başlangıç için rastgele bir x_0 değeri seçilir. Bu noktada her bir x_0 değerinin gerçekleşme olasılığı birbirine eşittir.
- Rastgele seçilen değerler yardımı ile fonksiyonun o noktadaki değeri hesaplanır.
- Eğer $f(x_{n+1}) < f(x_n) \Rightarrow x_n$ ile x_{n+1} yer değiştirir.
- Bu süreç kullanıcı tarafından belirlenen şartlar sağlanana kadar tekrarlanır. Örneğin, maksimum iterasyon sayısı sağlanıncaya kadar tekrar eder.

4.3.2. Sıralı Rastgele Arama

Bu tarz rassal arama yöntemlerinin bir diğeri olan *sıralı rastgele arama* yönteminin gerçekleştirilmesi de aşağıda gösterildiği gibidir. Normal tekrarsız arama yönteminden farklı olarak bu yöntemde hesaplanacak değerler rastgele olarak değil de tekrarlama olmadan fakat daha önce tanımlanmış bir sıralamada hesaplanır [78]. Sıralı Rastgele arama yönteminin gerçekleştirilmesi aşağıdaki gibidir:

- Başlangıç için rastgele bir x_0 değeri seçilir. Bu noktada her bir x_0 değerinin gerçekleşme olasılığı bir birine eşittir.
- Aday nokta olan x_{n+1} belirli bir üretici yardımı ile belirli bir sıraya göre oluşturularak yeni değer hesaplanır.
- Eğer $f(x_{n+1}) < f(x_n) \Rightarrow x_n$ ile x_{n+1} yer değiştirir.
- Bu süreç kullanıcı tarafından belirlenen şartlar sağlanana kadar. Örnek olarak, maksimum iterasyon sayısı sağlanana kadar tekrar eder.

4.3.3. İki Aşamalı Yöntemler

Bir diğer genel optimizasyon yöntemi ise iki aşamalı yöntemlerdir. Çoğu metasezgisel optimizasyon yöntemi iki aşamalı veya fazlı (two-phase) optimizasyon yaklaşımını kullanmaktadır. Bu tez çalışması kapsamında gerçekleştirilecek olan yöntem de iki aşamalı bir optimizasyon yaklaşımının ürünüdür. Bu tarz yaklaşımlarda örnekleme gerçekleştirildiğinde genel bir aşama tüm hesaplanabilir bölgeye yayılır. Yerel aşamada ise kısıtlanmış veya odaklanmış bir kısım üzerinden örnekleme gerçekleştirilir. Bu tarz iki aşamalı yöntemlerde genelden yerele doğru bir yayılım söz konusudur [78].

Bu sayede genel ve yerel aşamalı olmak üzere çok aşamalı bir yöntem olarak gerçekleştirilir. Genel aşamada, başlangıç noktalarına ait değerler genellikle muntazam dağılmış bir şekilde hesaplanır. Daha sonraki aşamada bu elde edilen değerler yerel faza ait başlangıç değerlerini oluşturur. Sonraki aşamada arama uzayı daraltılarak deterministik gradyent arama türü yöntemler yardımı ile yerel aşamaya ait hesaplamalar gerçekleştirilir. İki aşamalı stokastik genel optimizasyon yöntemlerinin gerçekleştirilmesi temel hatları ile aşağıdaki gibidir:

- Başlangıç için rastgele bir x_0 değeri seçilir. Bu noktada her bir x_0 değerinin gerçekleşme olasılığı bir birine eşittir.
- Genel aşamada, örnekleme dağılımı S 'e göre arama uzayı $X_n \in S$ oluşturulur.
- Yerel aşamada, aday noktalar $V_n \in S$ yerel örnekleme dağılımına veya yerel azalım yöntemlerine göre oluşturulur.
- Şimdiye kadar bulunan en iyi çözümün yanı sıra ilgilenen alandaki diğer parametreler güncellenir.
- Bu süreç kullanıcı tarafından belirlenen şartlar sağlanana kadar devam eder. Örnek olarak, maksimum iterasyon sayısı veya bağıl hata şartı sağlanana kadar tekrar eder.

Bu noktada üreticinin karakteristiği ve sıralama yapısı fonksiyonun genel optimal değerinin belirlenmesinde ya da o değere yakın değerlere yakınsamasında oldukça önemlidir.

Yıllar içinde geliştirilen metasezgisel optimizasyon yöntemlerinin başarımı genel olarak rastgele araştırma yöntemlerine göre daha yüksektir. Ayrıca önerilen yöntemlerde her zaman bir gelişim öne sürülür ancak bu başarımların artışı sadece belirli bir sınıf problemi kapsamaktadır.

Bu bağlamda iki problem ortaya çıkmaktadır. Birincisi, problemlerin sınıfının doğru olarak tanımlanmasıdır. Ancak bu kavramın tam olarak gerçekleştirilmesi imkansızdır. Bir diğer sorun ise metasezgisel yöntemler genel olarak kara kutu olarak adlandırılan problem türlerine uygulanır. Bu noktada kullanıcının problemin karakteristiği veya sınıfı hakkında yeterli bilgi sahibi olmadığı gözlemlenebilir ve uygun olmayan algoritmaların kullanılması sonucu başarısız sonuçların elde edilmesiyle sonuçlanabilir.

4.4. Rastgele Sayı Üreteçleri

Metasezgisel yöntemlerin temelini oluşturan stokastik problem çözücü yöntemler genel olarak sıralı rastgele sayı üreten yöntemler kullanmaktadır. Bu sayede deterministik olmayan problemlerin çözümü için yeni ve başarılı yollar bulabilirler [79]. Bu tarz bir rassalığın kullanılmasının bir diğer nedeni herhangi zorlayıcı bir problem ile karşılaşılması durumunda rastgele olarak gerçekleştirilen bu çözümün gerçek çözüme yakınsamanın tek yolu olmasıdır. Bu noktada bazen rassalık kullanımı bir çeşit entelektüel tembellikten veya daha temelde mevcut olan teorik araçların yetersizliğinden kaynaklanabilir. Bu tarz yöntemlerin tanımlanması sürecinde başlangıç koşulları başlangıç şartları olarak belirlenen bazı parametreler (x_0) vardır. Başlangıç şartları genellikle rastgele olacak şekilde seçilmektedir ve arama uzayının her noktasında bulunabilirler. Başlangıç şartlarını ve optimizasyon yönteminin başarımını doğrudan etkileyen bu rassal seçim bir rastgele sayı üretici vasıtasıyla gerçekleştirilir.

Rastgele sayı üreteçleri genel olarak ikiye ayrılırlar.

- Sözde rastgele sayı üreteçleri (Pseudo-Random Number Generators)
- Gerçek rastgele sayı üreteçleri (True Random Number Generators)

Sözde sayı üreteçleri, tam olarak rastgele olarak adlandırılmayan ve elde edilmesi için çeşitli matematiksel formüller veya önceden hesaplanmış değerler kullanan yapılardır. Sözde sayı üreteçleri verimli ve deterministik bir karakteristiğe sahiptir [80]. Verimlilik, kısa bir sürede veya hesaplama hızına bağlı olarak yeni sayıların üretilmesine imkan sağlaması anlamına gelmektedir.

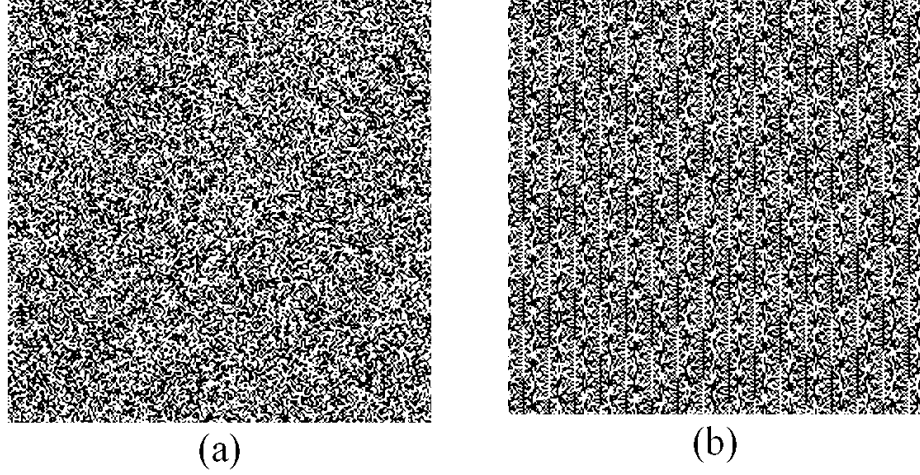
Deterministik olması, dizideki başlangıç noktası biliniyorsa bu sayısı dizisinin birebir aynısının istenildiği zaman üretilebileceği anlamına gelmektedir. Verimlilik kavramı eğer uygulamanızın çok sayıda hesaplama ihtiyacı varsa çok yararlı bir özelliktir. Deterministik olması ise yönteminizin ilerleyen aşamalarda tekrar kullanılabilmesine imkan sağlar. Sözde sayı üreteçleri aynı zamanda periyodiklerdir ve belirli bir süre sonra oluşturduğu sayılar kendini tekrar eder. Fakat günümüz sözde sayı üreteçleri için bu süreç oldukça uzundur ve pratikte göz ardı edilebilir. Bu özellikleri nedeniyle, sözde sayı üreteçleri birçok sayının ihtiyaç duyulduğu ve aynı sıralamanın kolayca oluşturulması gereken uygulamalar için uygundur.

Simülasyon ve modellemeler bu tarz uygulamalara örnektir. Ayrıca metasezgisel optimizasyon yöntemlerinin gerçekleştirilmesi sürecinde ihtiyaç duyulan rastgele sayıların üretilmesinde sözde sayı üreteçleri kullanılır. Fakat sözde sayı üreteçleri öngörülebilir olmaları nedeniyle, veri şifreleme gibi hayati önem arz eden konularda kullanılmaları uygun değildir. Bu tarz yüksek güvenlik istenen durumlarda, gerçek sayı üreteçlerinin kullanılması uygundur [81]. Gerçek sayı üreteçleri, fiziksel olaylardan rassallığı ayrıştırarak bu bilgilerin bilgisayar ortamına aktarılmasını sağlar.

Fiziksel olaylardan rastgelelik bilgisinin ayrıştırılması çok kolaydır. Fakat bu noktada belirtilen fiziksel olayın gerçekten rastgelelik içerip içermediği iyi analiz edilmelidir. Fiziksel ortamdan alınan bir beyaz gürültü (white noise) gibi sinyal türleri doğru tanımlanmaları ile yüksek başarımlı bir gerçek sayı üretici elde edilebilir [82].

Gerçek sayı üreteçlerinin gerçekleştirilmesi veya kayıt altına alınması oldukça zordur. Ayrıca bu tarz sayı üreteçlerinin belirli bir sürede üretebileceği sayı miktarı, dış ortamında bulunan ve gerçek sayı üreticisine kaynak olan verilerin toplanıp örneklenmesi süreci ile sınırlıdır.

Şekil 4.2’de gerçek ve sözde sayı üreteçlerinin karşılaştırmasını içeren bir ikilik tabandaki çıktılara ait resimlemeler gösterilmektedir. Verilenlerde görüldüğü üzere gerçek rastgele sayı üreticinde herhangi bir belirli örüntü veya desen bulunmazken, sözde sayı üreticinde birbirini tekrar eden belirgin bir örüntü bulunmaktadır. Bu bağlamda sözde rastgele sayı üreteçlerinin güvenlik gibi uygulamalar için uygun olmadığı açık bir şekilde gösterilmektedir.



Şekil 4.2. (a) Gerçek rastgele sayı üretici (b) Sözde rastgele sayı üretici

Tablo 4.1’de gerçek rastgele sayı üreticiler ile sözde rastgele sayı üreticilerin karşılaştırılması bulunmaktadır.

Tablo 4.1. Sözde ve Gerçek Rastgele Sayı Üreteçlerinin Karşılaştırılması

<i>Karakteristik</i>	<i>Sözde Rastgele Sayı Üreteçleri</i>	<i>Gerçek Rastgele Sayı Üreteçleri</i>
Verim	Mükemmel	Zayıf
<i>Karakteristiği</i>	Deterministik	Stokastik
<i>Periyodiklik</i>	Periyodik	Periyodik Değil

Tablo 4.2’de ise farklı rastgele sayı üreteçlerinin hangi uygulama alanına uygun olduğu gösterilmektedir.

Tablo 4.2. Farklı rastgele sayı üreteçlerinin kullanım alanlarının karşılaştırılması

<i>Kullanım Alanı</i>	<i>Uygun olan Rastgele Sayı Üretici</i>
Piyango ve çekilişler	Gerçek Rastgele Sayı Üretici
Oyunlar	Gerçek Rastgele Sayı Üretici
Rastgele Örnekleme	Gerçek Rastgele Sayı Üretici
Simülasyon ve Modelleme	Sözde Rastgele Sayı Üretici
Güvenlik ve Şifreleme	Gerçek Rastgele Sayı Üretici
Güzel Sanatlar	Gerçek ve Sözde Rastgele Sayı Üretici

5. METASEZGİSEL OPTİMİZASYON YÖNTEMLERİ

Metasezgisel optimizasyon yöntemleri doğadan esinlenmiş optimizasyon yöntemlerine verilen genel bir isimdir. Metasezgisel yöntemler, yerel iyileştirme süreçleri ile daha üst düzey çözümler arasındaki etkileşimi düzenleyen algoritmalarlardır. Ayrıca çözüm alanında sağlam bir araştırma yeteneği sağlarken, yerel optimum noktalarında bulunan tıkanmaların önüne geçmeyi sağlayan mekanizmalar içerirler.

Metasezgisel yöntemler, sezgisel yöntemlerin farklı türdeki problemlere uyarlanmasını hedefleyen optimizasyon yöntemlerine verilen genel bir isimdir ve ufak orandaki yenilemelerle farklı türdeki optimizasyon problemlerine adapte olan yapılardır [78]. Kara kutu olarak adlandırılan problemlerin çözüm sürecinde sıklıkla kullanılmaktadır.

Metasezgisel optimizasyon yöntemleri deterministik olmayan yöntemlerdir ve yaklaşık çözümler sunarlar. Arama uzayında bulunan tuzak alanlarda takılı kalmamak için çeşitli bütünlük mekanizmaları kullanırlar.

Metasezgisel kavramının temel konsepti, soyut bir uzayın açıklığa kavuşturmasına izin vermesidir [83]. Bu yöntemler kimi zaman doğadaki genlerin gelişim sürecinin modellenmesi üzerinden oluşturulurken kimi zaman bazı karınca türlerinin avlanma davranışından esinlenilir. Metasezgisel yöntemler arasında en çok bilinen ve bu tarz yöntemlere temel oluşturan yaklaşım Genetik Algoritmasıdır (GA).

5.1. Genetik Algoritması

1950 ve sonrasında birçok bilim adamı evrim kavramının mühendislik alanında bir çözüm aracı olabileceğine ait ilişkin çalışmalarda bulunmuşlardır. Bütün bu sistemlerde ana fikir, aday popülasyonlar arasından doğal genetik ve seçim süreçleri üzerinden en uygun popülasyonunun bulunmasını sağlayan çözümlerin elde edilmesini sağlamaktır. 1960'lı yıllarda Rechenberg, "Evrimsel Stratejileri" adı altında uçak kanat kesitlerinin gerçek değerler vasıtasıyla optimizasyonunu sağlayan bir yöntem önermiştir. Bu süreçten sonra benzer amaçlar ile çeşitli evrimsel optimizasyon yaklaşımları bulunmuştur. Fakat, evrimsel optimizasyona modern olarak temel olan yöntem ise Genetik Algoritmasıdır [84]. Genetik algoritması (Genetic algorithm), 1960'lı yıllarda John Holland tarafından bulunmuş ve geliştirilmiştir. Evrimsel stratejileri ve evrimsel programlamanın aksine, Holland'ın amacı belirli sorunları çözmek amacıyla algoritmalar tasarlamak yerine, doğada olduğu gibi uyum fenomenini incelemek ve doğal adaptasyon sistemini bilgisayar ortamında gerçekleştirerek farklı türde problemlerin

çözümüne imkan sağlamaktır. Bu bağlamda Genetik Algoritmasını (GA), modern anlamdaki metasezgisel yöntemlerin temeli olarak tanımlanabilir. Genetik algoritması, John Holland tarafından 1975 yılında Doğal ve Yapay Sistemlerde Adaptasyon (Adaptation in Natural and Artificial Systems) adı altında biyolojik evrim sürecini, Genetik Algoritması adı altında bir uyum ve teori çerçevesinde ele almıştır [85].

Genetik algoritması bir kromozom popülasyonundan en uygun veya bir diğer deyişle en gürbüz bireyin hayatta kalması temeli üzerine oluşturulmuştur. Mutasyon, çaprazlama v.b gibi kavramlar kullanılarak herhangi bir probleme ait genel maksimum ve minimum değerin veya bu değere yaklaşık değerlerin belirlenmesini sağlar. Bu bağlamda, Genetik algoritmasının kullanım alanları bulunduğu günden bu güne artmıştır. Günümüzde, örüntü tanımadan, havacılık mühendisliği ve finans alanına birçok alanda kullanılmaktadır.

GA çok farklı tipteki amaç fonksiyonu üzerinden başarılı olmaktadır. Bu amaç fonksiyonları, doğrusal ve doğrusal olmayan aynı zamanda durağan veya durağan olmayan şeklinde de olabilir. Ayrıca, sürekli zamanlı veya ayrık zamanlı sistemlerin belirli orandaki gürültü içeren sistemlerinde de optimizasyon sürecini başarı ile gerçekleştirmektedir. Bu avantajlarının yanında GA bazı dezavantajlar içerir. Amaç veya uygunluk fonksiyonunun tanımlanması, popülasyon sayısının seçimi veya mutasyon, çaprazlama parametrelerin uygun seçimi gibi konular GA'nın dezavantajları arasındadır. Holland iki farklı optimizasyon yönteminin beraber çalıştırılarak bu parametrelerinde optimize edilmesinin bu sorunu ortadan kaldıracağını öngörmesine rağmen pratik problemlerde bu sürecin hesaplama karmaşıklığını oldukça yüksek oranda artırması nedeniyle gerçekleştirilmesi oldukça zordur. Belirtilen bu dezavantajlarına rağmen GA'sı günümüz doğrusal optimizasyon algoritmaları arasında en çok tercih edilen ve üzerinde en çok çalışmalar yapılan yöntemdir. GA kromozom yapıları üzerinden gerçekleştirilir ve kromozomlar, genlerin birleşiminden oluşur [86]. Genetik Algoritması gerçekleştirilmesi sürecinde çeşitli terminolojiler bulunmaktadır. GA'nın en temel formu, üç adet operatör içerir. Bunlar; seleksiyon (selection), çarpazlama (crossover) ve mutasyondur (mutation).

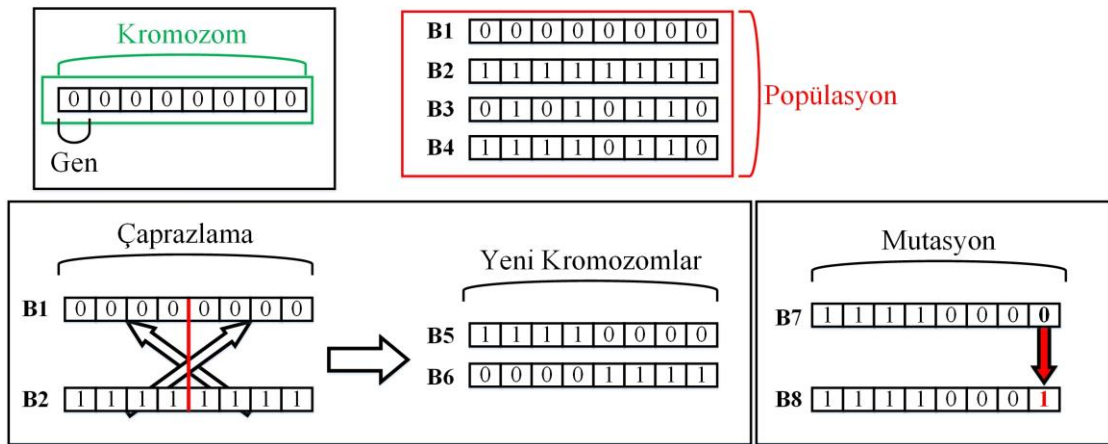
- Seleksiyon: Bu parametre popülasyon içindeki kromozomları seçer. Kromozom taklidi, çoğaltmak için seçilme olasılığını gerçekleştirecek şekilde artar.
- Çaprazlama: Bu parametreler rastgele bir yer seçer ve seçilen yerlerle beraber alt dizinleri değiştirir. Örnek olarak iki farklı birey olan 1. ve 2.bireyin çaprazlanması

sonucu yeni iki adet birey elde edilir. Eşitlik 4.1’de çaprazlama sonucu elde edilen yeni bireylerin gösterimi bulunmaktadır.

$$\underbrace{10000100}_{1.birey} \text{ ve } \underbrace{11111111}_{2.birey} \Rightarrow \text{Çaprazlama Sonucu} \Rightarrow \underbrace{10011111}_{3.birey} \text{ ve } \underbrace{11100100}_{4.birey} \quad (4.1)$$

- **Mutasyon:** Bu rastgele operatör ile her seferinde kromozomlarda bulunan bir adet bitin değeri değiştirilir. Mutasyon işlemi, genellikle kromozon yapısı üzerinde çok düşük bir olasılık değeri ile gerçekleştirilir ($p_m = 0.001$). Genellikle doğada gerçekleşen mutasyonun ilerleyen bireylerde olumlu özellikler gösterme olasılığı çok düşüktür. Bu noktada mutasyon kavramı genellikle genler üzerinde istenmeyen etkileri modellemek üzere tanımlanırken, bu algoritmada yerel minimum veya maksimum değerlerinin aşılması amacıyla kullanılmaktadır.

Şekil 5.1’de GA’na ait temel süreçlerin temsili gösterimi bulunmaktadır. Bu gösterimde kromozomlar 8 bit üzerinden kodlanmıştır. Temsil edilebilecek en büyük rakam $2^8 = 256$ şeklindedir ve örnek kodlama için karar değişkenlerinin sınırlamaları $ub_i \geq x_i \geq lb_i \Rightarrow 256 \geq x_i \geq 0, i = 1, \dots, n$ şeklinde değişir. İlgili şekilde gösterildiği gibi kromozom gruplarından oluşan yapılar, popülasyonları temsil etmektedir.

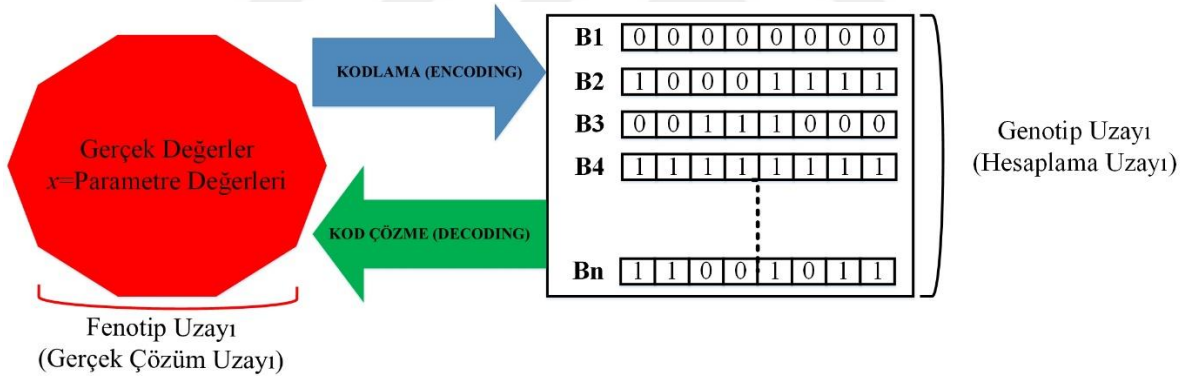


Şekil 5.1. Genetik Algoritmasına ait temel süreçlerin gösterimi

GA'nın çözüm sürecinin temelini oluşturan bir diğer süreç ise kodlama (encoding) ve kod çözme (decoding) kavramlarıdır. Eşitlik 5.2'de kodlama ve kod çözme süreçlerini gösteren dönüşüm bulunmaktadır.

$$\underbrace{10110101}_{\text{Genotip}} \Rightarrow x = 181_{\text{Fenotip}} \quad (5.2)$$

Şekil 5.2'de GA'nın gerçekleştirilmesi sürecinde kullanılan kodlama ve kod çözme kavramlarının karşılık geldiği uzaylar gösterilmiştir [85]. Şekilde görüldüğü gibi gerçek değerlerin GA'na uygun olacak şekilde kodlanması ve gerçek çözüm uzayında yeniden oluşturulması için kod çözme sürecine tabi tutulması gerekmektedir. Bu süreçte kodlamanın kaç bit değerler üzerinden oluşturulacağı sınırlar veya hesaplama kapasitesi üzerinden kullanıcı tarafından ya da hafıza alanının testi sonucu çeşitli uyarlamalı yöntemler tarafından belirlenir [87].



Şekil 5.2. GA'nda kodlama ve kod çözme

GA'nın gerçekleştirilmesi süreci aşağıda gösterildiği gibidir.

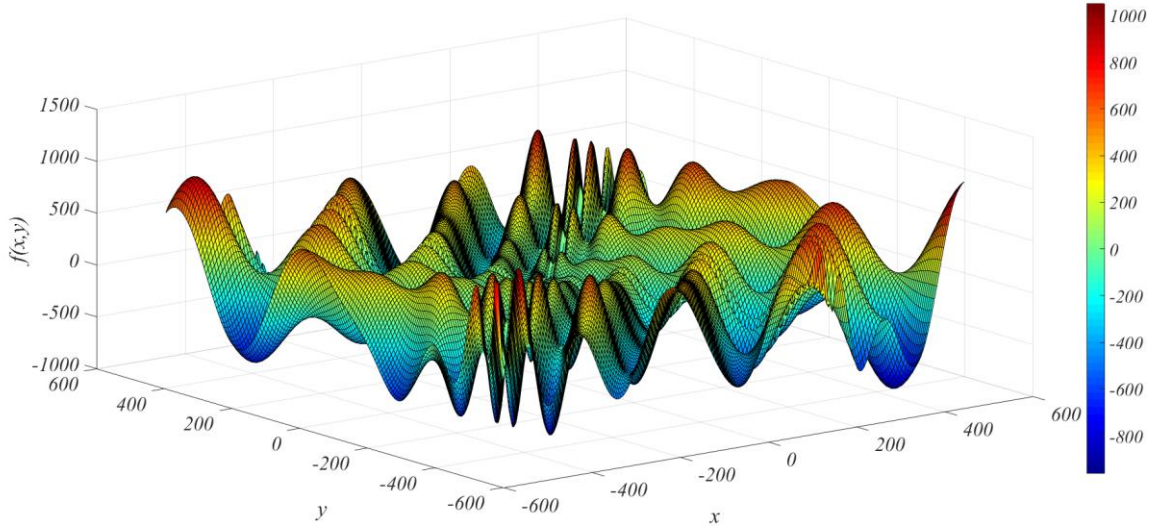
- İlk aşamada amaç veya optimize edilecek fonksiyon belirlenir.
- Başlangıç değerleri kodlanarak kromozomlara dönüştürülür. Bu aşamada aynı zamanda kısıtlar ve sınırlar belirlenir.
- Başlangıç popülasyonu oluşturulur ve Çaprazlama (p_c) ve mutasyon (p_m) olasılıkları kullanıcı tarafından belirlenir. Ayrıca maksimum jenerasyon veya iterasyon (it_{maks}) sayısı kullanıcı tarafından belirlenir.

- Çaprazlama ve mutasyon ile yeni çözüm oluşturulur. Eğer $p_c > rand$ ise çaprazlama gerçekleştirilir. Eğer $p_m > rand$ ise mutasyon gerçekleştirilir.
- Yeni çözümlerden en iyisi seçilir ve yeni jenerasyon oluşturulur. Bu sonuç yeni iterasyon için başlangıç şartlarını oluşturur.
- İstenilen kriterler veya maksimum iterasyon sayısına ulaşılan kadar bu süreç devam eder.
- Çıktıya ait genotip değerlerinin kodu çözülür ve sonuç görselleştirilir.

GA'nın gerçekleştirilmesi süreci, Eşitlik 5.3'de yumurta tutucu (egg holder) fonksiyonu üzerinden gösterilmiştir [88].

$$f(x, y) = -(y + 47) \sin\left(\sqrt{\left|y + \frac{x}{2} + 47\right|}\right) - x \sin(\sqrt{|x - (y + 47)|}) \quad (5.3)$$

İlgili fonksiyonun 3 boyutlu gösterimi Şekil 5.3'de elde edilmiştir. Fonksiyon konveks olmayıp çok sayıda yerel minimum noktası içermektedir. Bu nedenle metasezgisel bir yöntemin başarımının test edilmesi için oldukça uygun bir karakteristikleriğe sahiptir.



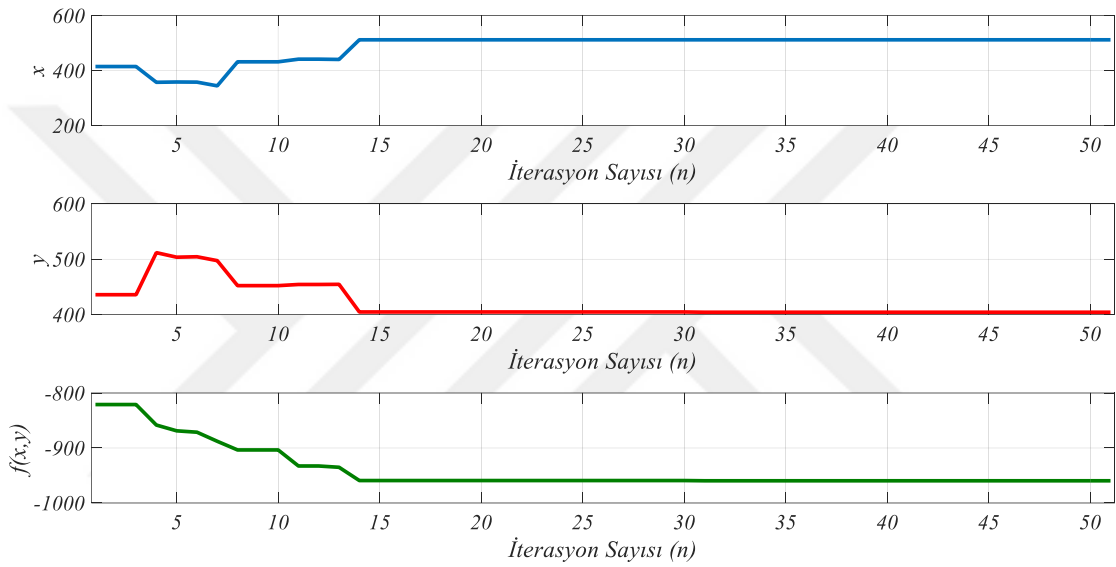
Şekil 5.3. Egg holder fonksiyonun değişimi

Literatürde egg holder olarak geçen fonksiyonun tanım aralığı ve genel minimum değeri ile bu değeri tanımlayan değişkenler, Eşitlik 5.4'de verildiği gibidir.

$$x, y \in [-512, 512], \text{ ve } f_{\min}(x_{opt}, y_{opt}) = -959.6407 \Rightarrow x_{opt} = 512 \text{ ve } y_{opt} = 404.2319 \quad (5.4)$$

GA'sı gerçekleştirilen doğrusal olmayan optimizasyon sürecine ait parametreler ise maksimum iterasyon sayısı $it_{maks} = 50$, çaprazlama olasılığı $p_c = 0.9$ ve mutasyon olasılığı $p_m = 0.1$ olarak seçilmiştir.

İlgili parametreler kullanılarak gerçekleştirilen GA optimizasyonunun, değişkenlere ve fonksiyon değerine bağlı değişimi Şekil 5.4'de gösterildiği gibidir.



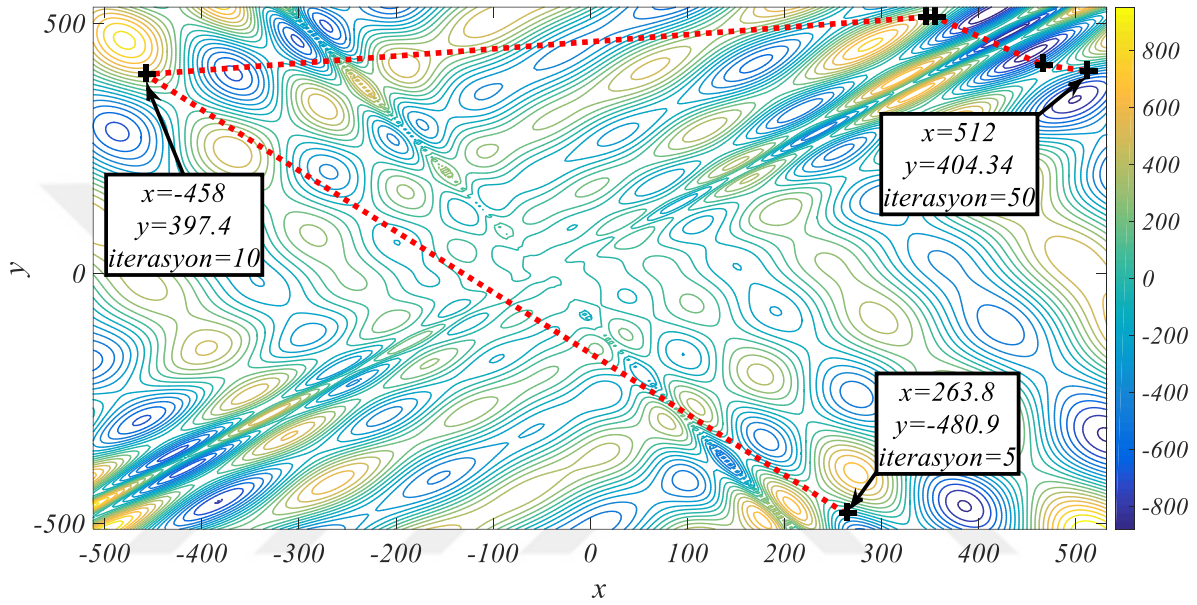
Şekil 5.4. Egg holder fonksiyonuna ait parametrelerin iterasyon sayısına bağlı değişimi

GA ile optimizasyon işlemi sonucunda Eşitlik 5.5'de gösterilen sonuçlar elde edilmiştir.

$$x, y \in [-512, 512], \text{ ve } f_{\min_{ga}}(x_{ga}, y_{ga}) = -959.6269 \Rightarrow x_{ga} = 512 \text{ ve } y_{ga} = 404.3416 \quad (5.5)$$

İlgili eşitlikte gösterildiği gibi değişken değerleri ve optimal sonuçlar yüksek başarımla elde edilmiştir. GA'sının gerçekleştirilmesi sürecinde parametre seçimi oldukça önemlidir. Bunlardan en önemlisi algoritmanın karakteristiğini belirleyen çaprazlama olasılığı p_c ve mutasyon olasılığı p_m parametreleridir. Çarpazlama olasılığının doğru seçilmemesi optimal çözüm arama sürecinin başarısız olmasına ya da karşılaştırma sonucu elde edilen yeni nesillerin tam olarak belirlenememesine neden olmaktadır.

Şekil 5.5’de, Şekil 5.3’de gösterilen test fonksiyonunun GA ile optimizasyon sürecinin eş yükselti eğrileri üzerinden gösterimi bulunmaktadır. Şekil 5.5’de görüldüğü gibi optimal çözüm noktasından uzak bir şekilde başlayan GA çözümü, ilerleyen iterasyon sayısına bağlı olarak optimal sonuca yakınsamıştır. Bu noktada, ilgili problemin başarı ile çözülmesini sağlayan temel etkenler, başlangıç şartlarının başarı ile seçimi ve GA’sına ait parametrelerin uygun bir şekilde tanımlanmasıdır.



Şekil 5.5. GA’sı sonucu bulunan optimal değerlerin fonksiyon üzerinde gösterimi

Mutasyon olasılığının düşük olması problemin türüne bağlı olarak yerel minimum noktalarında, istenmeyen takılmalara neden olurken, bu olasılığın yüksek olması rastgele arama gibi sadece rassal süreçlerden oluşan bir arama sisteminin gerçekleşmesine neden olur. Bu noktada, olasılık fonksiyonlarının gerçekleştirilmesi sürecinde rastgele sayı üreteçlerine ihtiyaç duyulmaktadır. Bunlara ek olarak literatürde bu olasılıkların problemin türüne göre uyarlamalı veya farklı şekilde kullanıldığı birçok çalışma bulunmaktadır.

5.2.Karınca Kolonisi Algoritması

Sürü davranışı kaynaklı problem çözme yöntemleri nispeten yeni sayılabilecek optimizasyon yöntemleri arasında sayılabilir. Bu noktada böceklerin sosyal davranışlarını temel alan çok sayıda yöntem bulunmaktadır. Karıncaların besin bulma süreçlerinin matematiksel modellenmesinden oluşturulan karınca kolonisi algoritması (KKA) bu yöntemler arasında en

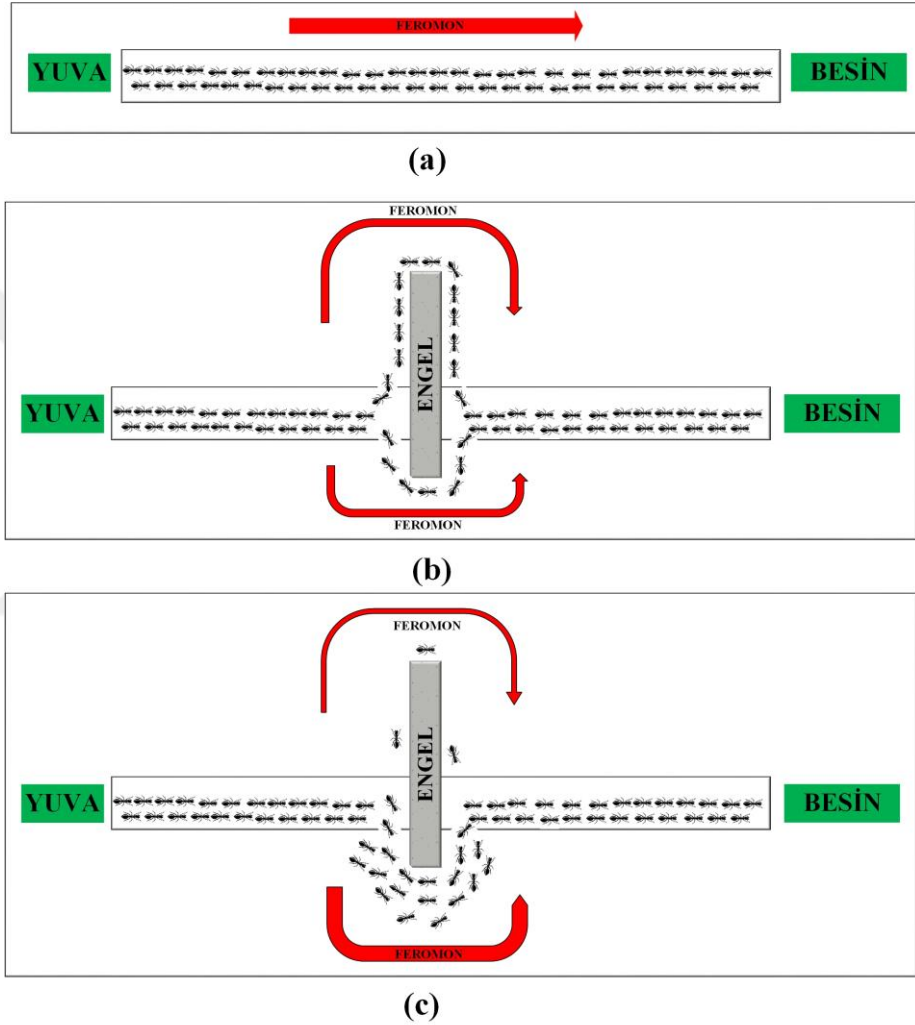
başarılarından ve en çok bilinenlerden biridir [89]. Bu algoritma, bazı karınca kolonilerin farklı mensuplarının izlemesi gereken kısa yolları belirlerken kullandığı feromon adlı kimyasalın değişimi üzerinden modellenmiştir. KKA'sı da problemleri bu yaklaşım ile çözmeyi amaçlamaktadır.

1990'lı yılların başlarında ilk KKA yöntemi Marco Dorigo tarafından 1992 yılında doktora tezi çalışması kapsamında gerçekleştirilmiştir. Karınca sistemi (ant system) olarak adlandırılan bu yaklaşım araştırmacılar tarafından çok ilgi görmüş ve kısa sürede çeşitli yenilikçi yaklaşımlar ile desteklenmiştir [90]. Fakat bu davranışın bilinmesi çok daha önceki zamanlara dayanmaktadır. 1950'li yıllarda Fransız entomolog Pierre-Paul Grasse bazı karınca türlerinin önemli uyarı olarak adlandırılan tepkiler gösterdiğini belirlemiştir. Bu uyarı sayesinde karınca kolonisi içerisindeki canlıların performanslarında değişim olduğunu gözlemlemiş ve bu farklılığın bir iletişim türü olduğunu belirlemiştir. KKA'sı da tam olarak bu feromon kimyasalının problem çözmeye etkisi üzerinden modellenmiştir. Bir yol üzerindeki feromon oranının fazla olması o yolun besine ulaşılacak en kısa yol olduğu hakkında bilgi vermektedir. Feromon salgısının yoğun olduğu yolu takip eden karıncalar besine hedef besine en kısa yoldan ulaşacaktır ve bu sayede koloninin enerji kaybı ve ulaşma süresi en aza indirgenecektir.

KKA veya sürü davranışı yöntemlerin temelinde, stigmerji (stigmergy) olarak adlandırılan kavram bulunmaktadır. Bu kavram, bir aksiyon sonucu ortamda oluşturulan izin bir sonraki eylem üzerindeki değişikliği olarak tanımlanmaktadır. Bir anlamda, dolaylı bir şekilde sağlanan koordinasyon yapısı olarak düşünebiliriz. Bir diğer deyişle, aynı ortamda bulunan iki bireyden birinin ortamda gerçekleştirdiği değişime bağlı olarak etkileşime girmektedir ve bu sayede diğer birey değişen ortama göre davranışını veya tepkisini değiştirmesi stigmerji kavramı olarak tanımlanmaktadır. Metasezgisel yöntemlerin temelinde bu kavram yatmaktadır. Bu kavram tekil olarak büyük bir değişkenlik ve başarımlar gösteremeyecek olan canlıların doğada hayatta kalması veya besin bulması süreçleri üzerinde oldukça etkilidir [91].

Karınca kolonisi üzerinden detaylandırdığımızda kör olan karıncalar, oluşturulan feromon ile etkileşime girerek yönlerini değiştirirler. Ayrıca feromonlar sadece yakın bölgedeki karıncalar tarafından gözlemlenebilir. Bu sayede koloniye yerel bir bütünlük veya kararlılık sağlar. Bunlara ek olarak feromon salgısının koloni üzerindeki etkisi olarak, ne kadar fazla karınca aynı yolu izlerse aynı rota üzerinde o kadar fazla yönelim olmaktadır. Bu etki, pozitif yapılı bir geri besleme olarak düşünülebilir [92]. Bir yol ne kadar çok önceki karıncalar tarafından tercih edilirse, sonraki süreçlerde ortama yeni gelen karıncalar tarafından seçilme olasılığı artmaktadır. Bu olasılıksal değişim KKA'nın temelini oluşturmaktadır.

Şekil 5.6’da karıncaların yuvalarından besine ulaşma sürecini resimlenmiştir. İlk aşamada engelsiz yol üzerinde hareket eden karınca sürüsü, ortamda oluşan engel sonucunda bir karar vermeye ihtiyaç duyarlar. Bu yönelim kararı kısa olan yol üzerinden feromon salgısı sayesinde ortam uyarlamalı gerçekleştirilir. Kısa yol üzerinde görüldüğü gibi feromon salgısı daha yoğunur [93].



Şekil 5.6. Feromon yapısı ve karıncaların besin bulması süreci

Şekil 5.6’da görüldüğü gibi yuva ve besin arasında oluşan bir engelin karıncaların yol bulması üzerinde uyarlamalı bir etkisi vardır. Bir önceki karınca tarafından salgılanan feromonlar sayesinde yuva ile besin arasındaki en kısa mesafe bilgisi belirlenmiştir.

Gerçek karınca yapısında feromonların buharlaşmak gibi bir özelliği vardır. Fakat metasezgisel olarak uyarlanan KKA’ında bu tarz bir dezavantaj söz konusu değildir. Bununla birlikte, KKA’sını gerçekleştiren yapay karıncalar ile gerçek karıncalar arasında bazı

noktalarda benzerlikleri bazı noktalarda olumlu farklılıkları bulunmaktadır. Bu benzerlikler ve farklılıklar Tablo 5’de gösterilmiştir [78].

Tablo 5.1. Gerçek ve yapay karıncalar arasındaki benzerlikler

<i>Benzerlikler</i>	<i>Farklılıklar</i>
İşbirliği yapan bütünsel bir yapı	Problem çözümünü ayrık zamanda gerçekleştirilmesi
Feromon izleri ve <i>sitgmerji</i> kavramı	Genel hafızalı bir yapı ile önceden gerçekleşen durumların bağlı bireysel bilgiler.
Olasılık tabanlı kararlar verme ve yerel stratejiler uygulama • Problem yapısına göre verilen ön bilgiler. • Karıncalar tarafından oluşturulan bilgiler bağlı olarak yerel değişiklikler	• Yapay karıncaların tam olarak kör olmamaları • Oluşturulan feromon oranı, çözümün kalitesine ait bir fonksiyonu temsil eder. • Feromonların biriktirilmesi süreci iteratif bir şekilde gerçekleşmektedir.

KKA’nın literatürde çok farklı uygulamaları bulunmaktadır. Fakat bunların temeli Marco Dorigo ve arkadaşları tarafından gerçekleştirilen karınca sistemi (ant system) algoritmasıdır. Bu yaklaşımında her bir iterasyonda feromon değerlerinin çözümünü oluşturan tüm karıncalar tarafından genel güncellenmesi söz konusudur. i ve j olarak tanımlanmış noktalar arasındaki feromon değeri (τ_{ij}) Eşitlik 5.6’da verildiği gibi güncellenir. İlgili eşitlikte, feromonların buharlaşma oranını ρ , toplam karınca sayısı m , i ve j noktaları arasında k karıncası tarafından oluşturulan feromon miktarı ise $\Delta\tau_{ij}^k$ ile tanımlanmıştır.

$$\tau_{ij} \leftarrow (1-\rho)\tau_{ij} + \sum_{k=1}^m \Delta\tau_{ij}^k \quad (5.6)$$

$\Delta\tau_{ij}^k$ parametresinin karınca davranışına göre değişimi Eşitlik 5.7’de verildiği gibidir. İlgili eşitlikte, Q sabit bir değeri, L_k ise k karıncası tarafından oluşturulan yolun uzunluğunu temsil etmektedir.

$$\Delta\tau_{ij}^k = \begin{cases} Q / L_k & , eğer \ k \text{ karıncası dolaşım sürecinde} \\ & i \text{ ve } j \text{ noktaları arasındaki yolu kullanırsa} \\ 0 & , diğ er durumda \end{cases} \quad (5.7)$$

Çözümün gerçekleştirilmesi sürecinde, ziyaret edilecek yeni konum karıncalar tarafında stokastik bir biçimde seçilir. k karıncası, i noktasında kısmi bir çözüm (s^p) oluşturduktan sonra, yeni konum olan j ye gitme olasılığı Eşitlik 5.8’de gösterildiği gibidir.

$$p_{ij}^k = \begin{cases} \frac{\tau_{ij}^\alpha \eta_{ij}^\beta}{\sum_{c_{il} \in N(s^p)} \tau_{il}^\alpha \eta_{il}^\beta} , eğer \ c_{ij} \in N(s^p) \\ 0, & değı lse \end{cases} \quad (5.8)$$

İlgili eşitlikte, $N(s^p)$ hesaplanabilir bileşenlerin kümesi, l parametresi k karıncası tarafında daha önce ziyaret edilmemiş şehri ve c_{ij} sonlu sayıdaki çözüm bileşenlerini temsil etmektedir. α ve β parametreleri, sezgisel bilgi (η_{ij}) karşı bağıl önemini kontrol etmektedir. η_{ij} parametresi Eşitlik 5.9’da gösterildiği gibi tanımlanmaktadır.

$$\eta_{ij} = \frac{1}{d_{ij}} \quad (5.9)$$

İlgili eşitlikte, d_{ij} parametresi, i ve j şehirleri arasındaki mesafeyi temsil etmektedir. İlerleyen zamanlarda KKA’sı tabanlı çok sayıda çalışma gerçekleştirilmiştir. Fakat bunlardan en dikkat çekenleri, MAKS-MIN karınca sistemi (MAX-MIN Ant System) ve karınca koloni sistemidir (Ant Colony System). MAKS-MIN karınca sisteminde, sadece en iyi karıncaya ait feromon izleri güncellenir ve bu güncelleme feromon değerine bağılı olmaktadır. Bir diğ er yöntem olan karınca koloni sisteminde ise, geliştirme sürecinin sonunda yapılan feromon güncellenmesinden farklı olarak yerel bir feromon güncellemesi gerçekleştirilir. Bu güncellemeye çevirim dışı güncelleme adı da verilmektedir. Yerel feromon güncellemesi her bir gelişim sürecinde tüm karıncalar tarafından gerçekleştirilir.

KKA’nın uygulaması süreci genel hatları ile aşağıdaki gibidir.

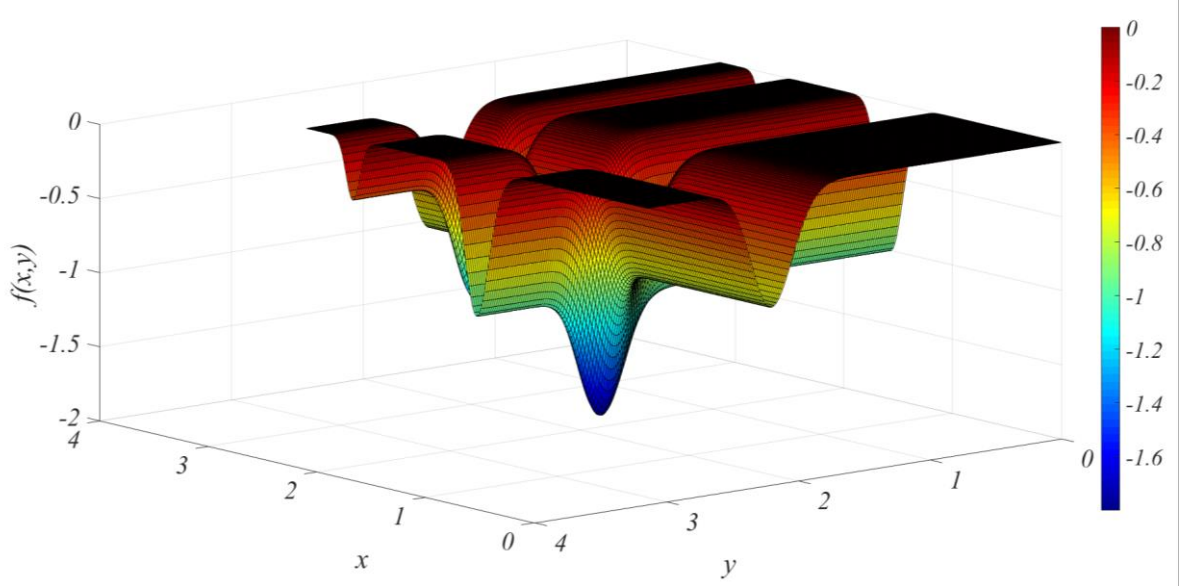
- Amaç fonksiyonu tanımlanır. $f(x)$, $x = (x_1, \dots, x_n)^T$ veya $c_{ij} \in \{1, 2, \dots, n\}$ şeklinde yönlendirme problemi olarak tanımlanabilir.
- Feromon buharlaşma oranı (ρ) tanımlanır.
- Yeni çözüm oluşturulur ve değerlendirilir.
- Daha iyi çözümler veya rotalar işaretlenir.
- Feromonlar güncellenir: $\tau_{ij} \leftarrow (1 - \rho)\tau_{ij} + \sum_{k=1}^m \Delta\tau_{ij}^k$
- Belirlenen durdurma kriterleri sağlanana kadar çözüm araştırma süreci devam eder.
-

KKA'sının karakteristiği bu tez çalışması kapsamında Michalewicz fonksiyonu üzerinden irdelenmiştir. İlgili fonksiyon tanımlandığı boyut sayısının faktöriyeli ($d!$) kadar yerel minimum içermektedir. Eşitlik 5.10'da ilgili fonksiyonun tanım bağıntısı gösterilmiştir. İlgili bağıntıda, $m=10$ olarak alınması önerilmektedir ve bu çalışmada da belirtilen değer alınmıştır. Ayrıca iki boyutlu Michalewicz fonksiyona ait tanım aralığı ve fonksiyonun minimum değer ile bu değeri sağlayan değişkenler Eşitlik 5.11'de gösterilmiştir.

$$f(x) = \sum_{i=1}^d \sin(x_i) \sin^{2m} \left(\frac{ix_i^2}{\pi} \right) \quad (5.10)$$

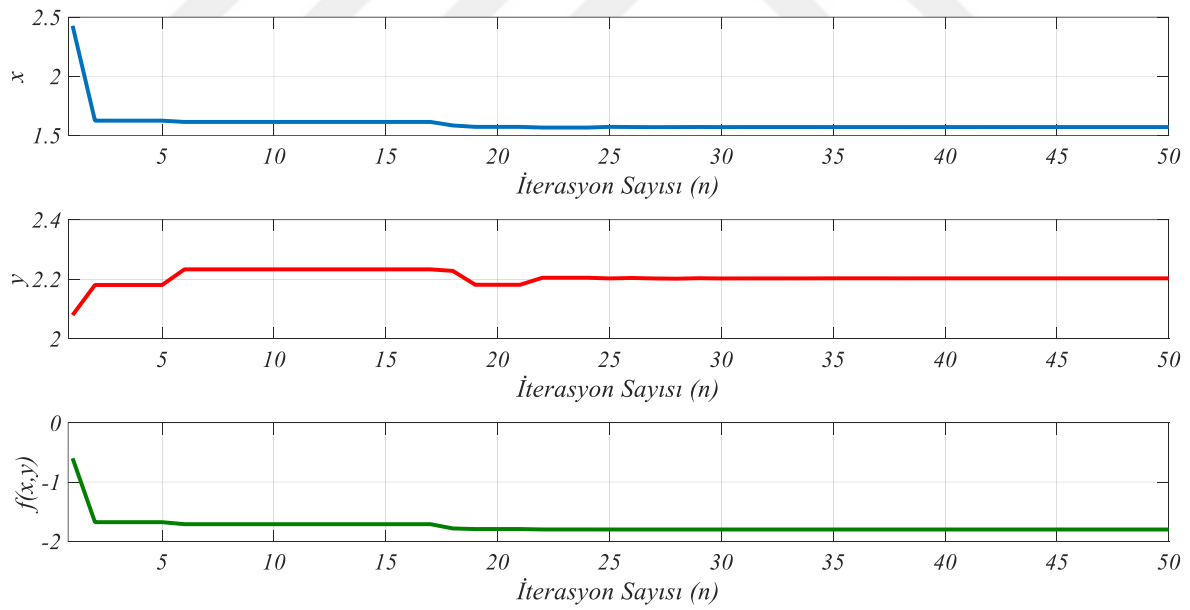
$$x, y \in [0, \pi], \text{ ve } f_{\min}(x_{opt}, y_{opt}) = -1.8013 \Rightarrow x_{opt} = 2.2029 \text{ ve } y_{opt} = 1.5707 \quad (5.11)$$

Şekil 5.7'de Michalewicz fonksiyonun tanım aralığında üç boyutlu değişimi gösterilmiştir [94].



Şekil 5.7. Michalewicz fonksiyonun üç boyutlu değişimi

Şekil 5.8’de değişkenlerin ve $f(x,y)$ fonksiyonunun KKA’sı sonucu minimum değerinin araştırılması sonucu iterasyon sayısına bağlı değişimi gösterilmiştir.

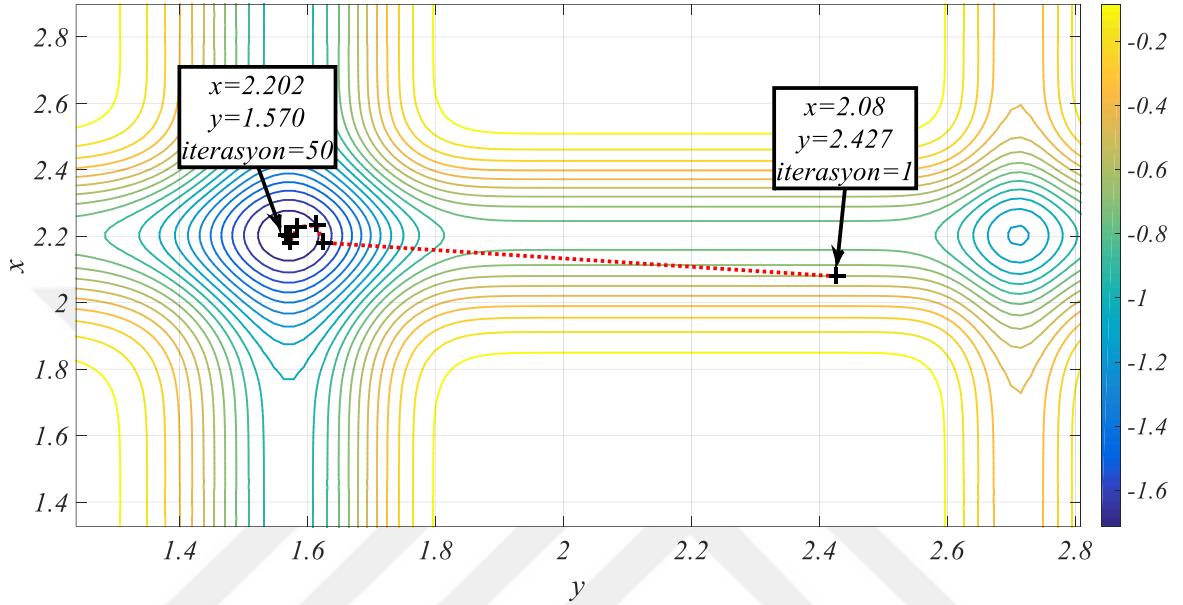


Şekil 5.8. Michalewicz fonksiyona ait parametrelerin iterasyon sayısına bağlı değişimi

Eşitlik 5.12’de Michalewicz fonksiyonunun KKA’sı ile çözümü sonucu elde edilen değerler bulunmaktadır.

$$x, y \in [0, \pi], \text{ ve } f_{\min_{kka}}(x_{kka}, y_{kka}) = -1.8013 \Rightarrow x_{kka} = 2.2029 \text{ ve } y_{kka} = 1.5708 \quad (5.12)$$

Şekil 5.9’da KKA’sına ait çözümlerin iterasyon sayısına bağlı değişimi gösterilmiştir. Şekilde görüldüğü gibi belirli bir iterasyon sayısından sonra çözüm uzayına yakınsanmıştır



Şekil 5.9. KKA’sının iterasyon sayısına bağlı değişimi

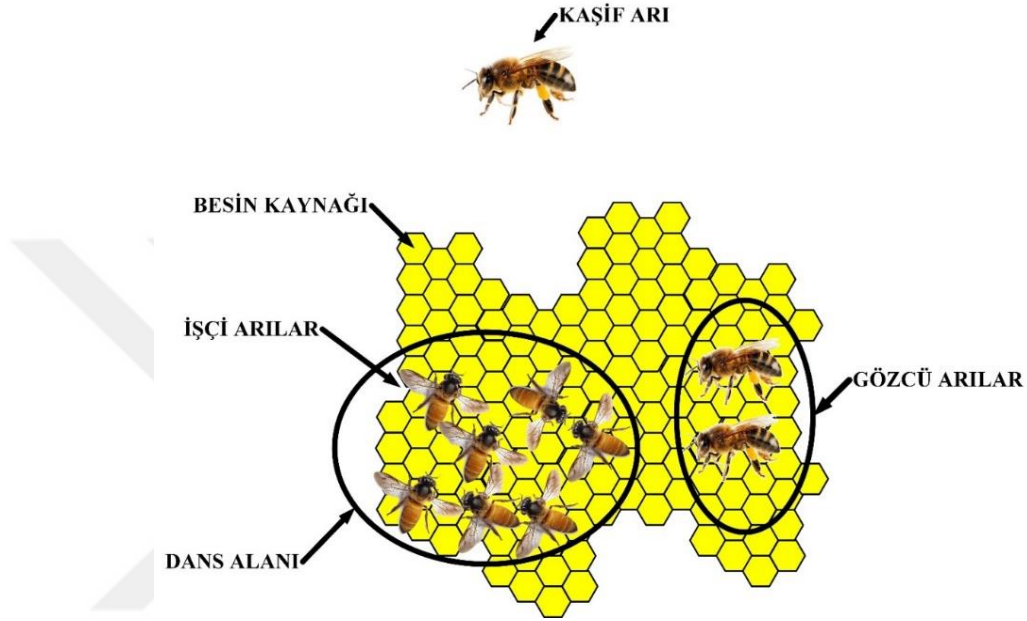
5.3. Yapay Arı Kolonisi

Yapay Arı Kolonisi (Artificial Bee Colony) yöntemi 2005 yılında *Derviş Karaboğa* tarafından geliştirilen sürü davranışı tabanlı metasezgisel bir algoritmadır [95]. İlgili yöntem bal arılarının akıllıca gerçekleştirdiği besin arama davranışından esinlenerek gerçekleştirilmiştir. Bir grup bal arısı sosyal etkileşim yolu ile belirli görevleri başarı ile gerçekleştirmektedir. Yapay Arı Kolonisi (YAK) yöntemi, 2005 yılında *Valery Tereshko* ve *Andreas Loengarov* tarafından bal arılarının beslenmesi üzerine geliştirilen model üzerinden oluşturulmuştur [96].

YAK algoritmasında üç farklı tür arı bulunmaktadır: *işçi*, *gözcü* ve *kâşif* arılardır. İşçi arılar, kaynaklar etrafında besin ararlar ve aynı zamanda bu bilgileri gözcü arılar ile paylaşırlar. Bu sayede daha yüksek kaliteli gıdanın daha düşük kaliteli gıdaya göre arılar tarafından seçilme şansı yükselir. Bu çözüm sistemi, KKA’sı kapsamında olduğu gibi bağıl olarak kısa olan rotanın

tercihi prensibine dayanır. Kaşif arılar, az sayıda işçi arının dönüştürülmüş şeklidir. Bu arılar besin kaynaklarını terk ederek yeni besin türleri ararlar [97].

YAK algoritmasında, sürülerin ilk yarısı işçi arıları diğer yarısı gözcü arıları kapsamaktadır. İşçi veya gözcü arıların sayısı sürüdeki çözüm sayısına eşittir. Temel bir arı kolonisi modeli Şekil 5.10’da gösterilmiştir [98].



Şekil 5.10. Arı kolonisi gösterimi

YAK algoritması rastgele dağıtılmış bir başlangıç nüfusu oluşturur ve bu başlangıç nüfusunun sayısı (SN) sürüdeki arıları temsil eder. Her bir işçi arı X_i , mevcut konumu yakınında Eşitlik 5.13’de gösterilen denklem ile aday çözüm (V_i) oluşturur.

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \quad (5.13)$$

$X_i = \{x_{i,1}, x_{i,2}, \dots, x_{i,D}\}$ kümesi, sürüdeki i ’nci çözümü temsil etmektedir. D parametresi ise boyut bilgisini vermektedir. İlgili eşitlikte, X_k rastgele seçilen aday çözümü temsil etmektedir ($i \neq j$) ve j parametresi ise $\{1, 2, \dots, D\}$ kümesindeki boyut indeksini, ϕ_{ij} ise $[-1, 1]$ arasında değişiklik gösteren rastgele sayıyı temsil etmektedir. Yeni aday çözüm (V_i) üretildikten sonra

açgözlü seçim yöntemi kullanır. Eğer V_i 'nin uygunluk değeri temeli X_i iyi ise X_i değeri V_i ile değiştirilir. Diğer türlü X_i değeri değiştirilmeden korunacaktır.

Tüm işçi arıların besin arama süreci bittikten sonra besin kaynakları ile ilgili bilgiyi gözcü arılar ile paylaşırlar. Gözcü arı tüm işçi arılardan alınan nektar bilgilerini değerlendirir ve nektar miktarına bağlı olarak bir olasılığı olan gıda kaynağını seçer. Bu olasılık seçimi bir rulet tekerliği seçim süreci gibidir ve olasılık tabanlı gösterimi Eşitlik 5.14'de olduğu gibidir.

$$p_i = \frac{fit_i}{\sum_{j=1}^{SN} fit_j} \quad (5.14)$$

İlgili eşitlikte, fit_i parametresi sürüdeki i 'nci çözümün uygunluk değeridir. Eğer önceden tanımlanan sayıda (limit) pozisyon bilgisi geliştirilemezse, ilgili besin kaynağı terk edilir. Terk edilen kaynak, kaşif arılar tarafından belirlenecek yeni besin kaynağı ile Eşitlik 5.15'de gösterildiği gibi güncellenir.

$$x_{ij} = lb_j + rand(0,1)(ub_j - lb_j) \quad (5.15)$$

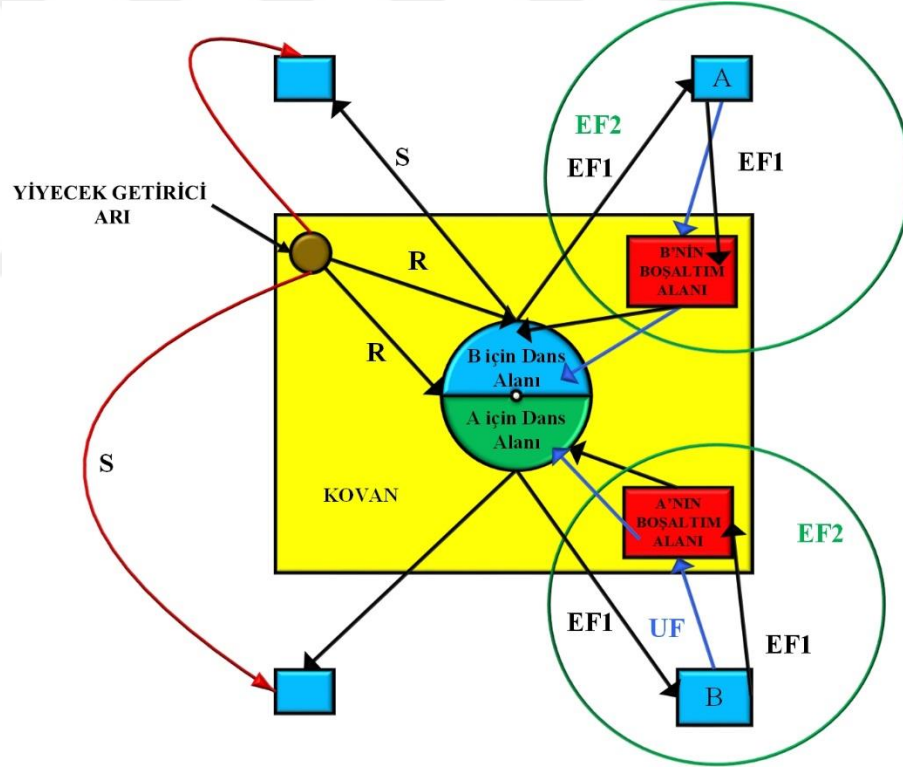
İlgili eşitlikte, rastgele sayı üretici normal dağılıma sahiptir ve $[0,1]$ arasında değişiklik gösterir. lb ve ub ise optimizasyon problemine ait alt ve üst sınırların j 'nci boyut üzerinden gösterimidir [98].

YAK algoritması kapsamında yapay arılar tarafından yapılan arama süreci aşağıda gösterildiği gibi özetlenebilir [97].

- Her bir işçi arı besin kaynaklarına yakın yerlerdeki besinlerin yerini belirler, kaynağından çıkartır ve karlılığını belirler.
- Her bir işçi arı kovanda bekleyen gözcü arılar ile bu bilgileri paylaşır verilen bilgiye bağlı olarak gıda kaynağının konumu seçilir.
- Her gözlemci, kendisi tarafından seçilen yerdeki bir gıda kaynağını belirler ve bu kaynağın karlılığını değerlendirir.
- Kaynaktan vazgeçilmesi sonucu işçi arı, kaşif bir arıya dönüştür ve yeni bir gıda kaynağını rastgele aramak üzere bir arayışa çıkar.

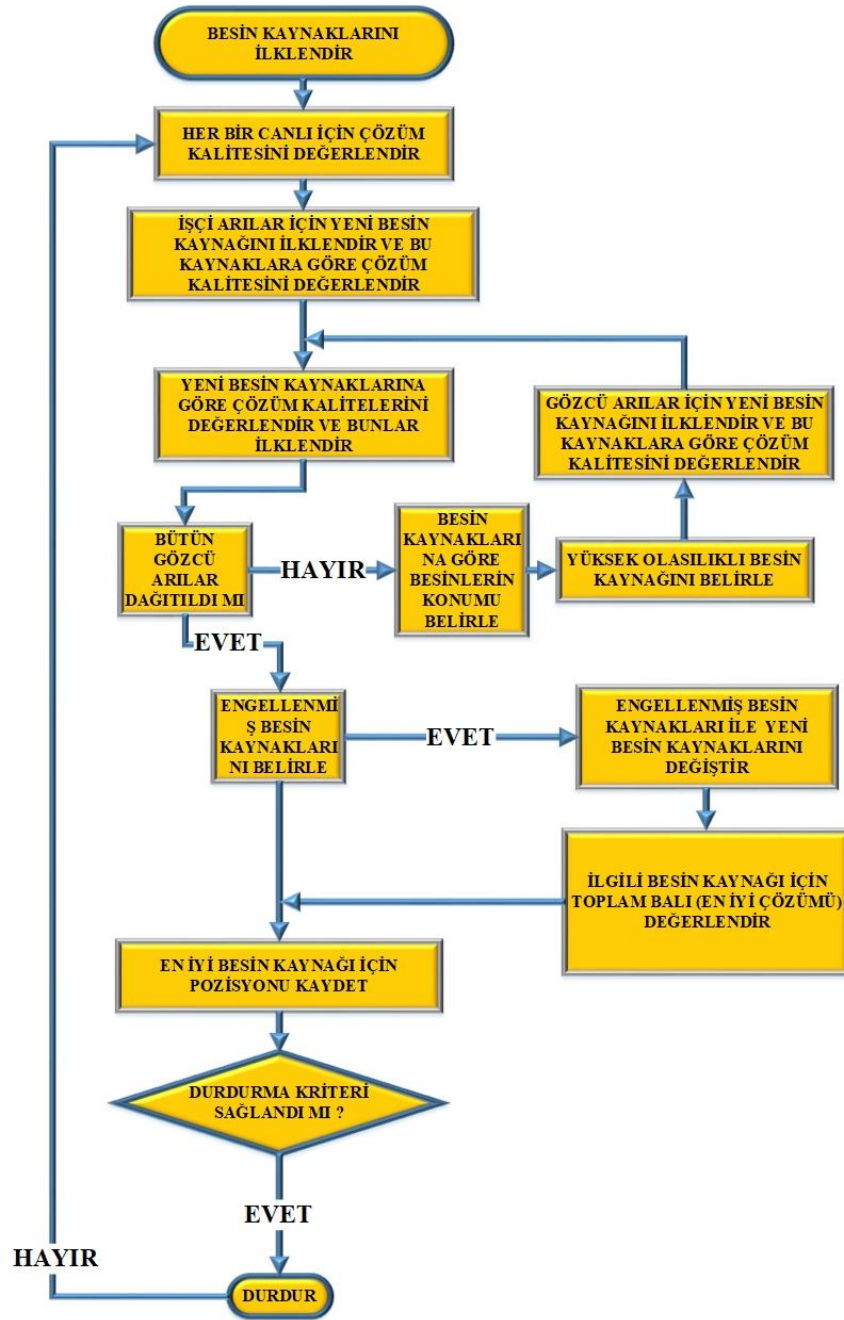
Şekil 5.11’de bal arılarının nektar toplama sürecindeki davranışı gösterilmiştir. İlgili şekle bağlı olarak, bal arısı besini bulduktan sonra bal arısı besinin konumunu kaydetmek için kendi yeteneklerini kullanır. Bu şekilde ilgili arı, gözcü bir arıya dönüşür. Toplayıcı arı kaynağından bir miktar nektar alır ve bu nektarı bir gıda deposuna boşaltarak kovanına geri döner. Besinleri boşalttıktan sonra aşağıda gösterilen süreç gerçekleştirilir [99].

- Yemek kaynağını (UF) terk ettikten sonra itaatsiz bir takipçi haline gelebilir.
- Aynı gıdanın kaynağına (EF1) dönmeden önce dans edip yuva arkadaşlarını ile birleşebilir.
- Diğer arılar ile birleşmeden besin kaynağından yem almaya devam edebilir (EF2).



Şekil 5.11. Bal arılarının nektar toplama süreci

Şekil 5.12’de YAK algoritmasının akış şeması gösterilmektedir.



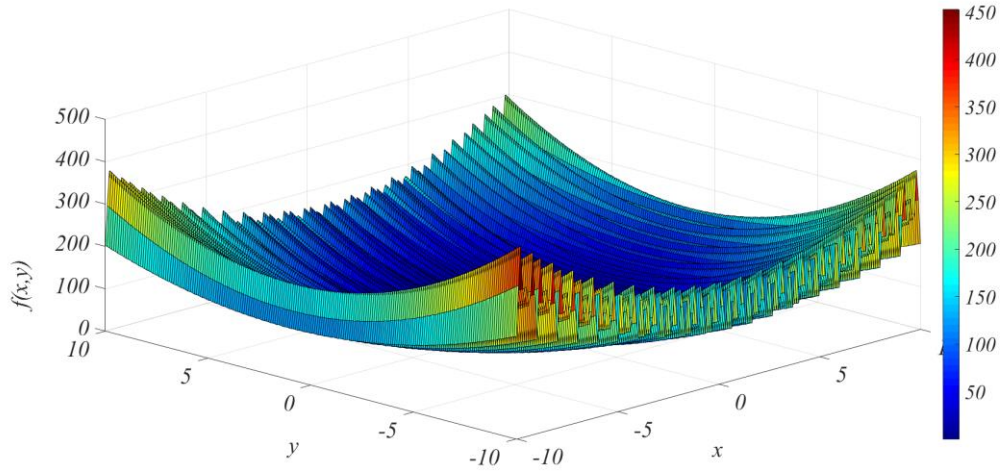
Şekil 5.12. YAK algoritmasına ait akış şeması

YAK’sinin karakteristiği bu tez çalışması kapsamında *Levy 13* fonksiyonu üzerinden irdelenmiştir. Eşitlik 5.16’de ilgili fonksiyonun tanım bağıntısı gösterilmiştir. Ayrıca 13 Numaralı *Levy 13* fonksiyona ait tanım aralığı ve fonksiyonun minimum değer ile bu değeri sağlayan değişkenler Eşitlik 5.17’de gösterilmiştir.

$$f(x, y) = \sin^2(3\pi x) + (x-1)^2 \left[1 + \sin^2(3\pi y) \right] + (y-1)^2 \left[1 + \sin^2(2\pi y) \right] \quad (5.16)$$

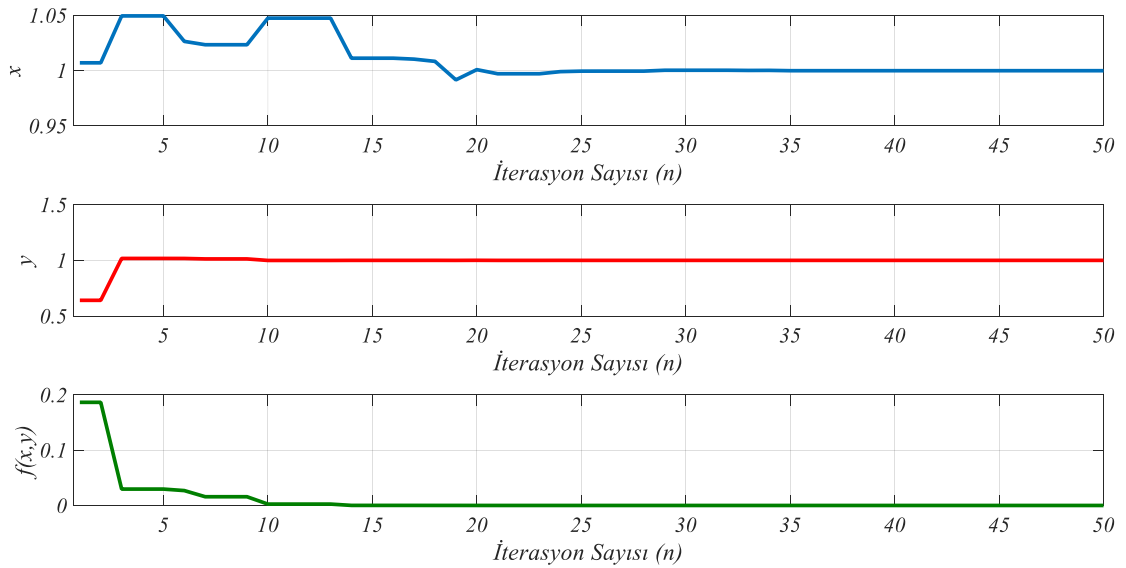
$$x, y \in [-10, 10], \text{ ve } f_{\min}(x_{opt}, y_{opt}) = 0 \Rightarrow x_{opt} = 1 \text{ ve } y_{opt} = 1 \quad (5.17)$$

Şekil 5.13’de *Levy 13* fonksiyonunun dağılımının grafiği elde edilmiştir [100].



Şekil 5.13. Levy 13 Fonksiyonun üç boyutlu değişimi

Şekill 5.14’de değişkenlerin ve $f(x,y)$ fonksiyonunun YAK’si algoritması vasıtasıyla minimum değerinin araştırılması sonucu iterasyon sayısına bağlı değişimi gösterilmiştir.



Şekil 5.14. Levy 13 fonksiyonuna ait parametrelerin iterasyon sayısına bağlı değişimi

Eşitlik 5.18’de *Levy 13* fonksiyonunun YAK’si ile çözümü sonucu elde edilen değerler bulunmaktadır.

$$x, y \in [-10, 10], \text{ ve } f_{\min_{YAK}}(x_{YAK}, y_{YAK}) = 2.196 \times 10^{-8} \Rightarrow x_{kka} = 1 \text{ ve } y_{kka} = 0.999 \quad (5.18)$$

İlgili eşitlikte görüldüğü gibi YAK algoritması ile fonksiyonun değerini minimize eden x ve y değerleri 50 iterasyon içinde başarı ile bulunmuştur.

5.4. Parçacık Sürü Optimizasyonu

Parçacık sürü optimizasyonu (PSO) algoritması, balık ve kuş sürülerinin doğadaki akın ve eğitim davranışları temel alınarak 1995 yılında, Russell Eberhart ve James Kennedy tarafından önerilmiştir. Sürü zekası temel alınarak oluşturulmuştur. Geliştirildiği 1995 yılından beri PSO yöntemleri daha geniş çalışma alanlarına yayılmış ve popüler bir araştırma konusu haline gelmiştir. PSO sürü zekası olarak adlandırılan genel bir fenomenin temeli olmuştur ve optimizasyon alanında popülerliği olan geniş bir çalışma alanına öncü olmuştur [101].

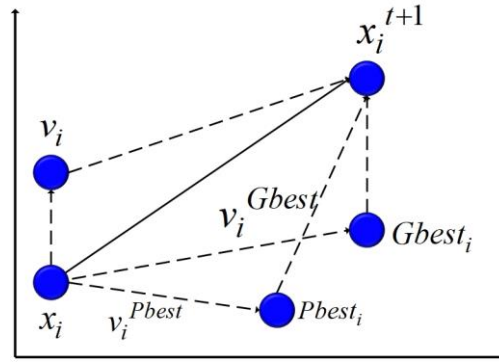
PSO algoritması, optimizasyon alanındaki hemen hemen tüm problemlere uygulanmıştır. PSO sürü halinde hareket eden canlıların aslında hedeflerine ya da besin kaynaklarına erişmek için rastgele olmayan belirli bir karakteristiğe sahip bir takım hareketler olduğu fenomeni üzerinden geliştirilmiştir. Arama süreci, GA’sın da olduğu gibi jenerasyon tabanlıdır ve her bir bireye parçacık (particle) adı verilmektedir. Parçacıklardan oluşan gruplara da sürü (swarm) adı verilir. Sürüdeki bütün parçacıklar kendi bulunduğu pozisyonunu süredeki en iyi pozisyona göre günceller. Bu bağlamda, PSO algoritması amaç fonksiyonu üzerinden arama uzayında parçacıklar yardımı ile bir rota ayarlaması gerçekleştirir. Sürüdeki parçacıkların hareketi iki temel bileşen içerir: stokastik ve deterministik bileşenler. Her bir parçacık geçmişteki en iyi konumuna bağlı olarak mevcut genel en iyiye doğru yönelim gösterir [102].

Parçacığın daha önceki bulunduğu konumdan daha iyi bir konum bulması ile pozisyon bilgileri güncellenir. Her bir iterasyon sürecinde n adet parçacıktan oluşan yapı da her zaman anlamlı bir en iyi vardır. Algoritmanın amacı, sınırlı sayıda iterasyon sürecinde veya en iyi çözümün geliştirilemeyeceği bir aşamaya kadar olabilecek en iyi çözümün parçacıklar yardımı ile bulunmasını sağlamaktır [103].

PSO ile GA bazı yönlerden benzerlikler gösterir. Başlangıç durumunda rastgele bir popülasyon üzerinden başlatılır ve genel optimal değer aranırken bu başlangıç değerleri güncellenir. Fakat GA'sının gerçekleştirilmesi sürecinde tanımlanan, çaprazlama veya mutasyon gibi parametreler bulunmaktadır. Bu şekilde PSO algoritmasının GA'sı na göre daha az kullanıcı tarafından tanımlanması gereken parametresi bulunmaktadır ve bu algoritmanın karmaşıklığının az olması açısından oldukça büyük bir avantajdır.

PSO algoritmasının bir diğer avantajı, gradyent bilgisinin hesaplanmasına ihtiyaç duymadığından kolay bir şekilde uygulanabilmesidir. PSO algoritması, bulanık mantık kontrol sistemleri, yapay sinir ağlarının eğitilmesi gibi çok geniş alanlarda uygulanmaktadır [104].

PSO algoritmasına göre parçacık noktalarının güncellenmesi süreci Şekil 5.15'de gösterilmiştir. İlgili şekilde, x_i parametresi geçerli konumu, x_i^{t+1} güncellenmiş konum, V_i geçerli hız, V_i^{t+1} güncellenmiş hız, V_i^{Pbest} en iyi konuma bağlı hız bilgisi, V_i^{Gbest} bütün parçacık popülasyonu arasındaki en iyi çözümü, $Pbest_i$ i 'nci parçacığa ait en iyi bir önceki pozisyonu, $Gbest_i$ bütün sürü içerisindeki en iyi parçacığı temsil etmektedir [74].



Şekil 5.15. PSO algoritması ile arama noktasını güncelleme süreci

PSO algoritmasının temelini oluşturan güncellenmiş hız vektörü Eşitlik 5.19'de gösterildiği gibi elde edilir.

$$v_i^{t+1} = v_i^t + \alpha \xi_1 [g^* - x_i^t] + \beta \xi_2 [x_i^{*(t)} - x_i^t] \quad (5.19)$$

İlgili eşitlikte, i 'nci parçacığa ait konum vektörü x_i , hız vektörü v_i olarak tanımlanmaktadır ve güncel en iyi parçacık ise $x_i^{*(t)}$ ile tanımlanır. $g^* \approx \min \{ f(x_i) \}, (i=1,2,...,n)$ ise t değerindeki güncel genel en iyi olarak tanımlanır. ξ_1 ve ξ_2 parametreleri 0 ve 1 arasında değer alan rastgele vektörleri temsil etmektedir. α ve β parametreleri, öğrenme veya ivme sabiti olarak tanımlanmaktadır ve genellikle $\alpha \approx \beta \approx 2$ olacak şekilde alınır.

Sürüyü oluşturan parçacıklar problem uzayına mümkün olduğu kadar eşit dağılmalıdır. Bu sayede belirli bir problem için önem arz eden çoğu bölgeyi örnekleyebilirler. Bir parçacığın başlangıç değeri sıfır olarak ($v_i^t = 0, t=0$) alınabilir ve ilerleyen iterasyonlar ile güncel ve anlamlı bir parametreye yakınsayabilir ve yeni pozisyon Eşitlik 5.20'de gösterildiği gibi güncellenebilir. v_i değeri herhangi bir değer olabilir ve genellikle $v_i \in [0, v_{\max}]$ şeklinde tanımlanır.

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (5.20)$$

PSO algoritmasının gerçekleştirilmesine ait sözde kodlar aşağıda sıralandığı gibidir.

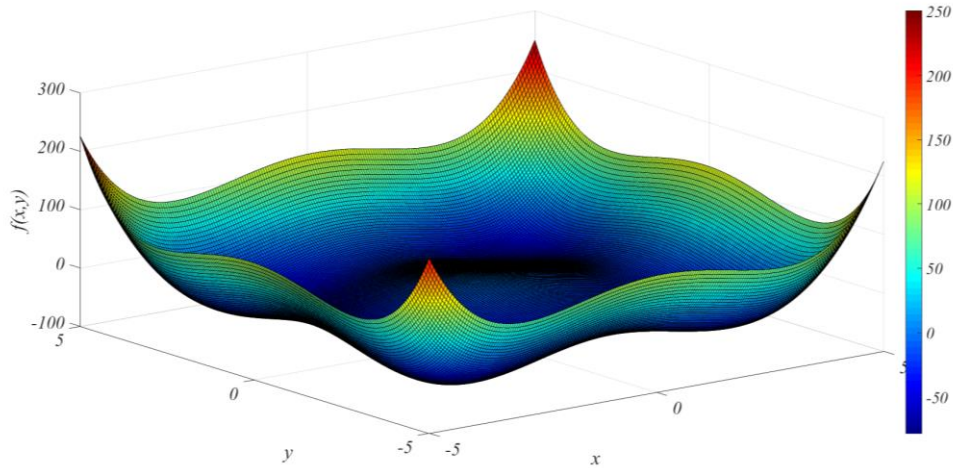
- Amaç fonksiyonu tanımlanır. $f(x), x = (x_1, ..., x_d)^T$
- n adet parçacığa ait başlangıç konumları x_i ve hızları ilklendirilir v_i .
- $\min \{ f(x_1, ..., f(x_n)) \}, t=0$ eşitliği üzerinden genel en iyi g^* bulunur.
- Yeni hız değeri Eşitlik 5.19'da gösterildiği gibi güncellenir.
- Yeni konum bilgisi Eşitlik 5.20'de gösterildiği gibi belirlenir.
- x_i^{t+1} noktasındaki yeni amaç fonksiyonu değerlendirilir.
- Her bir x_i^* parçacığı için güncel en iyiler belirlenir.
- Güncel genel en iyi (g^*) belirlenir.
- İterasyon sayısı artırılır ve güncel çözümün kullanıcı şartlarını karşılayıp karşılamadığına bakılır.
- Çıktı olarak son sonuçlar (g^* ve x_i^*) elde edilir.

PSO algoritmasının karakteristiđi bu tez alıřması kapsamında *Styblinski-Tang* fonksiyonu zerinden irdelenmiř ve Eřitlik 5.21’de tanım bađıntısı verilmiřtir. Ayrıca 13 Numaralı *Styblinski-Tang* fonksiyona ait tanım aralıđı ve fonksiyonun minimum deđer ile bu deđerı sađlayan deđerkenler Eřitlik 5.22’de gsterilmiřtir.

$$f(x) = \frac{1}{2} \sum_{i=1}^d (x_i^4 - 16x_i^2 + 5x_i) \quad (5.21)$$

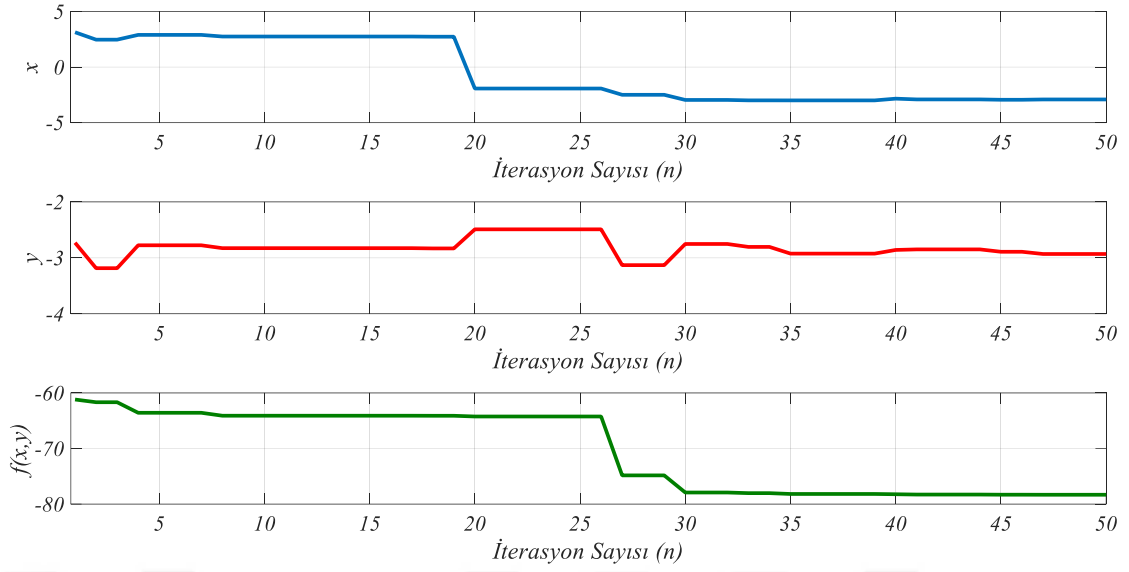
$$x, y \in [-5, 5], \text{ ve } f_{\min}(x_{opt}, y_{opt}) = -78.3323 \Rightarrow x_{opt} = -2.9035 \text{ ve } y_{opt} = -2.9035 \quad (5.22)$$

řekil 5.16’da *Styblinski-Tang* fonksiyonun tanım aralıđında  boyutlu deđerimi gsterilmiřtir [105].



řekil 5.16. Styblinski-Tang fonksiyonun  boyutlu deđerimi

řekil 5.17’de deđerkenlerin ve $f(x,y)$ fonksiyonunun PSO algoritması sonucu minimum deđerinin arařtırılması sonucu iterasyon sayısına bađlı deđerimi gsterilmiřtir.

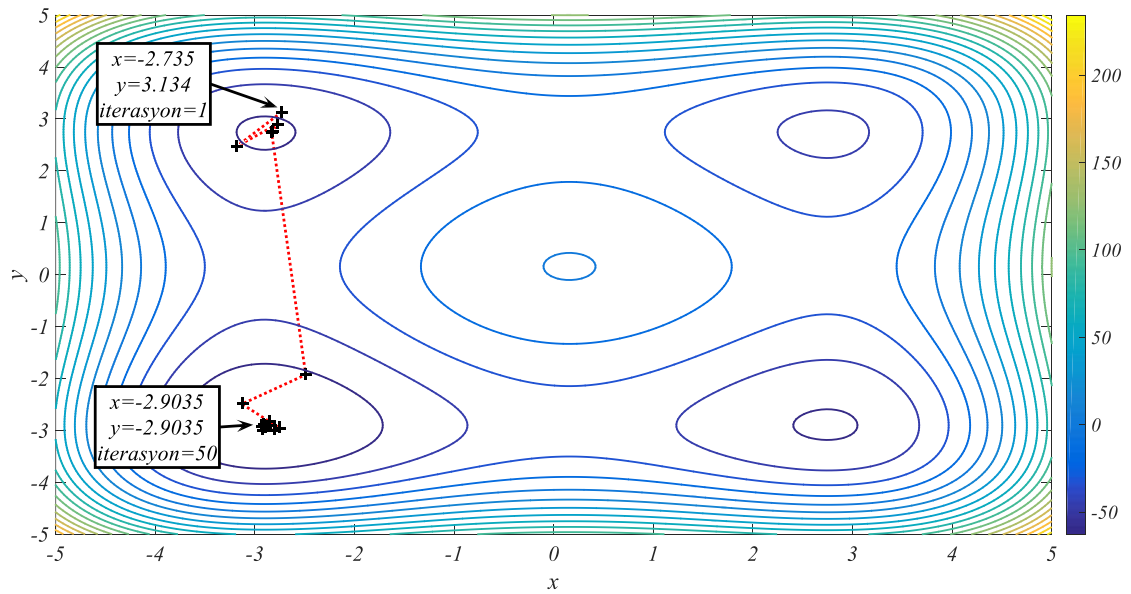


Şekil 5.17. Styblinski-Tang fonksiyona ait parametrelerin iterasyon sayısına bağlı değişimi

Eşitlik 5.23’de *Styblinski-Tang* fonksiyonunun PSO algoritması ile çözümü sonucu elde edilen değerler bulunmaktadır.

$$x, y \in [-5, 5] \text{ ve } f_{\min_{\text{psa}}}(x_{\text{psa}}, y_{\text{psa}}) = -78.332 \Rightarrow x_{\text{psa}} = -2.9035 \text{ ve } y_{\text{psa}} = -2.9035 \quad (5.23)$$

Şekil 5.18’de PSO algoritmasına ait çözümlerin iterasyon sayısına bağlı değişimi gösterilmiştir. Şekilde görüldüğü gibi belirli bir iterasyon sayısından sonra çözüm uzayına yakınsanmıştır.



Şekil 5.18. PSO algoritmasının iterasyon sayısına bağlı değişimi

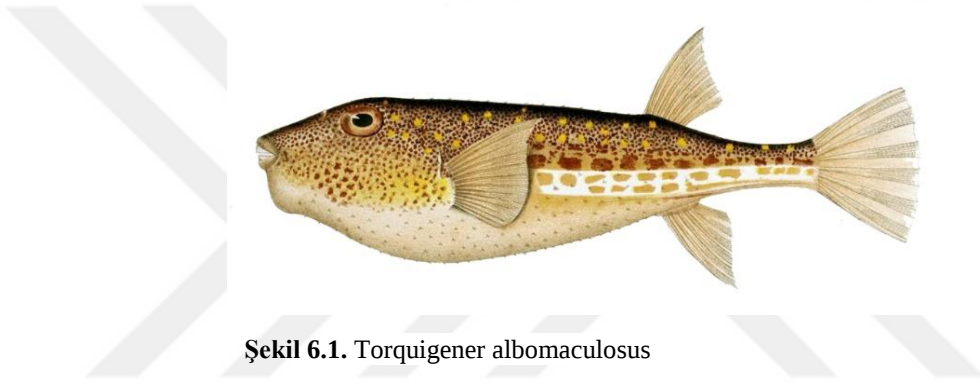
PSO algoritması diğer metasezgisel optimizasyon algoritmaları gibi çeşitli avantajlar ve dezavantajlar içermektedir. Klasik yöntemlere göre avantajları şu şekildedir:

- Türev işlemine ihtiyaç duymaz ve bu sayede daha kararlı bir davranış göstermektedir.
- Düşük adım aralıklarına imkan sağlar.
- Paralel çalışmaya uygundur.

PSO algoritması akıllı bir yapıdadır. Bu sayede bilimsel araştırma ve mühendislik uygulamalarında kolaylıkla kullanılabilir. Mutasyon hesaplaması ve örtüşen yapıda hesaplamalar içermemektedir. İşlemsel yükü GA'sına göre daha düşüktür. Dezavantaj olarak ise optimal çözümü yakınsama hızı klasik yaklaşımlara göre daha yavaştır.

6. BALON BALIĞI VE OPTİMİZASYON

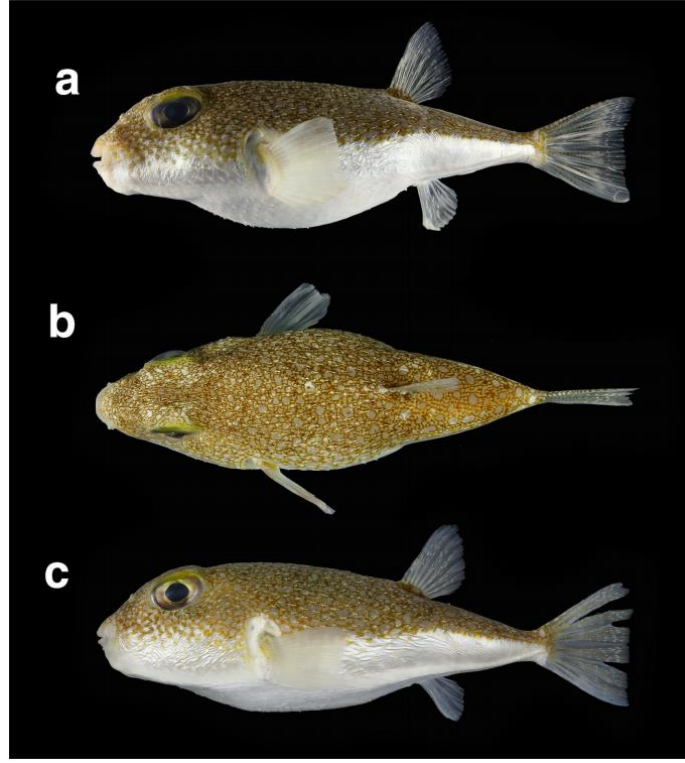
Tez çalışmasının bu bölümünde, metasezgisel optimizasyon yönteminin gerçekleştirilmesine ilham olan balon balığı sınıfına ait özgün balık cinsi hakkında çeşitli bilgiler verilecektir. Bu çalışmada davranışlarından esinlenen balon balığı cinsi Hint ve Pasifik okyanusunda yaşamakta ve genel bilimsel adı “Torquigener” olarak bilinmektedir. Tez çalışmasına ilham veren balık türünün bilimsel adı ise “Torquigener albomaculosus” olarak bilinmektedir. Şekil 6.1’de Torquigener albomaculosus balık türüne ait resim gösterilmektedir. Bu balık cinsinin özelliği deniz tabanında dairesel yapılar inşa edebilme yeteneğidir [106].



Şekil 6.1. Torquigener albomaculosus

Etmolojik olarak ilgili balık türünün ismini oluşturan Latince kelimelerin kökeni sırasıyla, *torquere*=döndürmek, *generare* = doğum, yarış ve *albomaculosus*=vücudundaki çok sayıdaki beyaz noktaların adıdır. Bu balık türü, denizlerde ve tropik ortamlarda yaşamaktadır ve yaşadıkları suyun derinliği 10-27 metre arasında değişmektedir. *Torquigener albomaculosus* olarak adlandırılan balık türü Kuzeybatı Pasifik bölgesinde bulunan Japonya'daki Ryukyu adası civarında bulunmaktadır. Balığın türünün maksimum uzunluğu erkeklerde 12 cm iken dişilerde bu uzunluk 9.1 cm değerindedir [107].

Kum altında büyük dairesel geometrik yapılar oluşturabilme yeteneği bu türü benzersiz kılmaktadır. Bu dairesel yapıların dişilerin eş seçiminin şekillenmesinde büyük rol oynamaktadır. İnce kum taneleri kullanarak yumurtlama hazırlığı için yuvalar oluştururlar. Şekil 6.2’de *Torquigener albomaculosus* türünün farklı anatomik yönlerden gösterimi bulunmaktadır [106].



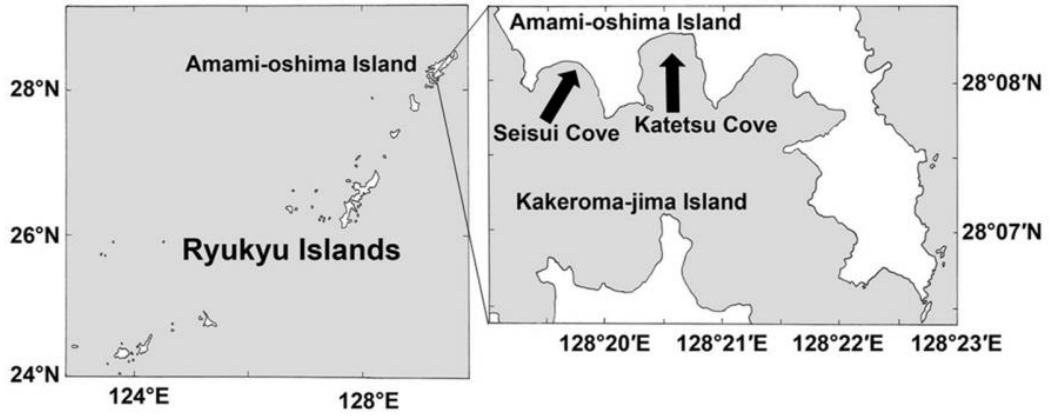
Şekil 6.2. a) Holotipin lateral görünümü, erkek, 87.8 mm b) Holotipin sırta ait görünümü c) Holotipin paratip görünümü, dişi, 90.5mm [106]

6.1. Balon Balığı ve Dairesel Yapılar

Torquigener albomaculosus balık türüne ait yumurtalama yuvaları oluşturulması ve erkekler tarafından karşı cinsin etkilenmesi için oluşturulan dairesel yapılar ilk olarak 1995 yılında güney Japonya da Amami-Oshima adası civarında gözlemlenmiştir. Daha sonra bu dairesel yapılar eğlence amacıyla dalış yapan dalgıçlar tarafından da gözlemlenmiştir. İlk aşamada ilgili yapıların kökeni belirsiz olup bir organizma ya da doğal bir fenomen tarafından ortaya çıkarıldığı düşünülmüştür ve belirtilen bu dairesel yapılara gizemli daire (mystery circle) adı verilmiştir. Yaklaşık 2 metre çapında olan bu dairesel yapılar bir süre daha gizemini korumuşlardır. 2011 yılında bu 2 metrelik dairesel yapıların 12 cm boyutlarında bir balon balığı türü olan Torquigener albomaculosus tarafından 10 ve 27 metre derinlikler arasında yapıldığı gözlemlenmiştir. İlgili yapılar iki kenarlı ve yuvarlak bir oluk ile birbirinden farklı karakteristiklerde oluşturulmaktadır. Deniz tabanına oluşturulan bu dairesel yapıların genel amacının karşı cinsi etkilemek olduğu düşünülmektedir. 2011 yılında su altı fotoğrafçısı Yoji Okata bu balık türünün oluşturduğu dairesel yapıların oluşum sürecine ait detaylı fotoğraflar elde etmiştir. 2012 yılının Temmuzun da ise Kawase ve arkadaşları bu balon balığı türüne ait gizemli dairesel yapıların oluşum sürecini video olarak kaydetmiştir ve bu araştırma tez

çalışmamıza da ilham olan 2013 yılında *Deniz Balon Balığı Üreme Sistemlerinde Büyük Geometrik Dairesel Yapıların Rolü (Role of Huge Geometric Circular Structures in the Reproduction of a Marine Pufferfish)* olarak isimlendirilen makalenin oluşmasını sağlamıştır. Bu araştırmalar sonucunda yeni bir balon balığı türü olan *Torquigener albomaculosus* isimli balık türü literatürde yer bulmuştur [107].

Şekil 6.3’de *Torquigener albomaculosus* balık türünün dairesel yapılar oluştururken gözlemlendiği konum aşağıda detaylı bir şekilde gösterilmektedir.



Şekil 6.3. Gözlem yapılan konumların detaylı gösterimi [106]

Şekil 6.4’de erkek *Torquigener albomaculosus* isimli balık türünün deniz tabanında oluşturduğu ve su altı fotoğrafçısı Hiroshi Kawase tarafından görüntülenen dairesel yapının bir örneği gösterilmektedir. Şekilde görüldüğü gibi ilgili dairesel yapı farkı bir takım örüntülerden oluşmaktadır ve tez çalışmamızın ilerleyen kısımlarında detaylı bir şekilde değinilecektir. İlgili makalede, bu örüntülerin rastgele bir süreç ile oluşmadığı belirli bir hiyerarşi ile olduğu ve çeşitli oranlar ve sayılar ışığında oluşturulan sıra dışı bir örüntü olduğu gözlemlenmiştir. Bu dairesel yapıyı ortaya çıkartan davranış bütünü bizim metasezgisel optimizasyon algoritmamıza ilham vermektedir. Dairesel yapıların oluşum sürecinin başarı ile bir algoritma olarak tanımlanabilmesi ile tamamen yenilikçi bir çalışma bu tez kapsamında ortaya çıkartılabilecektir. Bu sayede sadece belirli bir yöntemin ortaya çıkması sağlanmayacaktır, ayrıca matematiksel dünyaya aktarılan bu doğal fenomenin farklı türdeki optimizasyon problemlerinin çözümünde yol göstereceği öngörülmektedir.

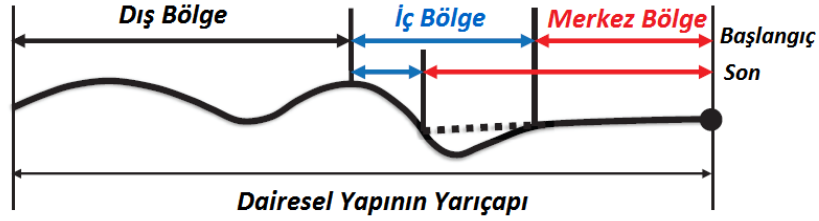


Şekil 6.4. Erkek balon balığı tarafından deniz tabanında oluşturulmuş dairesel yapı

Hiroshi Kawase ve arkadaşlarının gerçekleştirdiği araştırma ve gözlem sürecinde 10 adet erkek balığın üreme süreci *Seisui* ve *Katetsu* bölgelerinde gözlemlenmiştir. Bu balıklardan *Seisui* bölgesinde bulunanlar göre sırasıyla S1 ,S2 ,S3, S4, S5 ve S6 olarak etiketlenmişlerdir. *Katetsu* bölgesinde bulunan bir diğer balık grubu ise K1,K2,K3 ve K4 olarak etiketlenmiştir. Her bir balığın dairesel yapıyı tamamlaması 7-9 gün arası sürmektedir ve her bir balık tarafından oluşturulan dairesel yapılar farklılıklar göstermektedir. Dairesel yapıların oluşturulması çeşitli aşamalardan oluşmaktadır. İlk aşamada dairesel yapı genel hatları ile oluşturulmaktadır ve erkek balıklar kumun altında yüzgeçleri yardımıyla doğrusal hareketler ile tepcikler meydana getirirler. Sonraki aşamada ise dairenin dışından içine doğru radyal bir yönde çeşitli yönlerde yüzerek deniz dibi üzerinde radyal bir biçimde hizalanmış tepeler ve vadiler oluştururlar. Sonraki aşamada dış bölgeden iç bölgeye geçiş yaparlar ve iç bölgeden içeriye doğru geçişler yaparlar veya her iki bölgeden sürekli olarak geçerler [107].

Balıklar iç kısma yüzerken yüzgeç hareketlerini geçici olarak askıya almaktadırlar. Bu sayede ilgili dairesel yapıda koniklikler ve dış bükeylikler oluşmaktadır. Balık türü merkez kısımda ise rassal bir şekilde yüzmektedir ve bu sayede kumu düz bir alan oluşturmak üzere karıştırmaktadırlar. Sonuç olarak deniz dibinde düz bir alana sahip, radyal olarak hizalanmış tepelere ve vadilere sahip dairesel bir yapı oluşturulur. Bu süreç sonucunda oluşturulan dairesel yapının bir bütün olarak yarıçap değerleri 79-105 cm arasında değişirken, son merkez bölgenin yarıçap değeri 31-50 cm arasında değişmektedir. Ayrıca ilgili dairesel yapıya temel oluşturan tepelerin sayısı 24-32 arasında değişmektedir. Oluşturulan dairesel yapıların genel olarak 3 farklı bölümden oluştuğu söylenebilir.

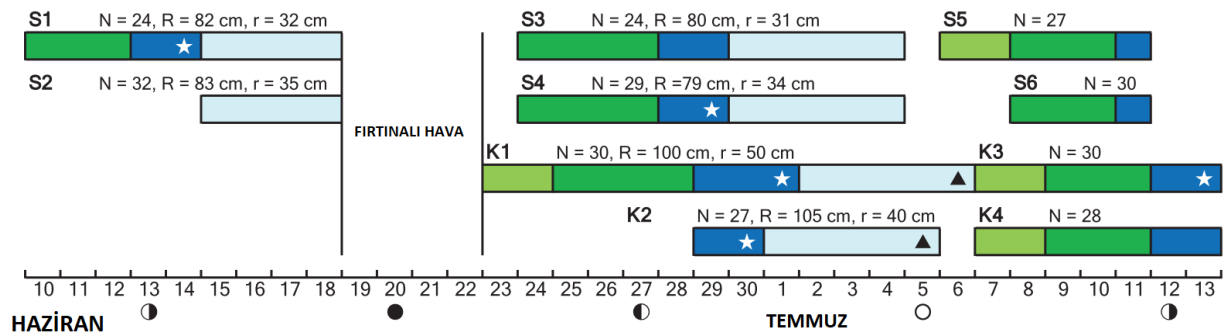
Şekil 6.5’de bu dairesel yapının detaylı gösterimi bulunmaktadır. Bu bölgeler dış (D), iç (İ) ve merkez (M) bölge olarak ayrılmıştır.



Şekil 6.5. Oluşturulan dairesel yapının detaylı gösterimi [107]

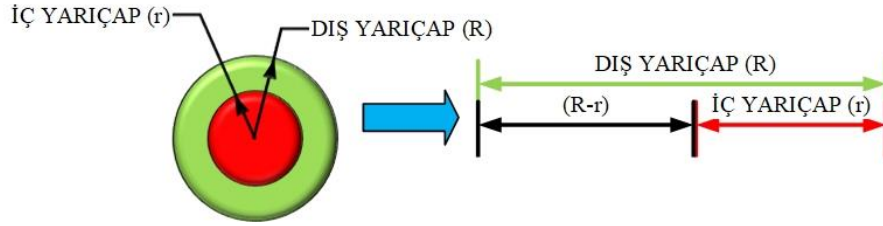
Hiroshi Kawase ve arkadaşlarının *Torquigener albomaculosus türü* 10 adet balık üzerinde yaptığı gözlemler sonucunda Şekil 6.6’da gösterilen detaylı bilgiye ulaşmaktadırlar.

İlgili şekilde farklı balıklara ait oluşturulan dairesel yapıların toplam tepe sayısı, yarıçap değeri ve iç yarıçap değeri gösterilmektedir. S1-S6’da Seisui bölgesinde bulunan balıkların oluşturduğu dairesel yapılara ait özellikler gösterilirken, K1-K4 balıklarında ise Katetsu bölgesinde bulunan balıkların oluşturduğu dairesel yapılara ait bazı sayısal bilgiler Şekil 6.6’da gösterilmektedir. Ayrıca ilgili şekilde açık yeşil sütunlar ilk süreci, yeşil sütunlar orta süreci ve mavi sütunlar son sürecin gerçekleştirime süresini gün olarak temsil etmektedir. Hiroshi Kawase ve arkadaşlarının araştırmasına göre dairesel yapıların oluşturulması süreci 3 farklı inşa aşamasından oluşmaktadır. Açık mavi sütunlar ise yumurta bakım süresini göstermektedir. Beyaz yıldızlar yumurtlamayı, katı üçgenler ise yumurtadan çıkmayı temsil etmektedir. Ayrıca ilgili şekilde N değeri tepe ve vadi çiftlerinin sayısını, R değeri dairesel yapının çapını, r değeri ise merkez bölgenin yarıçap değerini göstermektedir.



Şekil 6.6. Farklı balıkların oluşturduğu dairesel yapılara ait sayısal özellikler [107]

Bu tez çalışması kapsamında ilgili gözlem sonuçları irdelendiğinde daha detaylı bir şekilde gözlemlenen *Katetsu* bölgesindeki balıkların oluşturduğu dairesel yapılarda dairesel yapılara ait bölgelerin belirli bir orana (C_i) göre oluşturulduğu gözlemlenmektedir. İlgili oran Şekil 6.7’de temel bir dairesel yapı üzerinden gösterilmektedir.



Şekil 6.7. Yarıçap oranlarının karşılaştırılması

Dış yarıçap (R) ile merkez yarıçap arasındaki farkın merkez yarıçapa (r) oranı Eşitlik 6.1’de gösterildiği gibi hesaplanmaktadır.

$$c_i = \left(\frac{R_i - r_i}{r_i} \right) \quad (6.1)$$

K2 balığı için bu oran

$$c_{K2} = \left(\frac{R_{K2} - r_{K2}}{r_{K2}} \right) = \left(\frac{105 - 40}{40} \right) \Rightarrow \frac{65}{40} = 1.625 \text{ şeklinde hesaplanır.}$$

Bu oranı veren katsayısının yaklaşık olarak *altın oran* değerine (1.618) değerine yakınsadığı gözlemlenmektedir.

Ayrıca S3 balığı için bu oran

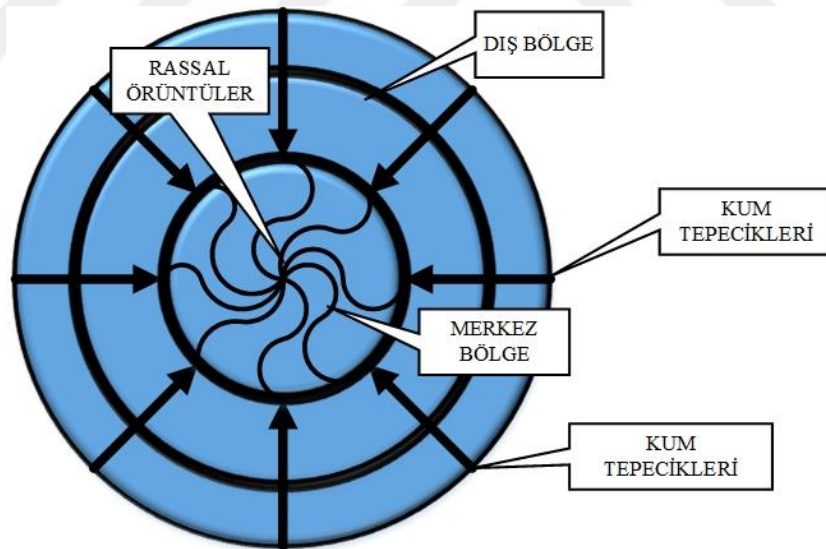
$$c_{S3} = \left(\frac{R_{S3} - r_{S3}}{r_{S3}} \right) = \left(\frac{80 - 31}{31} \right) \Rightarrow \frac{49}{31} = 1.562 \text{ şeklinde hesaplanır.}$$

Dikkat edilirse belirli bir kural içinde dikkat çekici dairesel yapılar oluşturulmaktadır. Son aşamada meydana gelen en dikkat çekici değişim, orta bölgede düzensiz bir örüntü oluşturulmasıdır. Bu örüntü balığın alt kısmında bulunan yüzgeç yardımı ile gerçekleştirilir.

Ayrıca merkez bölgede bulunan bu düzensiz örüntü ince kum taneleri yardımı ile gerçekleştirilir. Bu aşamada iç kısımdan dış bölgeye doğru her hangi bir yönelim gözlenmemektedir. Merkez bölge erkek balığın yüzgeç hareketleri ile Şekil 6.8’de gösterildiği gibi bir miktar daha genişletilir. Bu sayede ince kum tanecikleri dairesel yapının orta kısmına doğru bir yönelim göstermektedir. Bu aşamada ayrıca erkekler, radyal olarak hizalanmış dorukları kabuk ve mercan parçaları ile süslerler ve bu davranışın gerçekleştirilme sıklığı, diğer aşamalardan çok daha fazladır.

Dişilerin, erkek balıkların yuvasını sadece bu aşamada ziyaret ettikleri gözlemlenmiştir. Bir dişinin yuvaya yaklaşmasına bağlı olarak erkek balon balığı yuvanın merkez bölgesindeki kum parçacıklarını karıştırmaya devam etmektedir. Dişi balığın yuvaya girmesine bağlı olarak erkek, merkez bölgeye çekilerek dişiye doğru hızlı hamleler yapmaktadır. Dişi yuvada birkaç dakika durduktan sonra ayrılmakta ve tekrar aynı yuvayı sıklıkla ziyaret etmektedir.

Şekil 6.8’de balon balığının oluşturduğu dairesel yapının temsili bir gösterimi bulunmaktadır. Bu tez çalışması için önem arz eden bölgeler şekilde gösterildiği gibi detaylandırılmıştır [109].

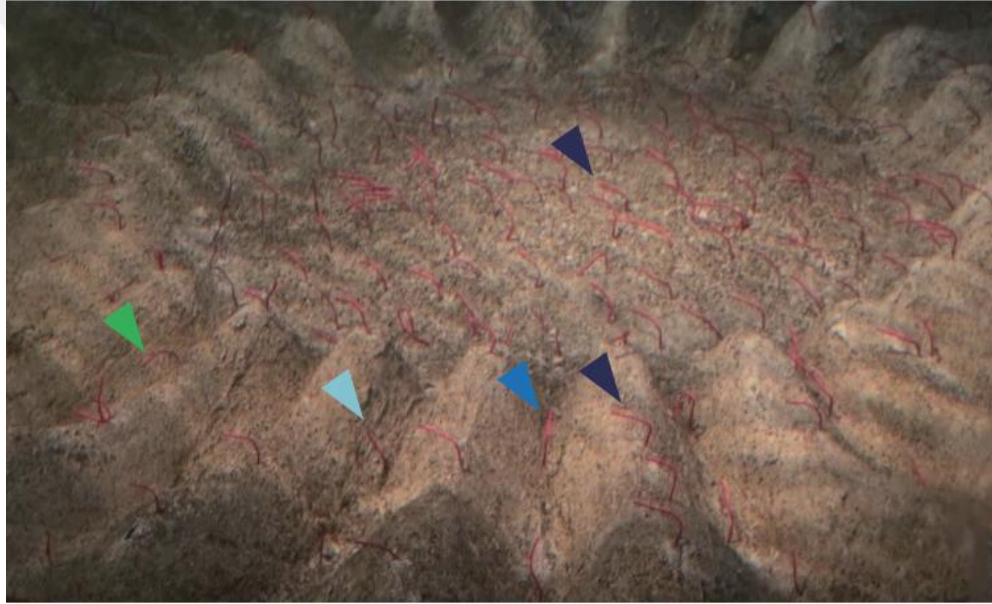


Şekil 6.8. Balon balığı tarafından yapılan dairesel yapının temsili gösterimi

Oluşturulan bu dairesel yapı yumurtlama sonrası akıntının etkisi ile çökmekte ve düzleşmektedir. Sonuç olarak dairesel yapının amacı karşı cinsi etkileyecek 2 metre çapında etkileyici bir mimari içeren bir yapı oluşturmaktadır. Yumurtalar yuvadan çıktıktan sonra erkek balık yuva bölgesinden ayrılmakta ve kısa bir süre sonra üreme döngüsü için gözlem alanına tekrar dönmektedirler.

Bu işlemlerden sonra erkek balon balığı eski yuva yerine geri dönmekte ve farklı bir alanda yeni bir dairesel yapı oluşturarak yaşam döngüsünü devam ettirmektedir. Bu etkileyici dairesel yapılar ile ilgili bir diğer sıra dışı fenomen ise bu yapıların su akışının dinamiğine olan etkisidir. Şekil 6.9’da dairesel yapının suyun akış dinamiğine etkisi üzerindeki etkisinin irdelenmesi amacıyla oluşturulmuş yapı gösterilmektedir.

İlgili şekilde akıntıya karşı yerleştirilen iplerin iç tarafa (mavi ok) döndüğü gözlemlenirken, cismin dış tarafında yerleştirilen iplerin (yeşil ok) akıntı yönünde bir yönelim gösterdiği gözlemlenmektedir. Tepenin üstünde ve dairesel yapının ortasında olan ipler (koyu mavi oklar) ise akıntı yönünde bir yönelim göstermektedir. Vadilerin ortasında bulunan ve akıntı yönünün tam ortasında bulunan iplerin (açık mavi ok) yönelimi ise tam olarak belli değildir.



Şekil 6.9. Dairesel yapı üzerinden su akış dinamiğinin incelenmesi [107]

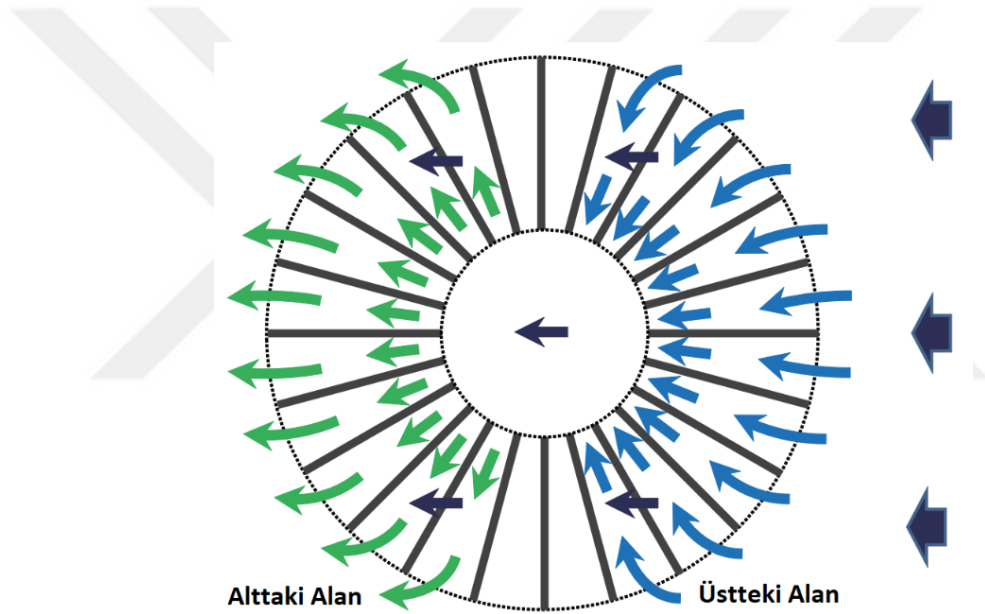
Bu bilgiler ışığında, su akışının vadilerin üst alanlarından merkezi bir yönde hareketi olduğu görülmektedir. Aynı şekilde merkezdeki akıntının ise vadilerden alt alanlara doğru bir yönelim gösterdiği gözlemlenmektedir.

Vadilerde üst ve alt alanlara ait su akışlarının orta noktalarında yer akışı oldukça az olmaktadır. Tepelerin üstündeki su akışının yönü ile dairesel yapının merkezindeki su akışının yönü aynıdır. Bütün bu bilgilere ek olarak, tez çalışmamız için önem arz eden bir gözlem daha ortaya çıkmaktadır. Bu sıradışı dairesel yapının merkez alanındaki su akış hızı dış alandaki su akış hızına göre 24.2% oranında daha azdır. Bu bağlamda ilgili dairesel yapının karşı cinsi etkilerken kalıcılığını da sağlayabilmesi için özel bir tasarım dili kullanıldığı söylenebilir.

Dairesel yapıyı oluşturan alanlar irdelendiğinde, merkez alanın farklı ve rassal bir örüntü ihtiva ettiği gözlemlenmektedir. Burada karşı cins için etkileyici olan önemli alanın merkez bölge olduğu söylenebilir.

Akış hızı üzerinden gözlemlenen değişim de bu sonucu destekleyecek niteliktedir. Merkez bölgedeki akış hızının yavaşlaması aynı zamanda bu alanda bağıl olarak ince kum taneleri ile oluşturulan örüntülerin daha kalıcı olmasını sağlayacaktır.

Şekil 6.10'da dairesel yapıda gerçekleşen su akış yönünün temsili gösterimi vardır. İlgili şekilde mavi oklar üst alandan merkeze olan akış yönünü, yeşil oklar ise merkezden alt alana olan suyun akış yönünü temsil etmektedir. Koyu renk ile gösterilen oklar ise tepelerin üstündeki suyun genel akış yönünü göstermektedir.

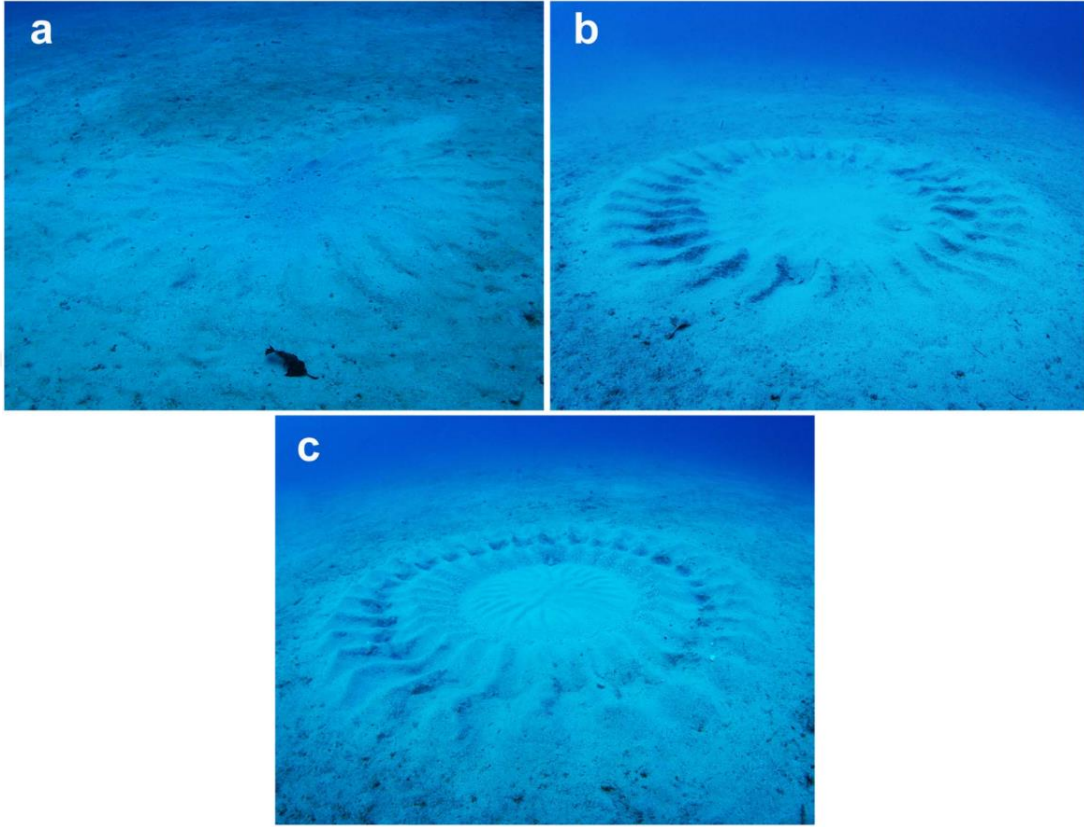


Şekil 6.10. Su akış yönünün temsili gösterimi [107]

Hiroshi Kawase ve arkadaşlarının Torquigener albomaculosus balık türüne ait dairesel yapılar üzerinde yaptığı araştırmalar sonucunda, üç aşamalı bir inşa ortaya çıkmaktadır. İlk aşamada (erken aşama) radyal olarak hizalanmış tepeler ve vadiler yuva alanının dış kısmında gerçekleştirilir.

İkinci aşamada (orta aşama), dairesel yapıya ait tepeler kabuk parçaları ile dekore edilir. Son aşamada ise yuvadan elde edilen ince kum taneleri ile düzgün olmayan örüntüler oluşturulur.

Bu üç aşamanın gerçekleştirilmesi süreci Şekil 6.11’de detaylı bir şekilde gösterilmektedir. İlgili şekilde, dairesel yapının gerçekleştirilmesi aşamalar ile gösterilmektedir. Aşağıdaki dairesel yapın, K1 etiketli balığın davranışlarının incelenmesi üzerinden gerçekleştirilmiştir.

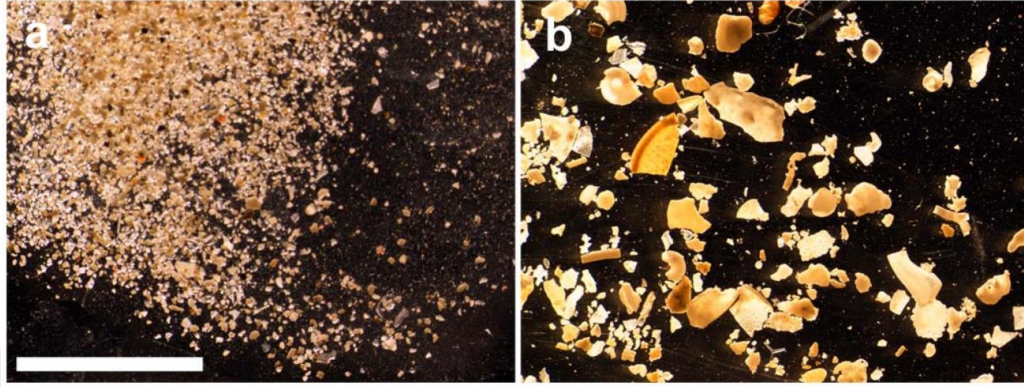


Şekil 6.11. a) Erken Aşama b) Orta Aşama c) Son Aşama [107]

Dişi balon balığının yuvayı ziyaretinden sonra, dairesel yapının çöküş süreci gerçekleşir. Oluşturulan dairesel yapının karakteristiği dişi balon balıklarının eş seçimi için önemli bir etkidir. Bu seçimi etkileyen çok farklı parametreler vardır. Bunlardan bazıları, dairesel yapının boyutu, simetri özelliği, tepelerin yükselti oranı, radyal olarak hizalanmış dorukların ve vadilerin sayısı, dekoratif cisimlerin boyutu, rengi, sayısı ve düzenlenmesi şeklindedir [110].

Dairesel yapıya ait önem arz eden bir diğer unsur ise elde edilen ince kumun miktarı ve kalitesidir. Belirtilen toprak yapısı ile bu etkileyici inşaatların orta kısmındaki düzensiz örüntülerin başarı ile gerçekleştirilmesine imkan sağlanabilecektir. Dahası radyal olarak hizalanmış ve yuvayı çevreleyen yapıların ince kum tanelerinin elde edilmesinde ve depolanmasında oldukça fazla önemi vardır.

Şekil 6.12’de yuva bölgesindeki kum tanelerinin fotomikrografları gösterilmiştir. Şekilde görüldüğü gibi erkek balon balığı dairesel yapıyı oluştururken farklı kalınlıklarda kum taneleri kullanmaktadır [107].



Şekil 6.12. a)Yumurtlamadan bir gün önce toplanan (Beyaz bar 5 mm) b)Yumurtlama sonrası

İlgili araştırmayı dişilerin etkilenmesi ve farkındalık boyutunda inceleyen çok sayıda da makale vardır. Fakat bu tez çalışmasında, oluşturulan dairesel yapıya ait nümerik bilgiler ve erkek balığın davranışı kullanılarak metasezgisel bir optimizasyon yöntemi önerilmiştir.

6.2. Balon Balığının Oluşturduğu Dairesel Yapıların İncelenmesi

Torquigener albomaculosus balık türünün gerçekleştirdiği dairesel yapılar belirli bir sistematik içinde oluşmaktadır. Bu şekilde merkezinde %24.2 oranında suyun akış hızını azaltan dairesel bir yapı ortaya çıkmaktadır. Ayrıca belirtilen yapı bazı özellikler barındırmaktadır. Bu özelliklerden bazıları aşağıda verilmiştir [111].

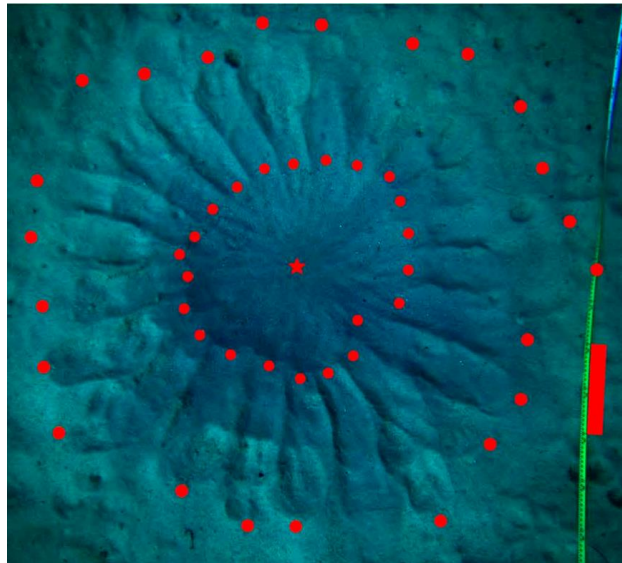
- Dairesel yapı su akıntılarına rağmen deniz tabanında mümkün olduğu kadar kalıcı olmalıdır.
- Tepe sayısının artması dairesel yapının kalıcı olmasını sağlamaktadır. Fakat yüksek sayıda tepe sayısı uzun bir uğraş sonucu ortaya çıkmaktadır.
- İç bölgedeki rassal örüntüler ince kum taneleri ile oluşturulmalıdır. Bu sayede oluşturulan yapı daha etkileyici bir görüntü oluşturacaktır.
- Dış bölge ve merkez bölge olacak şekilde ayrıştırılan bölgelerin yarıçapları arasında belirli bir oran olmalıdır.

Bu bilgiler ışığında, dairesel yapının merkez noktasının tıpkı gerçek hayattaki erkek balon balığının gerçekleştirdiği şekilde çeşitli seçimler sonucu belirlenmesi gibi değişken bir hale getirilmesi bu tez çalışmasına temel olan yeni bir metasezgisel optimizasyon yöntemine ortaya atılmasını sağlamıştır. Dairesel yapıya ait özniteliklerin nümerik hesaplamalarda çeşitli karşılıkları bulunmaktadır. Tepe sayısı, ilk aşamada hesaplanacak girdi sayısını temsil etmektedir. Kum yapısının değişkenliği(ince veya kalın) olması adım aralıklarını temsil etmektedir. İnce kum yapısı, küçük adım aralığını temsil etmektedir. Aynı şekilde kalın kum yapısı ise büyük adım aralığını temsil etmektedir.

İç ve dış yarıçaplar tıpkı gerçek dünyada olduğu gibi uygulanacak iki farklı karakteristiğe sahip problem uzayını temsil etmektedir. Rassal şekiller ise merkez bölgedeki davranışlar hakkında fikir vermektedir. Bu bağlamda algoritmanın gerçekleştirilmesi için bir takım parametrelere daha ihtiyaç vardır. Bunlardan bazıları problem uzayını daraltmak için yeni dairesel şeklin boyutunun belirli bir oranda küçültülmesi gerekmektedir [112]. Ayrıca merkez bölgede rassal yapıların belirli bir dağılım ışığında oluşturulması gerekmektedir [113].

Belirtilen dağılımı gerçekleştirmek için genel bir rassal dağılım yerine farklı bir karakteristiğe sahip bir dağılım gerekmektedir. Bu yüzden Levy uçuşu veya hareketi olarak adlandırılan dağılım, canlıların doğadaki besin arama sürecinin modellendiği optimizasyon algoritmalarında sıklıkla kullanılmaktadır. Bu dağılım ile dairesel yapıların merkez bölgesinde oluşturulan dairesel yapılar daha yüksek başarımla modellenilebilecektir.

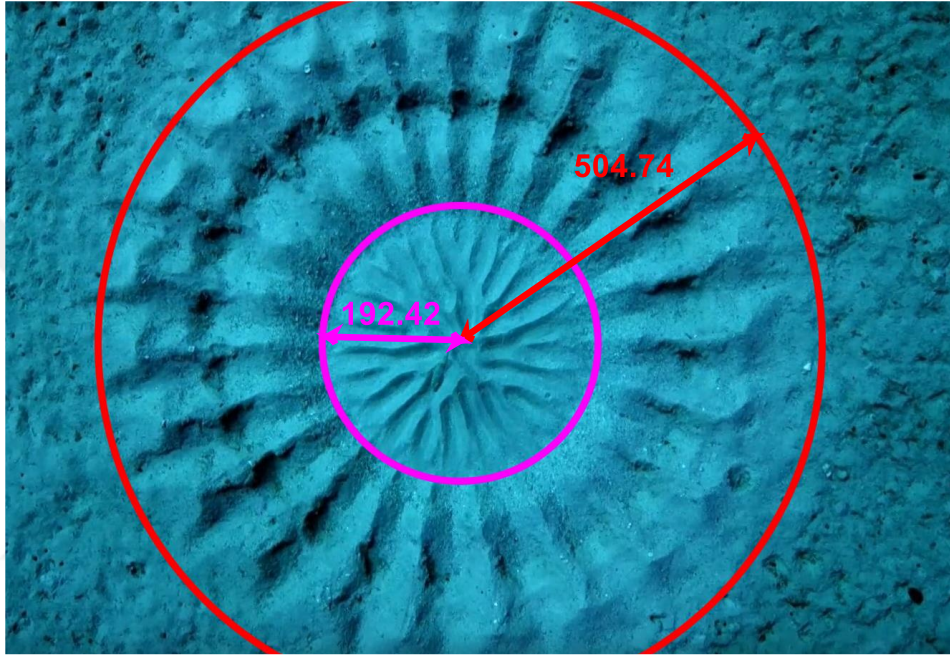
Şekil 6.13'de dairesel yapıya ait bir gösterim bulunmaktadır. İlgili yapıdaki tepe noktaları kırmızı ile işaretlenmiştir. Merkez noktası ise kırmızı yıldız ile belirtilmiştir.



Şekil 6.13. Erkek balon balığı tarafından gerçekleştirilen dairesel yapı (kırmızı bar=20 cm) [108]

İlgili dairesel yapılar daha önce belirtildiği gibi detaylı bir şekilde irdelenmişti. Fakat bu tez çalışmasında gözlemsel ölçümlerin yanı sıra imge işleme teknikleri ile dairesel yapıya ait örüntüler ve boyut bilgileri detaylı bir şekilde irdelenecektir. Bu şekilde oluşturulan örüntü hakkında daha kapsamlı bilgilere sahip olunacaktır.

Şekil 6.14’de örnek bir dairesel yapıya ait çemberler piksel değerleri üzerinden incelenmiştir [107].



Şekil 6.14.Farklı bölgelere ait yarıçap değerlerinin gösterilmesi

İlgili inceleme sonucunda dairelerin yarıçapları arasında daha önce belirtildiği gibi altın orana yakınsayan bir değer bulunmuştur. İlgili hesaplamalar Eşitlik 6.2’de gösterilmiştir.

$$c = \frac{R-r}{r} = \frac{504.7479 - 192.4237}{192.4237} = 1.6231 \quad (6.2)$$

Balon balığına ait dairesel yapılar yönteminin sözde (psuedo) kodlar, Tablo 6.1’de verilmiştir.

Tablo 6.1. Balon Balığının Dairesel Yapıları

Begin

Amaç fonksiyonu $f(x), x = (x_1, x_2, \dots, x_d)^T$

Başlangıç parametrelerini tanımla:

Tepe sayısı (t_s), maksimum dairesel yapı sayısı ($d_{\max}$)

Yarıçap küçültme oranı (α), Başlangıç yarıçapı ($r_{\text{başlangıç}}$),

Konum başlangıç bilgisini tanımla (x_i, y_i), Hata Kriteri (ε)

While ($d < d_{\max}$) veya hata kriterini sağla

Hesaplanmış maliyeti değerlendir.

Yarıçap değerini güncelle ($r_{\text{yeni}} = \alpha \times r_{\text{eski}}$)

Tepe konum bilgilerini kullanarak hesaplanmış maliyetleri değerlendir ($\text{adım}_{\text{dış}}$).

If ($r_{\text{yeni}} \leq \frac{r_{\text{başlangıç}}}{\beta}$)

Çözüme ait çözünürlük değeri artır ($\text{adım}_{\text{merkez}}$)

Merkez bölgede Levy dağılımı kullanarak rassal örüntüler oluştur

Bu rassal örüntülere karşılık gelen bölgelere ait maliyet değerlerini hesapla

End

Kum tepeleri vasıtasıyla merkez bölgeye yönelmiş desenler oluştur.

Maliyet değerlerini değerlendirerek en uygun konum bilgisini yeni başlangıç konumu olarak belirle

$l_{\text{best}_d} = l_{\text{initial}_{d+1}}$

$d++$

End while

İlgili sözde kodlarda, kullanıcı tarafından önceden tanımlanması gereken bir takım parametreler bulunmaktadır. Bunlar; tepe sayısı (t_s), maksimum dairesel yapı ($d_{\max}$) veya iterasyon ($it_{\max}$) sayısı, hata kriteri (ε), yarıçap küçültme oranı (α) ve çözüm uzayını kapsayacak şekilde belirlenmesi gereken başlangıç yarıçapı ($r_{\text{başlangıç}}$) değerleridir.

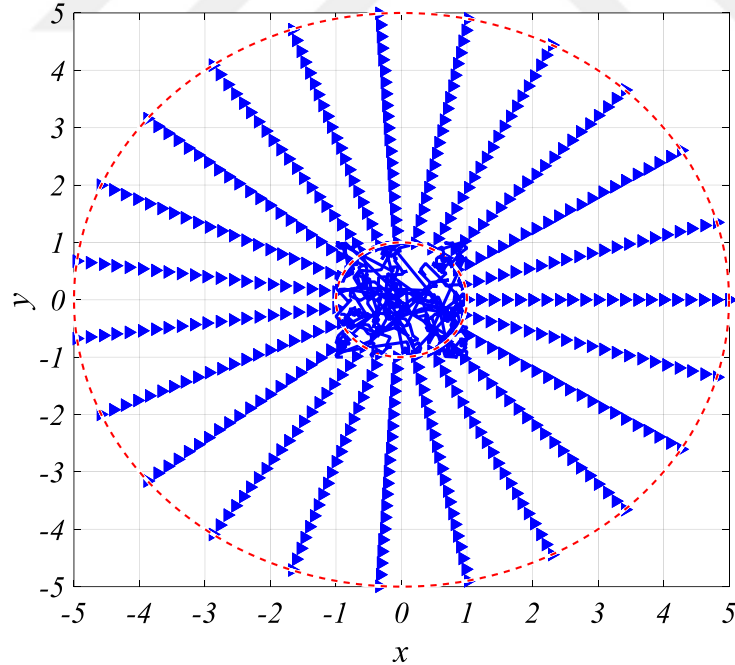
Bunların dışında dış ile merkez bölge arasındaki yarıçap oranı gerçek değerler ile benzerlik göstermesi için ($\beta = 2.625$) olarak seçilmiştir. Ayrıca dış ve merkez bölgedeki kumların boyutları arasındaki oran

$$\frac{adım_{merkez}}{adım_{dış}} = \frac{5}{1} \text{ olacak şekilde adım boyutu veya hesaplama çözünürlüğü olarak yöntem}$$

uyarlanmıştır. Bu sayede merkez bölgede daha hassas hesaplamalar yapılarak daha başarılı bir şekilde genel minimum değerlerinin yakınsanması sağlanmıştır. Yarıçap küçültme oranı (α) bu yöntemde, bir çeşit hassasiyet parametresi olarak tanımlanmıştır. Düşük (α) değeri hesaplama maliyetini artırmakta fakat daha başarılı sonuçlar vermekte iken yüksek (α) hesaplama maliyetini azaltmakta fakat çözüm kalitesini düşürmektedir.

Şekil 6.15’de balon balığına ait dairesel yapılardan esinlenerek ortaya çıkartılmış davranışı modelleyen şekil gösterilmektedir. İlgili şekilde ki ölçeklendirmeler gerçek dairesel yapıdan esinlenerek oluşturulmuştur. Dış dairede bulunan kesikli kırmızı çizgiler, dış sınırlara ait bölgeyi temsil etmektedir. İç kısımda bulunan kesikli kırmızı çizgiler ise rassal örüntünün oluşturulduğu bölgeyi temsil etmektedir. İlgili şekildeki tüm oranlar, adım aralıkları, tepe sayısı gibi değerler gerçek dairesel yapılardan alınmıştır.

Şekilde $x=[-5,5]$ ve $y=[-5,5]$ aralığında tanımlı bir bölge için gerçekleştirilmiş dairesel yapı gösterilmiştir.



Şekil 6.15. Balon Balığına ait dairesel yapıya ilişkin çizim

7. BALON BALIĞININ DAİRESEL YAPILARI YÖNTEMİ

Tez çalışmasının bu aşamasında önerilen yöntemin literatürde bulunan diğer metasezgisel optimizasyon yöntemleri ile olan karşılaştırılması yapılmıştır. Bu yöntemler sırasıyla, genetik algoritması (GA), karınca kolonisi yöntemi (KKA), parçacık sürü yöntemi (PSO), yapay arı kolonisi yöntemidir (YAK). İlgili yöntemler, literatürde bulunan ve yöntemlerin başarımını test etmek amacıyla sıklıkla kullanılan fonksiyonlar üzerinden performans analizine tabi tutulmuştur. Balon balığına ait dairesel yapılar yöntemi (BBDY) kısaltması ile tanımlanmıştır. Tablo 7.1’de test sürecinde kullanılan metasezgisel optimizasyon yöntemlerine ilişkin parametreler verilmiştir. Balon balığının dairesel yapılarında gösterilen başlangıç kum tepesi sayısı, diğer yöntemlerdeki popülasyon sayısına benzerlik göstermektedir. Yöntemlerin performansları, test fonksiyonların tanımlı olduğu aralıktaki 1000 farklı başlangıç noktası üzerinden irdelenmiştir.

Tablo 7.1. Performans Analizi Yapılan Yöntemlere İlişkin Parametreler

Genetik Algoritması (GA)	Karınca Kolonisi (KKA)	Parçacık Sürü (PSO)	Yapay Arı Kolonisi (YAK)	Balon balığı Dairesel Yapıları (BBDY)
Mutasyon Olasılığı=0.1	Yoğunlaşma Faktörü=0.5	Atalet Ağırlığı=1	$Limit = 29 \approx 0.6 \times 2 \times 24$	$\frac{Dış\ yarıçap}{İç\ yarıçap} = 2.625$
Çaprazlama Olasılığı=0.9	Mesafe Sapma Oranı=1	Atalet Ağırlığı Sönümlenme Oranı=0.99	Gözcü arı sayısı=12;	$\frac{Dış\ adım}{İç\ adım} = 5$
İterasyon Sayısı=50	İterasyon Sayısı=50	Kişisel Öğrenme Katsayısı=1.5	İşçi Arı Sayısı=12	$\alpha = 0.7$ (Yarıçap Küçültme Oranı)
Popülasyon Sayısı=24	Popülasyon Sayısı=24	Genel Öğrenme Katsayısı=2	Kaşif Arı Sayısı=1;	İterasyon Sayısı=50
		İterasyon Sayısı=50	İterasyon Sayısı=50	Başlangıç Kum Tepesi Sayısı(Popülasyon)=24
		Popülasyon Sayısı=24	Popülasyon Sayısı=24	

Tabloda görüldüğü gibi popülasyon sayısı ve iterasyon sayısı bütün yöntemler için aynı alınmıştır. Diğer parametreler literatürde sıklıkla kullanılan veya önerilen parametreler üzerinden seçilmiştir. Hata değeri tüm yöntemler için ($\varepsilon = 10^{-16}$) olarak seçilmiştir.

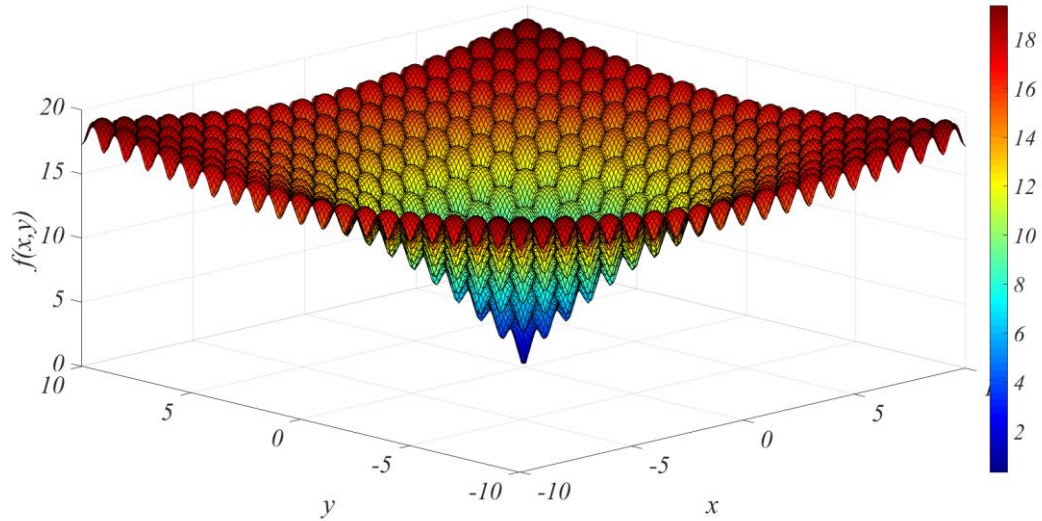
7.1. Balon Balığının Dairesel Yapıları Yöntemi ve Performans Analizi

Önerilen yöntem ilk aşamada, Ackley fonksiyonu üzerinden analiz edilmiştir. Ackley fonksiyonunun değişimi Şekil 7.1’de gösterilmiştir. Ackley fonksiyonun matematiksel tanımı Eşitlik 7.1’de gösterilmiştir.

$$f(x_1, \dots, x_n) = -a \exp\left(-b \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{d} \sum_{i=1}^n \cos(cx_i)\right) + a + \exp(1)$$

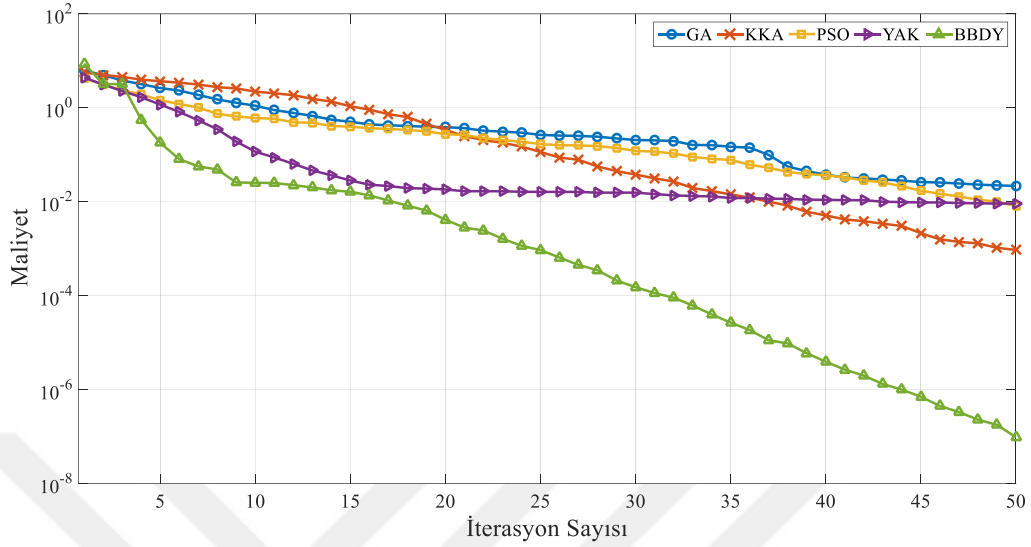
$$n = 2, a = 20, b = 0.2 \text{ ve } c = 2\pi$$

$$\min(f(0,0)) = 0 \text{ ve } x_i = [-32.768, 32.768]$$
(7.1)



Şekil 7.1. Ackley Fonksiyonu

Şekil 7.2’de yöntemlere ait maliyet değerlerinin (Ackley) iterasyona bağlı değişimi gösterilmiştir.



Şekil 7.2. Maliyet fonksiyonunun (Ackley) iterasyon sayısına bağlı değişimi

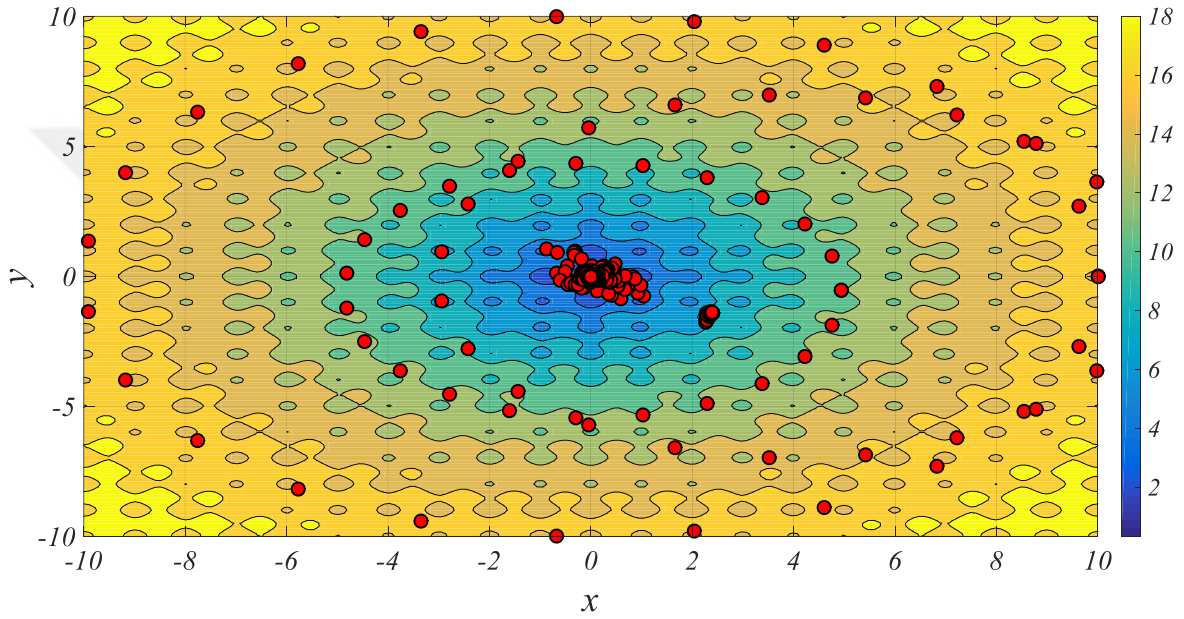
Ackley fonksiyonuna ait performans sonuçları maliyet fonksiyonunun değişimi üzerinden Tablo 7.2’de gösterilmektedir.

Tablo 7.2. Ackley Fonksiyonuna ait performans sonuçları

Yöntemin Adı	GA	KKA	PSO	YAK	BBDY
Ortalama Değer	0.757143867	0.985387772	0.479493940	0.308311672	0.323358411
Standart Sapma	1.293633803	1.528643151	0.826110015	0.845061145	1.357154924
Minimum	0.021071727	0.000940903	0.008049962	0.009141214	0.000000097
Maksimum	6.209772233	5.674273550	4.243884172	4.421479737	8.640585760
Hesaplama Süresi (saniye)	0.063582165	0.094538341	0.198038961	0.287973641	0.020777331

Şekil 7.2 ve Tablo 7.2’de görüldüğü gibi tez çalışması kapsamında önerilen yöntemin hem zaman hem performans değerleri bakımından diğer yöntemlere daha başarılı olduğu açıkça gözlemlenmektedir. Şekil 7.3’de BBDY yöntemi ile optimize edilen Ackley fonksiyonu ve eşyükselti eğrisine ait gösterim bulunmaktadır.

İlgili şekilde kırmızı noktalar, BBDY yöntemi ile hesaplanan noktaları göstermektedir. Şekilde görüldüğü gibi BBDY yöntemi ile global minimum değeri başarı ile yakınsanmaktadır.

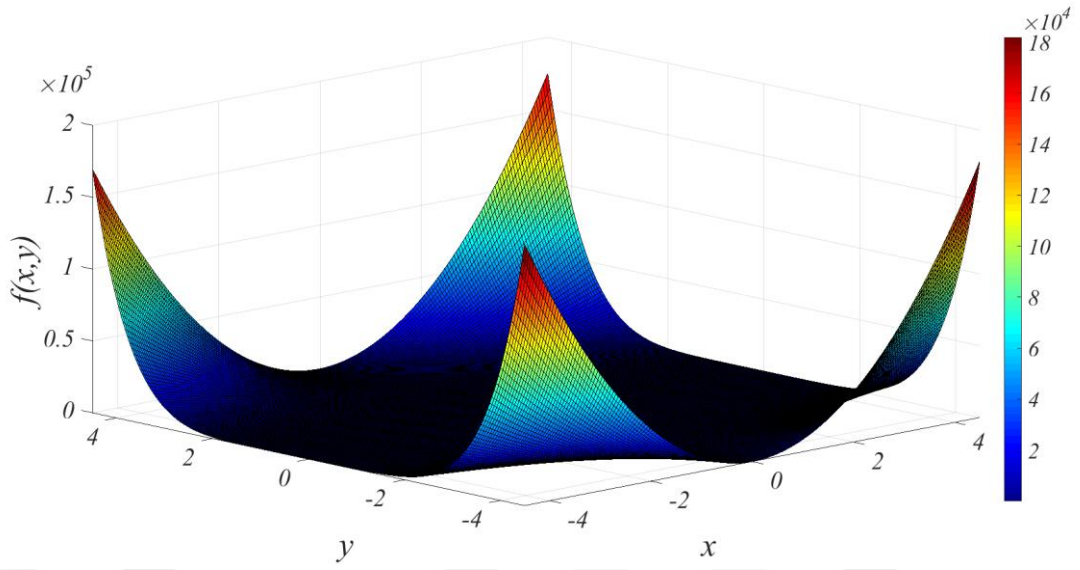


Şekil 7.3. Ackley Fonksiyonu için Dairesel Yapıların Değişimi

Önerilen yöntem ikinci aşamada, Beale fonksiyonu üzerinden analiz edilmiştir. Fonksiyonun matematiksel tanımı Eşitlik 7.2’de, değişimi ise Şekil 7.4’de verilmiştir.

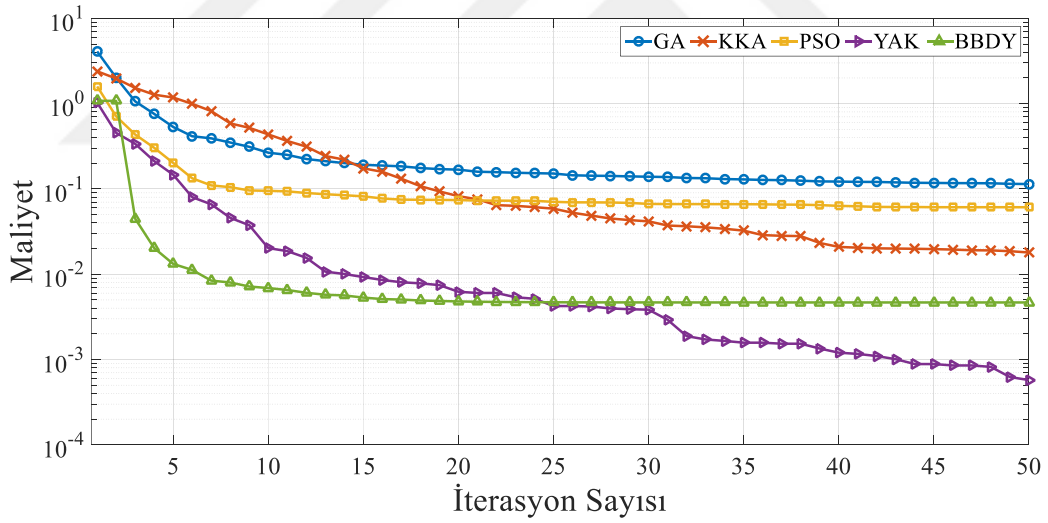
$$f(x, y) = (1.5 - x + xy)^2 + (2.25 - x + xy^2)^2 + (2.625 - x + xy^3)^2$$

$$\min(f(3, 0.5)) = 0 \text{ ve } x_i = [-4.5, 4.5] \quad (7.2)$$



Şekil 7.4. Beale Fonksiyonu

Şekil 7.5’de yöntemlere ait maliyet değerlerinin iterasyona bağlı değişimi gösterilmiştir.



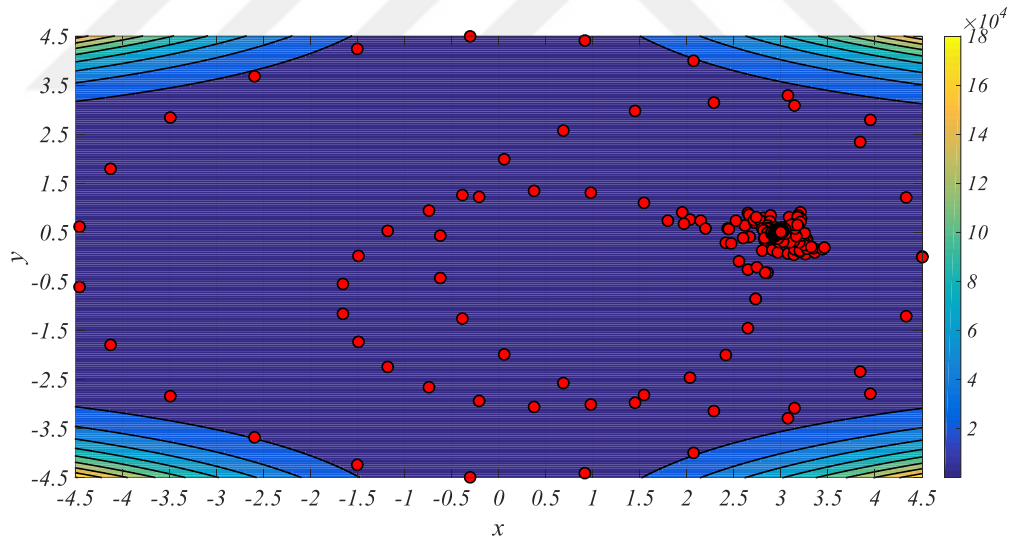
Şekil 7.5. Maliyet fonksiyonunun (Beale) iterasyon sayısına bağlı değişimi

Beale fonksiyonuna ait performans sonuçları maliyet fonksiyonunun değişimi üzerinden Tablo 7.3’de gösterilmektedir.

Tablo 7.3. Beale Fonksiyonuna ait performans sonuçları

Yöntemin Adı	GA	KKA	PSO	YAK	BBDY
Ortalama Değer	0.317850414	0.292679578	0.131273852	0.052090713	0.049411156
Standart Sapma	0.618365306	0.526794789	0.236823412	0.164580236	0.212205882
Minimum	0.113132362	0.017981063	0.060990956	0.000574572	0.004667255
Maksimum	4.086170322	2.393319746	1.590842750	1.029667120	1.078125000
Hesaplama Süresi (saniye)	0.052253282	0.080197372	0.167987968	0.249770197	0.015355722

Şekil 7.6’da BBDY yöntemi ile optimize edilen Beale fonksiyonu ve eşyükselti eğrisine ait gösterim bulunmaktadır. Şekilde gösterilen kırmızı noktalar BBDY yöntemi ile hesaplanan bölgeleri göstermektedir. Şekilde görüldüğü gibi $\min(f(3,0.5))=0$ değerine başarı ile yakınsanmaktadır.



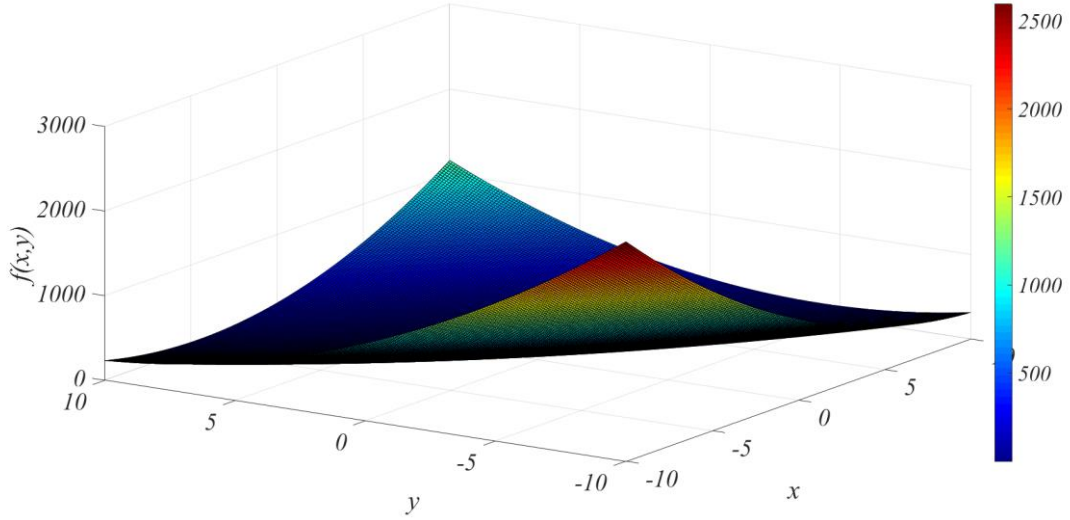
Şekil 7.6. Beale Fonksiyonu için Dairesel Yapıların Değişimi

İlgili şekil ve performans sonuçları incelendiğinde BBDY yönteminin global minimum değerini diğer yöntemlere göre daha hızlı bir şekilde yakınsadığı gözlemlenmektedir. Önerilen yöntem ikinci aşamada, Booth fonksiyonu üzerinden analiz edilmiştir. Fonksiyonun matematiksel tanımı Eşitlik 7.3’de, değişimi ise Şekil 7.7’de verilmiştir.

$$f(x, y) = (x + 2y - 7)^2 + (2x + y - 5)^2$$

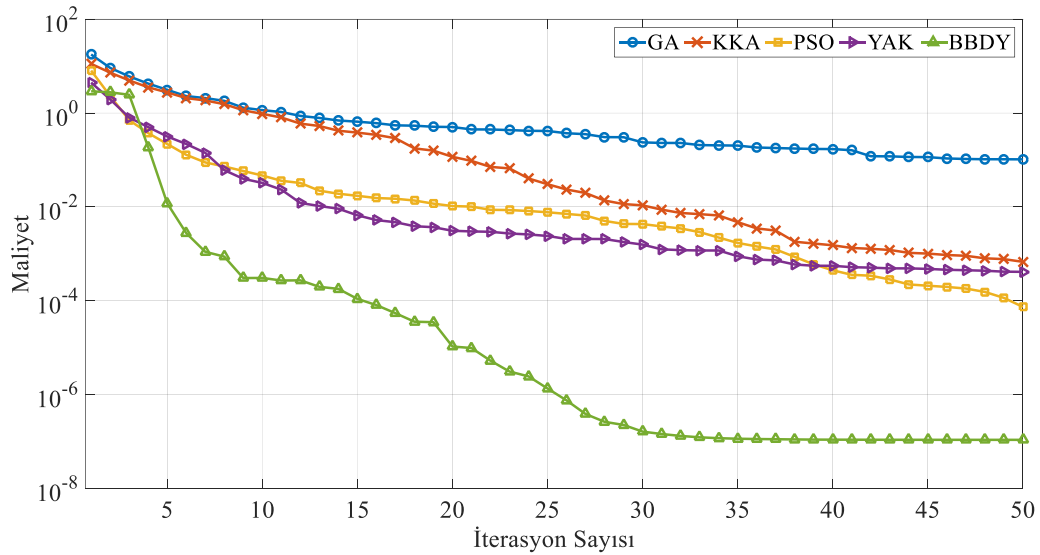
$$\min(f(1, 3)) = 0 \text{ ve } x_i = [-10, 10]$$

(7.3)



Şekil 7.7. Booth Fonksiyonu

Şekil 7.8’de yöntemlere ait maliyet değerlerinin (Booth) iterasyona bağlı değişimi gösterilmiştir.



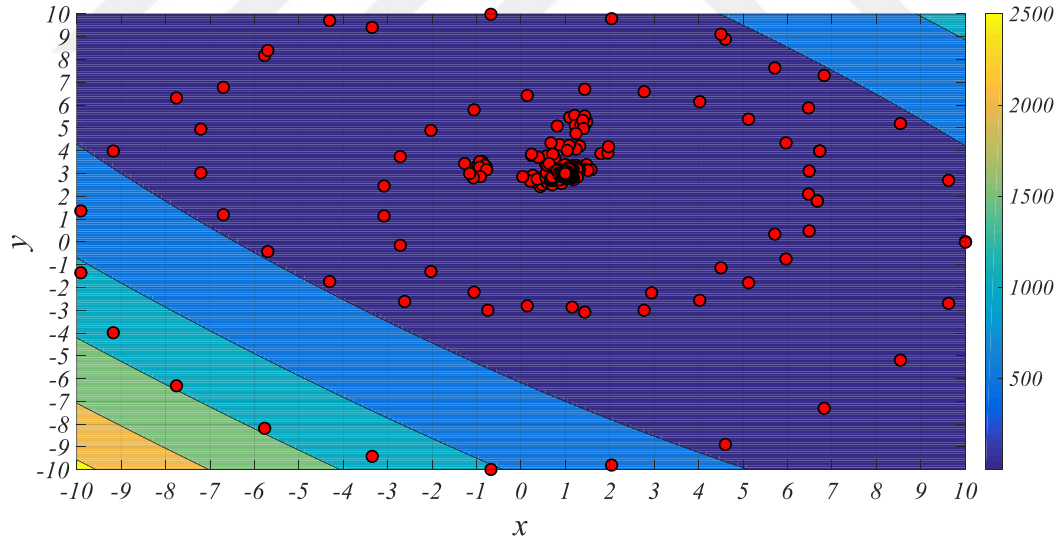
Şekil 7.8. Maliyet fonksiyonunun (Booth) iterasyon sayısına bağlı değişimi

Booth fonksiyonuna ait performans sonuçları maliyet fonksiyonunun değişimi üzerinden Tablo 7.4’de gösterilmektedir.

Tablo 7.4. Booth Fonksiyonuna ait performans sonuçları

Yöntemin Adı	GA	KKA	PSO	YAK	BBDY
Ortalama Değer	1,231734649	0,833710543	0,247377175	0,173645019	0,167215370
Standart Sapma	2,863093767	2,039656278	1,179803338	0,692157013	0,653311063
Minimum	0,087962254	0,000674960	0,000073887	0,000412919	0,000000108
Maksimum	17,747999173	11,231413856	8,074182613	4,476604780	2,920649758
Hesaplama Süresi (saniye)	0,050385137	0,075602002	0,157720842	0,240142760	0,015991347

Şekil 7.9’da BBDY yöntemi ile optimize edilen Booth fonksiyonu ve eşyükselti eğrisine ait gösterim bulunmaktadır. Şekilde gösterilen kırmızı noktalar BBDY yöntemi ile hesaplanan bölgeleri göstermektedir. Şekilde görüldüğü gibi $\min(f(1,3)) = 0$ değeri başarı ile yakınsanmaktadır.

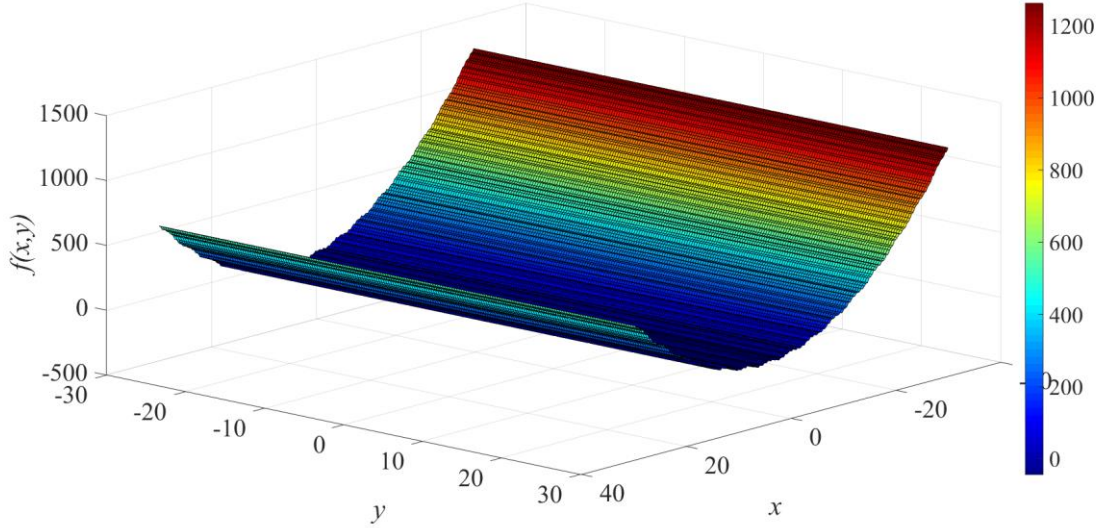


Şekil 7.9. Booth Fonksiyonu için Dairesel Yapıların Değişimi

Önerilen yöntem sonraki aşamada, Chichinadze fonksiyonu üzerinden analiz edilmiştir. Chichinadze fonksiyonuna matematiksel tanımı Eşitlik 7.4’de gösterilmiştir. Chichinadze fonksiyonunun değişimi Şekil 7.10’da gösterilmiştir.

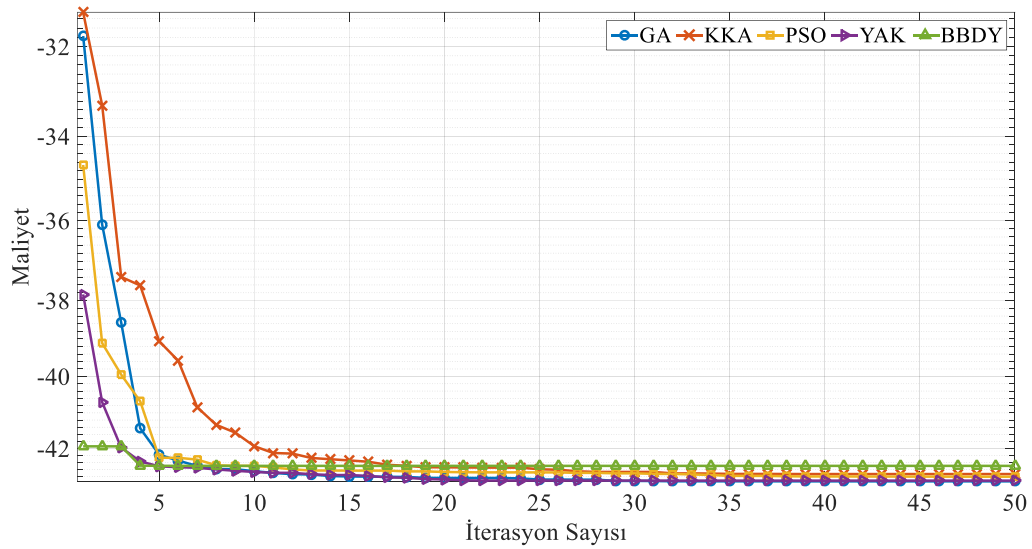
$$f(x, y) = x^2 - 12x + 8\sin\left(\frac{5}{2}\pi x\right) + 10\cos\left(\frac{1}{2}\pi x\right) + 11 - 0.2 \frac{\sqrt{5}}{e^{-\frac{1}{2}(y-0.5)^2}} \quad (7.4)$$

$$\min(f(6.189866, 0.5)) = -42.94438701 \text{ ve } x_i = [-30, 30]$$



Şekil 7.10. Chichinadze Fonksiyonu

Şekil 7.11’de yöntemlere ait maliyet değerlerinin (Chichinadze) iterasyona bağlı değişimi gösterilmiştir.



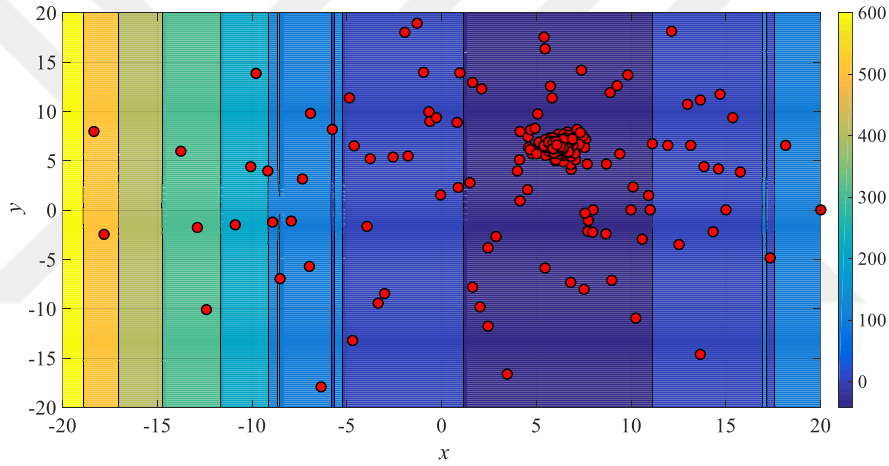
Şekil 7.11. Maliyet fonksiyonunun (Chichinadze) iterasyon sayısına bağlı değişimi

Chichinadze fonksiyonuna ait performans sonuçları maliyet fonksiyonunun değişimi üzerinden Tablo 7.5’de gösterilmektedir.

Tablo 7.5. Chichinadze Fonksiyonuna ait performans sonuçları

Yöntemin Adı	GA	KKA	PSO	YAK	BBDY
Ortalama Değer	-42,3780409	-41,7617804	-42,3549873	-42,6793243	-42,4634964
Standart Sapma	1,8897445	2,3069675	1,3089857	0,7811840	0,1337420
Minimum	-42,9443323	-42,7331496	-42,8096659	-42,9235283	-42,4971782
Maksimum	-31,7729466	-31,2647508	-34,6688089	-37,8416008	-41,9394630
Hesaplama Süresi (saniye)	0,0509133	0,0773506	0,1580293	0,2460726	0,0226007

Şekil 7.12’de Chichinadze fonksiyonunun çözüm sürecine ilişkin dairesel yapıların değişimi gösterilmektedir.

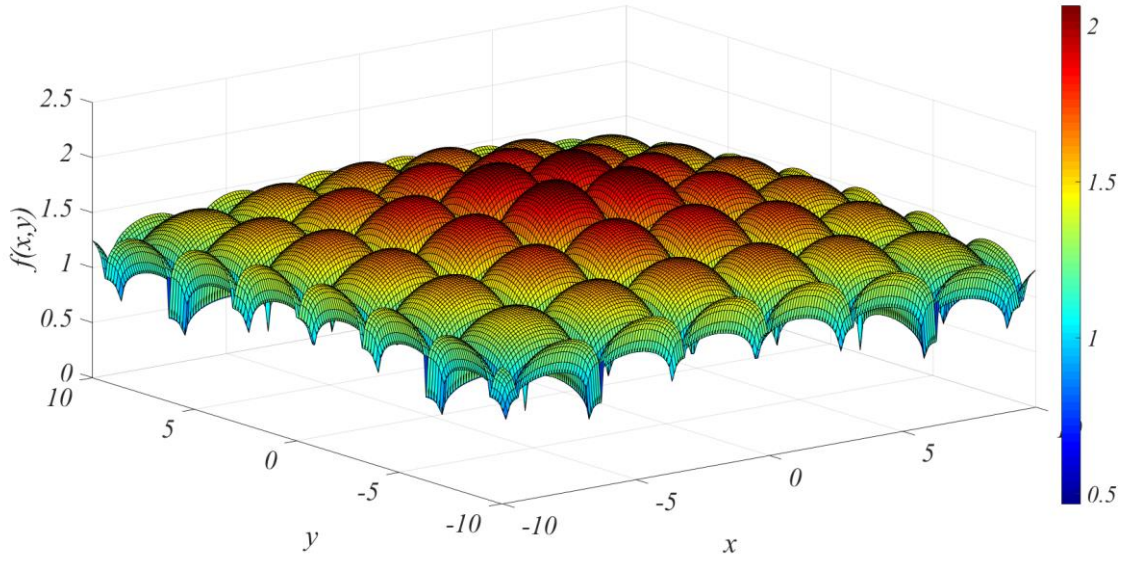


Şekil 7.12. Chichinadze Fonksiyonu için Dairesel Yapıların Değişimi

Önerilen yöntem sonraki aşamada, Crownedcross fonksiyonu üzerinden analiz edilmiştir. Crownedcross fonksiyonunun değişimi Şekil 7.13’de gösterilmiştir. Crownedcross fonksiyonuna matematiksel tanımı Eşitlik 7.6’da gösterilmiştir.

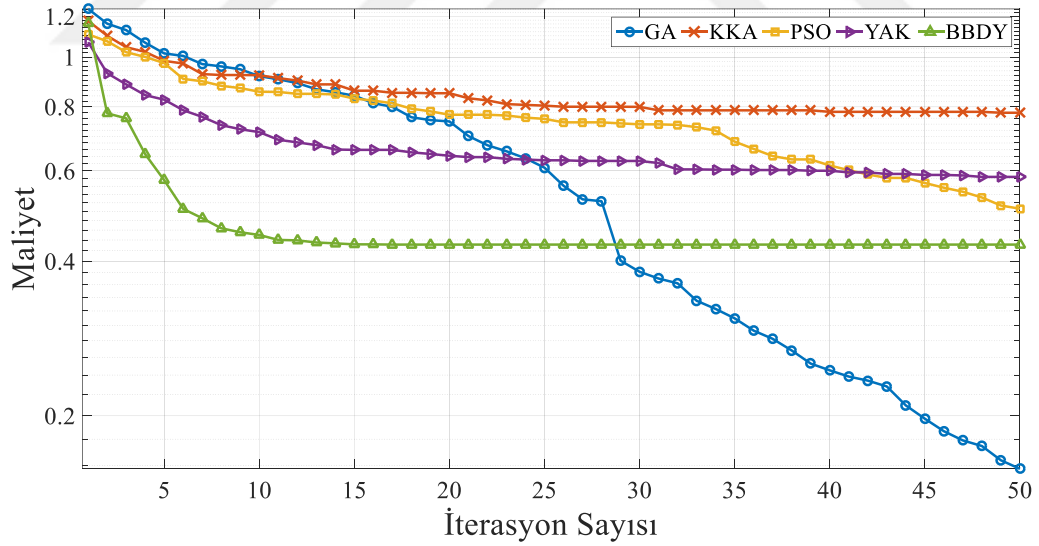
$$f(x, y) = 0.0001 \left(\left| e^{\left| 100 - \frac{\sqrt{x^2 + y^2}}{\pi} \right|} \sin(x) \sin(y) \right| + 1 \right)^{0.1} \quad (7.6)$$

$\min(f(0,0)) = 0.0001$ ve $x_i = [-10, 10]$



Şekil 7.13. Crownedcross Fonksiyonu

Şekil 7.14’de yöntemlere ait maliyet değerlerinin (Crownedcross) iterasyona bağlı değişimi gösterilmiştir.



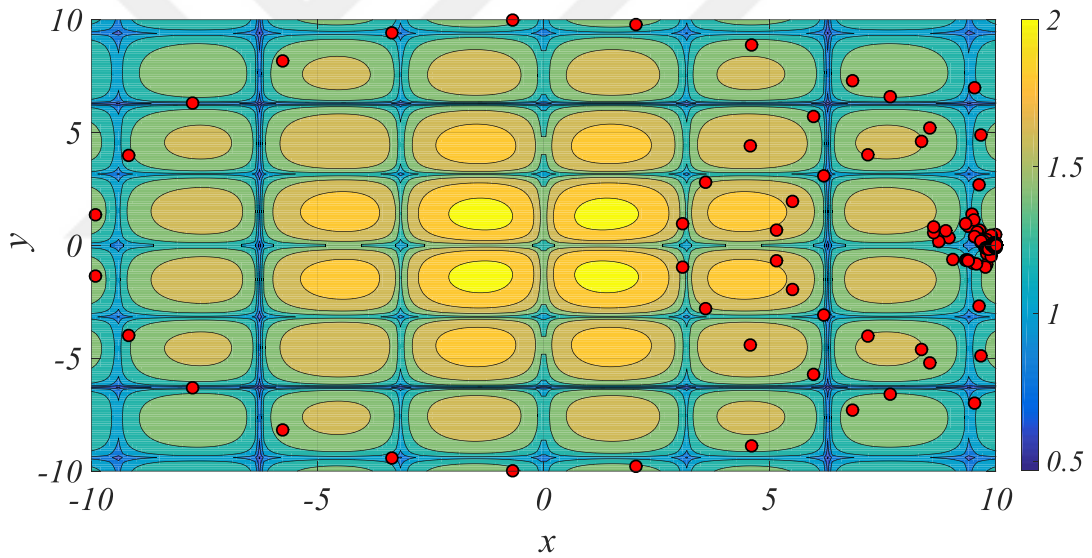
Şekil 7.14. Maliyet fonksiyonunun (Crownedcross) iterasyon sayısına bağlı değişimi

Crownedcross fonksiyonuna ait performans sonuçları maliyet fonksiyonunun değişimi üzerinden Tablo 7.6’da gösterilmektedir.

Tablo 7.6. Crownedcross Fonksiyonuna ait performans sonuçları

Yöntemin Adı	GA	KKA	PSO	YAK	BBDY
Ortalama Değer	0,578037611	0,849019796	0,754346382	0,663024823	0,470982285
Standart Sapma	0,328345758	0,091018868	0,144049209	0,099315535	0,124726352
Minimum	0,157092109	0,778842048	0,506360016	0,584004171	0,431049513
Maksimum	1,241036951	1,176512243	1,107219266	1,069211831	1,163489827
Hesaplama Süresi (saniye)	0,058425458	0,080009549	0,161810432	0,25221979	0,016335637

Şekil 7.15’de BBDY yöntemi ile optimize edilen Crownedcross fonksiyonu ve eşyüksekti eğrisine ait gösterim bulunmaktadır. Şekilde gösterilen kırmızı noktalar BBDY yöntemi ile hesaplanan bölgeleri göstermektedir.

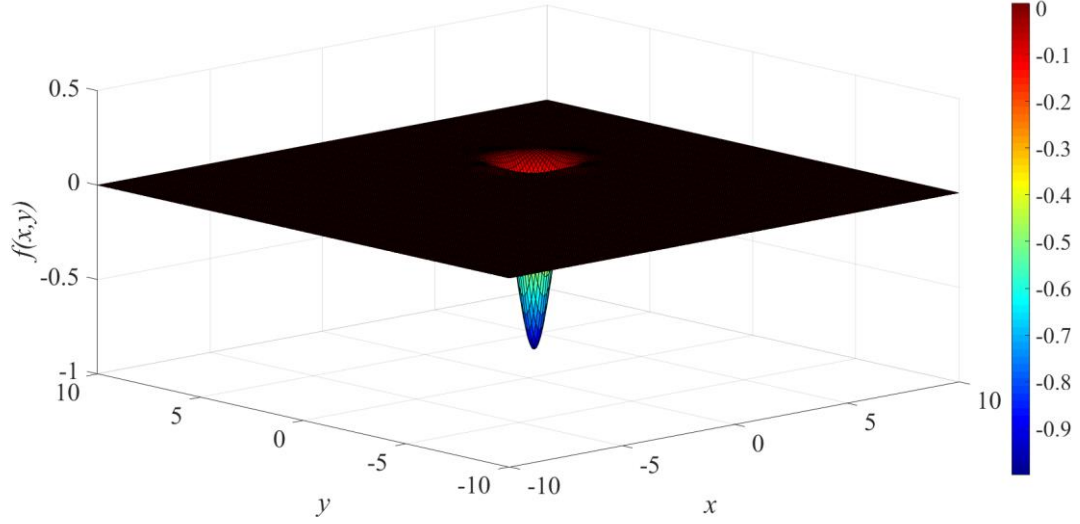


Şekil 7.15. Crownedcross Fonksiyonu için Dairesel Yapıların Değişimi

Önerilen yöntem sonraki aşamada, Easom fonksiyonu üzerinden analiz edilmiştir. Easom fonksiyonuna matematiksel tanımı Eşitlik 7.7’de gösterilmiştir.

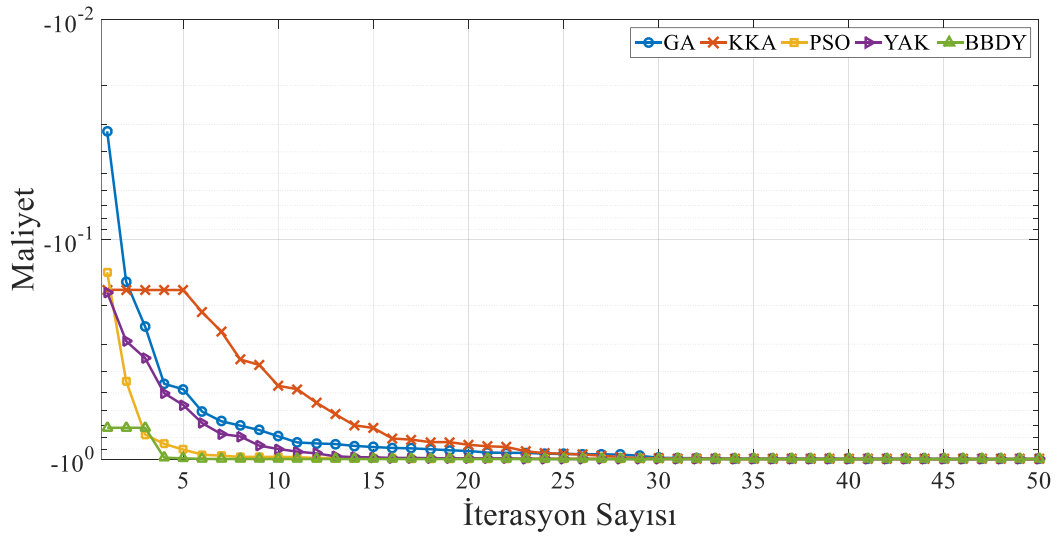
$$f(x, y) = -\cos(x) \cos(y) \exp(-(x - \pi)^2 - (y - \pi)^2)$$
$$\min(f(\pi, \pi)) = -1 \text{ ve } x_i = [-100, 100]$$
(7.7)

Easom fonksiyonunun değişimi Şekil 7.16’da gösterilmiştir.



Şekil 7.16. Easom Fonksiyonu

Şekil 7.17’de yöntemlere ait maliyet değerlerinin (Easom) iterasyona bağlı değişimi gösterilmiştir.



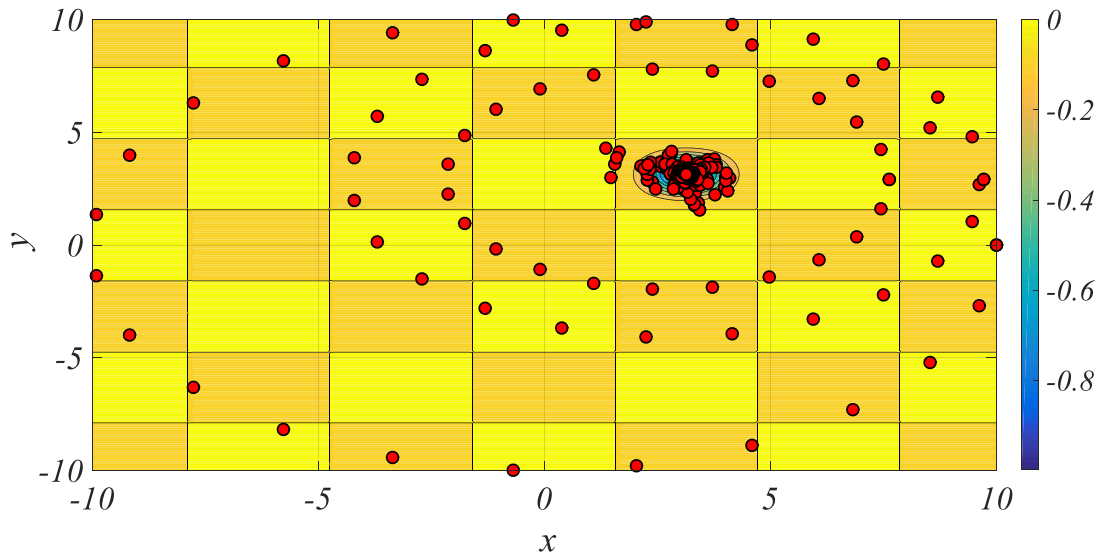
Şekil 7.17. Maliyet fonksiyonunun (Easom) iterasyon sayısına bağlı değişimi

Easom fonksiyonuna ait performans sonuçları maliyet fonksiyonunun değişimi üzerinden Tablo 7.7’de gösterilmektedir.

Tablo 7.7. Easom Fonksiyonuna ait performans sonuçları

Yöntemin Adı	GA	KKA	PSO	YAK	BBDY
Ortalama Değer	-0,865505074	-0,783705112	-0,956610529	-0,91252199	-0,982688022
Standart Sapma	0,223241857	0,297335003	0,14582312	0,196802875	0,066984993
Minimum	-0,9996095	-0,999994667	-0,999967943	-0,999976099	-1
Maksimum	-0,032244211	-0,16976038	-0,141251182	-0,174417106	-0,720369575
Hesaplama Süresi (saniye)	0,053211544	0,077774669	0,168575212	0,248784263	0,017559056

Şekil 7.18’de BBDY yöntemi ile optimize edilen Easom fonksiyonu ve eşyükselti eğrisine ait gösterim bulunmaktadır. Şekilde gösterilen kırmızı noktalar BBDY yöntemi ile hesaplanan bölgeleri göstermektedir. Şekilde görüldüğü gibi $\min(f(\pi,\pi))=-1$ değeri başarı ile yakınsanmaktadır.



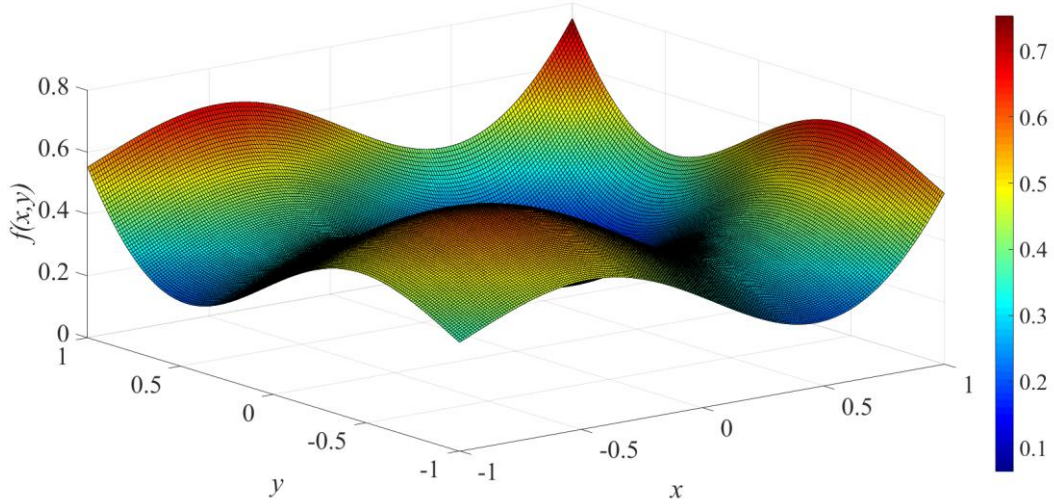
Şekil 7.18. Easom Fonksiyonu için Dairesel Yapıların Değişimi

Önerilen yöntem sonraki aşamada, Giunta fonksiyonu üzerinden analiz edilmiştir. Giunta fonksiyonuna matematiksel tanımı Eşitlik 7.8’de gösterilmiştir.

$$f(x, y) = 0.6 + \sum_{i=1}^n \left[\sin^2\left(1 - \frac{16}{15}x\right) - \frac{1}{50} \sin\left(4 - \frac{64}{15}x\right) - \sin\left(1 - \frac{16}{15}x\right) \right] \quad (7.8)$$

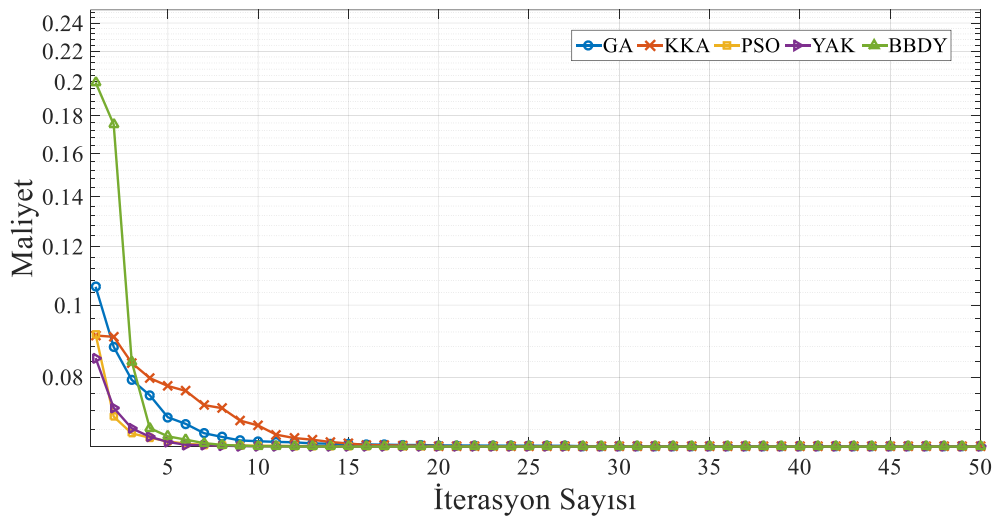
$$\min(f(0.46732, 0.46732)) = 0.06447042 \quad \text{ve} \quad x_i = [-1, 1]$$

Giunta fonksiyonunun değişimi Şekil 7.19’da gösterilmiştir.



Şekil 7.19. Giunta Fonksiyonu

Şekil 7.20’de yöntemlere ait maliyet değerlerinin (Giunta) iterasyona bağlı değişimi gösterilmiştir.



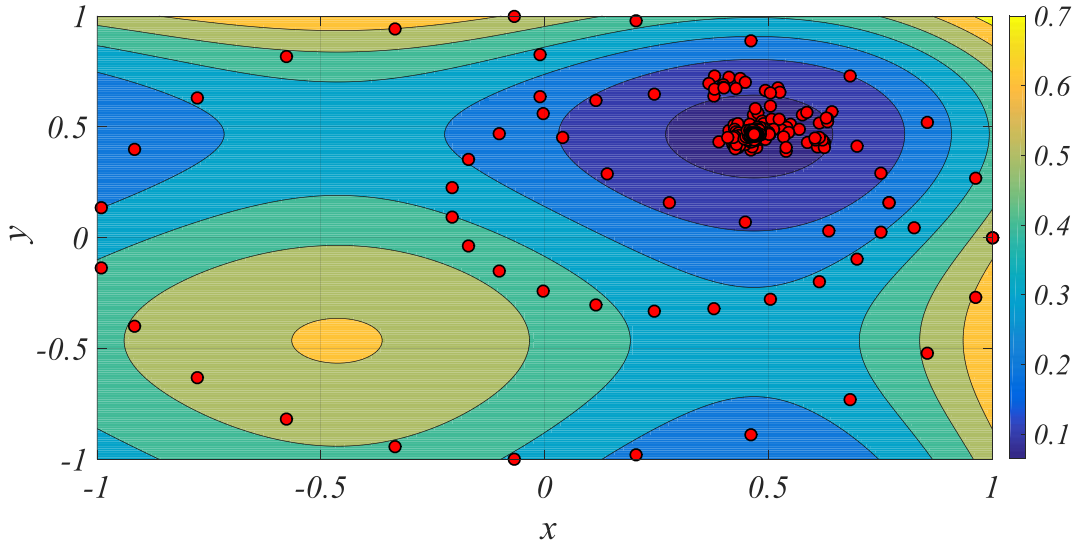
Şekil 7.20. Maliyet fonksiyonunun (Giunta) iterasyon sayısına bağlı değişimi

Giunta fonksiyonuna ait performans sonuçları maliyet fonksiyonunun değişimi üzerinden Tablo 7.8’de gösterilmektedir.

Tablo 7.8. Giunta Fonksiyonuna ait performans sonuçları

Yöntemin Adı	GA	KKA	PSO	YAK	BBDY
Ortalama Değer	0,066729004	0,067422862	0,065294714	0,065196766	0,069946694
Standart Sapma	0,006955208	0,006516171	0,003884449	0,00311447	0,024496669
Minimum	0,064470606	0,064470423	0,064470538	0,064470573	0,064470421
Maksimum	0,105849928	0,090986615	0,091336774	0,084840312	0,199546881
Hesaplama Süresi (saniye)	0,057675763	0,084308361	0,178862127	0,27736736	0,01108026

Şekil 7.21’de BBDY yöntemi ile optimize edilen Easom fonksiyonu ve eşyüksekti eğrisine ait gösterim bulunmaktadır. Şekilde gösterilen kırmızı noktalar BBDY yöntemi ile hesaplanan bölgeleri göstermektedir. Şekilde görüldüğü gibi $\min(f(0.4673,0.4673))=0.06447042$ değeri başarı ile yakınsanmaktadır.



Şekil 7.21. Giunta Fonksiyonu için Dairesel Yapıların Değişimi

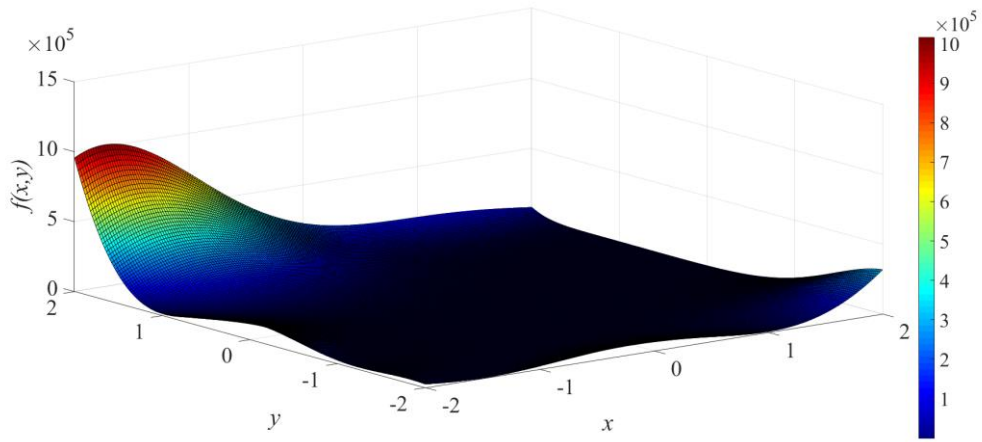
Önerilen yöntem sonraki aşamada, Goldstein-Price fonksiyonu üzerinden analiz edilmiştir. Goldstein-Price fonksiyonuna matematiksel tanımı Eşitlik 7.9’da gösterilmiştir.

$$f(x, y) = [1 + (x + y + 1)^2(19 - 14x + 3x^2 - 14y + 6xy + 3y^2)] \dots$$

$$\dots \times [30 + (2x - 3y)^2(18 - 32x + 12x^2 + 48y - 36xy + 27y^2)] \quad (7.9)$$

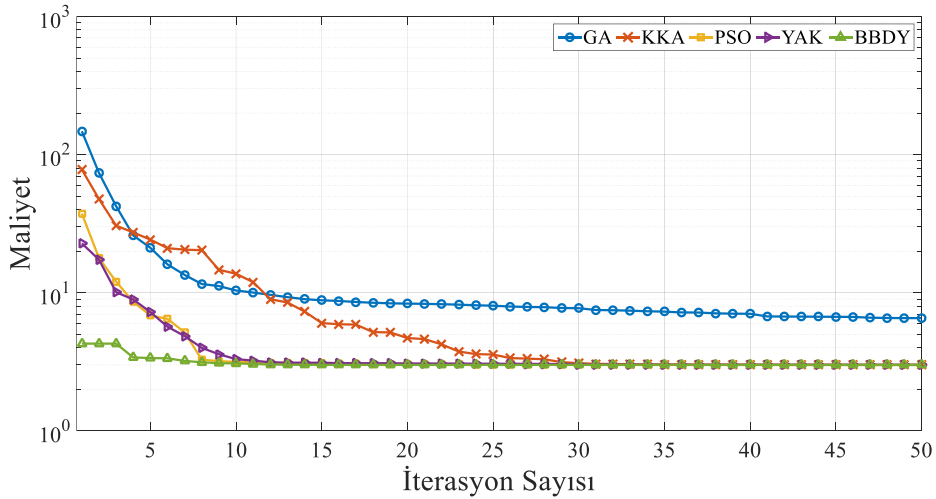
$$\min(f(0, -1)) = 3 \text{ ve } x_i = [-2, 2]$$

Goldstein-Price fonksiyonunun değişimi Şekil 7.22’de gösterilmiştir.



Şekil 7.22. Goldstein-Price Fonksiyonu

Şekil 7.23’de yöntemlere ait maliyet değerlerinin (Goldstein-Price) iterasyona bağlı değişimi gösterilmiştir.



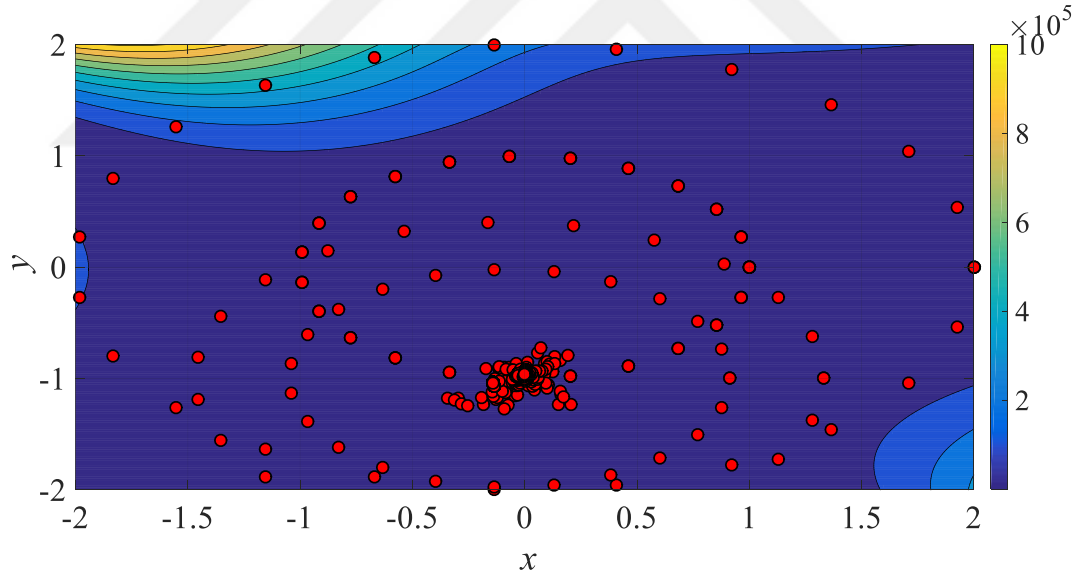
Şekil 7.23. Maliyet fonksiyonunun (Goldstein-Price) iterasyon sayısına bağlı

Goldstein-Price fonksiyonuna ait performans sonuçları maliyet fonksiyonunun değişimi üzerinden Tablo 7.9’da gösterilmektedir.

Tablo 7.9. Goldstein-Price Fonksiyonuna ait performans sonuçları

Yöntemin Adı	GA	KKA	PSO	YAK	BBDY
Ortalama Değer	13,49735085	9,276603119	4,500431728	4,193553282	3,111066614
Standart Sapma	21,95839518	13,42881219	5,416303248	3,618908037	0,312868012
Minimum	6,543245311	3,000110062	3,000174941	3,005925601	3,00000346
Maksimum	147,6467332	77,93042115	37,4383654	22,78509734	4,280645444
Hesaplama Süresi (saniye)	0,054544152	0,07927905	0,165958513	0,26442085	0,0144147

Şekil 7.24’de dairesel yapıların (Goldstein-Price) iterasyona bağlı değişimi gösterilmiştir.

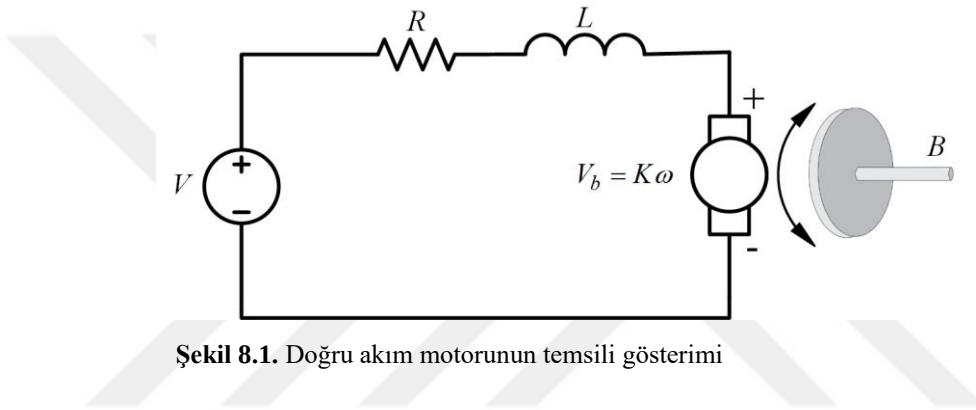


Şekil 7.24. Goldstein-Price Fonksiyonu için Dairesel Yapıların Değişimi

8. BALON BALIĞININ DAİRESEL YAPILARI İLE KONTROLÖR TASARIMI

Tez çalışmasının bu bölümünde, balon balığının dairesel yapıları yaklaşımı ile oransal ve integral (*PI*) kontrolör parametrelerinin ayarlanması süreci irdelenmiştir. İlk aşamada, bir doğru akım motorunun giriş gerilimi ile açısal hızı arasındaki transfer fonksiyonu elde edilmiştir. Sonraki aşamada, elde edilen transfer fonksiyonu, *PI* kontrolöre ait oransal (*P*) ve integral (*I*) değerleri, uygun bir maliyet fonksiyonu tanımlanarak BBDY yöntemi ile belirlenmiştir [114].

Doğru akım motoruna ilişkin şematik gösterim Şekil 8.1’de verilmiştir.



Doğru akım motora ilişkin tork değeri armatür akımı (*i*) arasındaki ilişki Eşitlik 8.1’de gösterildiği gibidir.

$$T = Ki(t) \quad (8.1)$$

Zıt elektromotor kuvveti (emf), V_b ile açısal hız arasındaki ilişki Eşitlik 8.2’de gösterilmiştir.

$$V_b(t) = K\omega(t) = K \frac{d\theta(t)}{dt} \quad (8.2)$$

Şekil 8.1’de gösterilen doğru akım motoruna istinaden Kirchhoff ile Newton kanunun birleşimi sonucu Eşitlik 8.3 ve 8.4’de gösterilen denklem takımları elde edilmektedir.

$$J \frac{d^2\theta(t)}{dt^2} + B \frac{d\theta(t)}{dt} = Ki(t) \quad (8.3)$$

$$L \frac{di}{dt} + Ri = V - K \frac{d\theta}{dt} \quad (8.4)$$

Laplace dönüşümü kullanılarak yukarda gösterilen eşitlikler, Eşitlik 5 ve 6'da gösterildiği gibi yazılabilir.

$$Js^2\theta(s) + bs\theta(s) = KI(s) \quad (8.5)$$

$$LsI(s) + RI(s) = V(s) - Ks\theta(s) \quad (8.6)$$

Doğru akım motoruna ait akım değeri Eşitlik 8.7'de gösterildiği gibi elde edilmektedir.

$$I(s) = \frac{V(s) - Ks\theta(s)}{R + Ls} \quad (8.7)$$

Eşitlik 8.5 kullanılarak, Eşitlik 8.8'de gösterilen genel tanım bağıntısı elde edilir.

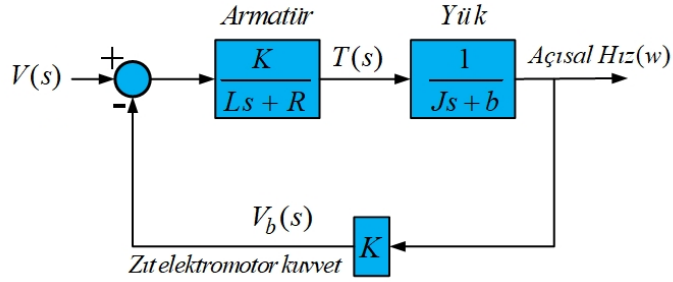
$$Js^2\theta(s) + bs\theta(s) = K \frac{V(s) - Ks\theta(s)}{R + Ls} \quad (8.8)$$

Doğru akım motorunun transfer fonksiyonunu elde edilmesi için bir takım sabitlere ihtiyaç duyulmaktadır. Bu parametreler Tablo 8.1'de gösterilmiştir.

Tablo 8.1. Doğru akım motoruna ait parametreler

Rotorda bulunan eylemsizlik momenti	$J=0.01 \text{ kg.m}^2$
Mekanik sistemdeki sönümlenme(sürtünme)	$b=0.1 \text{ Nms}$
Zıt elektromotor kuvvet sabiti	$K=0.01 \text{ Nm/A}$
Elektriksel Direnç	$R=1\Omega$
Elektriksel İndüktans	$L=0.5 \text{ H}$

İlgili sisteme ilişkin transfer fonksiyonu Şekil 8.2’de gösterilmiştir.



Şekil 8.2. Doğru akım motora ait transfer fonksiyonu

Eşitlik 8.9’da giriş gerilimi ($V(s)$) ile açı değeri ($\theta(s)$) arasındaki transfer fonksiyonu gösterilmiştir.

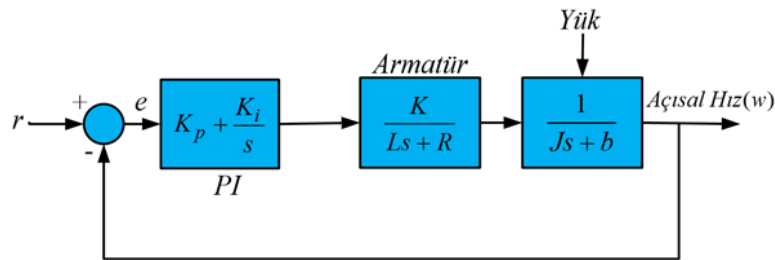
$$G_a(s) = \frac{\theta(s)}{V(s)} = \frac{K}{s[(R + Ls)(Js + b) + K^2]} \quad (8.9)$$

Eşitlik 8.10’da giriş gerilimi ($V(s)$) ile açısal hız değeri ($\omega(s)$) arasındaki transfer fonksiyonu gösterilmiştir.

$$G_v(s) = \frac{\omega(s)}{V(s)} = \frac{K}{(R + Ls)(Js + b) + K^2} \quad (8.10)$$

İlgili sistemin giriş gerilimi ile açısal hız değeri arasındaki ilişkiyi kontrol etmek amacıyla bir PI kontrolör tasarlanacaktır. Bu aşamada sisteme ait kontrolörün başarı ile tasarlanabilmesi için anlamı bir maliyet fonksiyonunun tanımlanması gerekmektedir.

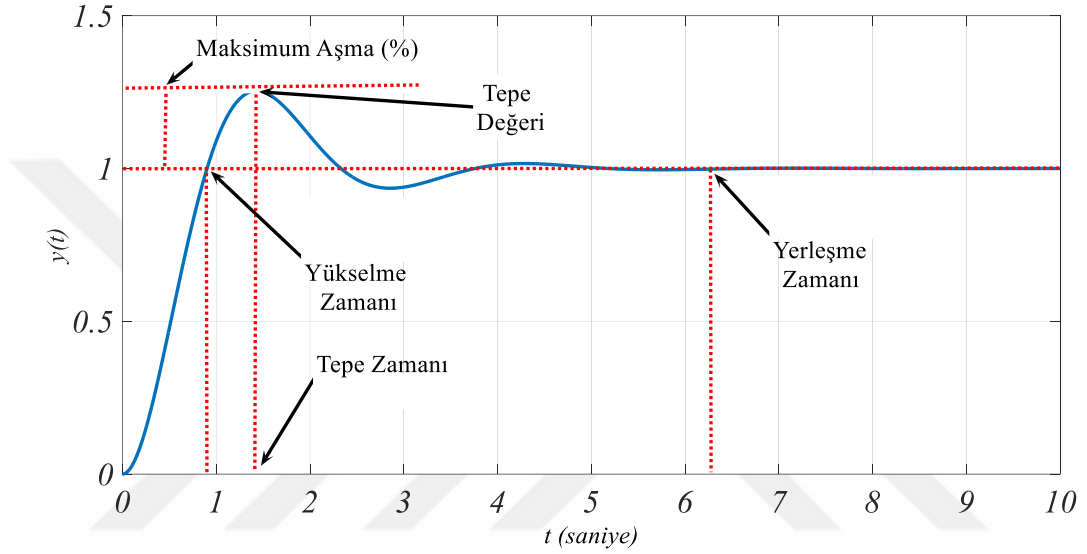
Şekil 8.3’de PI kontrolör yapısı ile doğru akım motoruna ilişkin blok diyagramı verilmiştir.



Şekil 8.3. Kapalı çevrim hız kontrolüne ilişkin blok diyagramı

PI kontrolöre ait parametrelerin optimizasyonu için ilk aşamada, anlamlı bir maliyet fonksiyonu seçilmelidir [34]. Literatürde genellikle, birim basamak cevaba ait belirli parametrelerin minimize edilmesi üzerinden belirli bir sisteme ait birim basamak cevabı optimize edilmektedir [115].

İkinci mertebeden düşük sönümlü ($0 < \zeta < 1$) bir sisteme ait birim basamak cevabı ve performans kriterleri Şekil 8.4’de gösterilmektedir.

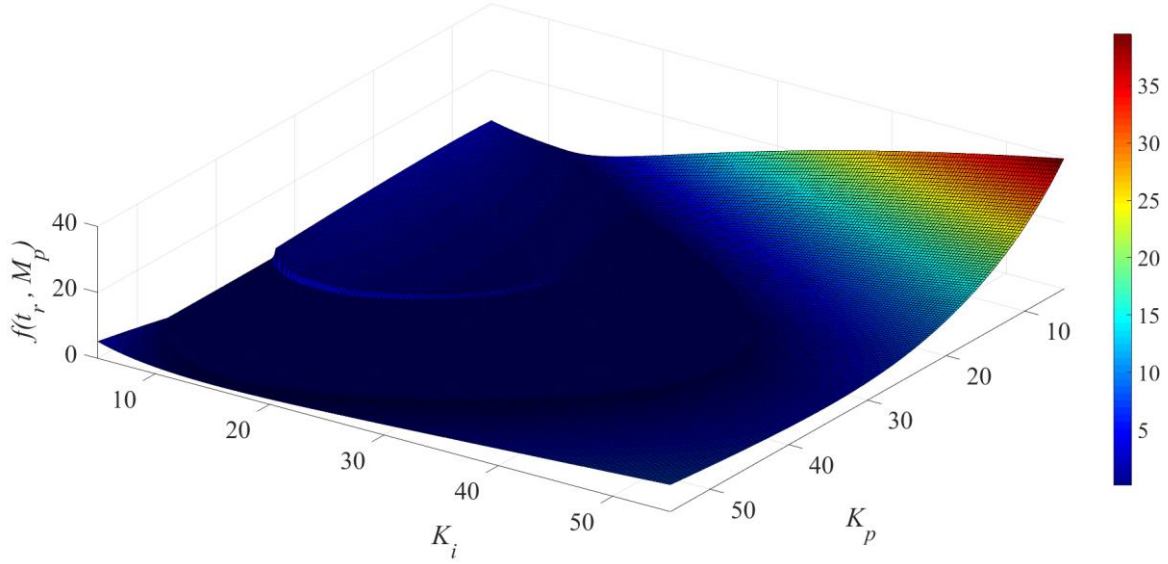


Şekil 8.4. Birim basamak cevabı ve analiz parametreleri

Kontrolör parametrelerin başarı ile optimizasyonu için anlamlı bir maliyet fonksiyonunun tanımlanması gerekmektedir. Bu parametreler genellikle sistemin birim basamak cevabına ait parametreler üzerinden gerçekleştirilir [116]. Bu tez çalışmasında ise deneysel sonuçlar ışığında en etkin maliyet fonksiyonunun Eşitlik 8.11’de gösterildiği gibi tanımlanması sonucu bulunmuştur. Bu parametreler, yükselme zamanı (t_r), maksimum aşma değeri (M_p) üzerinden tanımlanmıştır. İlgili problem için problem uzayı, K_p ve K_i parametrelerinin değerleri [5,55] arasında olacak şekilde sınırlandırılmıştır. Farklı metasezgisel optimizasyon yöntemleri için tüm parametreler, Tablo 7.1’de gösterildiği gibi seçilmiştir.

$$f(t_r, M_p) = [t_r + M_p] \quad (8.11)$$

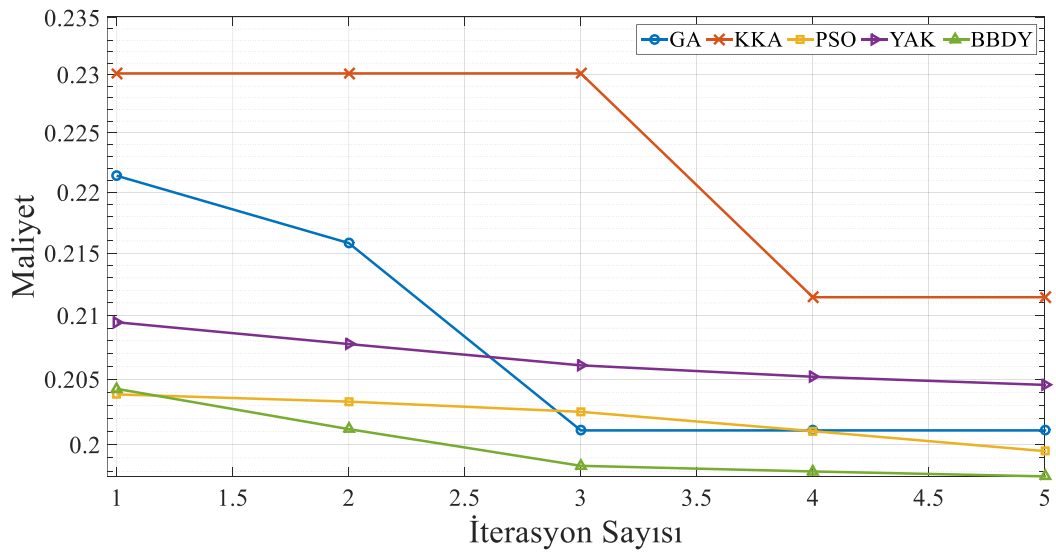
Şekil 8.5’de problemin maliyet fonksiyonunun ($f(t_r, M_p)$), K_p ve K_i parametreleri üzerinden modellenmesi gösterilmiştir.



Şekil 8.5. PI kontrolör maliyet uzayı

Tez çalışması kapsamında önerilen yönteminin gerçek zamanlı uygulamalardaki başarımını da irdelemek için, iterasyon sayısı (it)=5 olacak şekilde seçilmiştir ve yöntemler detaylı bir şekilde analiz edilmiştir. Tablo 8.2’de performans analiz sonuçlar gösterilmektedir.

Şekil 8.6’da maliyet fonksiyonlarının iterasyon sayısına bağlı değişimi gösterilmiştir.



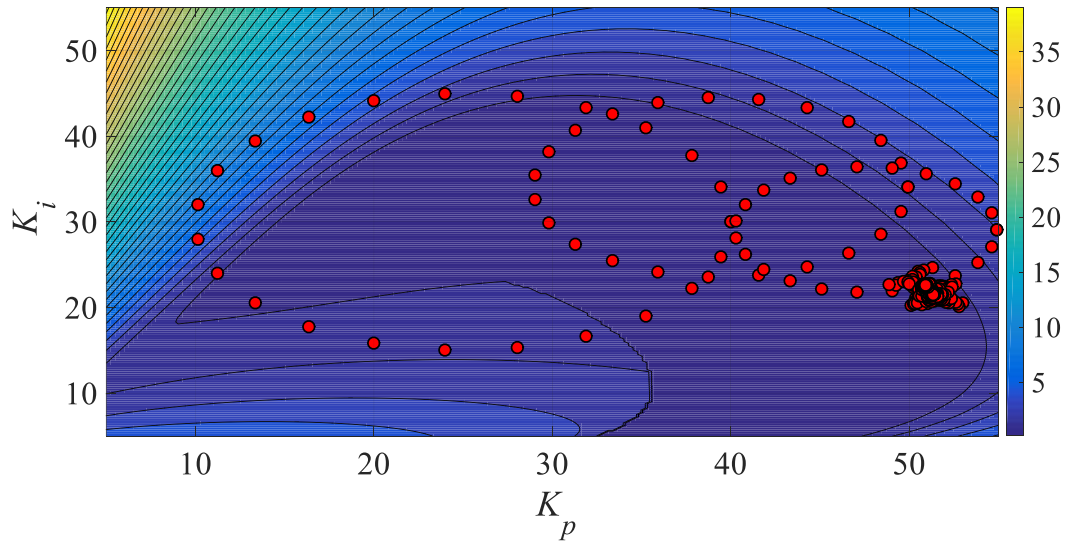
Şekil 8.6. Maliyet fonksiyonunun (PI) iterasyon sayısına bağlı değişimi

Tablo 8.2. PI kontrolör optimizasyon sonuçları

Yöntemin Adı	GA	KKA	PSO	YAK	BBDY
Ortalama Değer	0,20809013	0,22264524	0,20209240	0,20653962	0,19970052
Standart Sapma	0,00979674	0,01021635	0,00194291	0,00220095	0,00327654
Minimum	0,20108103	0,21145379	0,19996405	0,20493227	0,19768691
Maksimum	0,22138303	0,23010620	0,20351130	0,20895063	0,20546481
Hesaplama Süresi (saniye)	3,78512	4,34531	4,61329	8,06892	2,58486

Tablo 8.2 ve Şekil 8.6 irdelendiğinde BBDY yönteminin ve yüksek başarımla maliyeti azalttığı gözlemlenmiştir. Alternatif yöntemlere göre yüksek oranında bir hız artımı söz konusudur.

Şekil 8.7’de dairesel yapıların iterasyon sürecine bağlı değişimi gösterilmiştir.



Şekil 8.7. PI kontrolör maliyet fonksiyonu için dairesel yapıların değişimi

Şekil 8.7 irdelendiği zaman, optimal bölgenin başarı ile kestirildiği eş yükselti üzerinden görülmektedir.

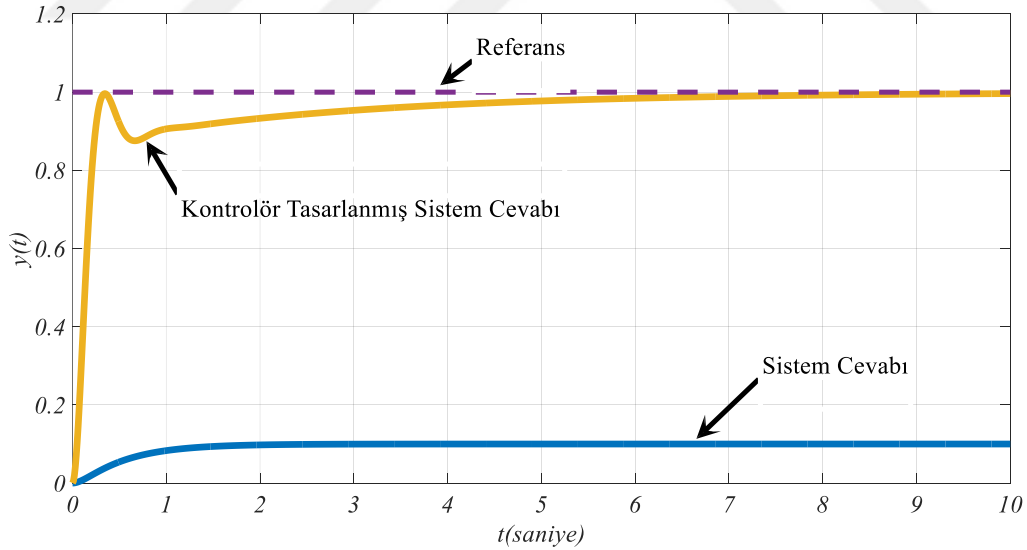
BBDY yöntemi ile *PI* kontrolör parametrelerin ayarlanması süreci sonucunda, $K_{p(BBDY)}=51,376$ ve $K_{i(BBDY)}=21,496$ olarak bulunmuştur. Tasarlanan *PI* kontrolöre ait transfer fonksiyonu Eşitlik 8.12’de gösterilmiştir.

$$G_c(s) = \frac{U(s)}{E(s)} = K_p + \frac{K_i}{s} = 51,376 + \frac{21,496}{s} \quad (8.12)$$

BBDY yöntemi ile oluşturulan kapalı çevrim sisteme ait transfer fonksiyonu ise Eşitlik 8.13’de gösterilmiştir.

$$G_s(s) = \frac{Y(s)}{R(s)} = \frac{0,5138s + 0,215}{0,005s^3 + 0,06s^2 + 0,6139s + 0,215} \quad (8.13)$$

Şekil 8.8’de doğru akım motoruna ait doğrudan birim basamak açılmal hız cevabı ve *PI* kontrolör yapısının BBDY yöntemi ile optimizasyonu sonucu elde edilen basamak cevabı gösterilmiştir.



Şekil 8.8. *PI* kontrolör tasarlanmış sisteme ait birim basamak cevabı

Kontrolörsüz ve *PI* Kontrolör tasarlanan sisteme ait birim basamak parametrelerin karşılaştırılması Tablo 8.3’de gösterilmiştir.

Tablo 8.3. PI kontrolörlü ve kontrolörsüz sisteme ait birim basamak parametreleri

Parametre	Yükselme Zamanı(sn)	Yerleşme Zamanı(sn)	Tepe Değeri
Kontrolörsüz Sistem	1,1351	2,0652	0,0999
PI kontrollü sistem	0,1940	4,8704	0,9963

Tablo 8.3 irdelendiğinde önerilen yöntemin yüksek başarımla kontrolör parametrelerini ayarladığı gözlemlenmiştir. Bu sayede sistem istenilen birim basamak cevabına sahip olmaktadır ve hız bilgisine ait referans değerini yakalamaktadır. Bu bilgiler ışığında tez çalışması kapsamında önerilen BBDY yönteminin kontrolör parametrelerinin uyarlamalı bir şekilde belirlenmesi sürecinde başarılı sonuçlar verdiği gözlemlenmiştir.

9. SONUÇLAR ve ÖNERİLER

Bu tez çalışmasında yeni bir metasezgisel optimizasyon yöntemi önerilmiştir. *Torquigener albomaculosus* olarak adlandırılan ve literatürde ilk olarak 1995 yılından sonra yer bulan erkek balon balığının karşı cinsi etkilemek amacıyla deniz tabanında oluşturduğu dairesel yapılardan esinlenilmiştir. Yöntemin başarımı literatürdeki diğer önem arz eden metasezgisel yöntemlerle karşılaştırılmıştır. Bunlara ek olarak tezde optimizasyon kavramı ve tarihçesi hakkında detaylı bilgiler verilmiştir.

Tamamen doğadan esinlenen bu yeni metasezgisel optimizasyon yönteminin başarımının artırılması için merkez bölgesindeki rassal örüntüler genel yaklaşımın aksine tek düze rastgele sayılar yardımı ile değil literatürde Levy uçuşu (Levy flight) olarak adlandırılan dağılım ile oluşturulmuştur. Bu sayede test fonksiyonlardaki başarımlar yaklaşık 3% oranında artırılmıştır. Bu dağılım daha çok metasezgisel yöntemlerdeki rassal süreçlerin modellenmesinde sıklıkla kullanılmaktadır. Bunlara ek olarak balon balığına ait dairesel yapılar imge işleme teknikleri ile irdelenmiştir. Bu analiz sonucunda oluşturulan dairesel yapının merkez bölgesi ile dış bölgesi arasındaki yarıçap oranının altın orana yakınsadığı yine bu tez kapsamında literatürde ilk defa gösterilmiştir. Deniz tabanındaki dairesel yapıların morfolojik özellikleri detaylı bir şekilde incelenmiş, önerilen yöntem oluşturulurken balon balığının oluşturduğu dairesel yapıların morfolojik özelliklerine sadık kalınmıştır ve nümerik parametreler bu bağlamda oluşturulmuştur. Bu şekilde büyük oranda doğadan esinlenilmiş bir metasezgisel optimizasyon yöntemi önerilmiştir.

Literatürde metasezgisel yöntemlerin başarımını irdelemek için sıklıkla kullanılan 8 adet test fonksiyonu üzerinden önerilen yöntemin başarımı diğer metasezgisel yöntemler ile karşılaştırılmıştır.

Sonuçlara ait detaylar ortalama, standart sapma, maksimum ve minimum değerler ve hesaplama süresi üzerinden değerlendirilmiştir. Ayrıca dairesel yapıların fonksiyon üzerindeki değişimi fonksiyonlar üzerinde detaylı bir şekilde gösterilmiştir. Bu süreç gerçekleştirilirken fonksiyonun tanımlı olduğu bölgeler içerisinde rassal bir şekilde 1000 farklı başlangıç değerleri seçilerek optimizasyon yönteminin yüksek doğruluğa sahip bir karakteristiği olduğu da gösterilmiştir.

Balon balığının farklı mevsimlerde, farklı parametrelere sahip dairesel yapılar oluşturduğu gözlemlenmiştir. Bu farklılıklar uygulanacak problemin davranışına bağlı olarak uyarlamalı bir şekilde değiştirilerek yöntemin başarımının artırılacağı ön görülmüştür.

Sonuçlarda görüldüğü gibi önerilen yöntem diğer yöntemlere göre daha hızlı bir şekilde sonucu yakınsamaktadır. Fonksiyonların çoğunda yüksek başarımlı gösteren yöntemin avantajları hesaplama süresi ve basit olmasıdır. Dezavantaj olarak da iterasyona bağlı olarak hesaplanan maliyet değerinin yüksek standart sapmaya sahip olduğu gözlemlenmiştir.

Önerilen yöntemin başarımı bu test fonksiyonlarına ek olarak bir doğru akım motorunun hız kontrolünü sağlamak amacıyla kullanılan *PI* kontrolör yapısının parametrelerinin uyarlamalı olarak belirlenmesi amacıyla kullanılmıştır. Bu bağlamda önerilen yöntemin yüksek başarımlı ile kontrolör parametrelerinin belirlenmesini sağladığı gözlemlenmiştir.

Tez çalışmasına ait sonuçlar irdelendiğinde balon balığına ait dairesel yapılar yönteminin yeni ve etkin bir metasezgisel yöntem olarak kullanılabileceği gösterilmiştir. Bu tez çalışması kapsamında yeni bir metasezgisel optimizasyon yönteminin önerilmesinin literatüre katkıda bulunacağı ve farklı alanlara uygulanacağı düşünülmektedir.

KAYNAKLAR

- [1] **Kvasov, D. E., & Sergeyev, Y. D.,** (2015). Deterministic approaches for solving practical black-box global optimization problems. *Advances in Engineering Software*, 80, 58-66.
- [2] **Rardin, R. L., & Rardin, R. L.,** (2016). *Optimization in operations research* (p. 919). Prentice Hall.
- [3] **Gass, S. I., & Assad, A. A.,** (2005). *An annotated timeline of operations research: An informal history* (Vol. 75). Springer Science & Business Media.
- [4] **Gandomi, A. H., Yang, X. S., Talatahari, S., & Alavi, A. H.,** (2013). Metaheuristic algorithms in modeling and optimization. In *Metaheuristic applications in structures and infrastructures* (pp. 1-24).
- [5] **Engelbrecht, A. P.,** (2006). *Fundamentals of computational swarm intelligence*. John Wiley & Sons.
- [6] **Kennedy, J.,** (2011). Particle swarm optimization. In *Encyclopedia of machine learning* (pp. 760-766). Springer US.
- [7] **Govan, A.,** (2006). Introduction to optimization. In North Carolina State University, SAMSI NDHS, Undergraduate workshop.
- [8] **Lange, K.,** (2004). Elementary optimization. In *Optimization* (pp. 1-17). Springer, New York, NY.
- [9] **Casquilho, M., & Buescu, J.,** (2011). A minimum distance: arithmetic and harmonic means in a geometric dispute. *International Journal of Mathematical Education in Science and Technology*, 42(3), 399-405.
- [10] **Dattorro, J.,** (2010). *Convex optimization & Euclidean distance geometry*. Lulu. com.
- [11] **Cardil, R.,** (2010). Kepler: The Volume of a Wine Barrel. *MatematicasVisuales*, posted, 8.
- [12] **Streck, S.,** (2010). The Fermat Problem. Pacific Lutheran University, Technical Report.
- [13] **Giaquinta, M., & Hildebrandt, S.,** (2013). *Calculus of variations II* (Vol. 311). Springer Science & Business Media.
- [14] **Grigoriev, V., Tahri, A., Varault, S., Rolly, B., Stout, B., Wenger, J., & Bonod, N.,** (2013). Optimization of resonant effects in nanostructures via Weierstrass factorization. *Physical Review A*, 88(1), 011803.
- [15] **Smith, W. D.,** (1992). How to find Steiner minimal trees in euclidean-space. *Algorithmica*, 7(1-6), 137-177.
- [16] **Bryson, A. E.,** (1996). Optimal control-1950 to 1985. *IEEE Control Systems*, 16(3), 26-33.
- [17] **Rardin, R. L., & Rardin, R. L.,** (2016). *Optimization in operations research* (p. 919). Prentice Hall.
- [18] **Lemaréchal, C.,** (2012). Cauchy and the gradient method. *Doc Math Extra*, 251, 254.
- [19] **Delavar, M. R., & Dragomir, S. S.,** (2017). On η -convexity. *Math. Inequal. Appl*, 20(1), 203-216.

- [20] **McCormick, G. P.**, (2014). Second order conditions for constrained minima. In *Traces and Emergence of Nonlinear Programming* (pp. 259-270). Birkhäuser, Basel.
- [21] **Abdul-Niby, M., Alameen, M., Salhieh, A., & Radhi, A.**, (2016). Improved Genetic and Simulating Annealing Algorithms to Solve the Traveling Salesman Problem Using Constraint Programming. *Engineering, Technology & Applied Science Research*, 6(2), 927-930.
- [22] **Lancia, G., & Serafini, P.**, (2018). Traveling Salesman Problems. In *Compact Extended Linear Programming Models* (pp. 155-164). Springer, Cham.
- [23] **Fiedler, M., Nedoma, J., Ramík, J., Rohn, J., & Zimmermann, K.**, (2006). *Linear optimization problems with inexact data*. Springer Science & Business Media.
- [24] **Yuan, J., Yu, Y., Gao, Y., Li, H., Ma, W., Ning, X., ... & Zhang, S.**, (2017). Three decades of progress in China's space High-Tech Program empowered by modern astrodynamics. *REACH-Reviews in Human Space Exploration*, 5, 1-8.
- [25] **Paulson, J. A., Harinath, E., Foguth, L. C., & Braatz, R. D.** (2018). Control and Systems Theory for Advanced Manufacturing. In *Emerging Applications of Control and Systems Theory* (pp. 63-79). Springer, Cham.
- [26] **Bellman, R.**, (2013). *Dynamic programming*. Courier Corporation.
- [27] **Bellman, R. E., & Dreyfus, S. E.**, (2015). *Applied dynamic programming*. Princeton university press.
- [28] **Pedroso, J. P., Rais, A., Kubo, M., & Murama, M.**, (2017). *Mathematical Optimization Documentation*.
- [29] **Yang, X. S.**, (2008). *Introduction to mathematical optimization. From Linear Programming to Metaheuristics*. Cambridge International Science Publishing.
- [30] **Harchaoui, Z., Juditsky, A., & Nemirovski, A.**, (2015). Conditional gradient algorithms for norm-regularized smooth convex optimization. *Mathematical Programming*, 152(1-2), 75-112.
- [31] **Wen, Z., & Yin, W.**, (2013). A feasible method for optimization with orthogonality constraints. *Mathematical Programming*, 142(1-2), 397-434.
- [32] **Marler, R. T., & Arora, J. S.**, (2004). Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization*, 26(6), 369-395.
- [33] **Yıldız, A. R.**, (2009). A novel particle swarm optimization approach for product design and manufacturing. *The International Journal of Advanced Manufacturing Technology*, 40(5-6), 617.
- [34] **Catalbas, M. C., & Gulten, A.**, (2017). A Novel Approach for Optimization of Nozzle Angle and Thrust Vectoring Controller via a Sub-Mutation Genetic Algorithm. *International Journal of Innovative Computing Information and Control*, 13(6), 1929-1940.
- [35] **Catalbas, M. C., & Dobrisek, S.**, (2017). 3D moving sound source localization via conventional microphones. *Elektronika ir Elektrotechnika*, 23(4), 63-69.
- [36] **Lyu, Z., & Martins, J. R.**, (2014). Aerodynamic design optimization studies of a blended-wing-body aircraft. *Journal of Aircraft*, 51(5), 1604-1617.

- [37] **Miller, H. R.**, (2011). Optimization: foundations and applications. John Wiley & Sons.
- [38] **Shang, Y. W., & Qiu, Y. H.**, (2006). A note on the extended Rosenbrock function. *Evolutionary Computation*, 14(1), 119-126.
- [39] **Bubeck, S.**, (2015). Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4), 231-357.
- [40] **Boyd, S., & Vandenberghe, L.**, (2004). Convex optimization. Cambridge university press.
- [41] **Costello, E.**, (2011). Length of a hanging cable. *Undergraduate Journal of Mathematical Modeling: One+ Two*, 4(1), 4.
- [42] **Bakr, M.**, (2013). Nonlinear optimization in electrical engineering with applications in MATLAB. Institution of Engineering and Technology.
- [43] **Nomura, T., Dede, E. M., Lee, J., Yamasaki, S., Matsumori, T., Kawamoto, A., & Kikuchi, N.**, (2015). General topology optimization method with continuous and discrete orientation design using isoparametric projection. *International Journal for Numerical Methods in Engineering*, 101(8), 571-605.
- [44] **Klodt, M., Schoenemann, T., Kolev, K., Schikora, M., & Cremers, D.**, (2008, October). An experimental comparison of discrete and continuous shape optimization methods. In *European Conference on Computer Vision* (pp. 332-345). Springer, Berlin, Heidelberg.
- [45] **Womersley, R.**, (2008). Local and global optimization.
- [46] **Rao, S. S., & Rao, S. S.**, (2009). Engineering optimization: theory and practice. John Wiley & Sons.
- [47] **Talbi, E. G.**, (2009). Metaheuristics: from design to implementation (Vol. 74). John Wiley & Sons.
- [48] **Antoniou, A., & Lu, W. S.**, (2007). Practical Optimization: Algorithms and Engineering Applications. Springer US.
- [49] **Hezam, I. M., Abdel-Baset, M., & Smarandache, F.**, (2015). Taylor Series Approximation to Solve Neutrosophic Multi-objective Programming Problem. *Infinite Study*.
- [50] **Rao, R.**, (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.
- [51] **Tuy, H., Hoang, T., Hoang, T., Mathématicien, V. N., Hoang, T., & Mathematician, V.**, (2016). Convex analysis and global optimization (Vol. 110). Berlin: Springer.
- [52] **Spall, J. C.**, (2005). Introduction to stochastic search and optimization: estimation, simulation, and control (Vol. 65). John Wiley & Sons.
- [53] **Papoulis, A., & Pillai, S. U.**, (2002). Probability, random variables, and stochastic processes. Tata McGraw-Hill Education.
- [54] **Butenko, S., & Pardalos, P. M.**, (2014). Numerical methods and optimization: An introduction. CRC Press.

- [55] **Refaeilzadeh, P., Tang, L., & Liu, H.,** (2016). Cross-validation. Encyclopedia of database systems, 1-7.
- [56] **Rosloniec, S.,** (2008). Fundamental numerical methods for electrical engineering (Vol. 18). Springer Science & Business Media.
- [57] **Conte, S. D., & De Boor, C.,** (2017). Elementary numerical analysis: an algorithmic approach (Vol. 78). SIAM.
- [58] **Killick, R., Fearnhead, P., & Eckley, I. A.,** (2012). Optimal detection of changepoints with a linear computational cost. Journal of the American Statistical Association, 107(500), 1590-1598.
- [59] **Rice, J. R.,** (2014). Numerical Methods in Software and Analysis. Elsevier.
- [60] **Anzt, H., Dongarra, J., Flegar, G., & Quintana-Ortí, E. S.,** (2018). Variable-Size Batched Gauss-Jordan Elimination for Block-Jacobi Preconditioning on Graphics Processors. Parallel Computing.
- [61] **Björck, Å.,** (2016). Numerical methods in matrix computations. Springer.
- [62] **Pratapa, P. P., Suryanarayana, P., & Pask, J. E.,** (2016). Anderson acceleration of the Jacobi iterative method: An efficient alternative to Krylov methods for large, sparse linear systems. Journal of Computational Physics, 306, 43-54.
- [63] **Edalatpour, V., Hezari, D., & Salkuyeh, D. K.,** (2017). A generalization of the Gauss–Seidel iteration method for solving absolute value equations. Applied Mathematics and Computation, 293, 156-167.
- [64] **Bragin, M. A., Luh, P. B., Yan, J. H., Yu, N., & Stern, G. A.,** (2015). Convergence of the surrogate Lagrangian relaxation method. Journal of Optimization Theory and applications, 164(1), 173-201.
- [65] **Isaacson, E., & Keller, H. B.,** (2012). Analysis of numerical methods. Courier Corporation.
- [66] **Stoer, J., & Bulirsch, R.,** (2013). Introduction to numerical analysis (Vol. 12). Springer Science & Business Media.
- [67] **Combettes, P. L., & Pesquet, J. C.,** (2015). Stochastic quasi-Fejér block-coordinate fixed point iterations with random sweeping. SIAM Journal on Optimization, 25(2), 1221-1248.
- [68] **Akram, S., & ul Ann, Q.,** (2015). Newton Raphson Method. International Journal of Scientific & Engineering Research, 6(7).
- [69] **Conte, S. D., & De Boor, C.,** (2017). Elementary numerical analysis: an algorithmic approach (Vol. 78). SIAM.
- [70] **Clerc, M.,** (2015). Guided randomness in optimization (Vol. 1). John Wiley & Sons.
- [71] **Bergstra, J., & Bengio, Y.,** (2012). Random search for hyper-parameter optimization. Journal of Machine Learning Research, 13(Feb), 281-305.
- [72] **Fu, M. C.,** (2016). Handbook of simulation optimization. Springer.

- [73] **Kallenberg, O.**, (2017). Random measures, theory and applications. Springer International Publishing.
- [74] **Yang, X. S.**, (2010). Firefly algorithm, Levy flights and global optimization. In Research and development in intelligent systems XXVI (pp. 209-218). Springer, London.
- [75] **Kamaruzaman, A. F., Zain, A. M., Yusuf, S. M., & Udin, A.**, (2013). Levy flight algorithm for optimization problems-a literature review. In Applied Mechanics and Materials (Vol. 421, pp. 496-501). Trans Tech Publications.
- [76] **Wilk, G., & Włodarczyk, Z.**, (2000). Interpretation of the nonextensivity parameter q in some applications of Tsallis statistics and Lévy distributions. Physical Review Letters, 84(13), 2770.
- [77] **Andradóttir, S.**, (2015). A review of random search methods. In Handbook of Simulation Optimization (pp. 277-292). Springer, New York, NY.
- [78] **Yang, X. S.**, (2014). Nature-inspired optimization algorithms. Elsevier.
- [79] **Gentle, J. E.**, (2013). Random number generation and Monte Carlo methods. Springer Science & Business Media.
- [80] **L'Ecuyer, P.**, (2010). Pseudorandom number generators. Encyclopedia of Quantitative Finance, 10, 9780470061602.
- [81] **Hida, T., Kuo, H. H., Potthoff, J., & Streit, L.**, (2013). White noise: an infinite dimensional calculus (Vol. 253). Springer Science & Business Media.
- [82] **Qi, B., Chi, Y. M., Lo, H. K., & Qian, L.**, (2010). High-speed quantum random number generation by measuring phase noise of a single-mode laser. Optics letters, 35(3), 312-314.
- [83] **Kaveh, A.**, (2016). Advances in metaheuristic algorithms for optimal design of structures. Springer.
- [84] **Mitchell, M.**, (1998). An introduction to genetic algorithms. MIT press.
- [85] **Pal, S. K., & Wang, P. P.**, (2017). Genetic algorithms for pattern recognition. CRC press.
- [86] **Sastry, K., Goldberg, D. E., & Kendall, G.**, (2014). Genetic algorithms. In Search methodologies (pp. 93-117). Springer, Boston, MA.
- [87] **Reeves, C.**, (2003). Genetic algorithms. In Handbook of metaheuristics (pp. 55-82). Springer, Boston, MA.
- [88] **Czerniak, J. M., & Zarzycki, H.**, (2017). Artificial acari optimization as a new strategy for global optimization of multimodal functions. Journal of Computational Science, 22, 209-227.
- [89] **Dorigo, M., & Birattari, M.**, (2011). Ant colony optimization. In Encyclopedia of machine learning (pp. 36-39). Springer, Boston, MA.
- [90] **Dorigo, M.**, (2007). Ant colony optimization. Scholarpedia, 2(3), 1461.
- [91] **Stützle, T.**, (2009, April). Ant colony optimization. In International Conference on Evolutionary Multi-Criterion Optimization (pp. 2-2). Springer, Berlin, Heidelberg.

- [92] **Dorigo, M., & Blum, C.,** (2005). Ant colony optimization theory: A survey. *Theoretical computer science*, 344(2-3), 243-278.
- [93] **Cai, Q., Zhang, D., Zheng, W., & Leung, S. C.,** (2015). A new fuzzy time series forecasting model combined with ant colony optimization and auto-regression. *Knowledge-Based Systems*, 74, 61-68.
- [94] **Molga, M., & Smutnicki, C.,** (2005). Test functions for optimization needs. *Test functions for optimization needs*, 101.
- [95] **Karaboga, D., & Basturk, B.,** (2007, June). Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems. In *International fuzzy systems association world congress* (pp. 789-798). Springer, Berlin, Heidelberg.
- [96] **Tereshko, V., & Loengarov, A.,** (2005). Collective decision making in honey-bee foraging dynamics. *Computing and Information Systems*, 9(3), 1.
- [97] **Karaboga, D., & Ozturk, C.,** (2011). A novel clustering approach: Artificial Bee Colony (ABC) algorithm. *Applied soft computing*, 11(1), 652-657.
- [98] **Pan, Q. K., Tasgetiren, M. F., Suganthan, P. N., & Chua, T. J.,** (2011). A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem. *Information sciences*, 181(12), 2455-2468.
- [99] **Karaboga, D.,** (2010). Artificial bee colony algorithm. *scholarpedia*, 5(3), 6915.
- [100] **Malan, K. M., & Engelbrecht, A. P.,** (2014, December). Particle swarm optimisation failure prediction based on fitness landscape characteristics. In *Swarm Intelligence (SIS), 2014 IEEE Symposium on* (pp. 1-9). IEEE.
- [101] **Eberhart, R., & Kennedy, J.,** (1995, October). A new optimizer using particle swarm theory. In *Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on* (pp. 39-43). IEEE.
- [102] **Kennedy, J.,** (2011). Particle swarm optimization. In *Encyclopedia of machine learning* (pp. 760-766). Springer US.
- [103] **Poli, R., Kennedy, J., & Blackwell, T.,** (2007). Particle swarm optimization. *Swarm intelligence*, 1(1), 33-57.
- [104] **Du, K. L., & Swamy, M. N. S.,** (2016). Particle swarm optimization. In *Search and Optimization by Metaheuristics* (pp. 153-173). Birkhäuser, Cham.
- [105] **Czarnecki, W. M., Osindero, S., Jaderberg, M., Swirszcz, G., & Pascanu, R.,** (2017). Sobolev training for neural networks. In *Advances in Neural Information Processing Systems* (pp. 4281-4290).
- [106] **Matsuura, K.,** (2015). A new pufferfish of the genus *Torquigener* that builds “mystery circles” on sandy bottoms in the Ryukyu Islands, Japan (*Actinopterygii: Tetraodontiformes: Tetraodontidae*). *Ichthyological research*, 62(2), 207-212.
- [107] **Kawase, H., Okata, Y., & Ito, K.,** (2013). Role of huge geometric circular structures in the reproduction of a marine pufferfish. *Scientific reports*, 3, 2106.

- [108] **Kawase, H., Mizuuchi, R., Shin, H., Kitajima, Y., Hosoda, K., Shimizu, M., ... & Kondo, S.,** (2017). Discovery of an Earliest-Stage “Mystery Circle” and Development of the Structure Constructed by Pufferfish, *Torquigener albomaculosus* (Pisces: Tetraodontidae). *Fishes*, 2(3), 14.
- [109] **Zong, R., & Gong, Y.,** (2018). Possible courtship behaviour of Devonian fish: Evidence from large radial trace fossils in northwestern China. *Palaeogeography, Palaeoclimatology, Palaeoecology*.
- [110] **Kawase, H., Okata, Y., Ito, K., & Ida, A.,** (2014). Spawning behavior and paternal egg care in a circular structure constructed by pufferfish, *Torquigener albomaculosus* (Pisces: Tetraodontidae). *Bulletin of Marine Science*, 91(1), 33-43.
- [111] **Catalbas, M. C., Gulten, A.,** (2018). The Impressive Circular Structures on the Seabed as Nature-Inspired Optimization, XIII Balkan Conference on Operational Research, 22
- [112] **Catalbas, M. C., & Gulten, A.,** (2018, April). Circular structures of puffer fish: A new metaheuristic optimization algorithm. In *Electrical and Biomedical Engineering, Clean Energy and Green Computing (EBECEGC)*, 2018 Third International Conference on (pp. 1-5). IEEE.
- [113] **Mizuuchi, R., Kawase, H., Shin, H., Iwai, D., & Kondo, S.,** (2018). Simple rules for construction of a geometric nest structure by pufferfish. *Scientific reports*, 8(1), 12366.
- [114] **Babuška, R., & Stramigioli, S.,** (1999). *Matlab and Simulink for Modeling and Control*. Technical report, Delft University of Technology.
- [115] **Neath, M. J., Swain, A. K., Madawala, U. K., & Thrimawithana, D. J.,** (2014). An optimal PID controller for a bidirectional inductive power transfer system using multiobjective genetic algorithm. *IEEE Transactions on Power Electronics*, 29(3), 1523-1531.
- [116] **Glad, T., & Ljung, L.,** (2014). *Control theory*. CRC press.

ÖZGEÇMİŞ

Mehmet Cem ÇATALBAŞ

catalbas@firat.edu.tr mcemcatalbas@gmail.com

Telefon:04242370000/5216

FIRAT ÜNİVERSİTESİ MÜHENDİSLİK FAKÜLTESİ
ELEKTRİK VE ELEKTRONİK MÜHENDİSLİĞİ BÖLÜMÜ

ELAZIĞ

- 1987:** Elazığ'da doğdu
- 2004:** Elazığ Ahmet Kabaklı Anadolu Öğretmen Lisesi'nden Mezun oldu.
- 2009:** Fırat Üniversitesi Mühendislik Fakültesi Elektrik ve Elektronik Mühendisliği Bölümü'nden mezun oldu.
- 2009:** BARAN Elektrik firmasında Satın alma mühendisi olarak çalışmaya başladı.
- 2010:** Ankara Üniversitesi Proje Mühendisi olarak çalışmaya başladı.
- 2010:** Kilis 7 Aralık Üniversitesi Mühendislik Fakültesi Elektrik ve Elektronik Mühendisliği bölümünde araştırma görevlisi olarak çalışmaya başladı.
- 2011:** Hacettepe Üniversitesi Mühendislik Fakültesi Elektrik ve Elektronik Mühendisliği bölümünde araştırma görevlisi olarak çalışmaya başladı.
- 2014:** Hacettepe Üniversitesi Fen Bilimleri Enstitüsü, Elektrik ve Elektronik Mühendisliği Anabilim dalında “Temel bileşenler analizi ve kanonik korelasyon analizi ile imge tanıma ve sınıflandırma” isimli yüksek lisans tezini tamamladı.
- 2014:** Fırat Üniversitesi Mühendislik Fakültesi Elektrik ve Elektronik Mühendisliği bölümünde araştırma görevlisi olarak çalışmaya başladı.
- 2016:** Ljubljana Üniversitesi The Laboratory of Artificial Perception, Systems and Cybernetics'da araştırmacı olarak çalışıldı.
- 2017:** Paris West University Nanterre La Défense Laboratoire Energétique Mécanique Electromagnétisme biriminde araştırmacı olarak çalışıldı