



T.C.
NECMETTİN ERBAKAN
ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ



LEVY UÇUŞUYLA GELİŞTİRİLEN BALİNA
OPTİMİZASYON ALGORİTMASININ
KÜMELEME PROBLEMLERİNE
UYGULANMASI

Ayşe Nagehan MAT
YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

Şubat-2021
KONYA
Her Hakkı Saklıdır

TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

İmza

Ayşe Nagehan MAT

Tarih:

ÖZET

YÜKSEK LİSANS TEZİ

LEVY UÇUŞUYLA GELİŞTİRİLEN BALİNA OPTİMİZASYON ALGORİTMASININ KÜMELEME PROBLEMLERİNE UYGULANMASI

Ayşe Nagehan MAT

Necmettin Erbakan Üniversitesi Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Dr. Öğr. Üyesi Onur İNAN

2021, 86 Sayfa

Jüri

Dr. Öğr. Üyesi Onur İNAN
Prof. Dr. Sabri KOÇER
Dr. Öğr. Üyesi Murat KÖKLÜ

Birçok araştırmacı tarafından ele alınan kümeleme, veri topluluklarının denetimsiz olarak gruplara ayrılmasıdır. Kümelemede veriler, aralarındaki benzerlik veya farklılıklar kullanılarak gruplandırılmaktadır. Geleneksel ve sezgisel birçok algoritma kümeleme probleminde kullanılmakta olup, günümüzde yeni teknikler geliştirilmeye devam etmektedir. Bu tezde, balinaların avlanma davranışını taklit eden Balina Optimizasyon Algoritması (BOA) ve Levy Flight (LF) stratejisi birlikte kullanılarak daha etkili yeni bir kümeleme algoritması geliştirilmiştir. UCI Machine Learning Repository veri tabanından alınan 16 gerçek veri setinde (Dermatology, E. Coli, Balance, Cancer-Int, Cancer, Iris, Glass, Wine, Credit, Thyroid, Heart, Spect, Diabetes, Hepatit, Breast Tissue, Parkinson) geliştirilen BOA-LF algoritmasıyla kümeleme uygulaması yapılmıştır. Önerilen algoritmanın kümeleme aşamasında, uygunluk fonksiyonu olarak Toplam Karesel Uzaklık fonksiyonu kullanılmıştır. BOA-LF'nin kümeleme performansı, k-means, k-medoids ve bulanık c-means kümeleme algoritmalarının performansı ile karşılaştırılmıştır. Bu karşılaştırmaya göre BOA-LF, 11 veri setinde (Cancer-int, Iris, Wine, Credit, Thyroid, Heart, Spect, Diabetes, Hepatit, Breast Tissue, Parkinson) daha iyi kümeleme sonuçları elde etmiştir. Deneysel sonuçlar, BOA-LF'nin kümeleme problemindeki başarısını ve alternatif bir kümeleme algoritması olarak kullanılabilmesini göstermiştir.

Anahtar Kelimeler: Balina Optimizasyon Algoritması, Bulanık C-means, K-means, K-medoids, Kümeleme, Levy Flight

ABSTRACT

MS THESIS

**APPLICATION OF THE WHALE OPTIMIZATION ALGORITHM
ENHANCED WITH LEVY FLIGHT TO CLUSTERING PROBLEMS**

Ayşe Nagehan MAT

**THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE OF
NECMETTİN ERBAKAN UNIVERSITY
THE DEGREE OF MASTER OF SCIENCE
IN COMPUTER ENGINEERING**

Advisor: Asst. Prof. Dr. Onur İNAN

2021, 86 Pages

Jury

Asst. Prof. Dr. Onur İNAN

Prof. Dr. Sabri KOÇER

Asst. Prof. Dr. Murat KÖKLÜ

Clustering, which is dealt with by many researchers, is the uncontrolled grouping of data stacks. In clustering, data are grouped using similarities or differences between them. Many traditional and heuristic algorithms are used in clustering problems, and new techniques continue to be developed today. In this thesis, a more effective new clustering algorithm has been developed using the Whale Optimization Algorithm (WOA), which mimics the hunting behavior of whales, and Levy Flight (LF) strategy. With the developed BOA-LF algorithm, 16 real data sets taken from the UCI Machine Learning Repository database (Dermatology, E. Coli, Balance, Cancer-Int, Cancer, Iris, Glass, Wine, Credit, Thyroid, Heart, Spect, Diabetes, Hepatit , Breast Tissue, Parkinson) clustering was applied. In the clustering stage of the proposed algorithm, Sum of Squared Error function was used as a fitness function. The clustering performance of BOA-LF was compared with the performance of k-means, k-medoids and fuzzy c-means clustering algorithms. According to this comparison, BOA-LF achieved better clustering results in 11 data sets (Cancer-int, Iris, Wine, Credit, Thyroid, Heart, Spect, Diabetes, Hepatitis, Breast Tissue, Parkinson). Experimental results have shown that the success of BOA-LF in clustering problem and it can be used as an alternative clustering algorithm.

Keywords: Clustering, Fuzzy C-means, K-means, K-medoids, Levy Flight, Whale Optimization Algorithm

ÖNSÖZ

Yüksek lisans çalışmalarım boyunca, değerli bilgi ve tecrübeleriyle beni yönlendiren, her zaman destek olan tez danışmanım Dr. Öğr. Üyesi Onur İNAN'a ve Arş. Gör. Murat KARAKOYUN'a teşekkürlerimi sunarım.

Sevgili aileme manevi desteklerini her zaman hissettirdikleri için teşekkür ederim.

Türkiye Cumhuriyeti'nin kurucusu, ulu önder Mustafa Kemal ATATÜRK'ü saygıyla yad eder, sonsuz minnet ve şükranlarımı sunmayı bir borç bilirim.

Ayşe Nagehan MAT
KONYA-2021



İÇİNDEKİLER

ABSTRACT	v
ÖNSÖZ	vi
İÇİNDEKİLER	vii
SİMGELER VE KISALTMALAR	ix
ŞEKİLLER VE ÇİZELGELER	x
1. GİRİŞ	1
1.1. Tezin Amacı ve Yöntemi	1
1.2. Kaynak Araştırması	1
1.3. Tezin Kapsamı ve Organizasyonu	4
2. OPTİMİZASYON	5
2.1. Lokal Optimizasyon	6
2.2. Global Optimizasyon	6
2.3. Optimizasyon Algoritmaları	7
2.3.1. Sezgisel optimizasyon algoritmaları	7
2.3.2. Metasezgisel optimizasyon algoritmaları	8
2.4. Balina Optimizasyon Algoritması	11
2.4.1. Algoritmanın ilham kaynağı	11
2.4.2. Matematiksel modeli ve yapısı	12
2.4.3. BOA sözde kodu ve akış diyagramı	17
2.5. Levy Flight Arama Stratejisi	20
2.5.1. Levy uçuşu	20
2.5.2. Matematiksel modeli ve yapısı	23
3. KÜMELEME	25
3.1. Kümelemede Kullanılan Uzaklık Yöntemleri	27
3.2. Kümelemede Kullanılan Benzerlik Yöntemleri	29
3.3. Kümeleme Algoritmaları	30
3.3.1. Hiyerarşik kümeleme algoritmaları	31
3.3.2. Hiyerarşik olmayan kümeleme algoritmaları	35
3.3.3. Bulanık kümeleme algoritmaları	41
3.4. BOA ve Levy Flight Arama Stratejisinin Kümelemede Kullanılması	44
4. DENEYSEL SONUÇLAR	46
4.1. Veri Setleri	46
4.2. Kümeleme Sonuçlarının Değerlendirilmesi	48
4.2.1. Toplam karesel uzaklık yöntemi	48
4.2.2. Rand index ölçüm yöntemi	49
4.3. Kümeleme Sonuçlarının Karşılaştırılması	49
5. SONUÇLAR VE ÖNERİLER	67

5.1. Sonular	67
6. KAYNAKLAR	68
ÖZGEÇMİŐ	75



SİMGELER VE KISALTMALAR

Kısaltmalar

BOA	Balina Optimizasyon Algoritması (Whale Optimization Algorithm)
GA	Genetic Algorithm (Genetik Algoritma)
UCI	University of California, Irvine (Kaliforniya Üniversitesi, Irvine)
ACO	Ant Colony Optimization (Karıncı Kolonisi Optimizasyonu)
PSO	Particle Swarm Optimization (Parçacık Sürü Optimizasyonu)
ABC	Artificial Bee Colony (Yapay Arı Kolonisi)
LF	Levy Flight
FA	Firefly Algorithm (Ateşböceği Algoritması)
LFA	Levy Flight-Firefly Algorithm (Levy Flight-Ateşböceği Algoritması)
SA	Simulated Annealing (Benzetimli Tavlama)
TS	Tabu Search (Tabu Araması)
FIFO	First In First Out (İlk Giren İlk Çıkar)
LIFO	Last In First Out (Son Giren İlk Çıkar)
ESS	Sum of Squares Index E
SSE	Sum of Squared Error (Toplam Karesel Hata)
PAM	Partitioning Around Medoids
BCM	Bulanık C-Means
RI	Rand Index
TKU	Toplam Karesel Uzaklık
B	Best (En İyi)
W	Worst (En Kötü)
A	Average (Ortalama)
S	Standard Deviation (Standart Sapma)

ŞEKİLLER VE ÇİZELGELER

Şekiller

Şekil 2.1. Bir çözüm bölgesindeki lokal ve global minimum maksimum noktaları.....	6
Şekil 2.2. Metasezgisel yöntemler	10
Şekil 2.3. Temsili avlanma ve gerçek avlanma	12
Şekil 2.4. Arama ajanlarının iki boyutlu pozisyon vektörleri ve olası konumları	14
Şekil 2.5. Arama ajanlarının üç boyutlu pozisyon vektörleri ve olası konumları	14
Şekil 2.6. Daralan çevreleme mekanizması	15
Şekil 2.7. Spiral hareket etme	16
Şekil 2.8. Konumun spiral güncellenmesi	16
Şekil 2.9. Global arama.....	17
Şekil 2.10. BOA akış diyagramı	19
Şekil 2.11. Yemek seçiminde farklı kriterlere göre örnek bir dağılım	20
Şekil 2.12. Normal dağılım ve Levy dağılımı	21
Şekil 2.13. Bilgisayarda oluşturulmuş 1000 hamlelik bir Levy uçuşu	22
Şekil 3.1. Nesnelere ve kümeler arası uzaklık	25
Şekil 3.2. Kümeleme algoritmaları	31
Şekil 3.3. Dendrogram yapısına örnek	32
Şekil 3.4. Bütünleştirici ve bölücü hiyerarşik kümelemenin örnek gösterimi.....	33
Şekil 3.5. Kümeleme örneği	36
Şekil 3.6. Orjinal veri ve kümelenecek veri	37
Şekil 3.7. Farklı k değerleri için k-means kümeleme	37
Şekil 3.8. Elbow metodu ve en uygun k değeri (k=3)	39
Şekil 3.9. Silhouette metodu ve en uygun k değeri (k=5)	39
Şekil 3.10. K-means kümeleme adımları	40
Şekil 3.11. K-medoids kümeleme adımları	41
Şekil 3.12. Farklı k değerleri için bulanık c-means kümeleme	43
Şekil 3.13. BOA-LF akış diyagramı	45
Şekil 4.1. Dermatology yakınsama eğrisi	53
Şekil 4.2. E.coli yakınsama eğrisi	53
Şekil 4.3. Balance yakınsama eğrisi	54
Şekil 4.4. Cancer-int yakınsama eğrisi	54
Şekil 4.5. Cancer yakınsama eğrisi	55
Şekil 4.6. Iris yakınsama eğrisi	55
Şekil 4.7. Glass yakınsama eğrisi	56
Şekil 4.8. Wine yakınsama eğrisi.....	57
Şekil 4.9. Credit yakınsama eğrisi	57
Şekil 4.10. Thyroid yakınsama eğrisi	58
Şekil 4.11. Heart yakınsama eğrisi	58
Şekil 4.12. Spect yakınsama eğrisi	59
Şekil 4.13. Diabetes yakınsama eğrisi	60
Şekil 4.14. Hepatit yakınsama eğrisi	60
Şekil 4.15. Breast tissue yakınsama eğrisi	61
Şekil 4.16. Parkinson yakınsama eğrisi	61

Çizelgeler

Çizelge 2.1. BOA sözde kodu.....	18
Çizelge 3.1. Uzaklık metotları ve formülleri	29
Çizelge 3.2. BOA-LF sözde kodu.....	44
Çizelge 4.1. Veri setlerinin özellikleri	48
Çizelge 4.2. SSE sonuçları.....	51
Çizelge 4.3. SSE ortalama sıralama değerleri.....	62
Çizelge 4.4. RI sonuçları.....	64
Çizelge 4.5. RI ortalama sıralama değerleri	66



1. GİRİŞ

Global ve yerel düzeyde giderek yaygınlaşan dijitalleşme süreçlerinin bir sonucu olarak pek çok farklı alanda büyük ölçekli veriler elde edilmektedir. Büyük ölçekteki veriden daha fazla yararlanmak ve anlamlı bilgi çıkarmak için verilerin doğru ve hızlı şekilde işlenmesi büyük önem taşımaktadır. Bu nedenle veri madenciliği tekniklerinin uygulanmakta olduğu endüstri, bankacılık, pazarlama, tıp, mühendislik ve ekonomi gibi birçok alanda bilginin etkin bir şekilde kullanılması için sürekli yeni yöntemler geliştirilmektedir.

Veri madenciliği, makine öğrenmesi, yapay zeka ve istatistik gibi ileri çözümlene tekniklerini kullanarak verideki gizli örüntü ve ilişkileri açığa çıkarmaktadır. Veri madenciliği tekniklerinden biri olan kümeleme, benzer özellikteki verileri bir araya toplar ve veri topluluğunun kümelere/gruplara ayrılmasını sağlar. Literatüre bakıldığında, sezgisel algoritmaların kümelemede kullanılması, daha geleneksel kümeleme tekniklerine bir alternatif olarak ortaya çıkmaktadır (Cui 2017, Zhang ve diğ. 2010).

1.1. Tezin Amacı ve Yöntemi

Bu tezde, balina optimizasyon algoritmasının, az sayıda parametreye sahip olması ve yerel minimum tuzağının bulunmaması gibi avantajları, kümeleme problemi için seçilmesinde etkili olmuştur. Çalışmanın amacı, basit çözümlerle daha iyi sonuçlar elde etmek ve etiketlenmemiş verileri kümelemektir. BOA'nın global aramasını güçlendirmek ve mevcut yöntemlere kıyasla eksiksiz bir arama yapabilmek için Levy Flight arama stratejisiyle birlikte kullanılmıştır. Önerilen algoritma, UCI (Murphy ve Aha 1994) veri tabanından seçilen 16 gerçek veri setinde test edilmiştir. Algoritmanın kümeleme performansı, k-means, k-medoids ve bulanık c-means algoritmaları ile karşılaştırmalı olarak verilmiştir.

1.2. Kaynak Araştırması

Bu bölümde kümeleme, balina optimizasyon algoritması ve Levy Flight arama stratejisi için literatür incelemesi yapılmıştır ve araştırmacıların çalışmalarına kısaca değinilmiştir.

Selim ve Al-Sultan, kümeleme problemi için benzetilmiş tavlama yaklaşımını kullanmışlardır. Algoritmanın önceden belirlenmiş parametreleri tartışılmış ve kümeleme problemindeki global çözüme yakınsadığı gösterilmiştir (Selim ve Alsultan 1991).

Maulik ve Mukhopadhyay, birleşik bir kümeleme algoritması sundular. Çözüm kalitesini artırmak için benzetilmiş tavlama algoritmasını yapay sinir ağları ile birleştirdiler. Önerilen hibrit algoritma, üç gerçek mikrodizi veri setini kümelemede kullanıldı ve önerilen yaklaşımın sonuçları, yaygın olarak kullanılan bazı kümeleme algoritmaları ile karşılaştırıldı. Sonuçlar yeni algoritmanın üstünlüğünü gösterdi (Maulik ve Mukhopadhyay 2010).

Maulik ve Bandyopadhyay, kümeleme problemini çözmek için genetik algoritmaya dayalı bir yaklaşım sundular. Sentetik ve gerçek veri setlerini kullanarak yaklaşımın performansını değerlendirdiler (Maulik ve Bandyopadhyay 2000).

Shelokar ve diğ., karınca kolonisi optimizasyonuna (ACO) dayalı bir kümeleme algoritması önermiştir. Önerilen algoritma bazı yapay ve gerçek veri setlerinde test edilmiştir. Bu tekniğin genetik algoritma, benzetilmiş tavlama ve tabu araması gibi popüler algoritmalarla karşılaştırıldığında performansı umut vericiydi (Shelokar ve diğ. 2004).

Van der Merwe ve Engelbrecht, kümeleme problemini çözmek için parçacık sürü optimizasyonu (PSO) algoritmasını kullanmıştır. Sürü parçacıklarının k-means algoritmasıyla seçildiği PSO kümelemesi ve hibrit bir yöntem kullanmışlardır. Her iki yöntem de k-means algoritması ile karşılaştırılmış ve önerilen algoritmaların daha iyi sonuçlara sahip olduğu görülmüştür (Van der Merwe ve Engelbrecht 2003).

Cura, kümeleme probleminde etkili, ayarlanması kolay, küme sayısının bilindiği veya bilinmediği durumlarda uygulanabilir yeni bir PSO yaklaşımı sundu (Cura 2012).

Karaboga ve Ozturk, kümeleme problemini çözmek için yapay arı kolonisi algoritmasını (ABC) kullanmıştır. Test verileri üzerindeki sonuçlar, önerilen algoritmanın PSO algoritması ve diğer bazı yaklaşımlara kıyasla üstün performanslı olduğunu göstermiştir. Ayrıca yazarlar, ABC algoritmasının çok değişkenli kümeleme problemlerini çözmek için uygun olabileceğini bulmuşlardır (Karaboga ve Ozturk 2011).

Zhang ve diğ. , yapay bir arı kolonisi (ABC) kümeleme algoritması önermiştir. Bu algoritmada, seçim sürecinde açgözlü seçim yerine Deb's kuralı kullanıldı. İyi bilinen birkaç veri setinde test edilen algoritma, kümelemede popüler diğer sezgisel

tarama algoritmaları ile karşılaştırıldı. Kümelerin kalitesi açısından sonuçlar başarılıydı (Zhang ve diğ. 2010).

Armano ve Farmani, global optimum çözümü bulmada k-means algoritmasının etkinliğini artırmak için k-means ve ABC algoritmalarının bir arada kullanıldığı bir yöntem önermiştir (Armano ve Farmani 2014).

Karthikeyan ve Christopher, PSO algoritması ve ABC algoritmasının birleşimiyle bir algoritma önerdiler. Önerilen yaklaşımın performansı, diğer kümeleme algoritmaları ile karşılaştırılarak belirlendi (Karthikeyan ve Christopher 2014).

Mane ve Gaikwad, veri kümelemede PSO ve bulanık k-means algoritmalarını sırayla kullanan yeni, hibrit bir sıralı kümeleme yaklaşımı önermiştir. Deneysel sonuçlar, yeni yaklaşımın oluşturulmuş kümelerin kalitesini artırdığını ve yerel minimuma takılmaktan kaçındığını göstermektedir (Mane ve Gaikwad 2014).

Son zamanlarda, Mirjalili ve Lewis, balina optimizasyonu algoritması (BOA) olarak adlandırılan ve kambur balinaların kabarcık avlanma stratejisini taklit eden yeni bir metasezgisel optimizasyon algoritması tanıttı (Mirjalili ve Lewis 2016). BOA algoritması, 29 matematiksel karşılaştırma optimizasyon problemiyle test edildi ve algoritmanın performansı PSO (Eberhart ve Kennedy 1995), Diferansiyel Evrim (Storn ve Price 1997), Yerçekimi Arama Algoritması (Rashedi ve diğ. 2009) ve Hızlı Evrimsel Programlama (Yao ve diğ. 1999) gibi diğer metasezgisel algoritmalarla karşılaştırıldı. Karşılaştırmalar neticesinde, BOA'nın diğer iyi bilinen metasezgisel yöntemlerle rekabet edebildiği kabul edildi.

Nasiri ve Khiyabani, çalışmalarında BOA'yı kümeleme uygulamasında kullandı. Sonuçları iyi bilinen k-means, PSO, yapay arı kolonisi, diferansiyel evrim ve genetik algoritma gibi diğer popüler algoritmalar ile karşılaştırdılar. Kümeiçi mesafe fonksiyonu ve standart sapma değerleri, balina optimizasyon algoritmasının kümeleme problemini çözmede başarıyla uygulanabileceğini gösterdi (Nasiri ve Khiyabani 2018).

Canayaz ve Ozdag, BOA ile kümeleme uygulaması yaparak, önerdikleri yöntemde BEST adı verilen bir özellik vektörü elde ettiler. Elde edilen bu vektörün test verisi ile arasındaki Manhattan uzaklığını hesaplayarak kümeleme yaptılar. Sonuçlar BOA'nın kümeleme probleminde kullanılabileceğini, çalışma süresi açısından değerlendirildiğinde de yapay atom algoritmasına göre daha hızlı olduğunu gösterdi (Canayaz ve Ozdag 2017).

Levy Flight, rastgele bir uçuş sınıfıdır ve hareket eden mesafe için Gauss olmayan rastgele dağıtılmış adım boylarını içerecek şekilde genelleştirilmiş Brown

hareketi olarak tanımlanabilir (Al-Temeemy ve diğ. 2010). Levy Flight, sıvı dinamikleri, deprem analizi, floresan moleküllerinin difüzyonu, soğutma davranışı, gürültü gibi birçok doğal ve yapay durumu tasvir edebilir (Chen 2010). Levy Flight aynı zamanda Pereyra ve Batatia (Pereyra ve Batatia 2010) tarafından cilt dokusunda ultrason ve Al-Temeemy (Al-Temeemy ve diğ. 2010) tarafından ladar taramasında kullanılmıştır. Bu alanlara ek olarak, bilgisayar bilimlerindeki birçok alanda da önemli rol oynamıştır. Terdik ve Gyires (Terdik ve Gyires 2008) tarafından internet trafik modelleri, Chen (Chen 2010) tarafından gecikme ve kesintiye tolerans ağı, Sutantyo ve diğ. (Sutantyo ve diğ. 2010) tarafından çoklu robot arama prosedürü ve Rhee ve diğ. (Rhee ve diğ. 2011) tarafından insan hareketlilik alanlarında kullanılmıştır.

Ayrıca albatros, yaban arısı ve geyik gibi birçok hayvanın gıda arama davranışına benzeyen Levy Flight, doğaya özgü algoritmaların iyileştirilmesinde kullanılmıştır (Edwards ve diğ. 2007, Viswanathan ve diğ. 1996). Yang ve Deb, Cuckoo Search'te yeni guguk kuşu yaratma aşamasında Levy Flight dağılımını kullandılar (Yang ve Deb 2013). Ayrıca, Yang rastgeleliği geliştirmek için ateşböceği algoritmasını (FA) Levy Flight arama stratejisiyle birleştirerek yeni bir Levy Flight-Ateşböceği algoritması (LFA) tanıtmıştır (Yang 2010).

1.3. Tezin Kapsamı ve Organizasyonu

Bu tez çalışmasında, balina optimizasyon algoritması ve Levy Flight arama stratejisi birlikte kullanılarak kümeleme yapılmıştır. Birinci bölümde tezle ilgili genel bilgiler ve kaynak araştırması paylaşılmıştır. İkinci bölümde optimizasyon kavramı, optimizasyon algoritmaları, BOA ve Levy Flight arama stratejisi açıklanmıştır. Üçüncü bölümde kümeleme kavramları, kümeleme algoritmaları ve tezde gerçekleştirilen kümeleme uygulaması açıklanmıştır. Dördüncü bölümde uygulama sonuçları ve sonuçların değerlendirilmesinde kullanılan kriterler paylaşılmıştır. Beşinci bölümde uygulama sonuçlarının ve tez çalışmasının nihai bir değerlendirilmesi yapılmıştır. Son bölümde tez hazırlanırken faydalanılan kaynak eserlere yer verilmiştir.

2. OPTİMİZASYON

Bir problemin belli koşullar altında optimum çözüm ya da çözümlerinin bulunması işlemine optimizasyon denir. Optimizasyon problemindeki değişkenlerin en iyi değerleri, hedef fonksiyonunu maksimize veya minimize etmek amacıyla belirlenir.

Günümüzde hesaplama maliyetlerinin düşürülmesi, zamanın etkin kullanımı ve enerji verimliliği gibi pek çok farklı alanda optimizasyona ihtiyaç duyulmaktadır. Bu nedenle optimizasyon algoritmalarıyla ilgili yapılan çalışmaların popülerliği giderek artmaktadır.

Optimizasyon işleminin ilk aşamasında optimize edilecek karar değişkenleri tanımlanır. İkinci aşamada, alternatif çözümleri karşılaştırmak, çözüm kalitesini/uygunluğunu ölçmek ve optimizasyon algoritmasının arama uzayında yön bulmasını sağlamak için bir amaç fonksiyonu belirlenmelidir. Amaç fonksiyonu optimizasyon probleminin türüne (maksimize ya da minimize) göre seçilir. Minimizasyon problemlerinde amaç fonksiyonu bir maliyet fonksiyonudur ve alternatif çözümler arasından en düşük sonucu üreten en iyi çözümdür. Maksimizasyon problemlerinde amaç fonksiyonu bir kâr fonksiyonudur ve alternatif çözümler arasından en yüksek sonucu üreten en iyi çözümü ifade etmektedir.

Tanımlanması gereken bir diğer husus optimizasyon problemindeki parametrelerin alamayacağı değerleri belirten kısıtlama fonksiyonlarıdır.

Karar değişkenlerini, amaç fonksiyonunu ve kısıtlama fonksiyonlarını örneklemek amacıyla “Bir fabrikada 3 kg ham madde kullanılarak A ürünü, 5 kg ham madde kullanılarak B ürünü üretilmektedir ve A ürünü satıldığında K TL kar B ürünü satıldığında L TL kar elde edilmektedir. Maksimum 60 kg ham madde kullanılarak ve en az 4 tane A ürünü üreterek en fazla ne kadar kar elde edilir.” problemini ele alalım. Bu problemde A ürünü sayısına X_1 , B ürünü sayısına ise X_2 dersek; karar değişkenleri X_1 ve X_2 , kısıtlar Denklem 2.1 ve Denklem 2.2, amaç fonksiyonu ise Denklem 2.3’te yer almaktadır.

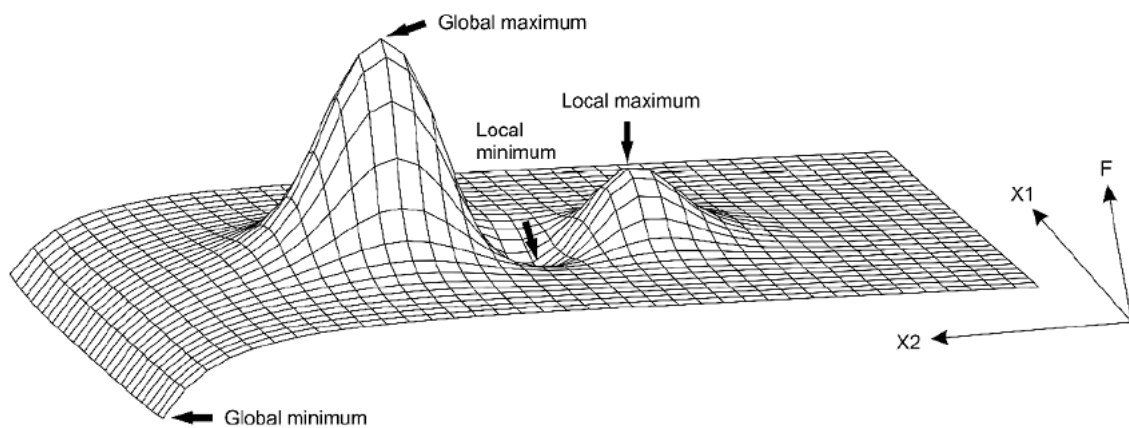
$$3.x_1 + 5.x_2 \leq 60 \quad (2.1)$$

$$x_1 \geq 4 \quad (2.2)$$

$$f(x) = K.x_1 + L. x_2 \quad (2.3)$$

Optimizasyon algoritmaları, lokal (yerel) ve global (küresel) olarak ikiye ayrılabilir (Yetkin 2016). Optimizasyon algoritmalarının genel amacı, bir amaç fonksiyonunun tanımlanmış arama/çözüm uzayında daha geniş tarama yaparak muhtemel tüm çözümleri değerlendirmesidir. Bu tarama, çözümün global minimuma veya maksimuma ulaşabilirliğini artırmaktadır.

Lokal optimizasyonda ise, bir başlangıç çözümü temel alınarak global optimizasyona göre daha dar bir arama/çözüm uzayında optimum çözüme ulaşılması hedeflenir. Global ve lokal optimizasyon için minimum ve maksimum noktalarının gösterimi Şekil 2.1’de verilmiştir (Yetkin 2016).



Şekil 2.1. Bir çözüm bölgesindeki lokal ve global minimum maksimum noktaları

2.1. Lokal Optimizasyon

Lokal optimizasyon algoritmaları, başlangıç parametrelerine bağlı olarak optimum çözüme ulaşmayı hedefler. Lokal optimizasyonda başlangıç parametreleri iyi bir şekilde belirlenmeli ve global optimum çözüme oldukça yakın değerler olarak seçilmelidir (Yetkin 2016).

2.2. Global Optimizasyon

Lokal optimizasyon algoritmaları belli dezavantajlara sahip olduğu için çeşitli sezgisel ve metasezgisel optimizasyon algoritmaları geliştirilmiştir. Bu algoritmalar, genellikle doğadan esinlenerek oluşturulur ve problem çözümünde kullanılabilecek gerekli matematiksel formüller elde edilerek optimizasyon problemine uygulanırlar. Ayrıca, arama/çözüm uzayı daha geniş olan algoritmalardır.

2.3. Optimizasyon Algoritmaları

Optimizasyon algoritmaları, deterministik optimizasyon algoritmaları ve deterministik optimizasyon algoritmalarına rastgelelik eklenerek oluşturulan stokastik optimizasyon algoritmaları olarak ikiye ayrılır. Stokastik optimizasyon algoritmaları ise, sezgisel algoritmalar ve metasezgisel algoritmalar olarak iki gruba ayrılır ve aralarındaki fark çok küçüktür. Sezgisel algoritmalar en iyi çözüme ulaşmak yerine yeterince iyi sayılabilen ve kolay elde edilen çözümler için kullanılmaktadır. Zor optimizasyon problemlerinin çözümünde, kesin olarak kullanılacak yöntemler olmaması ve çözüme makul bir zaman diliminde ulaşılması gerektiği için metasezgisel optimizasyon algoritmaları tercih edilir. Her problem için derinlemesine özelleşmeden sonuca ulaşmak amacıyla geliştirilen metasezgisel algoritmaları problem çözümüne bir üst seviyeden bakmaktadır. Genellikle spesifik bir yöntemin kullanıldığı ve istenilen sonuçların elde edilemediği problemlere uygulanırlar.

Sezgisel optimizasyon algoritmaları belli bir problem için özel olarak geliştirilen ve mantıksal olarak doğrulanabilen algoritmalarlardır. Sıralama algoritmaları, FIFO (First In First Out- İlk Giren İlk Çıkar) ve LIFO (Last In First Out- Son Giren İlk Çıkar) sezgisel algoritmalara örnektir. Metasezgisel optimizasyon algoritmaları ise, uygulama alanı sezgisel algoritmalara göre daha geniş olan algoritmalarlardır. Metasezgisel algoritmalar içerisinde sezgisel algoritmaları kullanan ve üzerinde küçük değişiklikler yapılarak çeşitli problemlere uygulanabilen algoritmalarlardır. Metasezgisel algoritmalar zor ve karmaşık problemler için makul süre içerisinde kabul edilebilir çözümler sunarlar (Talbi 2009).

2.3.1. Sezgisel optimizasyon algoritmaları

Sezgisel algoritmalar, problem çözümünde çeşitli alternatif seçeneklerden etkili olanlara karar vermek amacıyla kullanılan bilgisayar metodlarıdır. Sezgisel yöntemler, geleneksel optimizasyon algoritmalarının dezavantajlarını ortadan kaldırmak amacıyla geliştirilmiştir.

Problem karmaşıklığı arttıkça muhtemel tüm çözümlerin aranması imkânsız hale gelmektedir. Sezgisel algoritmalar, karmaşık yapıdaki problemleri daha basit hale getirerek, makul bir zaman dilimi içinde iyi uygulanabilir çözümler sunmayı

amaçlamaktadır. En iyi çözümü bulacaklarını garanti etmezler ancak en kısa zamanda en makul çözümü ortaya koyarlar (Yang 2010,Cura 2008).

Sezgisel algoritmalara ihtiyaç duyulmasının sebepleri aşağıda belirtildiği gibidir (Karaboğa 2011,Kıran 2014):

- Yapılarında fazla bir değişiklik gerektirmeden diğer problemlere uyarlanabilir.
- Basit olmaları nedeniyle karmaşık matematiksel hesaplamalar kullanmazlar.
- Problemin nasıl çözüleceğini bilmeden problemi çözebilirler. Bu nedenle, uzman deneyimi olmayan problemlerin çözümünde kullanılabilirler.
- Anlaşılabilirlik açısından karar vericiler için daha kolay algılanabilirler.
- Kesin çözüm bulunurken çözümün bir parçası olarak kullanılabilirler.

Genellikle gerçek dünya problemlerinin en zor kısımları (hangi amaçların ve hangi sınırlamaların kullanılacağı, hangi alternatiflerin test edileceği, problem verisinin nasıl toplanacağı) matematiksel tanımlamalarda ihmâl edilir. Model parametreleri belirlenirken kullanılan verinin hatalı olması, sezgisel yaklaşımın üretebileceği alt optimal çözümden daha büyük hatalara sebebiyet verebilir.

2.3.2. Metasezgisel optimizasyon algoritmaları

Metasezgisel optimizasyon algoritmaları, sezgisel yöntemlerin üstünde yer alan ve problem çözümünde kullanılacak yönteme karar veren algoritmalarlardır. Metasezgisel algoritmaların en belirgin özelliği optimizasyon yapmak için iyi bir başlangıç değerine ihtiyaç duymamasıdır. Metasezgisel algoritmaların uyarlanabilen operatörler ile arama uzayında daha etkin çözümler üretme özellikleri vardır. Bu algoritmalar, özellikle doğrusal olmayan ve analitik yollarla çözümü mümkün olmayan optimizasyon problemlerini araştırmada ve optimum çözüme ulaşmada sıkça kullanılmaya başlanmıştır (Yetkin 2016).

Metasezgisel algoritmaların optimizasyon süreci genel olarak şu şekildedir: İlk olarak bir başlangıç popülasyonu oluşturulur. Popülasyondaki bireylere, değişkenler için belirlenen alt ve üst sınırlar dikkate alınarak rastgele değerler atanır. Ayrıca, algoritmanın ihtiyacı olan diğer parametreler de belirlenir. Her birey için bu değerler kullanılarak uygunluk değeri hesaplanır. Daha sonra mevcut popülasyonun tamamı veya bir kısmı yeni nesil bireylerin üretilmesi için seçilir. Yeni nesil bireylerin üretimi

algoritmalar arasında farklılık gösterebilir. Örneğin; ebeveyn bireyler arasında çaprazlama yapılarak yeni nesil bireyler elde edilebileceği gibi, yeni bireyler diğer bireylerden bağımsız olarak sadece belli değişiklikler yapılarak üretilebilir. Ayrıca, yeni bireyler önceden elde edilmiş bilgiler kullanılarak da oluşturulabilir. Oluşturulan yeni bireyler kurallar çerçevesinde popülasyondaki ebeveyn bireylerin bir kısmının veya tamamının yerini alabilirler. Bahsedilen bu süreçler belirlenen maksimum iterasyon sayısına veya durdurma kriterine ulaşılan kadar döngüsel olarak tekrar eder. Optimizasyon süreci sonunda elde edilen sonuç, optimum veya istenilen kriteri sağlayan optimuma yakın bir değer olabilir (Başbuğ 2013).

Metasezgisel algoritmalarda keşif (exploration) ve sömürü (exploitation) olmak üzere iki temel bileşen bulunmaktadır:

- Keşif; arama uzayının tamamının araştırılarak farklı aday çözümlerin üretilmesidir.
- Sömürü ise, mevcut arama konumunda en iyi aday çözümün bulunduğu bölgeye odaklanılması ve oradaki bilginin kullanılmasını ifade etmektedir.

Sezgisel algoritmaların çoğu, keşif ve sömürü arasındaki dengeyi kurarak en iyi çözüme ulaşmayı hedefler (Afşar 2014).

Tek bir çözümle başlayıp işlemi operatörlerle ilerleten algoritmalar tek noktali yöntemler olarak adlandırılır. Tabu arama, ısı işlem gibi bütün yerel arama tabanlı algoritmalar bu grupta yer alır. Çok noktayla yani bir popülasyonla çözüme başlayıp farklı noktalarla optimizasyon yapan algoritmalar çok noktali ya da popülasyon tabanlı yöntemler olarak adlandırılır. GA ve PSO bu gruptaki algoritmalara örnektir (Alataş 2007,Kızılloluk 2013).

Problemin çözüm aşamasında amaç fonksiyonunu sabit tutan sezgisel algoritmalar sabit amaç fonksiyonlu, amaç fonksiyonunu değiştiren sezgisel algoritmalar değişen amaç fonksiyonlu olarak adlandırılırlar. Amaç fonksiyonunun değiştirilmesindeki hedef, yerel minimumdan kurtulmaktır (Alataş 2007,Kızılloluk 2013).

Metasezgisel algoritmaların çoğu tek bir komşuluk yapısında çalıştığı için tek komşuluk yapılı olarak gruplandırılabilir. Diğer metasezgisel algoritmalar, değişken arama işlemini düzenli bir şekilde değiştirerek birden fazla yerel arama metodu kullanır. Bu şekilde diğer çözüm uzaylarına ulaşmaya çalışan algoritmalar değişken komşuluk yapılı olarak gruplandırılabilir (Alataş 2007,Kızılloluk 2013).

Metasezgisel algoritmalar, çalışma esnasında elde ettikleri durumların bir kısmını veya tamamını hafızalarına kaydedebilir. Bu bilgiler algoritmanın sonraki aşamalarda vereceği kararlar için etkili biçimde kullanılabilir. Daha önceki durumların ya da en iyi durumların hatırlandığı algoritmalar hafızalı, hatırlanmadığı algoritmalar hafızasız olarak sınıflandırılır. PSO hafızalı algoritmalara, GA hafızasız algoritmalara örnek gösterilebilir (Alataş 2007,Kızıloluk 2013).

Metasezgisel algoritmaların birçoğu doğadan esinlenilerek geliştirilmiştir. Esinlendikleri alanlara göre; biyoloji tabanlı, kimya tabanlı, fizik tabanlı, matematik tabanlı, sosyal tabanlı, müzik tabanlı, spor tabanlı, sürü tabanlı ve hibrit olmak üzere kategorize edilebilir. Hibrit algoritmalar, birden fazla sezgisel algoritmanın bir arada kullanılmasıyla oluşmaktadır (Kızıloluk 2013,Canayaz 2015). Metasezgisel algoritmalar Şekil 2.2’de sınıflandırılmıştır.



Şekil 2.2. Metasezgisel yöntemler

Bir metasezgisel algoritmanın başarısı geçerli problem için çeşitlendirme (diversification) ve yoğunlaştırma (intensification) arasındaki dengeyi sağlayabilmesine bağlıdır. Mevcut metasezgisel algoritmalar arasındaki en önemli fark bu dengeyi sağlamak için geliştirdikleri yöntemlerdir (Birattari ve diğ. 2001).

En bilinen metasezgisel algoritmalar; Genetik Algoritma (GA), Karınca Kolonisi Optimizasyonu (KKO), Parçacık Sürü Optimizasyonu (PSO), Yapay Arı Kolonisi (YAK), Benzetimli Tavlama (BT) ve Tabu Araması (TA) algoritmalarıdır.

Metasezgisel optimizasyon algoritmaları, son yıllarda mühendislik uygulamalarında sıklıkla kullanılmaya başlanmıştır. Bu algoritmalar, dönüşümlerinin kolay olması, türev gerektirmeyen yapıları ve yerel minimumu aşabilmeleri gibi çeşitli avantajlara sahiptir. Ayrıca, farklı alanlardaki problemlere çözüm üretebildikleri için geniş kullanım yelpazeleri bulunmaktadır.

2.4. Balina Optimizasyon Algoritması

Balina Optimizasyonu Algoritması (BOA) 2016 yılında Mirjalili ve Lewis tarafından önerilmiştir. Bu bölümde algoritmanın ilham kaynağı, matematiksel modeli, algoritmanın yapısı, sözde kodu ve akış diyagramı paylaşılacaktır.

2.4.1. Algoritmanın ilham kaynağı

Balinalar, dünyanın en büyük memelileri olarak bilinmektedir. Yetişkin bir balina 180 ton ağırlık ve 30 metre uzunlukta olabilir. Kuzey, Minke, Katil, Buzul, Oluklu, Kambur ve Mavi balina olmak üzere 7 çeşit balina bulunur. Balinalar okyanus yüzeyinden nefes almak zorunda oldukları için beyinlerinin sadece yarısı uyur.

Hof ve Van Der Gucht'a göre, balinalar ile insan beyninin belli bölgeleri arasında benzerlik bulunmaktadır (Hof ve Gucht 2007). Benzer bölgelerdeki hücreler, insanların duyguları, sosyal davranışlarındaki akıl yürütme özellikleriyle ilgilidir ve insanları diğer canlılardan ayırır. Balinalarda bu hücreler insanlara göre iki kat daha fazla bulunmaktadır. Balinaların düşünebildiği, iletişim kurabildiği, öğrenebildiği ve hatta duygularının olduğu kanıtlanmıştır. Ancak insanlara kıyasla çok daha az zekidirler. Balinalar genellikle grup halinde gezmekle birlikte bazen de yalnız hareket ederler.

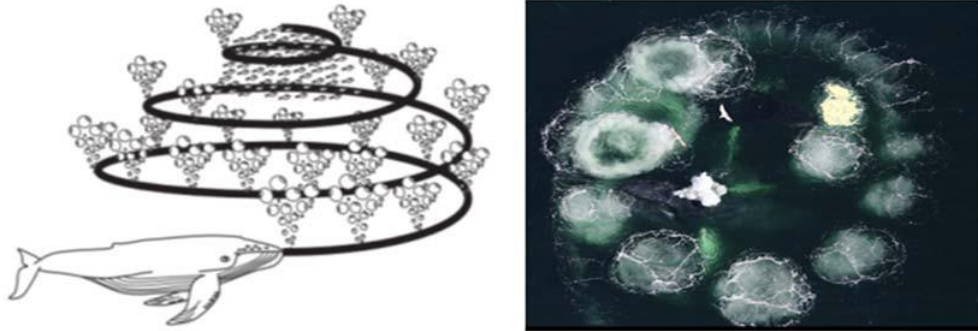
Bu çalışmada araştırılan kambur balinalar balina türlerinin en büyüğüdür. Kambur balinalar küçük ve kril balıkları bulmak için harika bir av yöntemine sahiptir (Mirjalili ve Lewis 2016). BOA kambur balinaların bu avlanma davranışından esinlenerek geliştirilmiştir.

2.4.2. Matematiksel modeli ve yapısı

Balina optimizasyon algoritması, bir metasezgisel optimizasyon algoritmasıdır. Algoritma, kambur balinaların avlanma davranışlarını taklit etmektedir ve avlanma esnasında kullandıkları kabarcık avlanma stratejisinden hareketle oluşturulmuştur (Mirjalili ve Lewis 2016).

Genellikle küçük balık sürüleriyle beslenen kambur balinalar, kendilerine özgü olan hava kabarcığı davranışıyla suyun altında soluk vererek hava kabarcığı bulutları meydana getirebilirler. Birbiriyle bağlantılı olan bu geniş hava kabarcığı kütleleri avları bir araya toplamada oldukça etkilidir. Daha sonra balina oluşan kabarcıklar içerisinde yüzeye doğru yükselmeye başlar. Yükselirken aynı zamanda kabarcık oluşturmaya devam eder ve avına yaklaştıkça kabarcık çemberini daraltıp hedef küçültür. Bu davranış, avın bulunmasında, hareketsiz hale getirilmesinde ve şaşırtılarak ele geçirilmesinde faydalı olduğu gibi avcının avından gizlenmesini de mümkün kılmaktadır (Goldbogen ve diğ. 2013).

Şekil 2.3'te kabarcık stratejisi ile avlanma yöntemi (sol) ve gerçek bir avlanma resmi (sağ) gösterilmektedir (Mirjalili ve Lewis 2016). Balina optimizasyon algoritmasında avlanma stratejisi üç bölüme ayrılmıştır: avın etrafını sarma, ava doğru hareket etme ve av arama.



Şekil 2.3. Temsili avlanma ve gerçek avlanma

2.4.2.1. Avın etrafını sarma

Balina optimizasyon algoritmasında, av ulaşılması gereken optimum çözümdür. Optimizasyon problemlerinde optimum çözüm bilinmediği için, algoritmanın ulaştığı en iyi sonuç veya onun yakınında bir nokta optimum çözüm olarak kabul edilir. Diğer çözümlerin konumları, belirlenen en iyi çözüm değerine göre güncellenir. Avın etrafını sarma davranışı, matematiksel olarak Denklem 2.4 ve 2.5'te gösterilmiştir (Mirjalili ve Lewis 2016).

$$\vec{D} = | \vec{C} \cdot \vec{X}^*(t) - \vec{X}(t) | \quad (2.4)$$

$$\vec{X}(t+1) = | \vec{X}^*(t) - \vec{A} \cdot \vec{D} | \quad (2.5)$$

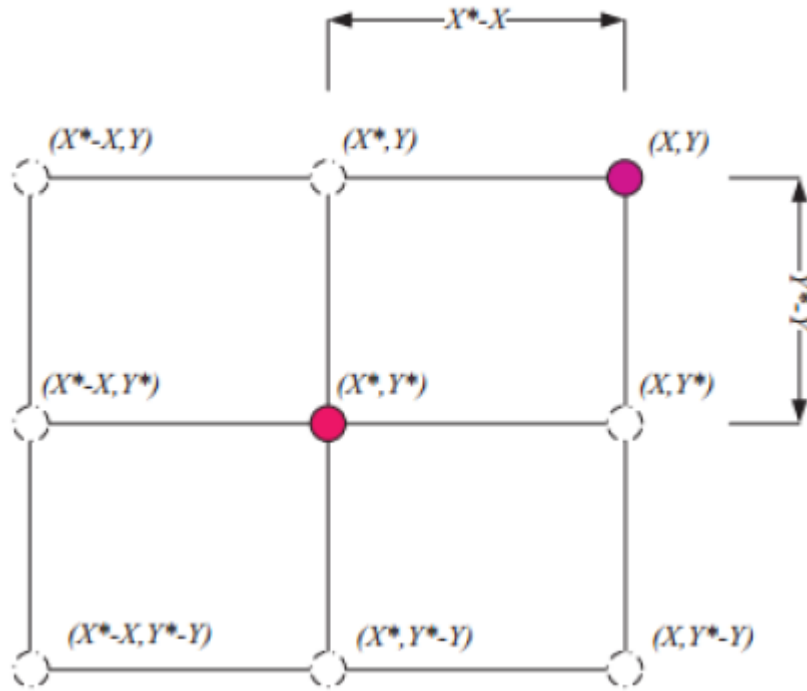
t , mevcut iterasyonu, \vec{A} ve \vec{C} yakınsama vektörlerini, \vec{X}^* en iyi çözüm vektörünü ifade etmektedir. Denklem 2.4 ve 2.5 optimizasyon problemindeki n boyutlu çözüm uzayının daha verimli olarak aranmasını sağlamaktadır.

$$\vec{A} = 2 \vec{\alpha} \cdot \vec{r} - \vec{\alpha} \quad (2.6)$$

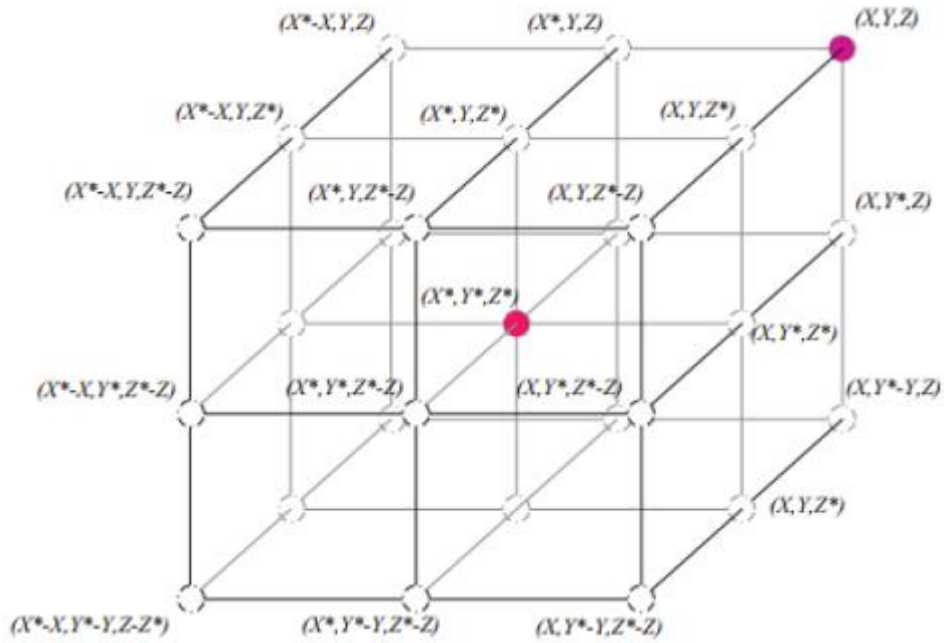
$$\vec{C} = 2 \cdot \vec{r} \quad (2.7)$$

$$a = 2 - m \cdot ((2) / M) \quad (2.8)$$

Denklem 2.6 ve 2.7'deki \vec{r} , rastgele bir vektörü, $\vec{\alpha}$ ise iterasyonlar süresince 2'den 0'a doğru lineer olan azalan bir vektörü temsil etmektedir (Mirjalili ve Lewis 2016). m , mevcut iterasyonu, M ise, maksimum iterasyonu göstermektedir. Arama ajanlarının pozisyon vektörleri ve olası konumları Şekil 2.4 ve Şekil 2.5'te gösterilmiştir.



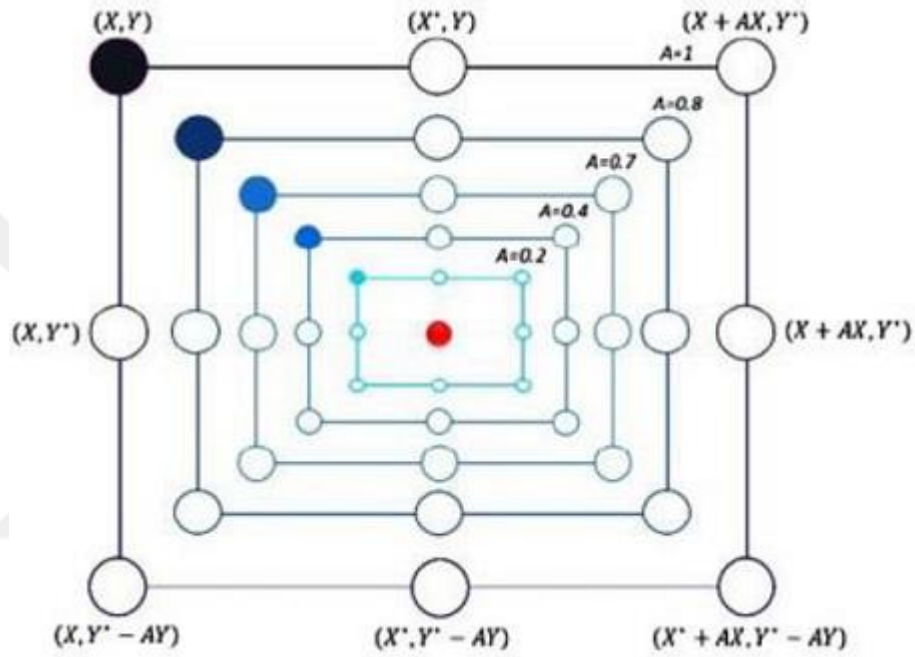
Şekil 2.4. Arama ajanlarının iki boyutlu pozisyon vektörleri ve olası konumları



Şekil 2.5. Arama ajanlarının üç boyutlu pozisyon vektörleri ve olası konumları

2.4.2.2. Ava doğru hareket etme

Bu aşama, avın etrafında oluşan çemberi daraltma ve spiral hareket olarak iki bölümde modellenmiştir. Denklem 2.6'daki α 'nın değeri azaltıldığında, avın etrafındaki çember de daralmaktadır. Şekil 2.6 (Khalilpourazari ve Khalilpourazary 2018) en iyi kambur balinanın konumuna (kırmızı nokta) göre kendi konumunu doğrusal olarak güncellemeye çalışan bir balinanın muhtemel pozisyonlarını ifade eder.



Şekil 2.6. Daralan çevreleme mekanizması

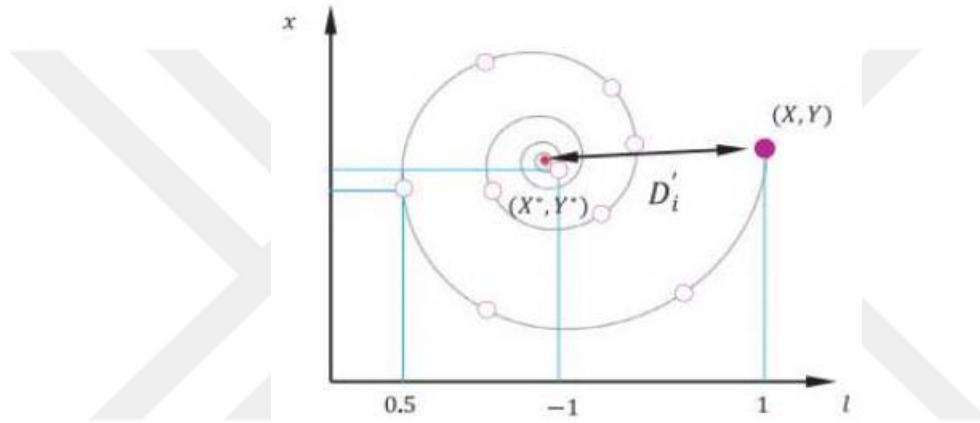
Gösterilen spiral hareket için hedeflenen konum (en iyi çözüm adayı) ile çözüm adayı arasındaki mesafe hesaplanmış ve Denklem 2.9 oluşturulmuştur (Tanyıldızı ve Cigalı 2017).

$$\vec{X}(t+1) = \vec{D}' \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}^*(t) \quad (2.9)$$

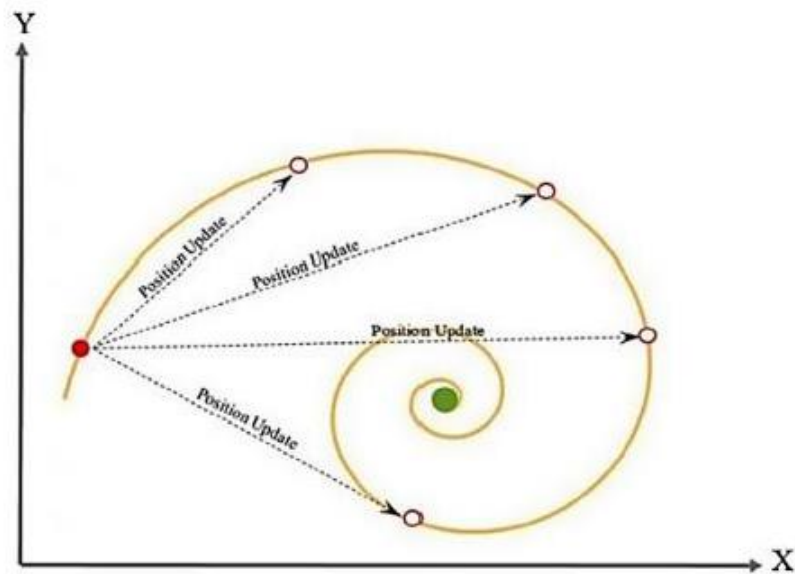
Denklem 2.9'daki $\vec{D}' = \vec{X}^*(t) - \vec{X}(t)$ 'yi ifade etmektedir. Bu ifade arama ajanı ile bilinen en iyi nokta arasındaki mesafeyi vermektedir. b , logaritmik spiral sabiti, l ise $[-1, 1]$ aralığında rastgele bir sayıyı ifade etmektedir. Algoritmanın, spiral hareket ve doğrusal hareketten hangisini seçeceği ise, Denklem 2.10'da gösterildiği gibi $\frac{1}{2}$ olasılıkla belirlenir. p , $[0, 1]$ aralığında rastgele bir sayıyı ifade etmektedir.

$$\vec{X}(t+1) = \begin{cases} \vec{X}^*(t) - \vec{A} \cdot \vec{D} , & p < 0,5 \\ \vec{D}' \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}^*(t) , & p \geq 0,5 \end{cases} \quad (2.10)$$

Doğadaki kumbur balinalar daralan çevreleme mekanizmasını ve spiral hareketi aynı anda gerçekleştirir. Yani, bir kumbur balina en iyi kumbur balınaya göre konumunu güncellerken çember daraltma hareketini veya spiral hareketi seçebilir. Bu nedenle BOA bu iki davranışı yüzde 50 olasılıkla kullanır. Şekil 2.7'de spiral hareket (Mirjalili ve Lewis 2016), Şekil 2.8'de en iyi çözümün konumu (Khalilpourazari ve Khalilpourazary 2018) gösterilmektedir.



Şekil 2.7. Spiral hareket etme



Şekil 2.8. Konumun spiral güncellenmesi

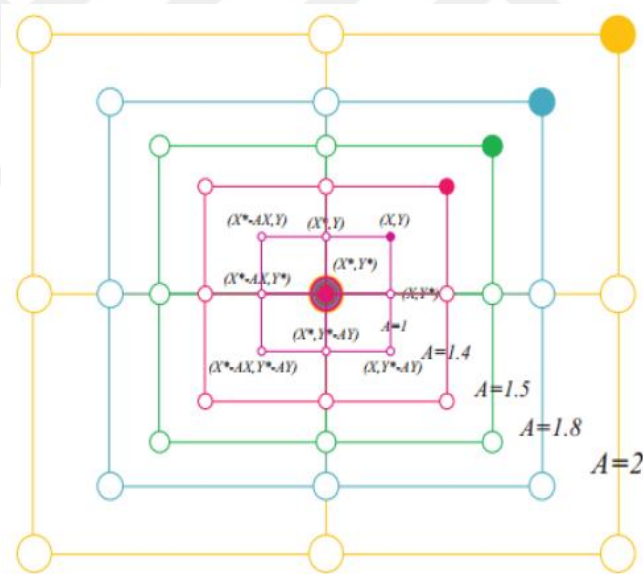
2.4.2.3. Av arama

Denklem 2.11 ve 2.12’de matematiksel modeli gösterilen global çözüm, çözüm adaylarının yeni konumlarını bilinen en iyi çözüm adayı üzerinden hesaplamak yerine, rastgele seçtiği bir çözüm adayının etrafında belirlemektedir.

$$\vec{D}^i = \vec{C} \cdot \overrightarrow{X_{rand}} - \vec{X} \quad (2.11)$$

$$\vec{X}(t+1) = \overrightarrow{X_{rand}} - \vec{A} \cdot \vec{D} \quad (2.12)$$

$\overrightarrow{X_{rand}}$, rastgele seçilen bir çözüm vektörünü göstermektedir. \vec{A} vektörünün değeri, global veya yerel arama için belirleyicidir. $\vec{A} > 1$ veya $\vec{A} < -1$ durumlarında en iyi noktadan daha uzak bir konum seçilebileceği için global arama olarak kabul edilir. Bu durumda Denklem 2.11 ve 2.12 uygulanmaktadır. Algoritmanın global arama davranışı Şekil 2.9’da gösterilmiştir (Mirjalili ve Lewis 2016).



Şekil 2.9. Global arama

2.4.3. BOA sözde kodu ve akış diyagramı

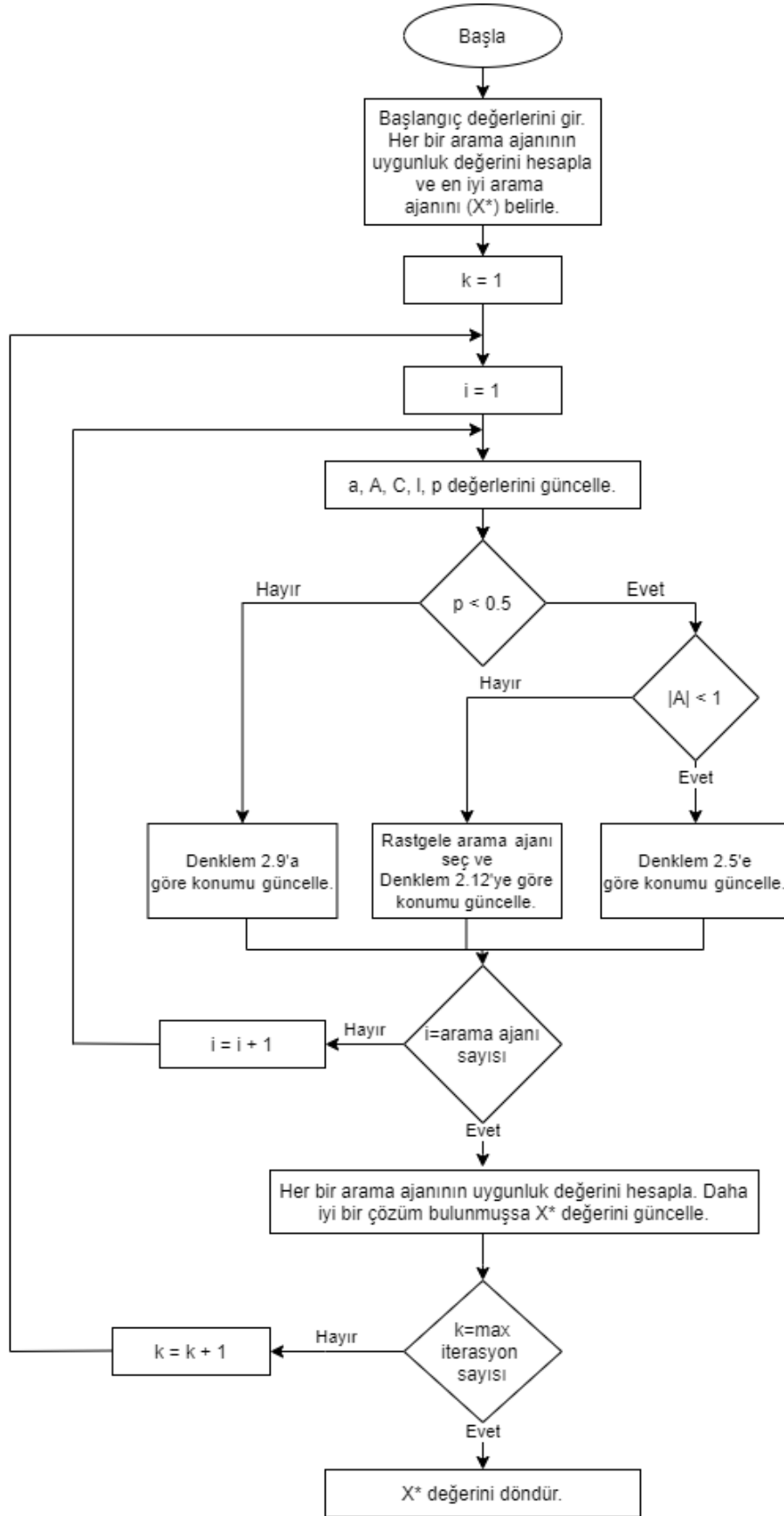
Balina Optimizasyon Algoritmasının sözde kodu Çizelge 2.1’de (Tanyıldızı ve Cigalı 2017), akış diyagramı Şekil 2.10’da (Doğan 2019) verilmiştir.

Çizelge 2.1. BOA sözde kodu

```
Başlangıç popülasyonu ayarla  $X_i$  ( $i = 1, 2, \dots, n$ )
Her bir arama ajanının uygunluk değerini hesapla
 $X^*$  = Bilinen en iyi arama ajanı
while ( $t <$  maksimum iterasyon sayısı)

for (her bir arama ajanı için)
  Güncelle  $a, A, C, l$ , ve  $p$ 
  if ( $p < 0.5$ )
    if ( $|A| < 1$ )
      Denklem 2.5 ile arama ajanı konumu güncelle
    else if ( $|A| \geq 1$ )
      Rastgele bir arama ajanı seç ( $\overrightarrow{X_{rand}}$ )
      Denklem 2.12 ile arama ajanını güncelle
    end if
  else if ( $p \geq 0.5$ )
    Denklem 2.9 ile arama ajanı konumu güncelle
  end if
end for

Kısıt dışına çıkan bireylere sınır değerini ver
Amaç fonksiyon değerlerini hesapla
Daha iyi çözüm bulunmuşsa en iyi ajanı güncelle
 $t = t + 1$ 
end while
Sonuç  $X^*$ 
```



Şekil 2.10. BOA akış diyagramı

2.5. Levy Flight Arama Stratejisi

Levy Flight 1937 yılında Fransız matematikçi Paul Levy tarafından tanıtılmıştır. Kendisinden yüzyıl kadar önce keşfedilen ve daha geleneksel olan Brownian hareketinin ötesine geçen istatistiksel bir arama stratejisidir.

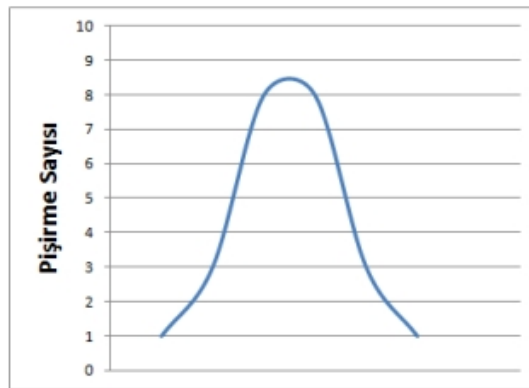
2.5.1. Levy uçuşu

Levy Flight, bazı hayvanların kullandığı yemek arama stratejilerinden biridir. Adım uzunluklarının bir sürekli olasılık dağılımına uygun olarak gerçekleştiği hareket örüntüsünü ifade eder. Yani, aslında sürekli bir olasılık dağılımıdır.

Levy uçuşu bir rastgele uçuş örneğidir. Rastgele uçuş, yere bıraktığımız bir topun yapacağı hareketin formüle edilebilen bir matematiksel modele sahip değildir. Bunun yerine, hareketlerin bir takım sınırlar veya kurallar nedeniyle belli bir olasılık dağılımına uyduğu istatistik bir modeli vardır.

Bir örnekle açıklamak gerekirse; annelerimiz genellikle ne yemek yapacaklarına rastgele olarak karar verirler. Bu yemeklerden bir kısmını yapmak çok zahmetliyken, bir kısmını yapmak ise çok kolaydır. Yapılacak yemeğin sağlıklı ve besleyici olması üzerinden başka bir değerlendirme yapmak da mümkündür. Şekil 2.11 örneğindeki annenin çalışan bir kadın olduğunu varsayalım. Zaman yönetimi onun için önemli olacaktır ancak çocuğunun iyi beslenmesini de istemektedir. Bu annenin hangi yemeği yapacağına yönelik kesin bir bağıntı geliştirmek ve tahminde bulunmak mümkün olmayacaktır. Ancak belli bir süre zarfında hangi yemekleri yaptığı incelenerek yemek seçimlerinin bir olasılık dağılımına uyup uymadığına bakabiliriz.

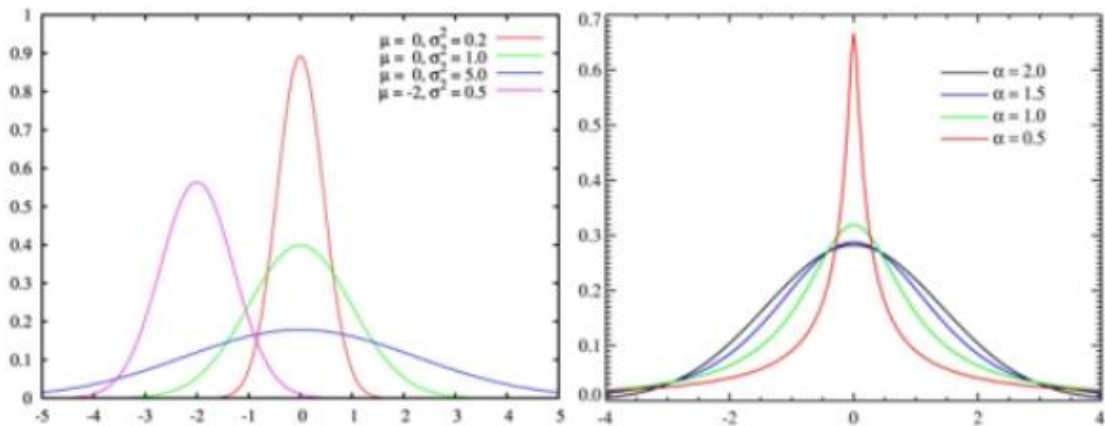
YEMEK NİTELİĞİ		N
Besleyicilik	Zamanlama	
Çok kötü	Çok iyi	1
Kötü	İyi	3
Fena Değil	İyi Sayılır	8
İyi Sayılır	Fena Değil	8
İyi	Kötü	3
Çok iyi	Çok kötü	1



Şekil 2.11. Yemek seçiminde farklı kriterlere göre örnek bir dağılım

Bu anne her ne kadar yemek yapmaya rastgele karar veriyorsa olsa da zaman ve besleyicilikle ilgili kaygıları nedeniyle ortaya şekildeki gibi bir tablo çıkıyorsa; bir ucunda besleyici ama zor, diğer ucunda kolay ama besleyici olmayan yemeklerin yer aldığı bir normal dağılım elde edebiliriz. Mesela omlet çok hızlı yapılabilir, ama tek başına yeteri kadar lezzetli ya da besleyici değildir. Bu yüzden incelenen zaman zarfında bir kez yapılmıştır. Diğer yandan güveç gibi bir yemek zengin içerikli ve besleyici olabilir; ancak yapımı çok zahmetlidir. Bu yemek de sadece bir kez yapılmış olabilir. Kıyaslama yapıldığında daha zor ama daha besleyici ya da daha besleyici ama daha zor yemeklerden optimum olanların daha sık yapılması beklenir.

Benzer şekilde Levy uçuşu olarak adlandırılan davranış, hayvanların iki yiyecek arama eylemi arasında kat ettikleri mesafelerin Levy dağılımı gibi sürekli olasılık dağılımlarına uygun olarak gerçekleşmektedir. Levy dağılımı tıpkı normal dağılım gibi sürekli olasılık dağılımı olmasına rağmen bu dağılımı normal dağılımdan farklı kılan bir takım varsayım, değer ve koşullardır.

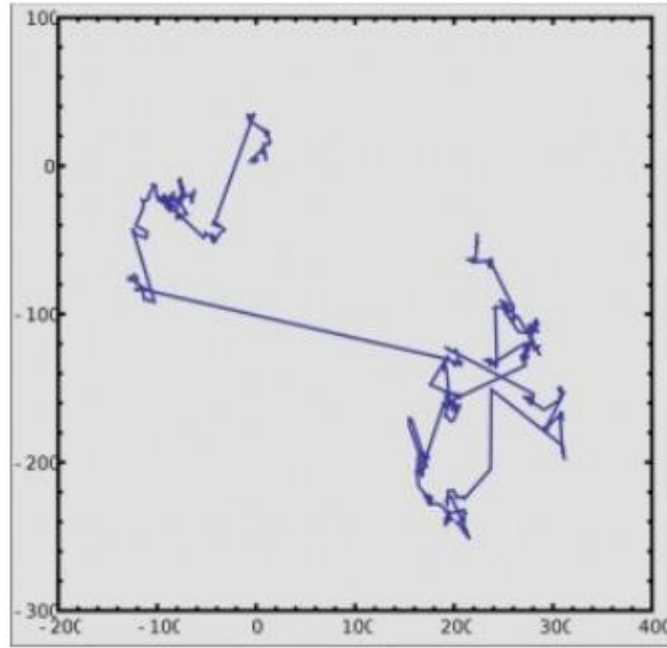


Şekil 2.12. Normal dağılım ve Levy dağılımı

Şekil 2.12’de gösterilen normal dağılımda μ Ortalama, σ^2 varyans değeridir. Anne-yemek örneğimizden devam edecek olursak, yemekler özellikleri bakımından birbirinden uzaklaşıp farklılaştıkça varyans büyüyecek ve grafiğimiz daha yayvan olacaktır. Anne iş yerinde sürekli mesaiye kalıp daha çabuk yapılan yemekleri seçseydi, bu defa da ortalama negatif bir sayı olacağından grafik sola doğru kayacaktı. Levy dağılımında α ölçeği dağılımı oluşturan değerlerin birbirinden ne kadar uzak ve farklı olduklarını temsil eder. Levy dağılımının doğası gereği α ölçeğindeki küçük değişiklikler dağılımda büyük değişimlere neden olmaktadır.

Yiyecek arayan bir bal arısını düşündüğümüzde arının enerjisinin ve çevredeki kaynakların sınırlı olduğunu görürüz. Arı çiçek ararken rastgele bir noktaya ışınlanmadığına göre önceki konumu ve sıradaki konumu birbirine bağlıdır. Peki arı belli bir alandaki çiçekleri gezdikten sonra sonraki konumunu neye göre ve nasıl belirleyecektir?

Bal arısı yiyecek arayacağı alana gelir ve bir çiçeği kontrol eder. Kısa uçuşlar yaparak bölge içindeki diğer çiçekleri gezer. Bu küçük alanda bir miktar kısa uçuşlar yaptıktan sonra rastgele bir yön seçer ve daha uzun bir uçuş yaparak başka bir bölgeye geçer. Geldiği yeni alanda da aynı şekilde kısa uçuşlar yapar ve akabinde tekrar bir uzun uçuş yapar. Yuvasına dönene kadar bu döngüyü takip eder. Şekil 2.13 bilgisayar ortamında oluşturulan örnek bir Levy uçuşunu göstermektedir.



Şekil 2.13. Bilgisayarda oluşturulmuş 1000 hamlelik bir Levy uçuşu

Araştırmalar arılar, ton balıkları ve köpek balıklarının da içinde yer aldığı pek çok hayvanın yiyecek ararken katettikleri mesafenin Levy dağılımına uygun olduğunu göstermiştir. Rassal yiyecek arama davranışlarında zaman ve enerji açısından optimum strateji çalışmaları ise, Levy dağılımı gibi olasılık dağılımlarına uygun davranış örüntülerinin en iyi strateji olabileceğini gösterdi (Viswanathan ve diğ. 1999). Bu araştırmalarda, hız ve konum bilgilerini ileten sensörleri hayvanlar üzerine yerleştirilerek gerçek veriler üzerinden kanıtlar elde edildi. Benzer şekilde 2010 yılında yapılan ve

milyonlarca konum verisini içeren okyanus arařtırmaları bazı okyanus balıklarının yařadıkları bölgeye göre Levy da da Brown dađılımlarını kullandığını göstermiştir (Humphries ve diđ. 2010).

2.5.2. Matematiksel modeli ve yapısı

Her nesilde bir dizi nesne kullanılarak, en iyi bilinen konumdan başlanır ve rastgele dađıtılmış mesafelerde yeni bir nesil üretilir. Ardından, yeni nesil en umut verici olanı seçmek için deđerlendirilir ve durma kriterleri sađlanana kadar bu iřlem tekrarlanır.

Genel olarak, hayvanların yiyecek arama davranışı bir tür rastgele harekettir. Bir sonraki hareket, mevcut pozisyona ve sonraki pozisyona geçme ihtimaline bađlı olduđundan, yapılan her rastgele hareketin önemi büyüktür. Yapılan son arařtırmalar, Levy Flight'ın, rastgele hareket modelinde en iyi arama stratejilerinden biri olduđunu göstermektedir (Pavlyukevich 2007, Reynolds ve Frye 2007, Shlesinger 2006).

Daha önceki çalıřmalar incelendiđinde, Levy Flight'ın orijinal halinin yanı sıra modifiye edilmiş haliyle de uygulamalarda kullanıldıđı görülmüřtür. Pek çok arařtırmacı, kesilmiş Levy Flight, düzgün kesilmiş Levy Flight, kademeli kesilmiş Levy Flight gibi modifiye edilmiş Levy Flight tekniklerini optimizasyon iřlemlerinde kullanmıştır. Bu çalıřmada Levy Flight tekniđi orijinal haliyle kullanılmıştır.

Levy Flight, Gauss dıřı rastgele süreçlerin bir sınıfıdır (Chechkin ve diđ. 2008). Rastgele hareket, Levy kararlı dađılımından yararlanılarak oluşturulur. Bu dađılım aslında basit bir güç yasası formülüdür. Matematiksel olarak, Levy dađılımının basit bir versiyonu Denklem 2.13'teki gibi tanımlanabilir (Yang ve Deb 2013, Yang 2010):

$$L(s, \gamma, \mu) = \begin{cases} \sqrt{\frac{\gamma}{2\pi}} \exp\left[-\frac{\gamma}{2(s-\mu)}\right] \frac{1}{(s-\mu)^{3/2}} & \text{if } 0 < \mu < s < \infty, \\ 0 & \text{if } s \leq 0 \end{cases} \quad (2.13)$$

μ konum veya yer deđiřtirme parametresi, $\gamma > 0$ parametresi dađılım ölçeđini kontrol eden ölçek parametresidir. Genel olarak, Levy dađılımını Denklem 2.14'te formülize edilen Fourier dönüřümü olarak tanımlanmalıdır.

$$F(k) = \exp[-\alpha|k|^\beta], \quad 0 < \beta \leq 2, \quad (2.14)$$

α , $[-1, 1]$ aralıđında bir parametredir ve çarpıklık veya ölçek faktörü olarak bilinir. Stabilité indeksi olan $\beta \in (0, 2]$, aynı zamanda Levy indeksi olarak da adlandırılır. İntegralin analitik formu, birkaç özel durum dıřında genel β için

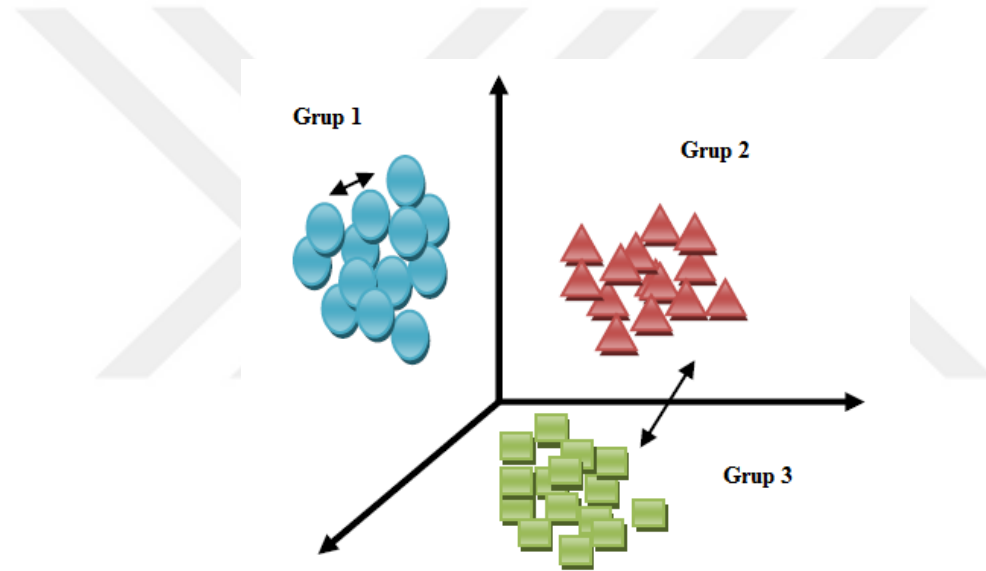
bilinmemektedir. β ve α parametreleri dağılımın belirlenmesinde büyük rol alırken, γ ve μ parametrelerinin etkisi küçüktür. β parametresi, olasılık dağılımının şeklini, özellikle kuyruk bölgesinde farklı olasılık dağılım şekilleri elde edecek şekilde kontrol eder. Bu nedenle, daha küçük olan β parametresi daha uzun kuyruk oluşturacağı için dağılımın daha uzun atlamalar yapmasına neden olur (Lee ve Yao 2001). Eğiklik parametresi α 'nın işareti, eğrinin yönünü gösterir.

Pozitif işaretli değerler sağ yönü ve negatif değerler sol yönü ifade eder. $\alpha = 0$ olduğunda, dağılım simetriktir. Son iki parametre, γ genişliği ve μ değişimi dağılımın tepe noktalarıdır (Al-Temeemy ve diğ. 2010). β parametresinin farklı değerleri dağılımı değiştirir. Daha küçük değerler için daha uzun atlamalar yaparken, daha büyük değerler için daha kısa atlamalar yapar.



3. KÜMELEME

Veri madenciliği uygulamalarından biri olan kümeleme, veri topluluğundaki elemanları benzerliklerine göre gruplandırma esasına dayanmaktadır. Kümeleme yöntemleri, hangi gruba ait olduğu bilinmeyen verilerin benzer özellikteki alt kümelere ayrılmasına yardımcı olan metotlardır (Özdamar 1999, Tatlıdil 1996). Kümelemenin esas amacı, nesnelerin sahip olduğu karakteristik özellikleri kullanarak gruplandırma yapmaktır. Kümeleme yapılırken, benzer bireylerin aynı grupta, farklı bireylerin ayrı grupta olmasına dikkat edilir. Şekil 3.1’de görüldüğü gibi aynı küme içindeki bireylerin birbirine olan uzaklığı çok az, farklı kümelerin birbirine olan uzaklığı ise daha fazladır (Hair ve diğ. 1998).



Şekil 3.1. Nesneler ve kümeler arası uzaklık

Kümeleme, veri setindeki değişkenlerin gruplandırılmaları ile ilgili kesin bilginin bulunmadığı bir popülasyon içinde yapılır. Bu popülasyondan alınan n tane verinin, p tane değişkene ilişkin gözlem sonuçlarına bakılır. Kümelemede benzer nitelikteki bireyler birbiriyle birleştirilerek kümelere ayrılır. Kümeleme yapılması, gözlem sonuçlarını çok az bir kayıpla bir araya toplamaya olanak sağlar (Lorr 1983).

Kümeleme analizinin amaçları, kullanım alanı ve amacına göre aşağıdaki şekilde sıralanabilir (Ball 1970):

- Doğru tiplerin belirlenmesi
- Veri araştırma (inceleme)
- Veri indirgeme

- Hipotez oluřturma
- Hipotez testi
- Gruplara dayalı tahmin
- Model oluřturmak

Arařtırmacılar birçok alanda, gruplandırma yapmadan kontrol edilemeyen büyük boyutlu verilerle çalışmaktadır. Kümeleme teknikleri bu tür verilerin indirgenmesinde kullanılabilir. Örneğın, pazarlama arařtırması alanındaki bir uygulamada, pazar testi için çok sayıda şehir kullanılabilmesine rağmen ekonomik faktörlere baėlı olarak şehir sayısının azaltılması gereklidir. Bu şehirlerden birbirine en çok benzeyenler küçük gruplara ayrılarak kümelendirildiėi takdirde her gruptaki bir şehir bir test pazarı olarak kullanılabilir (Everitt 1993).

Kümeleme analizinde N adet gözlemin her birinde p adet ölçümün yapıldığı Nx_p boyutlu veri matrisi ařaėıdaki gibi gösterilebilir (Çakmak 1999):

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & x_{22} & \dots & x_{2p} \\ \dots & \dots & \dots & \dots \\ x_{N1} & x_{N2} & \dots & x_{Np} \end{bmatrix}$$

Burada x_{ij} , j. deėişkenin i. birey ya da nesne için aldığı deėeri göstermektedir.

Kümeleme, aynı gruptaki nesnelere birbiriyle benzer, farklı gruptaki nesnelere birbirinden farklı olması fikrine dayanmaktadır. Nesnelere arasındaki bu iliřkiyi daha iyi anlamak amacıyla benzerlik ve farklılık (benzersizlik) kavramları kullanılmaktadır. İki nesnenin benzerliėi ne kadar fazlaysa, nesnelere arasındaki iliřki de o derece kuvvetlidir. Nesnelere arası iliřkinin zayıf olduėu örneklerde ise, nesnelere arasındaki farklılıėın fazla olduėu söylenebilir. Benzerlik/farklılık deėerlerini ölçmek için çeřitli metotlar bulunmaktadır ve bu metotlar kullanılarak nesnelere daha rahat şekilde gruplandırılabilir.

Nesnelere arasındaki benzerliėi ölçmede kullanılacak metot seėilirken, nesnelere ait özelliklerin veri tipleri dikkate alınır. Bir nesneye ait özelliėin aldığı deėerlerin sayısı sonlu ise, bu veri tipi kesikli (discrete) veri kategorisindedir. Ancak nesneye ait özellik birden çok aralıkta ve aralıklardaki her deėeri alabiliyorsa sürekli (continuous) veri kategorisindedir (Doėan 2002, Servi 2009).

3.1. Kümelemede Kullanılan Uzaklık Yöntemleri

Literatürde, uzaklık ölçme yöntemi olarak kullanılan bazı metotlar aşağıda paylaşılmıştır. Ölçme yönteminin belirlenmesinde verilerin sürekli veya kesikli olması etkili olmaktadır. Bu çalışmada, her bir veri örneği ile veri örneğinin ait olduğu küme merkezi arasındaki mesafe Öklid uzaklık fonksiyonuna göre hesaplanmıştır.

Öklid (Euclidean) uzaklığı

Yaygın olarak kullanılan Öklid uzaklık metodu, belirlenen iki nokta (nesne) arasındaki doğrusal uzaklığı Denklem 3.1'e göre hesaplar.

$$d(i, j) = \sqrt{\sum_{k=1}^n (x_{ik} - x_{jk})^2} \quad (3.1)$$

Eğer kümeleme analizinde kullanılacak değişkenler belirli bir önem derecesine göre ağırlıklandırılmış ise, Öklid uzaklık formülü Denklem 3.2'ye göre kullanılır.

$$d(i, j) = \sqrt{\sum_{k=1}^n w_k (x_{ik} - x_{jk})^2} \quad (3.2)$$

$d(i, j)$: i . ve j . nesnenin birbirine uzaklığı

x_{ik} : i . nesnenin k . niteliğinin değeri

x_{jk} : j . nesnenin k . niteliğinin değeri

$i = 1, 2, \dots, N$

$j = 1, 2, \dots, N$

$w_k = k$. niteliğin ağırlık değeri

N : Veri setindeki toplam nesne sayısı.

Manhattan uzaklığı

Manhattan uzaklık bulma yönteminde nesnelere arasındaki mutlak uzaklık değeri kullanılır ve Denklem 3.3'e göre hesaplanır.

$$d(i, j) = (|x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \dots + |x_{ip} - x_{jp}|) \quad (3.3)$$

$d(i, j)$: i . ve j . nesnenin birbirine uzaklığı

x_i : Veri setindeki i . nesne

x_j : Veri setindeki j . nesne

Pearson uzaklığı

Pearson uzaklık metotunda, iki nesne arasındaki uzaklık Denklem 3.4 kullanılarak bulunur. Bu formülde kullanılan S_n değeri, uzaklığın hesaplandığı değişkene ait varyanstır.

$$d(i, j) = \sqrt{(x_{i1} - x_{j1})^2/S_1^2 + (x_{i2} - x_{j2})^2/S_2^2 + \dots + (x_{ip} - x_{jp})^2/S_p^2} \quad (3.4)$$

$d(i, j)$: i . ve j . nesnenin birbirine uzaklığı

x_i : Veri setindeki i . nesne

x_j : Veri setindeki j . nesne

Minkowski uzaklığı

Minkowski metodu kullanılarak iki nesne arasındaki uzaklık, Denklem 3.5'e göre hesaplanır.

$$d(i, j) = [|x_{i1} - x_{j1}|^m + |x_{i2} - x_{j2}|^m + \dots + |x_{ip} - x_{jp}|^m]^{1/m} \quad (3.5)$$

$d(i, j)$: i . ve j . nesnenin birbirine uzaklığı

x_i : Veri setindeki i . nesne

x_j : Veri setindeki j . nesne

Bu yöntemde m değeri 1 seçilirse formül, Manhattan uzaklık metotuna formülüne, m değeri 2 seçilirse Öklid uzaklık metotuna dönüşür. Bu açıdan Minkowski metotunun, genel bir formül olduğu ve m değişkeninin farklı değerlerine göre yeni formüller türetileceği görülebilir.

Mesafe ölçmek için yaygın olarak kullanılan yukarıdaki uzaklık metotları değişken değerlerinin sürekli olması halinde uygulanmaktadır (Tatlıdil 1996, Doğan 2002). Çizelge 3.1'de sık kullanılan diğer uzaklık metotları görülmektedir (Duran ve Odell 1974).

Çizelge 3.1. Uzaklık metotları ve formülleri

Uzaklık Metodu	Matematiksel Gösterim
Sup-norm	$d_{\infty}(x_i, x_j) = \text{svp} \{ x_{ki} - x_{kj} \}$
Mahalanobis	$D^2(x_i, x_j) = (x_i - x_j)^T w^{-1} (x_i - x_j)$
B ₁ norm	$d_1(x_i, x_j) = \left(\sum_{k=1}^p x_{ki} - x_{kj} \right)$
B _p norm	$d_p(x_i, x_j) = \left(\sum_{p=1}^k x_{ki} - x_{kj} ^p \right)^{1/p}$

Değişkenlerin kesikli verilerden oluşması halinde, sürekli verilerin kümelenmesinde kullanılan uzaklık metot fonksiyonlarına, w_t katsayısı eklenmektedir. Bu katsayı Denklem 3.6'ya göre hesaplanır ve d_t değeri, t . değişkenin dağılım aralığını ifade etmektedir (Tatlıdil 1996).

$$w_t = \begin{cases} 1, & \text{nicel veriler için} \\ \frac{1}{d_t}, & \text{nitel veriler için} \end{cases} \quad (3.6)$$

3.2. Kümelemede Kullanılan Benzerlik Yöntemleri

Benzerlik kavramı, uzaklık kavramının tersi olarak iki veri arasındaki yakınlığı ifade etmektedir. Cosine Denklem 3.7, Overlap Denklem 3.8, Dice Denklem 3.9 ve Jaccard Denklem 3.10 kullanılarak ölçülen benzerlik yöntemlerindedir.

İki vektör arasındaki benzerliğin ölçülebilmesi için değerlerin pozitif ve sürekli olması gerekir. Vektörler birebir aynı olduğunda benzerlik 1; vektörlerden biri 0, diğeri 0'dan büyük olduğunda benzerlik 0 olacaktır. Yani, benzerlik ölçümünden elde edilen sonuç 0-1 aralığında bir değerdir.

$$\text{sim}(x_m, x_j)_{\text{COSINE}} = \frac{\sum_{i=1}^n x_{mi} x_{ji}}{\sqrt{\sum_{i=1}^n x_{mi}^2 * \sum_{i=1}^n x_{ji}^2}} \quad (3.7)$$

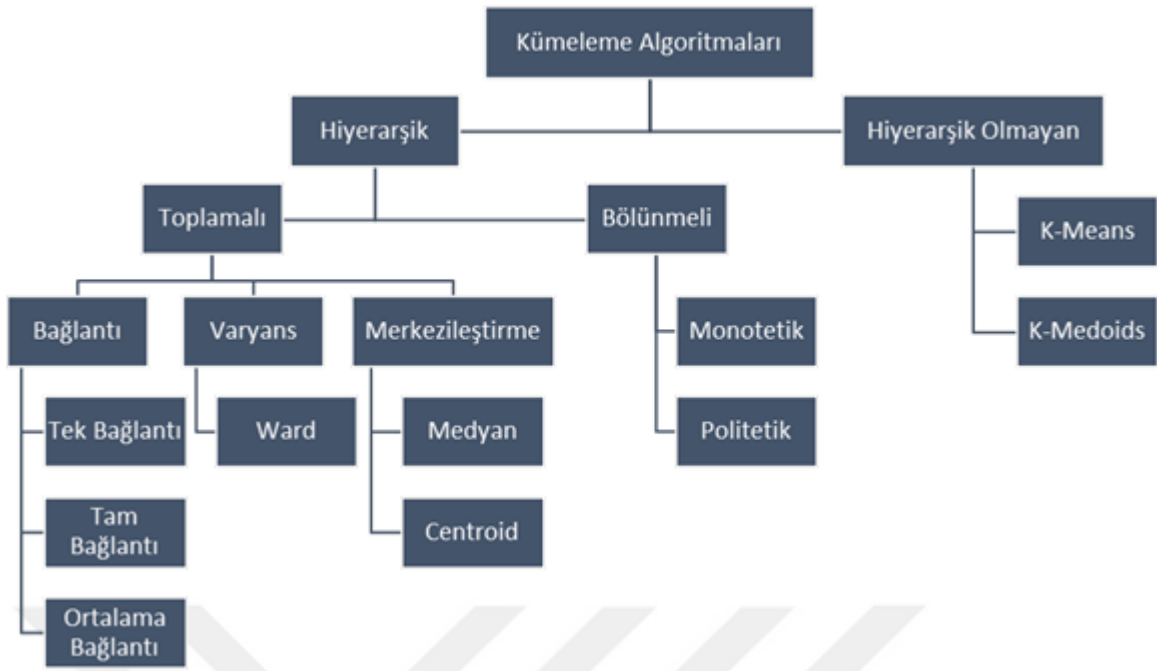
$$sim(x_m, x_j)_{OVERLAP} = \frac{\sum_{i=1}^n x_{mi} x_{ji}}{(\sum_{i=1}^n x_{mi}^2 + \sum_{i=1}^n x_{ji}^2)} \quad (3.8)$$

$$sim(x_m, x_j)_{DICE} = \frac{2 \sum_{i=1}^n x_{mi} x_{ji}}{\sum_{i=1}^n x_{mi}^2 + \sum_{i=1}^n x_{ji}^2} \quad (3.9)$$

$$sim(x_m, x_j)_{JACCARD} = \frac{\sum_{i=1}^n x_{mi} x_{ji}}{\sum_{i=1}^n x_{mi}^2 + \sum_{i=1}^n x_{ji}^2 - \sum_{i=1}^n x_{mi} x_{ji}} \quad (3.10)$$

3.3. Kümeleme Algoritmaları

Literatürde, kümeleme problemiyle ilgili birçok yaklaşım olsa da kümeleme algoritmaları temel olarak hiyerarşik ve hiyerarşik olmayan kümeleme algoritmaları olmak üzere ikiye ayrılmaktadır. Hiyerarşik kümeleme yöntemleri uygulanırken küme numarasının belirlenmesine gerek yoktur. Bu yöntemlerde, kümeleme işlemi bittikten sonra küme numarası belirlenir. Ancak hiyerarşik olmayan kümeleme yöntemlerinde kümeleme yapabilmek için küme numarası bilgisine ihtiyaç vardır. Özetle, hiyerarşik olmayan kümelemenin amacı, n adet veri örneğini k adet kümeye ayırmaktır. Zaman karmaşıklığı açısından bakıldığında, hiyerarşik kümeleme yöntemlerinin ikinci dereceden olduğu, hiyerarşik olmayan kümeleme yöntemlerinin ise doğrusal olduğu görülmektedir (Boushaki ve diğ. 2018, Frigui ve Krishnapuram 1999). Kümeleme algoritmaları Şekil 3.2’de genel olarak kategorize edilmiştir (Karakoyun 2015).



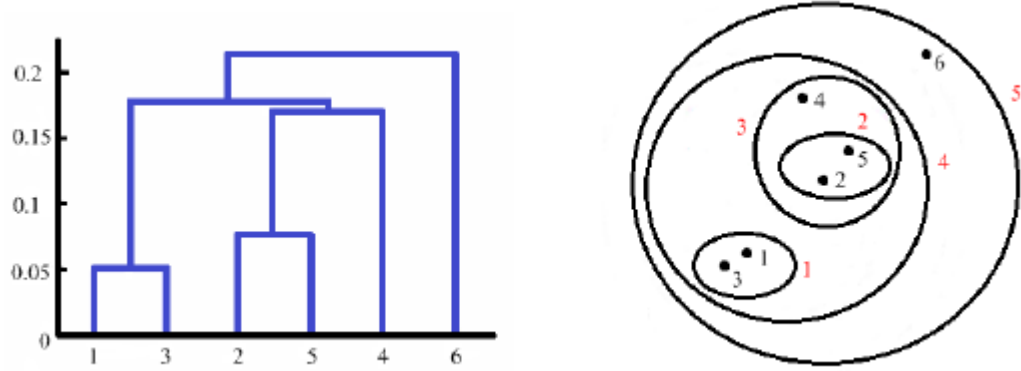
Şekil 3.2. Kümeleme algoritmaları

Kümeleme analizi yapılırken, ilk aşamada bir benzerlik veya uzaklık ölçütü seçilmelidir. Daha sonra hangi kümeleme tekniğinin (hiyerarşik veya hiyerarşik olmayan gibi) kullanılacağı belirlenmelidir. Sonraki adımda seçilen teknik için kullanılacak kümeleme yöntemi türü seçilir ve son aşamada kümeleme sonucunu yorumlayabilmek için küme sayısı belirlenir (Sharma 1996).

3.3.1. Hiyerarşik kümeleme algoritmaları

Hiyerarşik kümeleme algoritmalarında, nesnelere belirli adımlardan geçerek kümeler ayrılır ve nesnelere kümelerdeki aidiyeti ölçülür. Bu aidiyet ölçüsü, benzerlik veya farklılık özelliklerinden elde edilir. Hiyerarşik kümeleme tekniklerinde, kümeler peş peşe birleştirilir ve bir grup, diğer grupla birleştirildikten sonra kalan aşamalarda kesinlikle ayrılamaz (Fırat 1997).

Hiyerarşik kümelemede, nesnelere yakınlık ilişkisine göre kümeler oluşturulmaktadır. Oluşturulan kümelerle bir ağaç yapısı inşa edilir ve ağaç diyagramları ile gösterilen sonuçlara dendogram adı verilir (Lorr 1983, Everitt ve Dunn 1992). Ağaç yapısı, gövdeden yapraklara veya yapraklardan gövdeye doğru olabilir. Dendogram istenen seviyede bitirilerek kümeler elde edilmiş olur (Bilen 2009). Şekil 3.3'te (Bilen 2009) örnek bir dendogram gösterilmektedir.



Şekil 3.3. Dendogram yapısına örnek

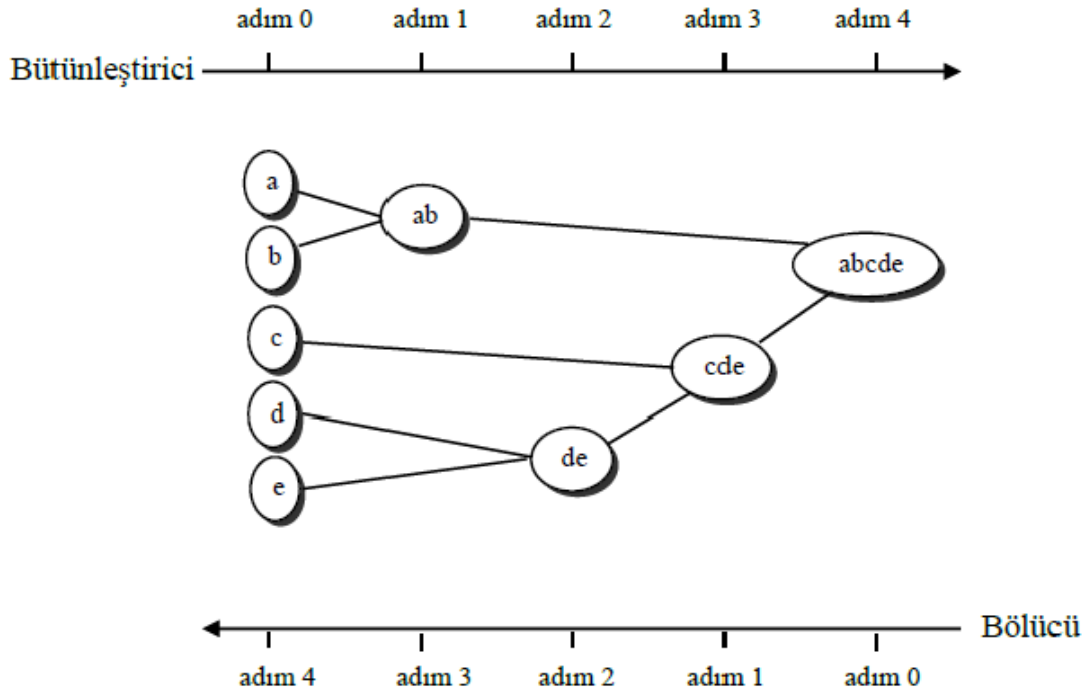
Hiyerarşik yapı yukarıdan-aşağıya doğru oluşturulduğunda toplamalı, aşağıdan-yukarıya oluşturulduğunda bölünmeli hiyerarşik kümeleme olarak sınıflandırılabilir (Bkz. Şekil 3.4 (Bilen 2009))

Toplamalı (aşağıdan yukarıya) yaklaşıma göre hiyerarşik kümeleme adımları:

- Her bir nesne için farklı bir grup oluştur.
- Merkezler arasındaki uzaklık, ortalama gibi kurallara göre grupları birleştir.
- Tüm nesnelere tek bir küme içinde toplanana kadar ya da belirlenen sonlandırma kriterine ulaşılan kadar birleştirme işlemine devam et.

Bölünmeli (yukarıdan aşağıya) yaklaşıma göre hiyerarşik kümeleme adımları:

- Aynı kümede bulunan tüm nesnelere başla.
- Bir kümeyi daha küçük kümelere ayır.
- Her nesne ayrı bir küme oluşturana kadar ya da belirlenen sonlandırma kriterine ulaşılan kadar ayırma işlemine devam et (Bilen 2009).



Şekil 3.4. Bütünleştirici ve bölücü hiyerarşik kümelemenin örnek gösterimi

3.3.1.1. Toplama teknikleri

Toplama tekniklerinde ilk olarak $\{ 1/2 [n (n-1)] \}$ olası gözlem (küme) çifti arasındaki benzerlik veya uzaklık matrisi hesaplanır. Daha sonra benzerlik veya uzaklık matrisine göre en yakın iki küme birleştirilir ve küme sayısı bir indirgenir. Benzerlik/uzaklık matrisi tekrar oluşturulur ve n birim aşamalı olarak sırayla $n, (n-1), (n-2), \dots, (n-r), \dots, 3, 2, 1$ kümeye yerleştirilir (Everitt 1974). Bu teknikler içerisinde tek bağlantı tekniği, tam bağlantı tekniği, ortalama bağlantı tekniği ve Ward's tekniği açıklanacaktır.

Tek bağlantı tekniği

Bu teknik ilk olarak Florek ve arkadaşları tarafından uygulanmıştır. Benzerlik veya uzaklık matrisinden yararlanılarak birbirine en yakın iki gözlem veya küme birleştirilmekte ve birleştirme işlemi tekrarlanmaktadır (Şentürk 1995, Fırat 1997). Birleştirme işleminde benzerlik (s_k) ve uzaklık (d_k) değerleri sırasıyla Denklem 3.11 ve 3.12 kullanılarak elde edilir (Everitt ve Dunn 1992, Şentürk 1995).

$$s_k(i, j) = \max(s_{ki}, s_{kj}) \quad (3.11)$$

$$d_k(i, j) = \min(d_{ki}, d_{kj}) \quad (3.12)$$

Tek bağlantı tekniği sonuçları bir dendogramda veya ağaç diyagramında gösterilebilir. Ağaç yapısındaki dallar, kümeleri ifade etmektedir (Everitt ve Dunn 2001, Şentürk 1995).

Tam bağlantı tekniği

Tam bağlantı tekniği, tek bağlantı tekniğine ters mantıkta çalışmaktadır. Bu teknikte, benzerlik veya uzaklık matrisinden yararlanılarak en yakın iki küme veya gözlem birleştirilmektedir. Birleştirme işleminde benzerlik (s_k) ve uzaklık (d_k) değerleri sırasıyla Denklem 3.13 ve 3.14 kullanılarak elde edilir (Tatlıdil 1996).

$$s_k(i, j) = \min(s_{ki}, s_{kj}) \quad (3.13)$$

$$d_k(i, j) = \max(d_{ki}, d_{kj}) \quad (3.14)$$

Tam bağlantı tekniğinin en önemli dezavantajı, aynı küme içinde yer alan gözlemler arası uzaklık belli bir değerden küçük olduğunda tüm kümeleri başarılı bir şekilde oluşturamamasıdır (Tatlıdil 1996).

Ortalama bağlantı tekniği

Ortalama bağlantı tekniği, Sokal ve Michener tarafından önerilmiştir. Bu teknikte, iki kümede yer alan gözlem çiftleri arasındaki ortalama fark iki küme arasındaki farkı ifade etmektedir (Everitt 1974). Farklı çeşitleri bulunan algoritmanın en yaygın kullanılan türünde gözlem çiftleri arasındaki uzaklığın aritmetik ortalaması hesaplanmaktadır. Ortalama bağlantı tekniği, yaygın olarak biyoloji biliminde kullanılmakla birlikte, sosyal bilimlerdeki kullanım oranı da giderek artmaktadır.

Ward's tekniği

Ward's tekniği, kümeler içindeki varyansı minimum değerde tutmayı amaçlar. Bu amaçla Denklem 3.15 kullanılır:

$$ESS = \sum_{i=1}^n x_i^2 - 1/n \left(\sum_{i=1}^n x_i \right)^2 \quad (3.15)$$

Burada x_i , i inci gözlemin skorudur (Aldenderfer ve Blashfield 1984). Kümeleme sürecinin ilk aşamasında, her bir gözlem bir kümeyi ifade ettiğinden ESS (Sum of Squares Index E) değeri sıfırdır. Diğer aşamalar, ESS’de minimum artışla sonuçlanan gözlemlerin elde edilmesiyle devam etmektedir.

3.3.2. Hiyerarşik olmayan kümeleme algoritmaları

Hiyerarşik olmayan kümeleme algoritmaları, küme sayısının kümeleme işleminden önce belirlendiği durumlarda kullanılmaktadır. Nesnelerin gruplandırılacağı küme sayısı belirlendikten sonra, kümeleme kriterlerine göre nesnelerin hangi kümeye ait olduğu tespit edilir ve atama işlemi yapılır (Everitt ve Dunn 1992, Tatlıdil 1996).

Şekil 3.5’te örnek bir kümeleme işlemi paylaşılmıştır.

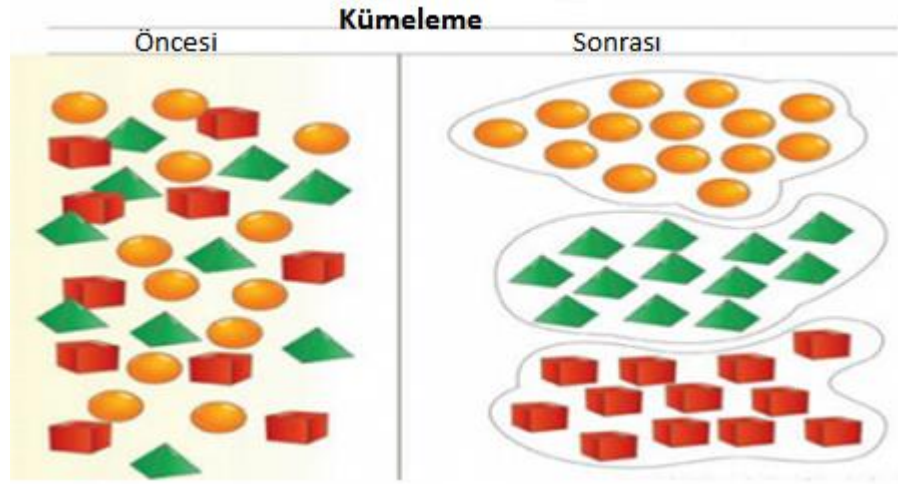
Hiyerarşik olmayan kümeleme algoritmaları, nesnelerin belirlenmiş bir k sayısına göre kümeler ayrılması için tasarlanmıştır. Küme sayısı (k), herhangi bir değer ya da kümeleme tekniğinin bir parçası olarak belirlenebilir. Bu durum, belirlenmesi gereken bir uzaklık/benzerlik matrisinin olmaması ve temel verinin bilgisayarın işlemleri boyunca depolanma mecburiyetinin olmayışıyla ilgilidir.

Hiyerarşik olmayan kümeleme algoritmalarında küme sayısını (k) belirlemek oldukça önemlidir. Sağlıklı bir kümeleme sonucunun elde edilmesi, doğru küme sayısının seçimi ile doğrudan bağlantılıdır. Küçük çaptaki verilerin kümelenmesinde k küme sayısı genellikle Denklem 3.16 kullanılarak bulunur (Atamer 1992).

$$k = \sqrt{\frac{n}{2}} \quad (3.16)$$

n : Kümelenen nesne sayısı

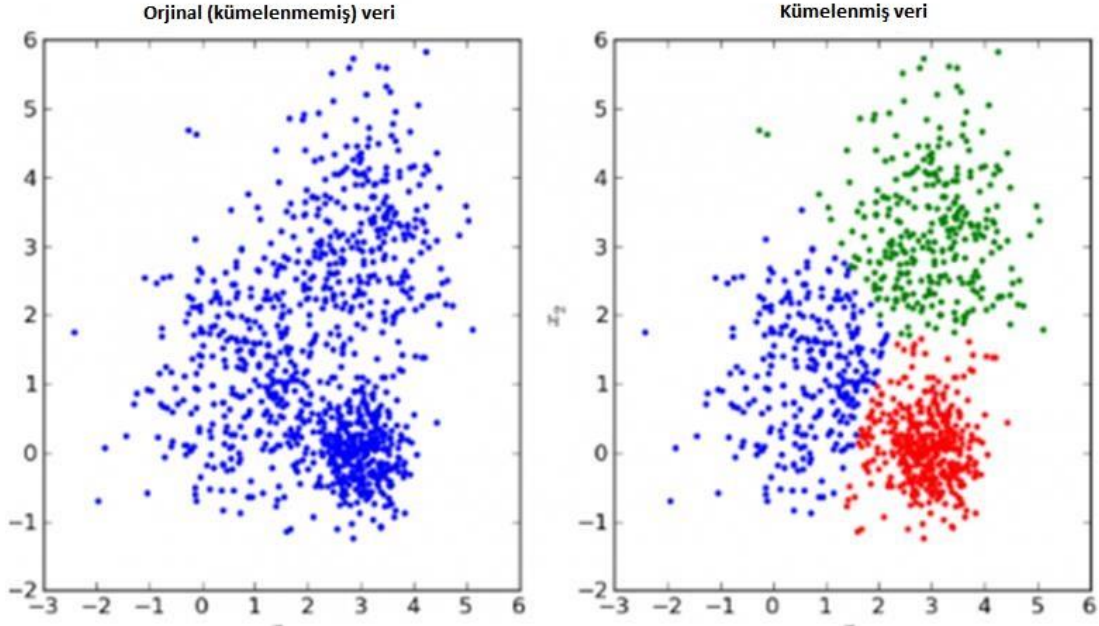
Hiyerarşik olmayan teknikler, hiyerarşik tekniklere göre daha hızlı olduğundan daha büyük veri kümelerine uygulanabilir (Johnson ve Wichern 1988).



Şekil 3.5. Kümeleme örneği

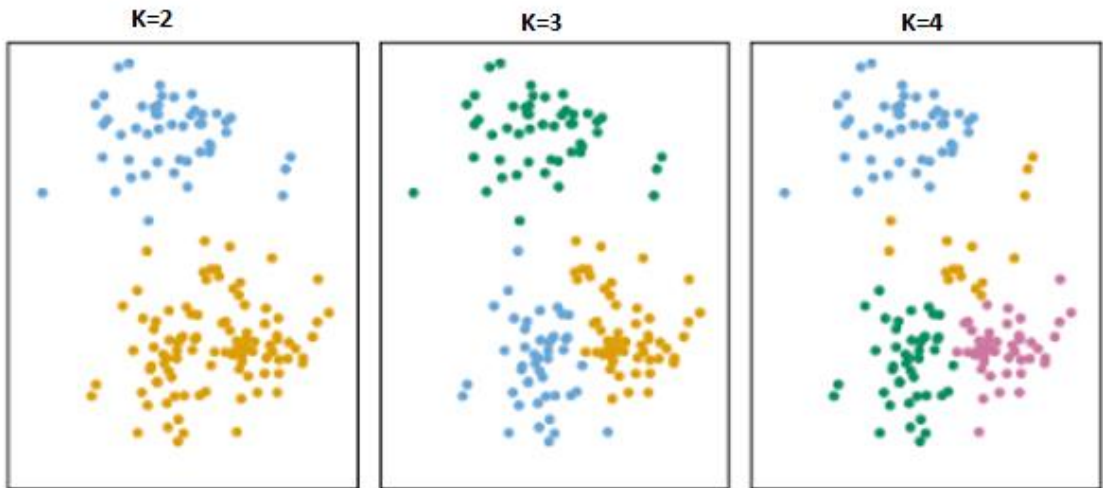
3.3.2.1. K-means kümeleme algoritması

J.B. MacQueen tarafından 1967'de geliştirilen k-means en eski kümeleme algoritmalarından biridir (MacQueen 1967). K-ortalamalar olarak adlandırılmasının nedeni, nesnelerin k adet kümeye ayrılması ve küme merkezinin kümede yer alan nesnelerin ortalama değerine sahip olmasıyla ilgilidir. Yaygın olarak kullanılmakta olan denetimsiz (eğitici-siz) öğrenme algoritmalarından birisi olan k-means küme ataması yaparken, her nesnenin yalnızca bir kümeye dâhil olmasına izin verir. Bu nedenle, keskin bir kümeleme algoritmasıdır. Küme merkezinin kümeyi temsil etmesi ana fikrine dayalı bir yöntemdir (Han ve Kamber 2006). Şekil 3.6'da verilerin kümeleme öncesi ve sonrası durumları gösterilmektedir.



Şekil 3.6. Orjinal veri ve kümelenmiş veri

Algoritmada kullanılan k parametresi verilerin kaç adet kümeye ayrılacağını göstermektedir. K parametresinin seçiminde kullanılan farklı analiz yöntemleri olsa da en iyi yöntem farklı k değerleri için kümeleme sonuçlarına bakarak karar vermektir. Çünkü farklı sayıda oluşacak kümeler, verideki farklı özellikleri ortaya çıkarabilecektir. Şekil 3.7'de k -means algoritmasının farklı k değerleri için elde ettiği kümeleme sonuçları görülmektedir.



Şekil 3.7. Farklı k değerleri için k -means kümeleme

K-means algoritması temelde n adet nesneyi k tane kümeye bölme prensibine dayanmaktadır. Burada kümelerin nesne bakımından mümkün olduğu kadar yoğun ve diğer kümelerden maksimum uzaklıkta olması amaçlanmaktadır. Yani, küme içindeki değerler birbirine en çok benzeyen nesnelere seçilmeli ve kümeler mümkün olduğunca birbirine benzememelidir. Küme benzerliği küme içindeki nesnelerin ortalama değeriyle ölçülür ve bu değer kümenin ağırlık merkezini ifade etmektedir (Rui ve Wunsch 2005).

K-means kümeleme yönteminin değerlendirilmesinde çoğunlukla karesel hata kriteri (SSE) kullanılır. SSE'nin en düşük değeri, en iyi sonucu ifade etmektedir. Nesnelerin küme merkezlerine olan uzaklıklarının karelerinin toplamı Denklem 3.17 ile hesaplanmaktadır (Tan ve diğ. 2006).

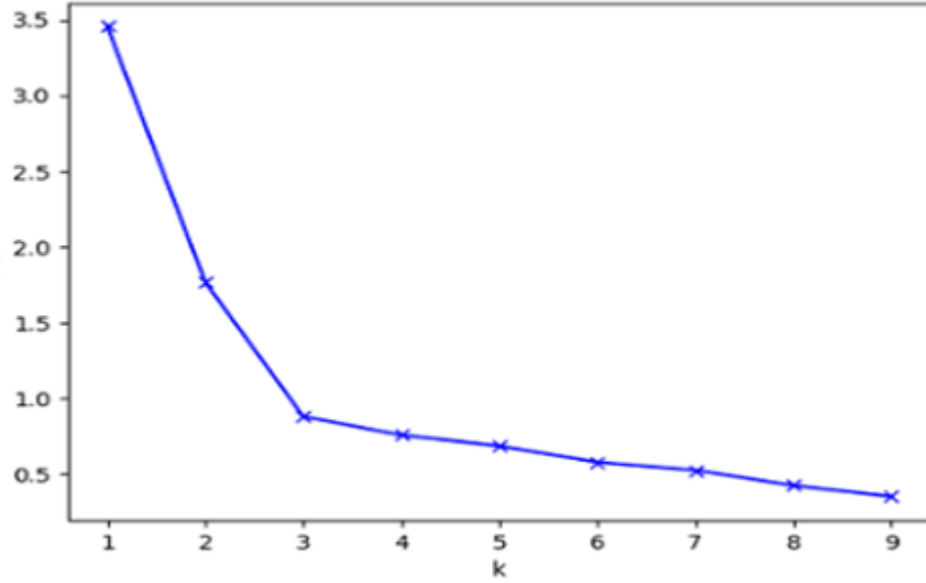
$$SSE = \sum_{i=1}^K \sum_{x \in C_i} dist^2(m_i, x) \quad (3.17)$$

K-means algoritmasının işlem adımları aşağıdaki gibidir:

1. Adım: Küme merkezleri belirlenir.
2. Adım: Merkez dışında kalan veriler mesafelerine göre sınıflandırılır.
3. Adım: Her bir veri kendisine en yakın kümeyle atanır.
4. Adım: Verilerin tamamı en yakın kümeyle atandığında k adet küme için yeniden merkezler hesaplanır.
5. Adım: Küme merkezleri kararlı bir hale gelene kadar 2. ve 3. Adımlar tekrar edilir.

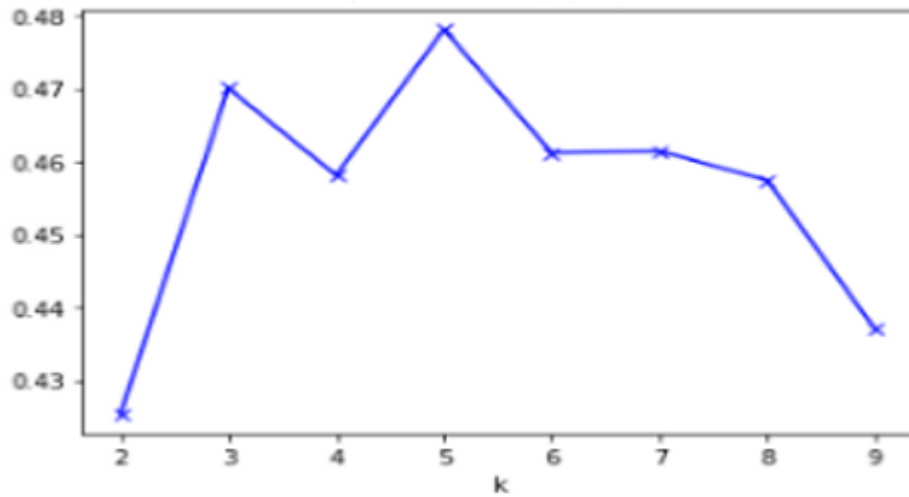
Algoritmada k sayısının belirlenmesi önemli bir problemdir. Uygun k değerini seçmek için çeşitli metotlar bulunmaktadır. Bu metotlardan Elbow ve Silhouette metodu kısaca açıklanacaktır.

Elbow metotunda; verilerin farklı k değerlerine göre küme merkezine olan uzaklıklarının karesel toplamı hesaplanır. Bu değerler grafik haline getirilir ve grafik üzerinde toplamlar arasındaki farkın azalmaya başladığı dirsek noktası en uygun k değeri olarak belirlenmektedir. Elbow metodu aynı zamanda dirsek metodu olarak da bilinmektedir. Şekil 3.8'de Elbow metotuna göre en uygun k değeri gösterilmektedir.



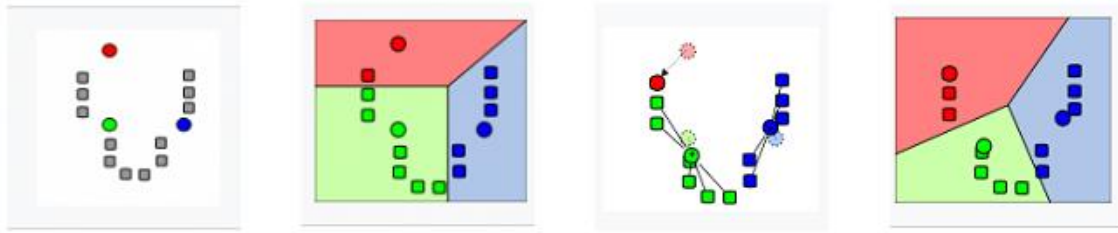
Şekil 3.8. Elbow metodu ve en uygun k değeri (k=3)

Silhouette metotunda; her bir k sayısı için küme farklılıkları incelenir ve oluşan kümeler -1 ile 1 aralığında değerler verir. 1'e en yakın olan değer en uygun k değeri olarak belirlenmektedir. Şekil 3.9'da Silhouette metotuna göre en uygun k değeri gösterilmektedir.



Şekil 3.9. Silhouette metodu ve en uygun k değeri (k=5)

Şekil 3.10'da k=3 değeri için yapılan örnek kümeleme uygulamasının adımları gösterilmiştir.



Şekil 3.10. K-means kümeleme adımları

3.3.2.2. K-medoids kümeleme algoritması

K-medoids algoritması, verinin çeşitli yapısal özelliklerini temsil eden k adet nesneyi bulma esasına dayanır (Dodge 2012). Temsilci nesne küme merkezine en yakın noktadır ve medoid olarak adlandırılır. Nesne grubunu k adet kümeye ayırırken amaç, benzer nesnelere bir araya toplamak ve farklı kümelerdeki nesnelere birbirine benzemediği bir yapı oluşturmaktır.

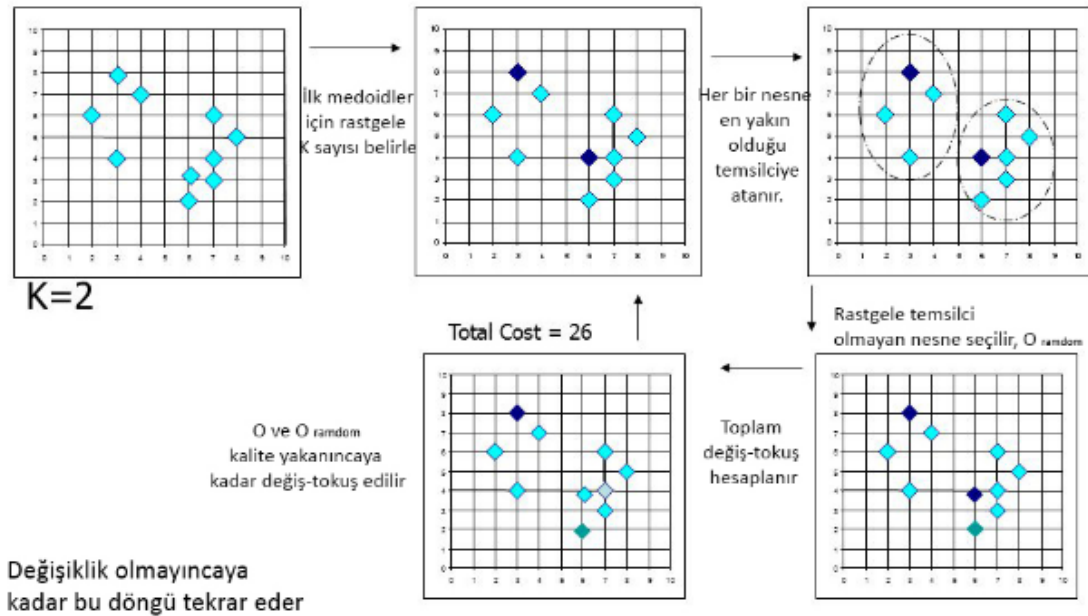
K-medoids algoritmasının birçok farklı türevi bulunmakla birlikte ilk ortaya atılan k -medoids algoritması PAM (Partitioning Around Medoids) algoritmasıdır. PAM algoritmasında, k -means algoritmasına benzer şekilde rastgele seçtiği k adet sayıyı küme merkezi olarak belirlenir ve her nesne kendisine en yakın olan temsilciye atanarak k adet küme oluşturulur. Kümeye katılan her yeni elemanla birlikte kümenin elemanları test edilerek küme gelişimine en fazla katkı sağlayabilecek nokta tespit edilir. Bu nokta tespit edilirken kriter olarak, nesne ile nesnenin ait olduğu kümenin temsilci nesnesi arasındaki ortalama benzersizlik maliyet fonksiyonu kullanılır (Işık ve Çamurcu 2007). Tespit edilen bu nokta kümenin yeni merkezi, kümenin eski merkezi de sıradan küme elemanı olacak şekilde bir yer değiştirme (swap) işlemi yapılır (Dinçer 2006).

K-medoids algoritmasının işlem adımları aşağıdaki gibidir:

1. Adım: K küme sayısı belirlenir.
2. Adım: K nesnelere başlangıç medoidleri olarak seçilir.
3. Adım: En yakın medoid x 'e sahip kümeye nesnelere atanır.
4. Adım: Amaç fonksiyonu hesaplaması yapılır.
5. Adım: Medoid olmayan y noktası rastgele olarak seçilir.
6. Adım: x ve y 'nin yer değiştirmesi amaç fonksiyonunu minimize ediyorsa, x ve y noktalarının yeri değiştirilir.

7. Adım: Küme merkezleri kararlı bir hale gelene kadar 3. Ve 6. Adım arasındaki işlemler tekrar edilir (Akın 2008).

K-medoids algoritmasında, başlangıçtaki merkez seçiminin ve verilerin işleme sırasının kümeleme üzerine etkisi yoktur. Ayrıca, merkezi elemanlar kümeyi temsil ettiği için gürültülü veriye karşı duyarlı değildir. Şekil 3.11’de k-medoids kümeleme adımları gösterilmiştir.



Şekil 3.11. K-medoids kümeleme adımları

3.3.3. Bulanık kümeleme algoritmaları

Bulanık kümeleme, kümeler birbirinden belirgin olarak ayrılamadığında ya da bazı nesnelere küme üyeliğinde kararsız bir yapı gösterdiğinde kullanılmaktadır. Bulanık kümelemede, diğer kümeleme yöntemlerine benzer şekilde uzaklık ölçümleri kullanılarak kümeleme yapılır. Kullanılan algoritma ve küme yapısına göre uygun bir uzaklık ölçütü belirlenir ve nesnelere küme üyelikleri hesaplanır. Bu üyelik değerleri yorumlama açısından kullanışlı olduğu için kümelemede avantaj yaratır. Bulanık kümeleme uzaklık kullanımı noktasında esneklik ve bilinen üyelik değerleri için sayısal optimizasyonla birleştirilebilir (Naes ve Mevik 1999).

Nesne ve küme sayısının fazla olması durumunda, çok fazla çıktı elde edilmesi nedeniyle bilginin tasnifi ve özetlenmesi dezavantaj olacaktır. Ayrıca, bulanık kümeleme algoritmaları karmaşık yapıdadır ve daha çok belirsizlik söz konusu olduğunda tercih edilmektedir (Şahinli 1999).

3.3.3.1. Bulanık c-means kümeleme algoritması

Bulanık c-means (BCM) algoritması, bulanık bölümlenmeli kümeleme teknikleri arasında en iyi bilinen ve yaygın olarak kullanılmakta olan bir yöntemdir. 1973 yılında Dunn tarafından ortaya atılmış olup, 1981'de Bezdek tarafından geliştirilmiştir (Höppner ve diğ. 1999).

K-means gibi yaygın bilinen kümeleme algoritmalarında her bir örnek sadece bir kümeye ait olabilmekte ve bu kümeleme yöntemleri katı kümeleme (hard clustering) olarak adlandırılmaktadır. Bulanık c-means algoritmasında ise, bir örneğin birden fazla küme için aitlik dereceleri vardır ve ılımlı kümeleme (soft clustering) olarak isimlendirilir. Örneğin, bir nesne %25 oranında A sınıfına, %75 oranında B sınıfına ait olabilmektedir. Bu algoritmalar, küme sınırlarındaki nesnelere tamamen belli bir kümeye dahil etmedikleri için katı kümelemeye göre daha güçlü ve doğal kümeleme yapabilmektedir (Cebeci ve Yıldız 2015).

Bulanık c-means algoritması amaç fonksiyonu mantığıyla çalışır. Algoritmadaki c değeri küme sayısını ifade etmektedir. Nesnelere iki veya daha fazla kümeye ait olabilir ve her bir küme için [0,1] aralığında birer üyelik değerine sahiptir. Bulanık mantık gereğince bu üyelik değerlerine bağlı olarak o kümeye aittirler. Bir nesnenin, tüm kümeler için hesaplanan üyelik değerleri toplamı 1 olmalıdır. Üyelik değeri en yüksek olan küme, nesnenin en yakın olduğu küme merkezini ifade etmektedir. Kümeleme işlemi amaç fonksiyonunun belirlenen değere yakınsamasıyla tamamlanmaktadır. Şekil 3.12'de (Cebeci ve Yıldız 2015) bulanık c-means algoritmasının farklı k değerlerindeki kümeleme sonuçları gösterilmiştir.

Bulanık c-means algoritmasının işlem adımları aşağıdaki gibidir:

1. Adım: C küme sayısı ve diğer başlangıç değerleri atanır.
2. Adım: U üyelik matrisi rastgele oluşturulur.
3. Adım: Küme merkezleri hesaplanır.
4. Adım: Bir sonraki üyelik değerleri hesaplanır.
5. Adım: Sistem kararlı bir hale gelene kadar 4. Adım tekrar edilir.

Algoritma, en küçük kareler yönteminin genellemesi olan Denklem 3.18'deki amaç fonksiyonunu minimize etmede kullanılmaktadır (Höppner ve diğ. 1999).

$$J_m = \sum_{i=1}^N \sum_{j=1}^C u_{ij}^m \|x_i - c_j\|^2, \quad 1 \leq m < \infty \quad (3.18)$$

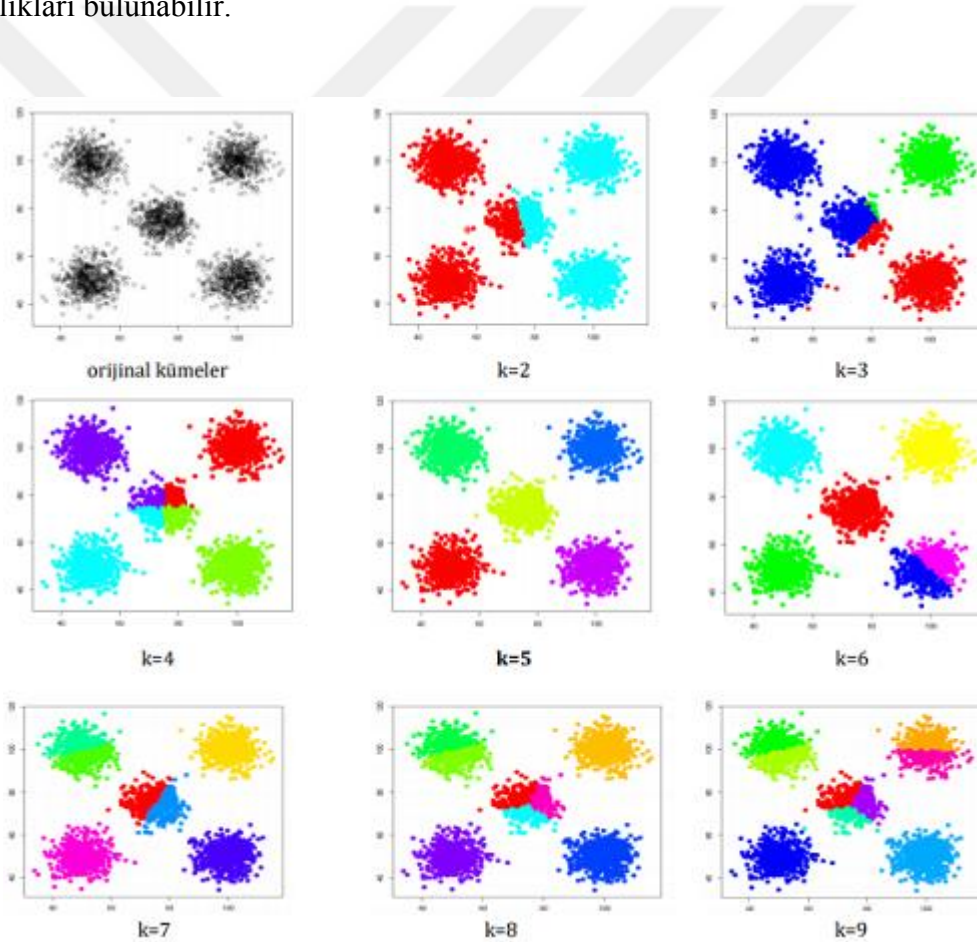
Rastgele oluşturulan U üyelik matrisi ile algoritma başlatılır. İkinci adımda merkez vektörleri Denklem 3.19 kullanılarak hesaplanır (Höppner ve diğ. 1999).

$$c_j = \frac{\sum_{i=1}^N u_{ij}^m x_i}{\sum_{i=1}^N u_{ij}^m} \quad (3.19)$$

U matrisi, hesaplanan küme merkezlerine göre Denklem 3.20 yardımıyla yeniden oluşturulur. U matrisinin önceki hali ile son durumdaki hali karşılaştırılır, aradaki fark ε 'den küçük olana kadar işlemler tekrar edilir (Moertini 2002).

$$u_{ij} = \frac{1}{\sum_{k=1}^C \left(\frac{\|x_i - c_i\|}{\|x_i - c_k\|} \right)^{2/(m-1)}} \quad (3.20)$$

Kümeleme tamamlandığında, U üyelik matrisindeki bulanık değerler kümeleme sonucunu göstermektedir. Bu değerler berraklaştırıldığı takdirde 0 ve 1 olarak karşılıkları bulunabilir.



Şekil 3.12. Farklı k değerleri için bulanık c-means kümeleme

3.4. BOA ve Levy Flight Arama Stratejisinin Kümelemede Kullanılması

BOA kümeleme algoritması arama ve sömürü yeteneklerine sahip olduğundan global bir iyileştirici olarak düşünülebilir. BOA algoritması bir dizi rastgele çözümler başlar ve her bir iterasyonda arama ajanlarının konumlarını yukardaki denklemlere göre günceller. Arama vektörü olan \vec{A} 'nın uyarlamalı değişimi, algoritmanın arama ve sömürü arasında kolay geçiş yapmasını sağlar. Bu durum bazı iterasyonların aramaya, geri kalanların sömürüye odaklanması anlamına gelir. Algoritmanın arama kabiliyeti, kullanılan pozisyon (Denklem 2.12) güncelleme mekanizmasından kaynaklanmaktadır. BOA'nın amacı, belirli bir amaç fonksiyonunun en iyi şekilde değerlendirilmesini sağlayan arama ajanını bulmaktır. Çalışmada, iterasyonlar boyunca hesaplanan BOA amaç fonksiyon değeri, Levy Flight amaç fonksiyon değeriyle kıyaslanmıştır. Levy Flight stratejisi uygulandıktan sonra bulunan konumun SSE değeri daha iyiye BOA konum değeri güncellenmiştir. Böylece, global aramanın verimliliği artırılarak daha iyi bir kümeleme sonucu elde edilmeye çalışılmıştır. Önerilen BOA-LF yönteminin sözde kod yapısı Çizelge 3.2'de, algoritmanın adımlarını gösteren akış diyagramı Şekil 3.13'te gösterilmiştir.

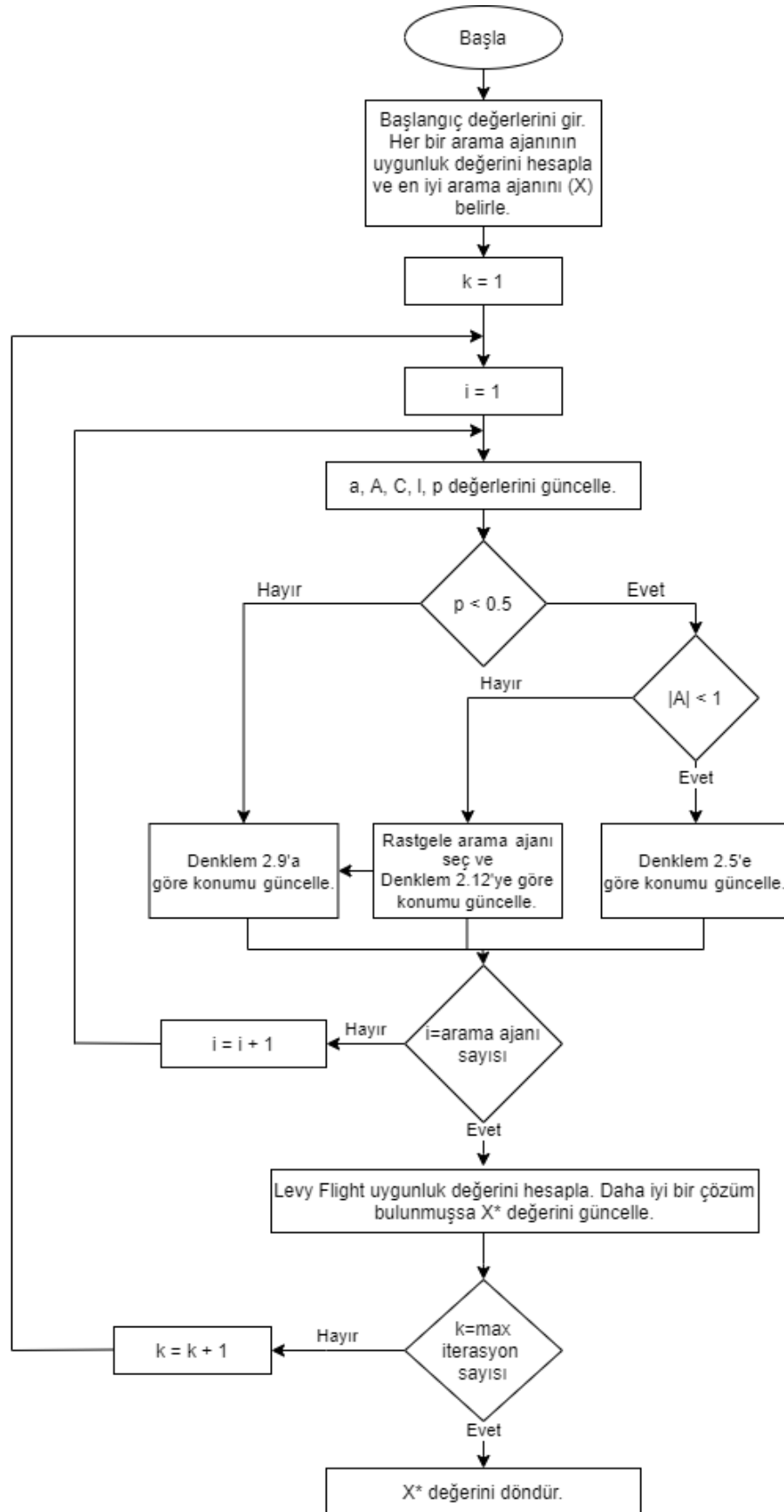
Çizelge 3.2. BOA-LF sözde kodu

```

Başlangıç popülasyonu ayarla  $X_i$  ( $i = 1, 2, \dots, n$ )
Her bir arama ajanının uygunluk değerini hesapla
 $X^*$  = Bilinen en iyi arama ajanı
while ( $t <$  maksimum iterasyon sayısı)

for (her bir arama ajanı için)
  Güncelle  $a$ ,  $A$ ,  $C$ ,  $l$ , ve  $p$ 
  if ( $p < 0.5$ )
    if ( $|A| < 1$ )
      Denklem 2.5 ile arama ajanı konumu güncelle
    else if ( $|A| \geq 1$ )
      Rastgele bir arama ajanı seç ( $\vec{X}_{rand}$ )
      Denklem 2.12 ile arama ajanını güncelle
    end if
  else if ( $p \geq 0.5$ )
    Denklem 2.9 ile arama ajanı konumu güncelle
  end if
end for
Levy flight amaç fonksiyon değerini hesapla
Daha iyi çözüm bulunmuşsa en iyi ajanı güncelle
Kısıt dışına çıkan bireylere sınır değerini ver
 $t = t + 1$ 
end while
Sonuç  $X^*$ 

```



Şekil 3.13. BOA-LF akış diyagramı

4. DENEYSEL SONUÇLAR

Son yıllarda literatüre kazandırılmış olan Balina Optimizasyon Algoritması ve Levy Flight arama stratejisi kullanılarak 16 farklı veri setinde (Dermatology, E. Coli, Balance, Cancer-Int, Cancer, Iris, Glass, Wine, Credit, Thyroid, Heart, Spect, Diabetes, Hepatit, Breast Tissue, Parkinson) kümeleme uygulaması yapılmıştır. Veri setleri, University of California, Irvine (UCI) açık erişimli veri tabanından alınmıştır. Kümeleme sonuçları, k-means, k-medoids ve bulanık c-means kümeleme algoritmalarının kümeleme sonuçlarıyla karşılaştırmalı olarak verilmiştir.

4.1. Veri Setleri

Dermatology veri seti, ciltte oluşan kızarıklık, kaşıntı, sertlik, kepeklenme, yumuşama, çatlama; kafa derisi, diz ve dirseklerde hastalığın görülme oranı gibi medikal özellikler kullanılarak oluşturulmuştur. Veri seti 34 özellik, 6 sınıf ve 366 örnekten oluşmaktadır. Sınıflar; Psoriasis 112, Seboreik Dermatit 61, Liken Planus 72, Pityriasis Rosea 49, Kronik Dermatit 52 ve Pityriasis Rubra Pilaris 20 adet örnekle temsil edilmektedir. Yaş niteliği eksik olan 8 adet veri örneğinde eksik veriler örneğin ait olduğu sınıfın ortalama yaş değeriyle güncellenmiştir.

E. Coli veri seti, 7 özellik, 5 sınıf ve 327 örnekten oluşmaktadır. Ayrıca, veri setinde yer alan referanslar veri setinin oluşturulmasında ve geliştirilmesinde öncülük yapmaktadır.

Balance veri seti, psikolojik deney sonuçları modellenerek oluşturulmuştur. Veri setinde 4 özellik, 3 sınıf ve 625 örnek bulunmaktadır. Özellikler sol ağırlık, sol mesafe, sağ ağırlık ve sağ mesafe şeklindedir. Sınıflar ise, denge skalasının durumuna göre sağda (R), solda (L) veya dengelenmiş (B) olarak belirtilmiştir.

Cancer-Int veri seti, hastalık bölgesindeki tümörlü alanın (tümörlü bölgenin kalınlığı, tümörlü hücrelerin şekil ve büyüklük olarak birbirine benzeme oranı vb.) özelliklerini kullanmaktadır. Bu özelliklerle tümörün iyi veya kötü huylu olması sonucu elde edilir. Veri seti, 30 özellik, 2 sınıf ve toplam 569 örnekten oluşmaktadır. Sınıf dağılımı, iyi huylu (benign) 357 ve kötü huylu (malignant) 212 örneğe sahiptir.

Cancer veri seti, 9 özellik, 2 sınıf ve toplam 699 örnekten oluşmaktadır. Veri setindeki özellikler kullanılarak tümör iyi veya kötü huylu olarak tespit edilmektedir.

Iris veri seti, örüntü tanıma literatüründe en iyi bilinen veri setlerinden biridir. Veri seti 4 özellik ve her biri 50 örnekle temsil edilen 3 sınıftan oluşmaktadır. Iris bitkisinin çanak ve taç yapraklarının uzunluk ve genişlik bilgileri özellik; Iris Setosa, Iris Versicolour ve Iris Virginica türleri de sınıf değeri olarak kullanılmaktadır.

Glass veri seti, suç mahallinde bulunan cam türlerinin kanıt olarak doğru sınıflandırılabilmesi için kriminolojik araştırmalarla bağlantılı olarak oluşturulmuştur. Veri setinde 9 özellik, 6 sınıf ve 214 adet örnek bulunmaktadır.

Wine veri seti, İtalya'da aynı bölgede yetiştirilen ancak üç farklı çeşitten türetilen şarapların kimyasal analiz sonuçlarıdır. Veri seti, 13 (bileşen miktarları) özellik, 3 (şarap türleri) sınıf ve toplam 178 örnek içermektedir.

Credit veri seti, kredi kartı uygulamalarıyla ilgili verilerden oluşmaktadır. Veri setinde 14 özellik, 2 sınıf (+ ve -) ve 690 adet örnek bulunmaktadır. Sınıf dağılımı, 307 pozitif, 383 negatif örnekten oluşmaktadır.

Thyroid veri seti, RT3U testinden elde edilen değerler ile öz sıvıdaki bazı değerlerin özellik olarak kullanılması ile oluşturulmuştur. Kişinin normal, hiper (yüksek) veya hipo (düşük) tiroid hastası olup olmadığını tahmin etmede kullanılmaktadır. Veri seti 5 özellik, 3 sınıf ve toplam 215 örnekten oluşmaktadır. Sınıf dağılımı, normal 150, hiper 35 ve hipo 30 örnek şeklindedir.

Heart veri setinde, 75 özellik bulunmakla birlikte deneylerde genellikle 13 özellik kullanılmaktadır. Hedef (goal) alanı, hastada kalp hastalığı olup olmadığını göstermektedir. Veri setinde toplam 270 örnek bulunmaktadır.

Spect veri seti, Tek Foton Emisyonlu Bilgisayarlı Tomografi (SPECT) görüntülerinin teşhis edilmesini ifade etmektedir. Örnek veriler, normal ve anormal olarak iki sınıfla temsil edilmektedir. Spect görüntü seti 22 özelliikle tanımlanan 267 örneğe sahiptir.

Diabetes veri seti, kayıt başına dört alandan oluşmaktadır. Her alan bir sekmeye ve her kayıt bir satır sonu ile ayrılmıştır. Veri seti, diyabet hastalarında kaydedilmiş 70 veri setinin dağılımını içermektedir. 2 sınıf, 8 özellik ve toplam 768 veriden oluşmaktadır.

Hepatit veri setinde, die ve live olmak üzere iki sınıf bulunmaktadır. Veri seti toplam 155 örnek ve 21 özellik içermektedir.

Breast Tissue veri seti, 9 özellik ve 106 örnekten oluşmaktadır. Veri setinde car (karsinom), fad (fibro-adenom), mas (mastopati), gla (glandüler), con (connector) ve adi (adipoz) olmak üzere toplam 6 sınıf bulunmaktadır.

Parkinson veri seti, 23'ü Parkinson hastası olan 31 kişiden alınan bir dizi biyomedikal ses ölçümünden oluşmaktadır. Tablodaki her sütun belirli bir ses ölçüsüdür. Her satır, bu kişilerden alınan 195 ses kaydından birine karşılık gelir. Durum sütununda 1 Parkinson hastalığını, 0 ise sağlıklı kişiyi göstermektedir. Çizelge 4.1, veri setleri hakkında özet bilgi içermektedir.

Çizelge 4.1. Veri setlerinin özellikleri

Veri seti	Sınıf Sayısı	Özellik Sayısı	Veri Sayısı
Dermatology	6	34	366
E. Coli	5	7	327
Balance	3	4	625
Cancer-Int	2	9	699
Cancer	2	30	569
Iris	3	4	150
Glass	6	9	214
Wine	3	13	178
Credit	2	14	690
Thyroid	3	5	215
Heart	2	13	270
Spect	2	22	267
Diabetes	2	8	768
Hepatit	2	21	155
Breast Tissue	6	9	106
Parkinson	2	22	195

4.2. Kümeleme Sonuçlarının Değerlendirilmesi

Kümeleme analiz sonuçlarının değerlendirilmesi için farklı yöntemler bulunmaktadır. Hangi yöntemin kullanılacağı veri türü ve kümeleme yöntemine göre belirlenmektedir. Yaygın olarak Rand Index Ölçüm (RI), Toplam Karesel Uzaklık (TKU) ve TKU yakınsama değeri kullanılmaktadır. Bu çalışmada, BOA ve Levy Flight kullanılarak geliştirilen kümeleme algoritmasının sonuçları toplam karesel uzaklık yöntemi kullanılarak değerlendirilmiştir.

4.2.1. Toplam karesel uzaklık yöntemi

Küme merkezi tabanlı kümeleme algoritmalarında kümeye ait nesnelere toplam uzaklıklarının minimize edilmesi gereklidir. Bu nedenle, küme merkezinin kümedeki nesnelere olan karesel uzaklık değeri minimizasyon problemi olarak değerlendirilir. Toplam karesel uzaklık değeri ne kadar küçükse küme merkezinin o kadar iyi belirlendiği söylenebilir (Wan ve diğ. 2012).

4.2.2. Rand index ölçüm yöntemi

N tane nesneden oluşan ve k adet kümeye ayrılacak olan bir veri setinde, gerçek küme bilgileri bilinen nesnelere E (Expected) ve kümeleme yapıldıktan sonra nesnelere O (Obtained) ile temsil edilsin. Veri setindeki her nesne, kendisi dışında geriye kalan tüm nesnelere kıyaslanır. $i, j = 1, 2, \dots, N$ olmak üzere veri setindeki her x_i ve x_j nesne çiftinin, E ve O durumlarında birbirine göre 4 farklı hali vardır.

1. x_i ve x_j nesne çiftinin O durumunda aynı kümeye dâhil olması ve E durumunda da bu nesne çiftinin aynı kümede olması.

2. x_i ve x_j nesne çiftinin O durumunda aynı kümeye dâhil olması ve E durumunda da bu nesne çiftinin farklı kümelere olması.

3. x_i ve x_j nesne çiftinin O durumunda farklı kümelere dâhil olması ve E durumunda da bu nesne çiftinin aynı kümede olması.

4. x_i ve x_j nesne çiftinin O durumunda farklı kümelere dâhil olması ve E durumunda da bu nesne çiftinin farklı kümelere olması.

Veri setindeki x_i ve x_j nesne çiftlerinin yukarıdaki 4 durum için elde edilen eşleşme değerleri sırasıyla a, b, c, d ile ve elde edilen toplamları da M ile gösterilsin. Bu durumda $M = a + b + c + d$ ile elde edilir.

Rand Index (RI), [0, 1] aralığında bir değer alır ve Denklem 3.20 ile hesaplanır. Bu değer birle yaklaştıkça E ve O durumlarının birbirine benzerliği artmaktadır. Bu iki durumun benzerliğinin fazla olması ise, kümelemenin başarılı bir şekilde yapıldığını göstermektedir (Rand 1971, Servi 2009).

$$RI = \frac{a + d}{M} \quad (3.20)$$

4.3. Kümeleme Sonuçlarının Karşılaştırılması

Çalışmada kullanılan kümeleme algoritmalarının performanslarını karşılaştırmak için farklı sayıda özellik ve küme sayısına sahip veri setleri seçilmiştir. Algoritmaların uygulaması, bir Intel® Core™ i5-2400 CPU @ 3.1 GHz işlemci, 4 GB RAM ve Windows 7 (64-bit) Professional işletim sistemi üzerinde yapılmıştır.

Tüm algoritmalarda ortak parametre olarak iterasyon sayısı 1000 ve popülasyon büyüklüğü 100 olarak belirlenmiştir. K-means, k-medoids ve bulanık c-means

algoritmaları parametre olarak sadece maksimum iterasyon sayısına sahiptir. Bununla birlikte, BOA algoritması bir dizi parametre içerir. α parametresi 2 değeriyle başlayıp 0'a doğru doğrusal bir şekilde azalmaktadır. r_1 ve r_2 parametreleri ise, [0, 1] aralığında rastgele üretilmektedir.

Uygulamada Denklem 3.17'de verilen SSE değeri, tüm algoritmalar için amaç fonksiyonu olarak kullanılmaktadır. Algoritmaların amacı, SSE değerini minimize eden en iyi küme merkezlerini bulmaktır. Algoritmalar 30 kez çalıştırılmış B(est) en iyi, W(orst) en kötü, A(verage) ortalama ve S(tdDev) standart sapma sonucu olarak Çizelge 4.2'de sunulmuştur. Her veri kümesi için, elde edilen en iyi ortalama sonuç kalın olarak işaretlenmiştir.

BOA-LF, ortalama değerler göz önüne alındığında 11 veri setinde (Cancer-int, Iris, Wine, Credit, Thyroid, Heart, Spect, Diabetes, Hepatit, Breast Tissue, Parkinson) en iyi sonucu elde etmiştir. Cancer veri setinde, k-means, BOA saf hali ve BOA-LF algoritması aynı değere ulaşarak birinci sırada yer almıştır.

Kalan 4 veri setinde (Dermatology, E.coli, Balance, Glass) k-means algoritması en iyi sonuçları üretmiştir. K-medoids ve bulanık c-means algoritmaları, diğer algoritmalarla kıyaslandığında hiçbir veri seti için en iyi sonuca ulaşamamıştır.

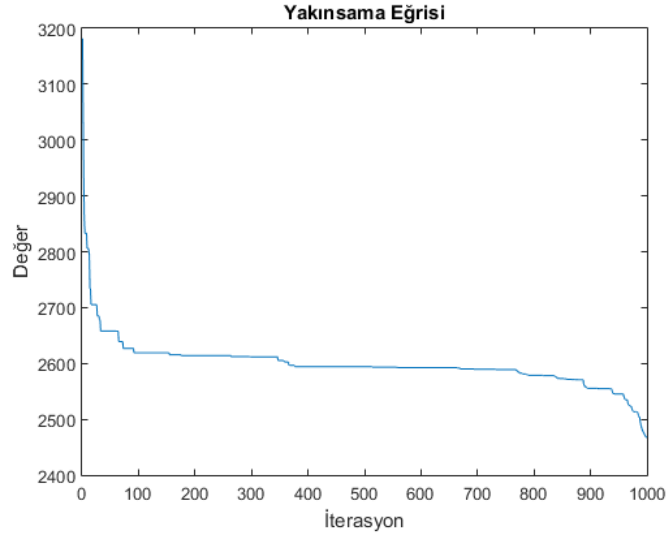
BOA-LF ve BOA saf halinin kümeleme sonuçları incelendiğinde yakın değerlere sahip oldukları tespit edilmiştir. Bu durum özellikle Iris, Spect ve Parkinson veri setlerinde belirgin bir haldedir ve kümeleme sonuçları arasında oldukça az bir fark bulunmaktadır. BOA-LF'nin başarılı olduğu diğer veri setlerinde (Cancer-int, Wine, Credit, Thyroid, Heart, Diabetes, Hepatit, Breast Tissue) ise, bu fark açılarak BOA saf hal kümeleme sonuçlarından daha iyi değerlere ulaşılmıştır. Her bir veri seti için elde edilen sonuçlar ve BOA-LF yönteminde elde edilen yakınsama eğrileri bu bölümde paylaşılmıştır.

Çizelge 4.2. SSE sonuçları

		K-means	K-medoids	B.C-means	BOA	BOA-LF
Dermatology	B	2034.0428	2830.0774	5196.3797	2476.1037	2409.4543
	W	2104.7138	3067.6218	5196.3797	2656.1172	2553.5648
	A	2064.199	2960.3799	5196.3797	2579.3799	2485.4799
	S	35.6053	87.3953	0	47.6974	39.4899
E. Coli	B	65.9877	130.7275	108.4402	71.6711	67.687
	W	69.6421	158.1678	108.4402	83.3333	75.3441
	A	68.0015	144.1519	108.4402	75.0888	72.6395
	S	1.7927	11.6843	0	2.2047	2.1524
Balance	B	1423.8514	1657.7469	1722.2446	1424.4026	1424.1544
	W	1425.8472	1709.7908	1722.2446	1434.4825	1431.4788
	A	1425.0116	1681.8454	1722.2446	1428.1721	1426.7978
	S	1.0599	23.5731	0	2.6198	1.6456
Cancer-Int	B	2986.9613	3455.4793	3286.1132	2964.7949	2964.433
	W	2988.4278	4460.8037	3286.1132	2967.4645	2964.7714
	A	2987.8412	3790.8969	3286.1132	2965.7767	2964.554
	S	0.80322	433.1914	0	0.71739	0.087534
Cancer	B	1.34e+154	1.79e+308	7.62e+156	1.34E+154	1.34e+154
	W	1.34e+154	1.79e+308	7.62e+156	1.34E+154	1.34e+154
	A	1.34e+154	1.79e+308	7.62e+156	1.34E+154	1.34e+154
	S	0	0	0	0	0
Iris	B	97.3259	191.6737	106.3591	97.0327	96.7007
	W	97.3462	239.8904	106.3591	98.989	97.673
	A	97.3381	215.3423	106.3591	97.7151	97.0511
	S	0.011116	17.0605	0	0.49302	0.25891
Glass	B	215.3564	304.1732	400.9819	283.5364	293.4393
	W	250.5733	349.1045	404.3579	406.8702	385.8969
	A	226.3667	330.9148	402.9559	332.7943	327.0969
	S	14.2828	18.2839	1.8028	33.4276	27.6554
Wine	B	16555.6794	17144.9288	17128.4579	16328.7025	16315.4289
	W	18436.9521	28200.8093	17128.4579	16426.5265	16423.3876
	A	18060.6975	22277.4703	17128.4579	16371.1249	16358.9337
	S	841.3307	4475.5945	0	24.1798	26.9812
Credit	B	777510.5758	584107.5622	759180.4699	556777.3207	556750.1635
	W	808744.4416	665014.7495	759180.4699	594034.5555	556924.7684
	A	802497.6684	620589.9931	759180.4699	558242.5354	556819.8377
	S	13968.2094	32888.9802	0	6763.0422	41.1228
Thyroid	B	2001.6358	2142.6923	2812.4999	1892.9851	1888.6516
	W	2017.0459	2402.473	2812.4999	2066.274	1988.6663
	A	2013.9639	2241.8421	2812.4999	1965.2538	1921.5825
	S	6.8916	101.7599	0	43.9114	21.0485

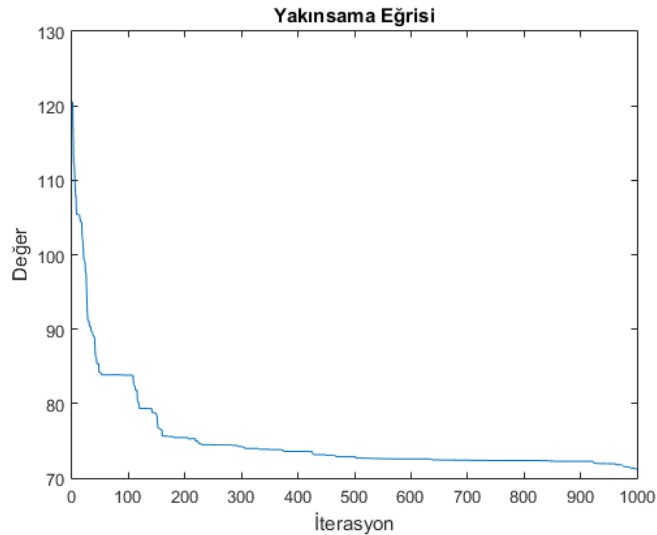
		K-means	K-medoids	B.C-means	BOA	BOA-LF
Heart	B	10700.8385	11574.288	13943.6643	10642.9101	10635.9571
	W	10700.8385	13907.6973	13943.6643	10731.264	10732.4421
	A	10700.8385	12619.013	13943.6643	10680.1755	10673.6712
	S	0	978.5486	0	19.4321	22.4637
Spect	B	557.5988	633.544	557.5988	554.5439	554.5435
	W	557.5988	633.544	557.5988	558.0749	554.5435
	A	557.5988	633.544	557.5988	554.7039	554.5435
	S	0	0	0	0.67138	0
Diabetes	B	74604.324	73172.0714	74604.324	72107.6455	72107.2278
	W	74604.324	73787.3376	74604.324	74172.3583	72133.3325
	A	74604.324	73389.0283	74604.324	72188.2534	72113.6223
	S	0	301.5164	0	375.0442	4.9244
Hepatit	B	9973.9731	10363.3165	12755.0747	9442.0413	9443.2764
	W	11383.7554	12306.6018	12755.0747	10176.3652	9484.8536
	A	10878.6548	10872.7204	12755.0747	9504.3466	9457.5178
	S	699.408	811.8676	0	181.2706	13.3116
Breast Tissue	B	130649.5537	173235.3868	145764.7385	126415.0569	125010.3574
	W	190427.1978	393722.6287	145764.7385	134541.9822	135803.8494
	A	143613.8846	274804.1355	145764.7385	129593.7168	127730.4009
	S	26179.5628	79907.5761	0	2473.768	2228.8195
Parkinson	B	17081.5962	17120.1081	17081.5962	16463.1067	16463.0297
	W	17081.5962	17120.1081	17081.5962	16464.9775	16465.088
	A	17081.5962	17120.1081	17081.5962	16463.9199	16463.6816
	S	0	0	0	0.54677	0.54867

Dermatology veri seti sonuçları incelendiğinde, k-means algoritması 2064.199 değeriyle en iyi kümeleme sonucunu elde etmiştir. Önerilen BOA-LF algoritması 2485.4799 değeriyle ikinci sırada yer almıştır. BOA saf hali ve k-medoids algoritmaları sırasıyla 2579.3799 ve 2960.3799 değerlerine ulaşırken, bulanık c-means algoritması 5196.3797 değeriyle son sırada yer almıştır. BOA-LF algoritmasının dermatology veri setindeki yakınsama eğrisi Şekil 4.1’de gösterilmiştir.



Şekil 4.1. Dermatology yakınsama eğrisi

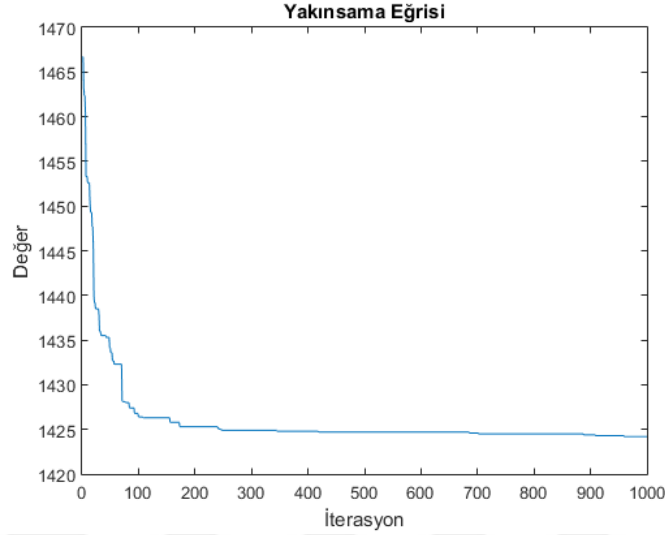
E.Coli veri seti sonuçları incelendiğinde, k-means algoritması 68.0015 değeriyle en iyi kümeleme sonucunu elde etmiştir. Önerilen BOA-LF algoritması 72.6395 değeriyle ikinci sırada yer almıştır. BOA saf hali ve bulanık c-means algoritmaları sırasıyla 75.0888 ve 108.4402 değerlerine ulaşırken, k-medoids algoritmasının 144.1519 değeriyle son sırada yer aldığı görülmektedir. BOA-LF algoritmasının e.coli veri setindeki yakınsama eğrisi Şekil 4.2’de gösterilmiştir.



Şekil 4.2. E.coli yakınsama eğrisi

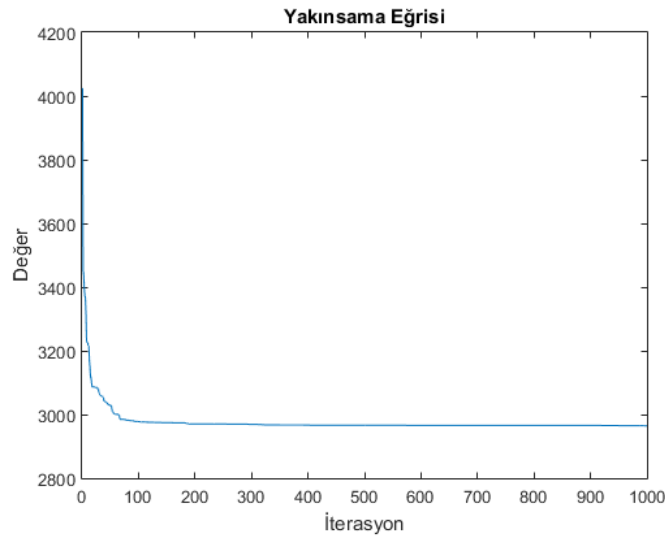
Balance veri seti sonuçları incelendiğinde, k-means algoritması 1425.0116 değeriyle en iyi kümeleme sonucunu elde etmiştir. Önerilen BOA-LF algoritması

1426.7978 değeriyle ikinci sırada yer almıştır. BOA saf hali ve k-medoids algoritmaları sırasıyla 1428.1721 ve 1681.8454 değerlerine ulaşırken, bulanık c-means algoritması 1722.2446 değeriyle son sırada yer almıştır. BOA-LF algoritmasının balance veri setindeki yakınsama eğrisi Şekil 4.3'te gösterilmiştir.



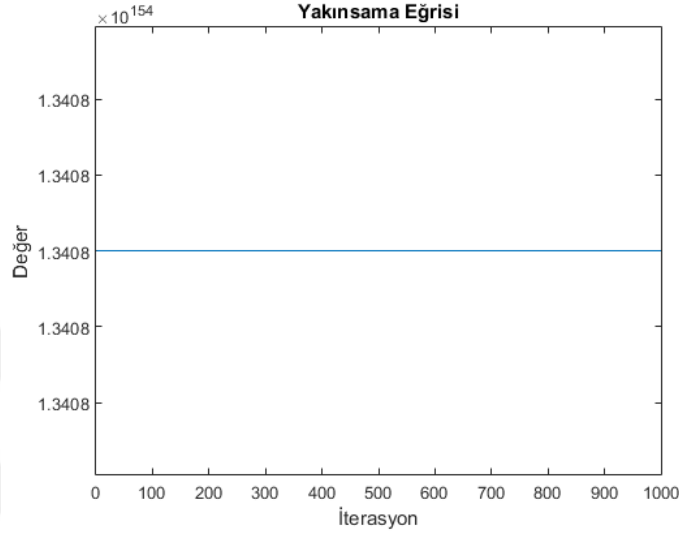
Şekil 4.3. Balance yakınsama eğrisi

Cancer-Int veri seti sonuçları incelendiğinde, önerilen BOA-LF algoritması 2964.554 değeriyle en iyi kümeleme sonucunu elde etmiştir. BOA saf hali 2965.7767 değeriyle ikinci sırada yer almıştır. K-means ve bulanık c-means algoritmaları sırasıyla 2987.8412 ve 3286.1132 değerlerine ulaşırken, k-medoids algoritması 3790.8969 değeriyle son sırada yer almıştır. BOA-LF algoritmasının cancer-int veri setindeki yakınsama eğrisi Şekil 4.4'te gösterilmiştir.



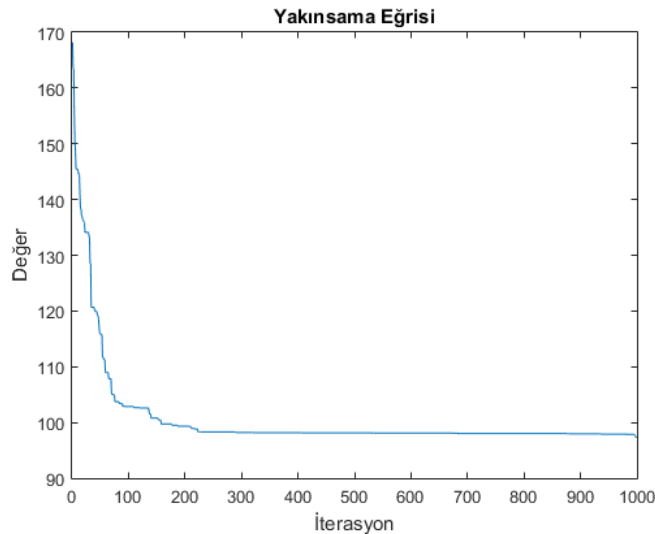
Şekil 4.4. Cancer-int yakınsama eğrisi

Cancer veri seti sonuçları incelendiğinde, önerilen BOA-LF algoritması, BOA saf hali ve k-means algoritması $1.34E+154$ en iyi kümeleme değerine ulaşarak birinci sırada yer almışlardır. İkinci sırada $1.79E+308$ değeriyle k-medoids algoritması ve son sırada $7.62E+156$ değeriyle bulanık c-means algoritması bulunmaktadır. BOA-LF algoritmasının cancer veri setindeki yakınsama eğrisi Şekil 4.5'te gösterilmiştir.



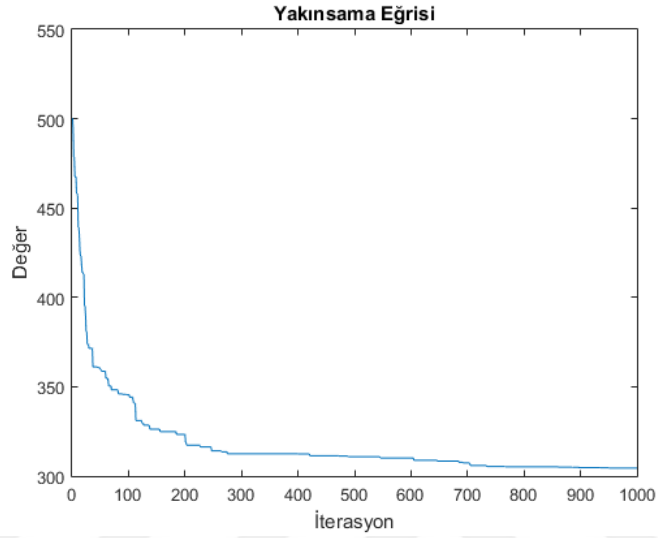
Şekil 4.5. Cancer yakınsama eğrisi

Iris veri seti sonuçları incelendiğinde, önerilen BOA-LF algoritması 97.0511 değeriyle en iyi kümeleme sonucunu elde etmiştir. İkinci sırada 97.3381 değeriyle k-means algoritması yer almaktadır. BOA saf hali ve bulanık c-means algoritmaları sırasıyla 97.7151 ve 106.3591 değerlerine ulaşırken, k-medoids algoritması 215.3423 değeriyle son sırada yer almıştır. BOA-LF algoritmasının iris veri setindeki yakınsama eğrisi Şekil 4.6'da gösterilmiştir.



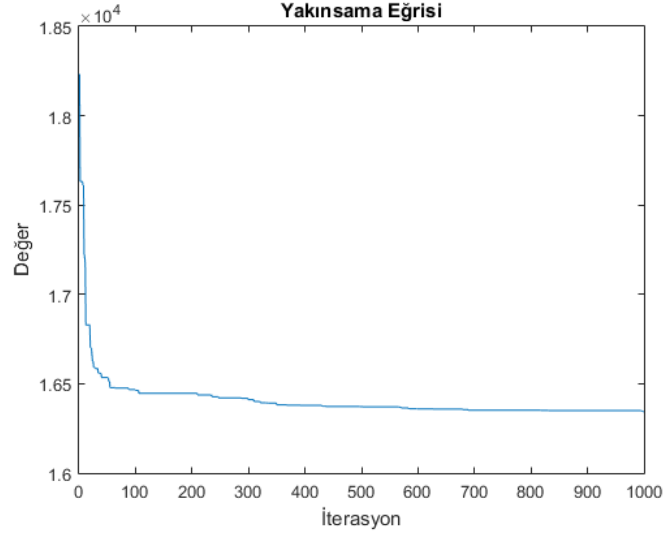
Şekil 4.6. Iris yakınsama eğrisi

Glass veri seti sonuçları incelendiğinde, k-means algoritması 226.3667 değeriyle en iyi kümeleme sonucunu elde etmiştir. Önerilen BOA-LF algoritması 327.0969 değeriyle ikinci sırada yer almıştır. K-medoids ve BOA saf hali sırasıyla 330.9148 ve 332.7943 değerlerine ulaşırken, bulanık c-means algoritmasının 402.9559 değeriyle son sırada yer aldığı görülmektedir. BOA-LF algoritmasının glass veri setindeki yakınsama eğrisi Şekil 4.7’de gösterilmiştir.



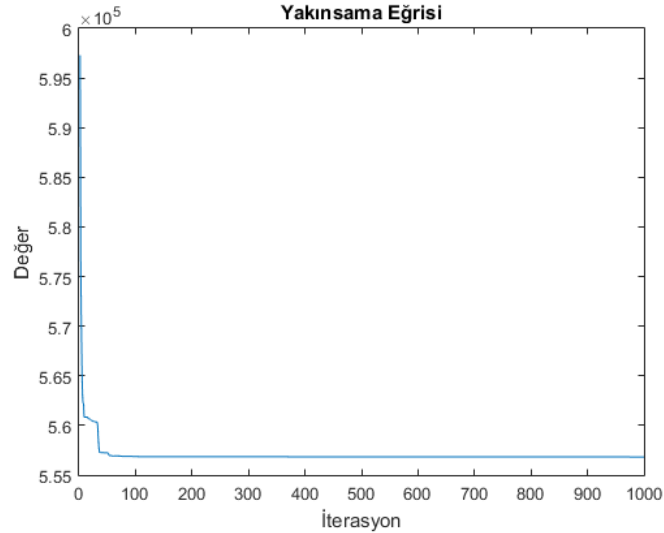
Şekil 4.7. Glass yakınsama eğrisi

Wine veri seti sonuçları incelendiğinde, önerilen BOA-LF algoritması 16358.9337 değeriyle en iyi kümeleme sonucunu elde etmiştir. BOA saf hali 16371.1249 değeriyle ikinci sırada yer almıştır. Bulanık c-means ve k-means algoritmaları sırasıyla 17128.4579 ve 18060.6975 değerlerine ulaşırken, k-medoids algoritması 22277.4703 değeriyle son sırada yer almıştır. BOA-LF algoritmasının wine veri setindeki yakınsama eğrisi Şekil 4.8’de gösterilmiştir.



Şekil 4.8. Wine yakınsama eğrisi

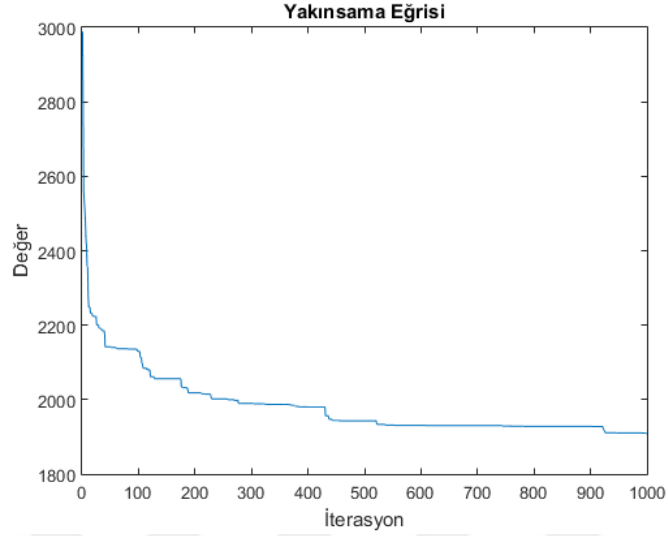
Credit veri seti sonuçları incelendiğinde, önerilen BOA-LF algoritması 556819.8377 değeriyle en iyi kümeleme sonucunu elde etmiştir. İkinci sırada 558242.5354 değeriyle BOA saf hali yer almaktadır. K-medoids ve bulanık c-means algoritmaları sırasıyla 620589.9931 ve 759180.4699 değerlerine ulaşırken, k-means algoritması 802497.6684 değeriyle son sırada yer almıştır. BOA-LF algoritmasının credit veri setindeki yakınsama eğrisi Şekil 4.9’da gösterilmiştir.



Şekil 4.9. Credit yakınsama eğrisi

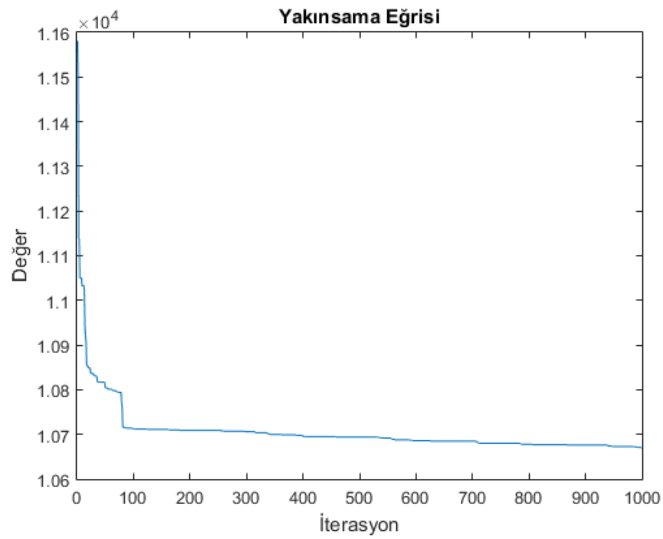
Thyroid veri seti sonuçları incelendiğinde, önerilen BOA-LF algoritması 1921.5825 değeriyle en iyi kümeleme sonucunu elde etmiştir. BOA saf hali 1965.2538 değeriyle ikinci sırada yer almıştır. K-means ve k-medoids algoritmaları sırasıyla

2013.9639 ve 2241.8421 değerlerine ulaşırken, bulanık c-means algoritmasının 2812.4999 değeriyle son sırada yer aldığı görülmektedir. BOA-LF algoritmasının thyroid veri setindeki yakınsama eğrisi Şekil 4.10'da gösterilmiştir.



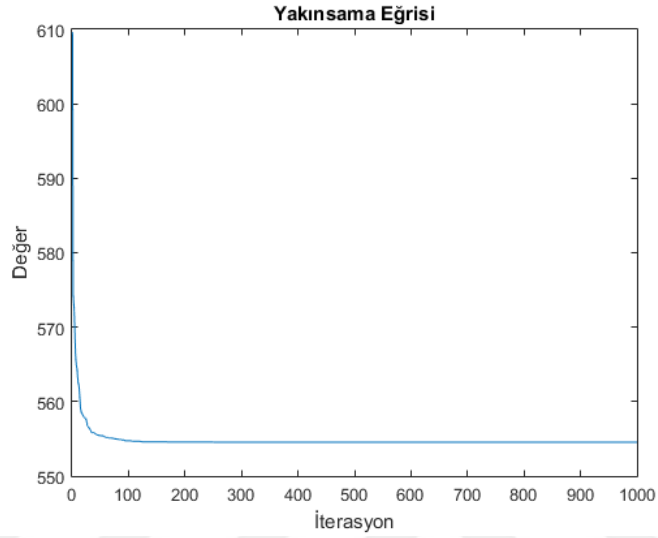
Şekil 4.10. Thyroid yakınsama eğrisi

Heart veri seti sonuçları incelendiğinde, önerilen BOA-LF algoritması 10673.6712 değeriyle en iyi kümeleme sonucunu elde etmiştir. İkinci sırada 10680.1755 değeriyle BOA saf hali yer almaktadır. K-means ve k-medoids algoritmaları sırasıyla 10700.8385 ve 12619.013 değerlerine ulaşırken, bulanık c-means algoritması 13943.6643 değeriyle son sırada yer almıştır. BOA-LF algoritmasının heart veri setindeki yakınsama eğrisi Şekil 4.11'de gösterilmiştir.



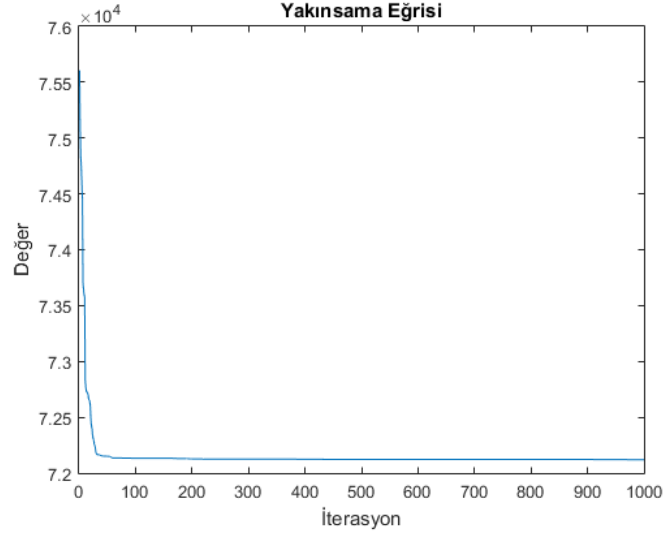
Şekil 4.11. Heart yakınsama eğrisi

Spect veri seti sonuçları incelendiğinde, önerilen BOA-LF algoritması 554.5435 değeriyle en iyi kümeleme sonucunu elde etmiştir. BOA saf hali 554.7039 değeriyle ikinci sırada yer almıştır. Bulanık c-means ve k-means algoritmaları 557.5988 değerini elde ederek aynı sonuca ulaşmıştır. Son sırada 633.544 değeriyle k-medoids algoritması yer almaktadır. BOA-LF algoritmasının spect veri setindeki yakınsama eğrisi Şekil 4.12’de gösterilmiştir.



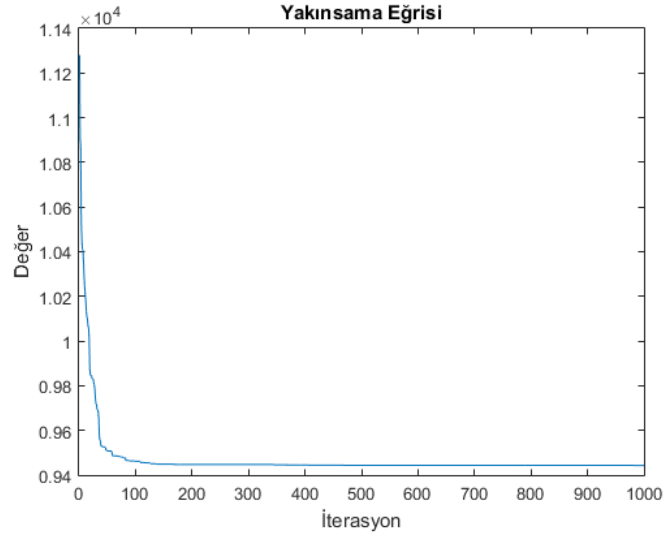
Şekil 4.12. Spect yakınsama eğrisi

Diabetes veri seti sonuçları incelendiğinde, önerilen BOA-LF algoritması 72113.6223 değeriyle en iyi kümeleme sonucunu elde etmiştir. İkinci sırada 72188.2534 değeriyle BOA saf hali yer alırken, k-medoids algoritması 73389.0283 değerine ulaşmıştır. K-means ve bulanık c-means algoritmaları 74604.324 değerini elde ederek aynı sonuca ulaşmış ve son sırada yer almıştır. BOA-LF algoritmasının diabetes veri setindeki yakınsama eğrisi Şekil 4.13’te gösterilmiştir.



Şekil 4.13. Diabetes yakınsama eğrisi

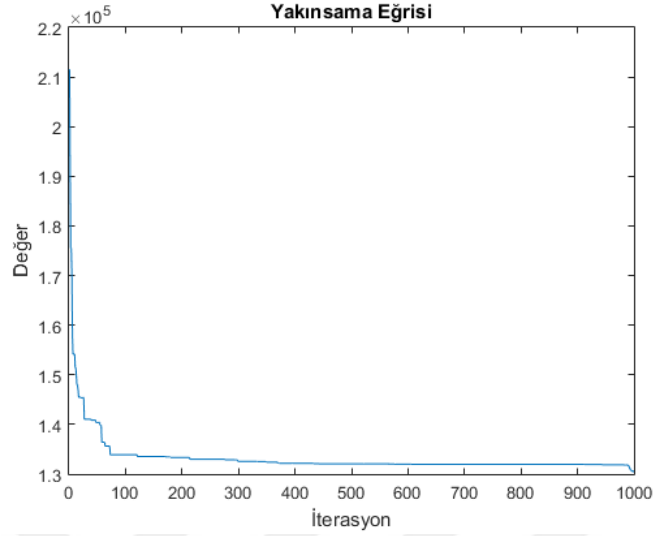
Hepatit veri seti sonuçları incelendiğinde, önerilen BOA-LF algoritması 9457.5178 değeriyle en iyi kümeleme sonucunu elde etmiştir. BOA-LF saf hali 9504.3466 değeriyle ikinci sırada yer almıştır. K-medoids ve k-means algoritmaları sırasıyla 10872.7204 ve 10878.6548 değerlerine ulaşırken, bulanık c-means algoritmasının 12755.0747 değeriyle son sırada yer aldığı görülmektedir. BOA-LF algoritmasının hepatit veri setindeki yakınsama eğrisi Şekil 4.14’te gösterilmiştir.



Şekil 4.14. Hepatit yakınsama eğrisi

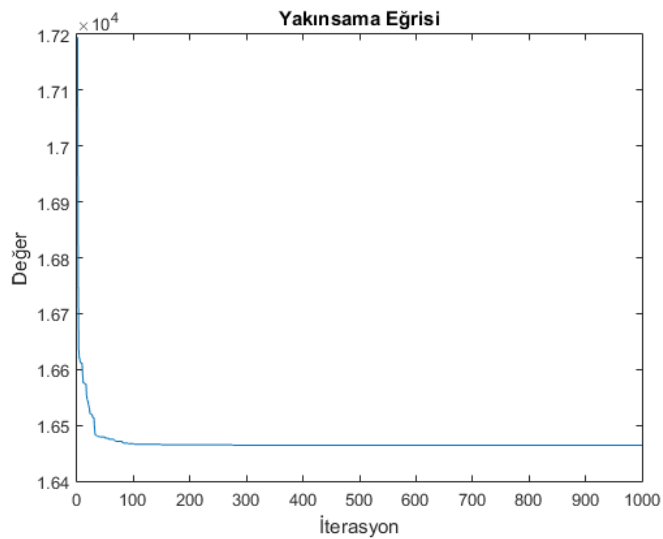
Breast Tissue veri seti sonuçları incelendiğinde, önerilen BOA-LF algoritması 127730.4009 değeriyle en iyi kümeleme sonucunu elde etmiştir. İkinci sırada 129593.7168 değeriyle BOA saf hali yer almaktadır. K-means ve bulanık c-means

algoritmaları 143613.8846 ve 145764.7385 değerlerini elde ederken, k-medoids algoritması 274804.1355 değeriyle son sırada yer almıştır. BOA-LF algoritmasının breast tissue veri setindeki yakınsama eğrisi Şekil 4.15'te gösterilmiştir.



Şekil 4.15. Breast tissue yakınsama eğrisi

Parkinson veri seti sonuçları incelendiğinde, önerilen BOA-LF algoritması 16463.6816 değeriyle en iyi kümeleme sonucunu elde etmiştir. BOA saf hali 16463.9199 değeriyle ikinci sırada yer almıştır. Bulanık c-means ve k-means algoritmaları 17081.5962 değerini elde ederek aynı sonuca ulaşmıştır. Son sırada 17120.1081 değeriyle k-medoids algoritması yer almaktadır. BOA-LF algoritmasının parkinson veri setindeki yakınsama eğrisi Şekil 4.16'da gösterilmiştir.



Şekil 4.16. Parkinson yakınsama eğrisi

Çizelge 4.3. SSE ortalama sıralama değerleri

	K-means	K-medoids	B.C-means	BOA	BOA-LF
Dermatology	2064.199 1	2960.3799 4	5196.3797 5	2579.3799 3	2485.4799 2
E. Coli	68.0015 1	144.1519 5	108.4402 4	75.0888 3	72.6395 2
Balance	1425.0116 1	1681.8454 4	1722.2446 5	1428.1721 3	1426.7978 2
Cancer-Int	2987.8412 3	3790.8969 5	3286.1132 4	2965.7767 2	2964.554 1
Cancer	1.34e+154 1	1.79e+308 2	7.62e+156 3	1.34E+154 1	1.34e+154 1
Iris	97.3381 2	215.3423 5	106.3591 4	97.7151 3	97.0511 1
Glass	226.3667 1	330.9148 3	402.9559 5	332.7943 4	327.0969 2
Wine	18060.6975 4	22277.4703 5	17128.4579 3	16371.1249 2	16358.9337 1
Credit	802497.6684 5	620589.9931 3	759180.4699 4	558242.5354 2	556819.8377 1
Thyroid	2013.9639 3	2241.8421 4	2812.4999 5	1965.2538 2	1921.5825 1
Heart	10700.8385 3	12619.013 4	13943.6643 5	10680.1755 2	10673.6712 1
Spect	557.5988 3	633.544 4	557.5988 3	554.7039 2	554.5435 1
Diabetes	74604.324 4	73389.0283 3	74604.324 4	72188.2534 2	72113.6223 1
Hepatit	10878.6548 4	10872.7204 3	12755.0747 5	9504.3466 2	9457.5178 1
Breast Tissue	143613.8846 3	274804.1355 5	145764.7385 4	129593.7168 2	127730.4009 1
Parkinson	17081.5962 3	17120.1081 4	17081.5962 3	16463.9199 2	16463.6816 1
Avg. Rank	2.62	3.93	4.12	2.31	1.25

Çizelge 4.3'te, her bir algoritma için 30 kez çalıştırılan amaç fonksiyonunun ortalama değeri ve bu değerlere göre, veri setlerindeki algoritma başarı sıralamaları verilmektedir. Ortalama sıralamalar arasında, BOA-LF 1.25 ile en iyi sıralama değerine sahiptir. 2.31 değeriyle BOA saf hali ikinci sırada yer almıştır. K-means, k-medoids ve

bulanık c-means algoritmaları için sırayla 2.62, 3.93 ve 4.12 sıralama değerleri elde edilmiştir.

Uygulama sonuçları SSE amaç fonksiyonu üzerinden değerlendirilmekle birlikte, 30 kez çalıştırılan algoritmaların Rand Index değerleri; B(est) en iyi, W(orst) en kötü, A(verage) ortalama ve S(tdDev) standart sapma sonucu olarak Çizelge 4.4'te ayrıca sunulmuştur. Her veri kümesi için, elde edilen en iyi ortalama sonuç kalın olarak işaretlenmiştir.

Çizelge 4.5'te, veri setlerindeki ortalama Rand Index değerleri ve bu değerlere göre algoritma başarı sıralamaları verilmektedir. Ortalama sıralama değerleri incelendiğinde, önerilen BOA-LF algoritması 2.0 ile en iyi sıralama değerine sahiptir. K-means, k-medoids, bulanık c-means ve BOA saf hali için sırayla 2.62, 2.56, 3.06 ve 2.68 sıralama değerleri elde edilmiştir. Rand Index sıralama sonuçlarında en iyi ve en kötü olarak belirlenen algoritmalar SSE sıralama sonuçlarıyla benzerlik göstermektedir. BOA-LF algoritması her iki tabloda da en iyi sıralama değerine ulaşarak birinci sırada yer almıştır. Aynı şekilde bulanık c-means algoritması her iki tablodaki en kötü sıralama değerini elde ederek sonuncu olmuştur.

Kümeleme sonuçları genel olarak değerlendirildiğinde, BOA'nın Levy Flight ile birlikte kullanılarak algoritmanın saf haline kıyasla iyileştirilmiş bir kümeleme başarısı elde ettiği görülmektedir. Kullanılan veri setlerinin küme sayısı, özellik sayısı ve örnek sayısı açısından çeşitli farklılıklara sahip olduğu ve uygulama sonuçlarında daha başarılı olan BOA-LF yönteminin veri seti parametrelerinden bağımsız şekilde bu değerleri elde ettiği tespit edilmiştir. Dolayısıyla, önerilen yöntemin veri setindeki parametrelerden etkilenmeksizin kümelemede başarılı alternatif bir algoritma olduğu söylenebilir.

Çizelge 4.4. RI sonuçları

		K-means	K-medoids	B.C-means	BOA	BOA-LF
Dermatology	B	0.71202	0.72334	0.50659	0.70608	0.70694
	W	0.68692	0.68699	0.20339	0.69301	0.69579
	A	0.69917	0.70919	0.44492	0.69893	0.69991
	S	0.010793	0.013597	0.13502	0.0037825	0.0031291
E. Coli	B	0.87469	0.85295	0.81366	0.87512	0.87387
	W	0.80277	0.69382	0.81366	0.69044	0.77743
	A	0.82342	0.75737	0.81366	0.82723	0.83821
	S	0.029751	0.064891	0	0.046848	0.034043
Balance	B	0.59482	0.60899	0.43173	0.62774	0.62577
	W	0.57178	0.55814	0.43173	0.53776	0.53981
	A	0.58456	0.58395	0.43173	0.58162	0.58474
	S	0.0093167	0.22282	0	0.22526	0.024044
Cancer-Int	B	0.92687	0.92687	0.92149	0.93229	0.93229
	W	0.92417	0.91881	0.92149	0.92958	0.93229
	A	0.92525	0.92203	0.92149	0.9322	0.93229
	S	0.0014773	0.0029463	0	0.00049554	0
Cancer	B	0.5324	0.62964	0.53329	0.56071	0.59956
	W	0.5324	0.50458	0.53329	0.50122	0.50143
	A	0.5324	0.55536	0.53329	0.5227	0.52682
	S	0	0.04929	0	0.016779	0.022586
Iris	B	0.88132	0.94199	0.89907	0.90013	0.88742
	W	0.87541	0.77492	0.89907	0.71912	0.87541
	A	0.87777	0.81007	0.89907	0.8788	0.88298
	S	0.0032404	0.073778	0	0.030568	0.0047235
Glass	B	0.69181	0.64855	0.61852	0.66455	0.65825
	W	0.60074	0.57466	0.59744	0.55049	0.55957
	A	0.6636	0.61834	0.60839	0.59409	0.59495
	S	0.035995	0.033488	0.010262	0.026732	0.025852
Wine	B	0.7218	0.7304	0.72996	0.73115	0.73115
	W	0.69531	0.65652	0.72996	0.72061	0.72061
	A	0.70061	0.70741	0.72996	0.72541	0.72553
	S	0.011846	0.029091	0	0.003917	0.0036908
Credit	B	0.50812	0.57161	0.50999	0.52432	0.52432
	W	0.50777	0.52432	0.50999	0.52432	0.52432
	A	0.50784	0.54432	0.50999	0.52432	0.52432
	S	0.0001557	0.020417	0	0	0
Thyroid	B	0.79272	0.68208	0.55357	0.74841	0.75909
	W	0.58372	0.5596	0.52097	0.5699	0.56563
	A	0.62552	0.61097	0.53293	0.61181	0.61784
	S	0.093468	0.049755	0.012598	0.035145	0.031824

		K-means	K-medoids	B.C-means	BOA	BOA-LF
Heart	B	0.51893	0.52327	0.52645	0.52811	0.52983
	W	0.51893	0.50195	0.5163	0.51759	0.51893
	A	0.51893	0.50977	0.52103	0.52466	0.52568
	S	0	0.0089379	0.0036471	0.0027893	0.0030635
Spect	B	0.6741	0.6741	0.6741	0.6741	0.6741
	W	0.6741	0.6741	0.6741	0.6741	0.6741
	A	0.6741	0.6741	0.6741	0.6741	0.6741
	S	0	0	0	0	0
Diabetes	B	0.54622	0.54622	0.54622	0.54622	0.54622
	W	0.54622	0.54622	0.54622	0.54622	0.54622
	A	0.54622	0.54622	0.54622	0.54622	0.54622
	S	0	0	0	0	0
Hepatit	B	0.667	0.74888	0.54888	0.60165	0.57039
	W	0.6134	0.55285	0.54888	0.56576	0.57039
	A	0.64678	0.59904	0.54888	0.57232	0.57039
	S	0.027765	0.084185	0	0.0080189	0
Breast Tissue	B	0.61823	0.76777	0.64574	0.7198	0.7198
	W	0.50062	0.69847	0.64574	0.67202	0.67413
	A	0.58498	0.74424	0.64574	0.71535	0.71417
	S	0.047678	0.027046	0	0.0088526	0.011392
Parkinson	B	0.63077	0.63077	0.63077	0.63077	0.63077
	W	0.63077	0.63077	0.63077	0.63077	0.63077
	A	0.63077	0.63077	0.63077	0.63077	0.63077
	S	0	0	0	0	0

Çizelge 4.5. RI ortalama sıralama değerleri

	K-means	K-medoids	B.C-means	BOA	BOA + LF
Dermatology	0.69917 3	0.70919 1	0.44492 5	0.69893 4	0.69991 2
E. Coli	0.82342 3	0.75737 5	0.81366 4	0.82723 2	0.83821 1
Balance	0.58456 2	0.58395 3	0.43173 5	0.58162 4	0.58474 1
Cancer-Int	0.92525 3	0.92203 4	0.92149 5	0.9322 2	0.93229 1
Cancer	0.5324 3	0.55536 1	0.53329 2	0.5227 5	0.52682 4
Iris	0.87777 4	0.81007 5	0.89907 1	0.8788 3	0.88298 2
Glass	0.6636 1	0.61834 2	0.60839 3	0.59409 5	0.59495 4
Wine	0.70061 5	0.70741 4	0.72996 1	0.72541 3	0.72553 2
Credit	0.50784 4	0.54432 1	0.50999 3	0.52432 2	0.52432 2
Thyroid	0.62552 1	0.61097 4	0.53293 5	0.61181 3	0.61784 2
Heart	0.51893 4	0.50977 5	0.52103 3	0.52466 2	0.52568 1
Spect	0.6741 1	0.6741 1	0.6741 1	0.6741 1	0.6741 1
Diabetes	0.54622 1	0.54622 1	0.54622 1	0.54622 1	0.54622 1
Hepatit	0.64678 1	0.59904 2	0.54888 5	0.57232 3	0.57039 4
Breast Tissue	0.58498 5	0.74424 1	0.64574 4	0.71535 2	0.71417 3
Parkinson	0.63077 1	0.63077 1	0.63077 1	0.63077 1	0.63077 1
Avg. Rank	2.62	2.56	3.06	2.68	2.0

5. SONUÇLAR VE ÖNERİLER

5.1. Sonuçlar

Doğru ve etkili bir kümeleme yapabilmek için çeşitli algoritmalar kullanılmakta ve alternatif yöntemler geliştirilmektedir. BOA, özellikle son yıllarda araştırmacılar tarafından sıklıkla kullanılan metasezgisel bir optimizasyon algoritmasıdır. Bu çalışmada, balinaların avlanma stratejisine dayanan Balina Optimizasyon Algoritması (BOA) ve Levy Flight (LF) arama stratejisi birlikte kullanılarak kümeleme uygulaması yapılmıştır. Levy Flight stratejisi, BOA algoritmasının daha iyi global arama yapabilmesi ve kümeleme sonuçlarının iyileştirilmesi için kullanılmıştır.

Önerilen yöntem, UCI veri ambarından alınan 16 gerçek veri seti (Dermatology, E. Coli, Balance, Cancer-Int, Cancer, Iris, Glass, Wine, Credit, Thyroid, Heart, Spect, Diabetes, Hepatit, Breast Tissue, Parkinson) üzerinde test edilmiştir. Algoritmada Toplam Karese Uzaklık fonksiyonu uygunluk fonksiyonu olarak kullanılmıştır ve iterasyonlar boyunca bu fonksiyonun minimize edilmesi amaçlanmıştır.

BOA-LF'nin kümeleme problemindeki performansı, üç temel kümeleme algoritması ile karşılaştırmalı olarak incelenmiştir. K-means, k-medoids ve bulanık c-means algoritmalarıyla karşılaştırıldığında, BOA-LF 11 veri setinde (Cancer-int, Iris, Wine, Credit, Thyroid, Heart, Spect, Diabetes, Hepatit, Breast Tissue, Parkinson) en iyi kümeleme sonucunu elde etmiştir.

Kullanılan veri setlerinin küme sayısı, özellik sayısı ve örnek sayısı bakımından çeşitli farklılıklara sahip olduğu görülmektedir. Uygulama sonuçları incelendiğinde daha başarılı olan BOA-LF yöntemi veri seti parametrelerinden bağımsız şekilde bu değerlere ulaşmıştır. Dolayısıyla, önerilen yöntemin veri setindeki parametrelerden etkilenmediği sonucunu çıkarabiliriz.

Önerilen optimizasyon temelli kümeleme algoritması, global arama yeteneği sayesinde mevcut arama uzayının dışına çıkmadan en iyi küme merkezlerini bulmaya odaklanmıştır. BOA-LF, birçok veri setinde standart kümeleme algoritmalarından daha iyi kümeleme başarısı elde ettiği için alternatif bir kümeleme algoritması olarak kullanılabilir. Ayrıca, önerilen algoritmanın sonuçları diğer yaklaşımlarla karşılaştırıldığında kolay uygulanabilir ve etkilidir. BOA-LF kümeleme algoritmasının diğer hibrit kümeleme yaklaşımlarıyla kıyaslanması veya farklı veri madenciliği yöntemlerinde kullanılması gelecekteki araştırmaların konusu olabilir.

6. KAYNAKLAR

- Afşar, B. (2014). Metasezgisel algoritmalarda çevrim içi ayarlama ile uyarlanabilir parameter kontrolü, Doktora Tezi, Ege Üniversitesi Fen Bilimleri Enstitüsü, İzmir.
- Akın, K. Y. (2008). Veri Madenciliğinde Kümeleme Algoritmaları ve Kümeleme Analizi. Doktora Tezi, Marmara Üniversitesi, Sosyal Bilimler Enstitüsü, İstanbul, 164s.
- Alataş, B. (2007). Kaotik haritalı parçacık sürü optimizasyonu algoritmaları geliştirme, Doktora Tezi, Fırat Üniversitesi Fen Bilimleri Enstitüsü.
- Aldenderfer, M. S., & Blashfield, R. K. (1984). Cluster Analysis Sage University Papers Series. Quantitative.
- Al-Temeemy, A. A., Spencer, J. W., & Ralph, J. F. (2010, July). Levy flights for improved ladar scanning. In 2010 IEEE International Conference on Imaging Systems and Techniques (pp. 225-228). IEEE.
- Armano, G., & Farmani, M. R. (2014). Clustering analysis with combination of artificial bee colony algorithm and k-means technique.
- Atamer, B. (1992). Kümeleme Analizi (Cluster Analysis) ve Kümeleme Analizinin İlaç Sektörüne Uygulanması, Yüksek Lisans Tezi.
- Ball, G.H. (1970). Classification Analysis, Menlo Park, Calif. Stanford Research Institute; Reproduced by National Technical Information Service, Springfield, p.
- Başbuğ, S. (2013). Metasezgisel algoritmalar ile anten dizi sentezi, Doktora Tezi, Erciyes Üniversitesi Fen Bilimleri Enstitüsü, Kayseri.
- Bilen, H. (2009). Bankacılık Sektöründe Personel seçimi ve Performans Değerlendirmesine ilişkin Veri Madenciliği uygulaması (Yüksek Lisans Tezi), Gazi Üniversitesi. Fen Bilimleri Enstitüsü, Ankara.
- Birattari, M., Paquete, L., Stützle, T., & Varrentrapp, K. (2001). Classification of metaheuristics and design of experiments for the analysis of components. Teknik Rapor, AIDA-01-05.
- Boushaki, S. I., Kamel, N., & Bendjeghaba, O. (2018). A new quantum chaotic cuckoo search algorithm for data clustering. Expert Systems with Applications, 96, 358-372.
- Canayaz, M. (2015). Cırcır böceği algoritması: Yeni bir meta-sezgisel yaklaşım ve uygulamaları, Doktora Tezi, İnönü Üniversitesi Fen Bilimleri Enstitüsü, Malatya.
- Canayaz, M., & Özdağ, R. (2017). Data clustering based on the whale optimization. Middle East Journal of Technic, 2(1), 178-187.

- Cebeci, Z., & Yıldız, F. (2015). Bulanık C-Ortalamlar Algoritmasının Farklı Küme Büyüklükleri için Hesaplama Performansı ve Kümeleme Geçerliliğinin Karşılaştırılması. Ulusal Zootekni Bilim Kongresi, 227-239.
- Chechkin, A. V., Metzler, R., Klafter, J., & Gonchar, V. Y. (2008). Introduction to the theory of Lévy flights. *Anomalous Transport*, 129-162.
- Chen, Y. (2010, October). Research and simulation on Levy flight model for DTN. In 2010 3rd International Congress on Image and Signal Processing (Vol. 9, pp. 4421-4423). IEEE.
- Cui, D. (2017). Application of whale optimization algorithm in reservoir optimal operation. *Advances in Science and Technology of Water Resources*, 37(3), 72-79.
- Cura, T. (2008). Modern sezgisel teknikler ve uygulamaları. Papatya Yayıncılık Eğitim.
- Cura, T. (2012). A particle swarm optimization approach to clustering. *Expert Systems with Applications*, 39(1), 1582-1588.
- Çakmak, Z. (1999). "Kümeleme Analizinde Geçerlilik Problemi ve Kümeleme Sonuçlarının Değerlendirilmesi", *Dumlupınar Üniversitesi Sosyal Bilimler Dergisi*, Sayı:3, Kasım,s.187-205.
- Dinçer, Ş. E. (2006). Veri madenciliğinde K-means algoritması ve tıp alanında uygulanması (Master's thesis, Kocaeli Üniversitesi, Fen Bilimleri Enstitüsü).
- Dodge, Y. (Ed.). (2012). *Statistical data analysis based on the L1-norm and related methods*. Birkhäuser.
- Doğan, C. (2019). Balina optimizasyon algoritması ve gri kurt optimizasyonu algoritmaları kullanılarak yeni hibrit optimizasyon algoritmalarının geliştirilmesi.(Master dissertation, Erciyes Üniversitesi Fen Bilimleri Enstitüsü).
- Doğan, İ. (2002). Kümeleme analizi ile seleksiyon. *Turkish Journal of Veterinary and Animal Sciences*, 26(1), 47-53.
- Duran, B.S. and Odell, P.L. (1974). *Cluster Analysis (Lecture Notes in Economics and Mathematical Systems, Econometrics; Managing Editors: M. Beckmann and H.P. Kunzi)*. Springer-Verlag: New York.
- Eberhart, R., & Kennedy, J. (1995, November). Particle swarm optimization. In *Proceedings of the IEEE international conference on neural networks* (Vol. 4, pp. 1942-1948).
- Edwards, A. M., Phillips, R. A., Watkins, N. W., Freeman, M. P., Murphy, E. J., Afanasyev, V., ... & Viswanathan, G. M. (2007). Revisiting Lévy flight search patterns of wandering albatrosses, bumblebees and deer. *Nature*, 449(7165), 1044.

- Everitt, B. (1974). *Cluster Analysis*. London:Heinemann Educational Books Ltd.
- Everitt, B. (1993). *Cluster Analysis for Applications*, Academic Press, New York.
- Everitt, B., Dunn G. (2001). *Applied Multivariate Data Analysis.*, Oxford University Press Inc., New York.
- Everitt, B.S., Dunn G. (1992). *Applied Multivariate Data Analysis*. Oxford University Press, New York.
- Fırat, S. Ü. (1997). Kümeleme analizi: İstihdamın sektörel yapısı açısından Avrupa ülkelerinin karşılaştırılması. *İÜ Sosyal Bilimler Dergisi*, 3(2), 50-59.
- Frigui, H., & Krishnapuram, R. (May 01, 1999). A robust competitive clustering algorithm with applications in computer vision. *Ieee Transactions on Pattern Analysis and Machine Intelligence*, 21, 5, 450-465.
- Goldbogen, J. A., Friedlaender, A. S., Calambokidis, J., Mckenna, M. F., Simon, M., & Nowacek, D. P. (2013). Integrative approaches to the study of baleen whale diving behavior, feeding performance, and foraging ecology. *BioScience*, 63(2), 90-100.
- Hair, J. F., Black, W. C., Babin, B. J., Anderson, R. E., & Tatham, R. L. (1998). *Multivariate data analysis* (Vol. 5, No. 3, pp. 207-219). Upper Saddle River, NJ: Prentice hall.
- Han, J., & Kamber, M. (2006). *Data mining: Concepts and techniques*. Amsterdam: Elsevier.
- Hof P.R., Gucht V.D. E. (2007). Structure of the cerebral cortex of the humpback whale, *Megaptera novaeangliae* (Cetacea, Mysticeti, Balaenopteridae). *Anat Rec*;290:1–31.
- Höppner, F., Klawonn, F., Kruse, R., & Runkler, T. (1999). *Fuzzy cluster analysis: methods for classification, data analysis and image recognition*. John Wiley & Sons.
- Humphries, N. E., Queiroz, N., Dyer, J. R., Pade, N. G., Musyl, M. K., Schaefer, K. M., ... & Sims, D. W. (2010). Environmental context explains Lévy and Brownian movement patterns of marine predators. *Nature*, 465(7301), 1066-1069.
- İşık, M., & Çamurcu, A. Y. (2007). K-means, k-medoids ve bulanık c-means algoritmalarının uygulamalı olarak performanslarının tespiti.
- Johnson, R. A., & Wichern, D. W. (1988). *Applied Multivariate Statistical Analysis*:(2nd Ed.) Prentice Hall, Englewood Cliffs, New Jersey.
- Karaboga, D., & Ozturk, C. (2011). A novel clustering approach: Artificial Bee Colony (ABC) algorithm. *Applied soft computing*, 11(1), 652-657.

- Karaboğa, D. (2011). Yapay zeka optimizasyon algoritmaları: tabu araştırma, ısıl işlem, genetik, karınca koloni, yapay bağışıklık, Diferansiyel gelişim, parçacık sürü optimizasyonu, yapay arı koloni algoritmaları. Nobel Yayın Dağıtım.
- Karakoyun, M. (2015). Kurbağa sıçrama algoritmasının kümeleme problemlerine uygulanması (Doctoral dissertation, Selçuk Üniversitesi Fen Bilimleri Enstitüsü).
- Karthikeyan, S., & Christopher, T. (2014). A hybrid clustering approach using artificial bee colony (ABC) and particle swarm optimization. *International Journal of Computer Applications*, 100(15).
- Khalilpourazari, S., & Khalilpourazary, S. (2018). SCWOA: an efficient hybrid algorithm for parameter optimization of multi-pass milling process. *Journal of Industrial and Production Engineering*, 35(3), 135-147.
- Kıran, M. S. (2014). Optimizasyon problemlerinin çözümü için yapay arı kolonisi algoritması tabanlı yeni yaklaşımlar (Doctoral dissertation, Selçuk Üniversitesi Fen Bilimleri Enstitüsü).
- Kızıloluk, S. (2013). Sosyal tabanlı sezgisel optimizasyon algoritmalarıyla sınıflandırma kurallarının keşfi, Yüksek Lisans Tezi, Munzur Üniversitesi Fen Bilimleri Enstitüsü, Tunceli.
- Lee, C. Y., & Yao, X. (2001, May). Evolutionary algorithms with adaptive lévy mutations. In *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No. 01TH8546) (Vol. 1, pp. 568-575)*. IEEE.
- Lorr, M. (1983). *Cluster analysis for social scientists*. Jossey-Bass Incorporated Pub.
- MacQueen, J. (1967, June). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability (Vol. 1, No. 14, pp. 281-297)*.
- Mane, S. U., & Gaikwad, P. G. (2014). Hybrid particle swarm optimization (HPSO) for data clustering. *International Journal of Computer Applications*, 97(19).
- Maulik, U., & Bandyopadhyay, S. (2000). Genetic algorithm-based clustering technique. *Pattern recognition*, 33(9), 1455-1465.
- Maulik, U., & Mukhopadhyay, A. (2010). Simulated annealing based automatic fuzzy clustering combined with ANN classification for analyzing microarray data. *Computers & Operations Research*, 37(8), 1369-1380.
- Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in engineering software*, 95, 51-67.
- Moertini, V. (2002). Introduction to five data clustering algorithm. *Integral*, 7(2), 87-96.

- Murphy, P.M., Aha, D.W. (1994). UCI Repository of machine learning databases, Irvine, CA: University of California, Department of Information and Computer Science.[<http://www.ics.uci.edu/~mlearn/MLRepository.html>].
- Naes, T., & Mevik, B. H. (1999). The flexibility of fuzzy clustering illustrated by examples. *Journal of Chemometrics*, 13(3-4), 435-444.
- Nasiri, J., & Khiyabani, F. M. (2018). A whale optimization algorithm (WOA) approach for clustering. *Cogent Mathematics & Statistics*, 5(1), 1483565.
- Özdamar, K. (1999). Paket Programları ile İstatistiksel Veri Analizi-1. 2. Baskı, Eskişehir: Kaan Kitapevi.
- Pavlyukevich, I. (2007). Lévy flights, non-local search and simulated annealing. *Journal of Computational Physics*, 226(2), 1830-1844.
- Pereyra, M. A., & Batatia, H. (2010, October). A Levy flight model for ultrasound in skin tissues. In *2010 IEEE International Ultrasonics Symposium* (pp. 2327-2331). IEEE.
- Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336), 846-850.
- Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. (2009). GSA: a gravitational search algorithm. *Information sciences*, 179(13), 2232-2248.
- Reynolds, A. M., & Frye, M. A. (2007). Free-flight odor tracking in *Drosophila* is consistent with an optimal intermittent scale-free search. *PloS one*, 2(4), e354.
- Rhee, I., Shin, M., Hong, S., Lee, K., Kim, S. J., & Chong, S. (2011). On the levy-walk nature of human mobility. *IEEE/ACM transactions on networking (TON)*, 19(3), 630-643.
- Rui, X., & Wunsch, D. (2005). Survey of clustering algorithms. *Ieee Transactions on Neural Networks*, 16, 3, 645-678.
- Selim, S. Z., & Alsultan, K. (1991). A simulated annealing algorithm for the clustering problem. *Pattern recognition*, 24(10), 1003-1008.
- Servi, T. (2009). Çok değişkenli karma dağılım modeline dayalı kümeleme analizi. *Yayımlanmamış Doktora Tezi, Çukurova Üniversitesi, Adana.*
- Sharma, S. (1996). *Applied multivariate techniques*. New York: J. Wiley.
- Shelokar, P. S., Jayaraman, V. K., & Kulkarni, B. D. (2004). An ant colony approach for clustering. *Analytica Chimica Acta*, 509(2), 187-195.
- Shlesinger, M. F. (2006). Mathematical physics: Search research. *Nature*, 443(7109), 281.

- Storn, R., & Price, K. (1997). Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4), 341-359.
- Sutantyo, D. K., Kernbach, S., Levi, P., & Nepomnyashchikh, V. A. (2010, July). Multi-robot searching algorithm using Lévy flight and artificial potential field. In *2010 IEEE Safety Security and Rescue Robotics* (pp. 1-6). IEEE.
- Şahinli, F. (1999). Kümeleme analizine fuzzy set teorisi yaklaşımı. Gazi Üniversitesi, Yayınlanmamış Yüksek Lisans Tezi, Ankara.
- Şentürk, A. (1995). Kümeleme Analizi ve Bir Uygulama, Yüksek Lisans, Uludağ üniversitesi, Bursa.
- Talbi, E.G. (2009). *Metaheuristics: from design to implementation* (Vol 74). Hoboken: John Wiley & Sons.
- Tan, P. N., Steinbach, M., & Kumar, V. (2006). Classification: basic concepts, decision trees, and model evaluation. *Introduction to data mining*, 1, 145-205.
- Tanyıldızı, E., & Cıgalı, T. Kaotik Haritalı Balina Optimizasyon Algoritmaları. *Fırat Üniversitesi Mühendislik Bilimleri Dergisi*, 29(1), 307-317.
- Tatlıdil, H. (1996). *Uygulamalı Çok Değişkenli İstatistiksel Analiz*, Cem Ofset Ltd. Şti. Eylül, Ankara.
- Terdik, G., & Gyires, T. (2008). Lévy flights and fractal modeling of internet traffic. *IEEE/ACM Transactions on Networking*, 17(1), 120-129.
- Van der Merwe, D. W., & Engelbrecht, A. P. (2003, December). Data clustering using particle swarm optimization. In *The 2003 Congress on Evolutionary Computation, 2003. CEC'03.* (Vol. 1, pp. 215-220). IEEE.
- Viswanathan, G. M., Afanasyev, V., Buldyrev, S. V., Murphy, E. J., Prince, P. A., & Stanley, H. E. (1996). Lévy flight search patterns of wandering albatrosses. *Nature*, 381(6581), 413.
- Viswanathan, G. M., Buldyrev, S. V., Havlin, S., Da Luz, M. G. E., Raposo, E. P., & Stanley, H. E. (1999). Optimizing the success of random searches. *nature*, 401(6756), 911-914.
- Wan, M., Wang, C., Li, L., & Yang, Y. (2012). Chaotic ant swarm approach for data clustering. *Applied Soft Computing*, 12(8), 2387-2393.
- Yang, X. S. (2010). Firefly algorithm, Levy flights and global optimization. In *Research and development in intelligent systems XXVI* (pp. 209-218). Springer, London.
- Yang, X. S. (2010). *Nature-inspired metaheuristic algorithms*. Luniver press.

- Yang, X. S., & Deb, S. (2013). Multiobjective cuckoo search for design optimization. *Computers & Operations Research*, 40(6), 1616-1624.
- Yang, X.-S. (2010). *Engineering optimization: An introduction with metaheuristic applications*. Hoboken, N.J: John Wiley.
- Yao, X., Liu, Y., & Lin, G. (1999). Evolutionary programming made faster. *IEEE Transactions on Evolutionary computation*, 3(2), 82-102.
- Yetkin, M. (2016). Metasezgisel Algoritmaların Jeodezi'de Kullanımı. *Geomatik Dergisi*, 1(1), 8-13.
- Zhang, C., Ouyang, D., & Ning, J. (2010). An artificial bee colony approach for clustering. *Expert Systems with Applications*, 37(7), 4761-4767.

