T.C.
ERCİYES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

YAPAY ARI KOLONİ ALGORİTMASI TABANLI GÖRÜNTÜ KÜMELEME YÖNTEMLERİNİN GELİŞTİRİLMESİ

(Doktora Tezi)

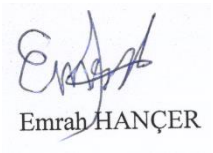
Hazırlayan
Emrah HANÇER

Danışman
Prof. Dr. Derviş KARABOĞA

Ocak 2016
KAYSERİ

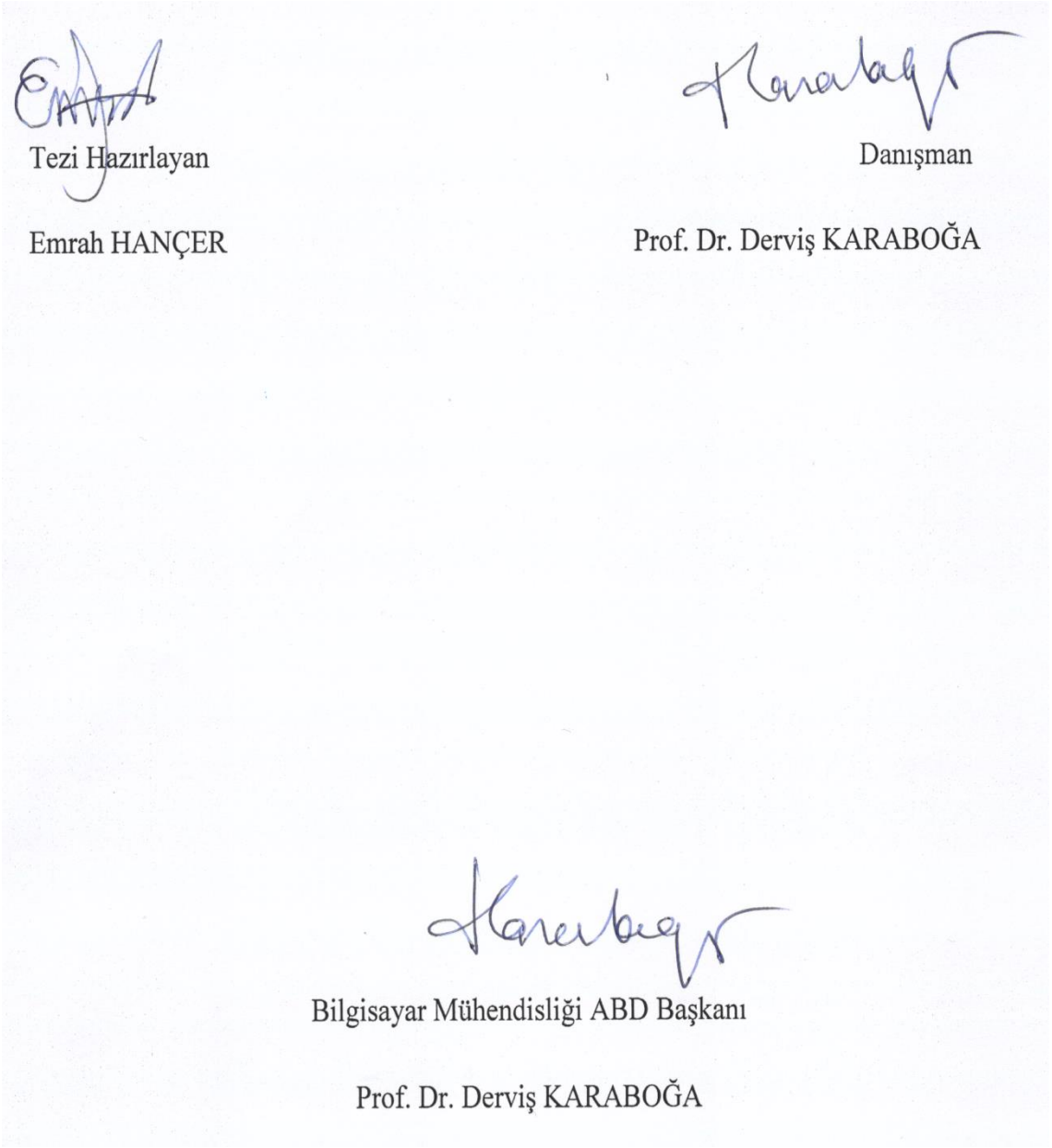
BİLİMSEL ETİĞE UYGUNLUK

Bu çalışmadaki tüm bilgilerin, akademik ve etik kurallara uygun bir şekilde elde edildiğini beyan ederim. Aynı zamanda bu kural ve davranışların gerektirdiği gibi, bu çalışmanın özünde olmayan tüm materyal ve sonuçları tam olarak aktardığımı ve referans gösterdiğimi belirtirim.



YÖNERGEYE UYGUNLUK

"**Yapay Arı Koloni Algoritması Tabanlı Görüntü Kümeleme Yöntemlerinin Geliştirilmesi"** adlı Doktora tezi, Erciyes Üniversitesi Lisansüstü Tez Önerisi ve Tez Yazma Yönergesi'ne uygun olarak hazırlanmıştır.



Tezi Hazırlayan

Emrah HANÇER

Danışman

Prof. Dr. Derviş KARABOĞA

Bilgisayar Mühendisliği ABD Başkanı

Prof. Dr. Derviş KARABOĞA

Prof. Dr. Derviş KARABOĞA danışmanlığında **Emrah HANÇER** tarafından hazırlanan "**Yapay Arı Koloni Algoritması Tabanlı Görüntü Kümeleme Yöntemlerinin Geliştirilmesi"** adlı bu çalışma, jürimiz tarafından Erciyes Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalında **Doktora** tezi olarak kabul edilmiştir.

21 / 01 / 2016

JÜRİ:

Danışman	:Prof. Dr. Derviş KARABOĞA	.....
Üye	:Prof. Dr. Mehmet Reşit TOLUN	.....
Üye	:Doç. Dr. Harun UĞUZ	.....
Üye	:Yrd. Doç. Dr. Celal ÖZTÜRK	.....
Üye	:Yrd. Doç. Dr. Fehim KÖYLÜ	.....

ONAY:
Bu tezin kabulü Enstitü Yönetim Kurulunun 02/02/2016 tarih ve 2016/08-07 sayılı kararı ile onaylanmıştır.

..... / ..... / .....

Prof. Dr. Kazım KEŞLİOĞLU

Enstitü Müdürü

ÖNSÖZ / TEŞEKKÜR

Danışmanım Prof. Dr. Derviş KARABOĞA başta olmak üzere, üzerimde emeği olan tüm hocalarıma teşekkürü bir borç bilirim.

Ayrıca, bu tez çalışmasına maddi destek veren YÖK Öğretim Elemanı Yetiştirme Programı'na ve çalışmalar için gerekli altyapıyı sağlayan Erciyes Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölüm Başkanlığı'na teşekkür ederim.

Son olarak ve en önemlisi, hayatımdaki en büyük lütuflardan biri olan, desteğini her daim yanımda hissettiğim ve beni yetiştirebilmek için her türlü zorluğun üstesinden gelen sevgili anneme minnet ve şükranlarımı sunarım.

Emrah HANÇER

Kayseri, 2016

YAPAY ARI KOLONİ ALGORİTMASI TABANLI GÖRÜNTÜ KÜMELEME YÖNTEMLERİNİN GELİŞTİRİLMESİ

Emrah HANÇER

Erciyes Üniversitesi, Fen Bilimleri Enstitüsü

Doktora Tezi, Ocak 2016

Danışman: Prof. Dr. Derviş KARABOĞA

KISA ÖZET

Kümeleme, bir verideki görünmeyen ve gizli bulunan bilginin elde edilmesi amacıyla verinin karakteristik özelliklerini değerlendirerek gruplarına ayırma işlemidir. Kümeleme mühendislikten istatistiğe kadar birçok alanda hayati bir öneme sahiptir. O alanlardan birisi de örüntü tanıma ve görüntü işlemedir. Kümeleme yöntemleri, örüntü tanıma ve görüntü işleme alanında görüntülerin segmentasyon ve analiz işlemlerinde kullanılan önemli tekniklerden biridir. Ancak, mevcut alışılagelmiş kümeleme yöntemlerinin yerele takılma ve başlangıç koşullarına bağımlılık gibi problemleri mevcuttur. Evrimsel hesaplama teknikleri bu problemlerin giderilmesinde kullanılabilecek küresel araştırma algoritmalardır. Evrimsel hesaplama tekniklerinden biri olan yapay arı koloni (ABC) algoritması, diğer algoritmalara nazaran daha yeni, kolay uygulanabilir, az parametre değeri içeren ve erken yakınsama problemleri az olan bir algoritmadır. ABC algoritması birçok farklı problemlere başarılı bir şekilde uygulanmıştır. Ancak, ABC algoritmasının görüntü kümeleme üzerindeki potansiyeli henüz tam olarak araştırılmış değildir.

Bu tez çalışmasının amacı, ABC'nin görüntü kümelemedeki performansını araştırmak ve geliştirmektir. Aynı zamanda bu çalışma farklı evrimsel hesaplama tekniklerinin de görüntü kümeleme üzerindeki kabiliyetini incelemekte ve değerlendirmektedir.

Tez çalışmasının birinci kısmında, ilk olarak beyin MRI görüntüleri için ABC ile kümeleme tabanlı tümör segmentasyon metodolojisi önerilmiştir. Elde edilen sonuçlar neticesinde, önerilen metodolojinin klasik kümeleme metodolojilerinden daha iyi tümör segmentasyonu sağladığı görülmüştür. İkinci olarak literatürdeki mevcut görüntü kümeleme kriterlerinin avantaj ve dezavantajları incelenerek yeni bir görüntü kümeleme kriteri ortaya konulmuş ve önerilen kümeleme kriteri diğer kümeleme kriterlerinden

daha iyi kümeleme performansı sağlamıştır. Üçüncü olarak ABC ile kümeleme tabanlı bir renk kuantalama yöntemi geliştirilmiştir. Bu kuantalama yöntemi ABC'nin bu alandaki ilk uygulaması olma niteliği taşımaktadır. Önerilen kuantalama yöntemi, RGB ve L*a*b* renk uzaylarında test edilmiş ve diğer yöntemlerden daha iyi performans gösterdiği görülmüştür.

Tez çalışmasının ikinci kısmında küme sayısı bilinmeyen verileri otomatik olarak kümeleyen ABC tabanlı yöntemler geliştirilmiştir. Geliştirilen yöntemler dışarıdan küme sayısının belirlenmesine ihtiyaç duymamakta ve verinin karakteristik yapısını analiz ederek kümeleme işlemini gerçekleştirmektedir. Önerilen bu yöntemler mevcut yöntemlerden kümeleme kalitesi ve optimum küme sayısını elde etme açısından daha iyi performans göstermiştir. Standart ABC'nin bir verideki optimum küme sayısının belirlenmesi problemi üzerindeki performansı ilk olarak bu tez çalışmasında incelenmiştir.

Anahtar Kelimeler: Kümeleme analizi, yapay arı koloni, evrimsel hesaplama, küme sayısının belirlenmesi.

DEVELOPMENT OF ARTIFICIAL BEE COLONY ALGORITHM BASED IMAGE CLUSTERING APPROACHES

Emrah HANÇER

Erciyes University, Graduate School of Natural and Applied Sciences

Ph.D. Thesis, January 2016

Supervisor: Prof. Dr. Derviş KARABOĞA

ABSTRACT

Clustering is the process of grouping through evaluating the characteristics of a data set to obtain unseen and invaluable information from the data. Clustering has an important role in a various number of fields from engineering to statistics. One of these fields is pattern recognition and image processing. Clustering is one of the important current techniques in image segmentation and image analysis processes of pattern recognition and image processing. However, existing clustering techiques suffer from problems, such as local optima and dependent on initial conditions. In order to alleviate these problems, evolutionary computation techniques that are global search algorithms can be used. Among evolutionary computation techniques, artificial bee colony (ABC) algorithm is quite new, can be easily implemented, includes less control parameters and less stagnation problems than other well-known algorithms. ABC has been succesfully applied to a various number of fields, but its potential for image clustering has not been fully investigated yet.

The overall goal of this thesis is to investigate and improve capability of the ABC algorithm for image clustering. Furthermore, this thesis also investigated and evaluated the capabilities of different evolutionary computation techniques for image clustering.

In the first stage of the thesis, an ABC clustering based tumour segmentation methodology for brain MRI images was first proposed. It was seen from the obtained results that the proposed methodology can provide better tumour segmentation than traditional methodologies. Second, a new image clustering criterion was introduced by investigating advantages and disadvantages of existing criteria. The proposed clustering criterion obtained better clustering quality than other criteria. Third, an ABC clustering based color quantization technique was developed. The proposed quantization technique

was tested on RGB and L*a*b* color spaces, and performed better than other quantization techniques. Furthermore, to our knowledge, this proposed technique is the first application of ABC algorithm on color quantization.

In the second stage of this thesis, ABC based automatic clustering techniques which automatically group the data without the knowlege of cluster number are improved. The improved techniques do not require the cluster number as a parameter and perform clustering by analyzing the characteristics of the data. The improved techiques performed better than existing techniques in terms of clustering quality and obtaining the optimum number of clusters. The potential of standard ABC for the determination of cluster number was first investigated in this thesis.

Keywords: Cluster anaylsis, artificial bee colony, evolutinary computation, determination of cluster number.

İÇİNDEKİLER

YAPAY ARI KOLONİ ALGORİTMASI TABANLI DİNAMİK KÜMELEME YÖNTEMLERİNİN GELİŞTİRİLMESİ

2. BÖLÜM

EVRİMSEL HESAPLAMA TEKNİKLERİ

3. BÖLÜM

KÜME SAYISINI PARAMETRE OLARAK ALAN ABC TABANLI GÖRÜNTÜ KÜMELEME YÖNTEMLERİ VE UYGULAMALARI

4. BÖLÜM

KÜRESEL EN İYİ ABC ALGORİTMASI İLE OTOMATİK KÜMELEME YÖNTEMİ

5. BÖLÜM

İKİLİ ABC VE K-MEANS'E DAYALI OTOMATİK KÜMELEME YÖNTEMLERİ

6. BÖLÜM

TARTIŞMA, SONUÇ VE ÖNERİLER

KISALTMA VE SİMGELER

ISODATA	Iterative Self-Organizing Data Analysis Technique
DYNOC	Dynamic Optimal Cluster Seeking
MML	Minimum Length Encoding
ABC	Yapay arı koloni algoritması (artificial bee colony)
MRABC	Modulation rate artificial bee colony
GABC	Gbest guided artificial bee colony
AMABC	Angle modulated artificial bee colony
DisABC	Discrete artificial bee colony
IDisABC	Improved discrete artificial bee colony
GBABC	Genetic binary artificial bee colony
PSO	Parçacık sürü optimizasyon (particle swarm optimization)
BPSO	Binary particle swarm optimization
QBPSO	Quantum inspired particle swarm optimization
NBPSO	Novel binary particle swarm optimization
DCPSO	Dynamic clustering particle swarm optimization
DE	Diferansiyel gelişim (differential evolution)
GA	Genetik algoritma (genetic algorithm)
GCUK	Genetic Clustering with an unknown number of clusters K
VGA	Variable string length genetic algorithm
CGA	Clustering genetic algorithm
NSGA II	Nondominated sorting genetic algorithm II
PESA II	Pareto envelope based selection algorithm II
MODE	Multi-objective differential evolution
DEMO	Differential evolution for multi-objective optimization

TABLOLAR LİSTESİ

ŞEKİLLER LİSTESİ

GİRİŞ

Veri (data) işlenmemiş gerçek ya da soyut bilgi olarak tanımlanmakta ve ölçüm, deney, gözlem ya da araştırma yolu ile elde edilmektedir [1]. Veri ayrıca belirli bir nesne, birey veya olguya ait bir temsildir. Veriler temel olarak nicel ve nitel olmak üzere ikiye ayrılmaktadır. Nicel veriler, ölçüm ya da sayım yolu ile toplanan ve sayısal bir değeri olan verilerdir. Nitel veriler, sayısal bir değeri olmayan veriler olarak bilinmektedir.

Veriler gerçek zamanlı uygulamalar ve hayatın birçok alanında hayati bir öneme sahip olmasına rağmen, tek başına bir anlam ifade etmemektedir. Verilerin bir anlam ifade etmesi ve ilişkili oldukları olguyu açıklaması, elle veya teknolojik araçlar ile işlenmesiyle mümkün olmaktadır. Böylece bir problemin çözümüne veya karar verme mekanizmalarına yardımcı olabilecek bir formata dönüşmektedir.

Verilerin analizi ve işlenmesi veri madenciliği (data mining) disiplininin konusudur. Veri madenciliği eldeki verilerden üstü kapalı, fazla net olmayan, ancak potansiyel olarak kullanışlı bilginin elde edilmesi sürecidir [2]. Veri madenciliği disiplini, makine öğrenmesi (machine learning), istatistik, matematik, meta-sezgisel algoritmalar ve veri tabanı sistemleri gibi çeşitli bilim dallarını bir arada kullanır. İstatistik ve matematik bilimi, nicel ve nitel verilerin özetlenmesini sağlayan araçları bünyesinde barındırır. Makine öğrenmesi, yapay zeka (artificial intelligence) disiplininin bir alt dalı olup, verinin ait olduğu olgu veya problemi veriye göre modelleme işlemini gerçekleştiren algoritmaları içerir. Meta-sezgisel algoritmalar, karmaşık problemlerde veriler üzerinden etkin çözümler geliştirmek için kullanılmaktadır. Veri tabanı sistemleri, verilerin yapısal (hiyerarşik) bir düzende bir arada tutulmasını sağlamaktadır.

Veriden anlamlı bilginin elde edilmesi işleminde makine öğrenmesi önemli bir yer tutmaktadır. Makine öğrenmesi yöntemleri temel olarak tahmin edici ve özetleyici (açıklayıcı) olmak üzere iki farklı gruba ayrılırlar [3]. Tahmin edici yöntemler, mevcut

verinin analiz edilmesi ve yeni durumlara uygun teşhisi ortaya koyabilecek modelin ortaya çıkarılmasını sağlar. Özetleyici yöntemler ise, mevcut verileri analiz ederek öz ve anlamlı bilgiyi çıkartır. Temel bilinen makine öğrenmesi yöntemleri şu şekildedir [3]:

1. Kümeleme analizi: Veri elemanları (instances) arasındaki benzerlikler dikkate alınarak veri elemanlarının alt gruplara (kümelere) ayrılması işlemidir. Verinin hangi sınıfa ait olduğu bilinmeden elemanlar arasındaki ilişki (örn. Öklid mesafesi) dikkate alınarak kümeleme işlemi gerçekleştirilir.
2. Ayrıklık tespiti: Veri elemanları arasında belirli bir kural setine veya genel eğilime göre aykırılık gösterenlerin ayıklanması işlemidir. Diğer bir tabirle gürültülü elemanların elimine edilmesini sağlar.
3. Birliktelik kurallarının keşfi: Bir verideki sıklıkla gerçekleşen olayların tespit edilmesi işlemidir. Bir başka ifade ile bir veri setindeki öznitelikler (features) arasındaki ortak özelliklerin tespit edilmesi amacıyla kullanılır (örn. Apriori algoritması [4]).
4. Regresyon analizi: Bir veri setindeki öznitelikler arasındaki ilişkinin matematiksel bir modelinin oluşturulmasıdır. Matematiksel model oluşturulurken bağımsız giriş özniteliklerin ve bağımlı çıkış özniteliklerinin sisteme girilmesi gereklidir.
5. Sınıflandırma: Bir veri setindeki elemanların öznitelikler ve sınıf isimleri (class labels) arasında bağlantı kurularak sınıflarına ayrılması işlemidir. Regresyon analizinde olduğu gibi çıkış değerlerinin (sınıf isimlerinin) bilinmesine ihtiyaç vardır.

Bu yöntemlerin ilk üçü özetleyici amaç için kullanılırken, diğer ikisi tahmin amacıyla kullanılmaktadır. Bu tez çalışmasında kümeleme analizi üzerine yoğunlaşılmıştır.

Kümeleme analizi, grupları kesin olarak bilinmeyen veri elemanlarının karakteristik özellikleri temel alınarak gruplara (kümelere) ayırma işlemi olarak tanımlanmaktadır. Karakteristik özelliklere göre oluşturulan kümelerde aynı kümede bulunan elemanlar birbirine benzer, aynı kümeye ait olmayan elemanların ise birbirinden farklı olması beklenir. Geometrik olarak değerlendirildiğinde ise aynı kümede bulunan elemanların birbirine yakın ve farklı kümede bulunan elemanların birbirinden uzak olması beklenir.

Kümeleme analizi çok geniş bir kullanım alanına sahiptir [5]. Mühendislik (konuşma tanıma, radar sinyal analizi, bilgi sıkıştırma vb.), bilgisayar bilimleri (web madenciliği, mekânsal veritabanı analizi, örüntü tanıma, görüntü analizi, görüntü segmentasyonu vb.), medikal bilimler (genetik, biyoloji, mikrobiyoloji, patoloji vb.) ve astronomi ve yer bilimleri (coğrafya, jeoloji, uzaktan algılama vb.) bu alanlardan sadece birkaçıdır. Bu tez çalışmasında kümeleme ile görüntü verilerinin analiz ve segmentasyon uygulamaları üzerinde yoğunlaşılacaktır. Ayrıca, bilinen temel veri setleri de geliştirilecek yöntemlerin performansını değerlendirmek amacıyla kullanılacaktır.

Farklı sınıflandırma türleri mevcut olmakla beraber, kümeleme algoritmaları temel olarak hiyerarşik (hierarchical) ve bölünmeli (partitional) olmak üzere iki sınıfta incelenmektedir [6]. Hiyerarşik yöntemler, kümeleri içeren bir hiyerarşik yapı (dendrogram) oluşturulması prensibine dayanır. Bu yöntemler kümeleri daha büyük kümeler oluşturmak suretiyle birleştirerek ("aşağıdan yukarı") veya daha küçük kümeler oluşturmak suretiyle bölerek ("yukarıdan aşağı") çalışmaktadır. Hiyerarşik kümeleme yöntemleri başlangıç koşullarından bağımsız ve küme sayısı parametresine ihtiyaç duymamaktadır. Ancak, her ne kadar küme sayısına ihtiyaç duymasa da kullanıcı tarafından bölünme veya birleştirme seviyesinin belirtilmesi gereklidir. Hiyerarşik yöntemlerin bir diğer dezavantajı da önceki aşamalarda bir kümeye atanmış bir elemanın daha sonraki aşamalarda güncellemesinin mümkün olmamasıdır. Bu durum kümeleme kalitesini olumsuz yönde etkilemektedir. Hiyerarşik yöntemlerden farklı olarak bölünmeli kümeleme yöntemleri, veri elemanlarını önceden belirlenmiş bir kümeleme kriterine göre bir kümeden diğer kümeye dağıtarak veya göndererek kümeleme işlemini gerçekleştirir. Bu yönüyle bölünmeli kümeleme, "NP-hard" optimizasyon problemi olarak da ele alınabilmektedir. Bölünmeli yöntemler başlangıç koşullarına bağımlıdır ve küme sayısının bir kontrol parametre olarak kullanıcı tarafından belirlenmesine ihtiyaç duymamaktadır. Ancak, bölünmeli yöntemler, kümeleme performansı olarak hiyerarşik yöntemlerden daha iyidir ve araştırmacılar tarafından daha fazla kabul görmektedir.

Geleneksel bölünmeli kümeleme yöntemlerin başlangıç koşullarından etkilenme ve yerele takılma gibi problemlerinin temel sebebi küresel araştırma yeteneklerinin zayıf olmasıdır. Bölünmeli kümeleme yöntemlerine küresel araştırma yeteneği kazandırmaya yönelik olarak evrimsel hesaplama teknikleri kullanılmıştır [7, 8]. Özellikle genetik

algoritma (GA) [9] ve parçacık sürü optimizasyon (PSO) algoritması [10] tabanlı geliştirilen birçok kümeleme yöntemi mevcuttur. Evrimsel hesaplama teknikleri ile dizayn edilen kümeleme yöntemlerinin bilinen alışılagelmiş yöntemlere göre daha iyi kümeleme kalitesi sağladığı görülmüştür. Her ne kadar araştırmacılar tarafından birçok evrimsel hesaplama tabanlı yöntem geliştirilmiş olsa da bu yöntemlerin görüntüler üzerindeki uygulamaları hususunda giderilmesi gereken ciddi eksiklikler söz konusudur: 1) çalışmaların çoğunluğu kümeleme kriteri olarak sadece ortalama karesel hataya (mean square error) odaklanmıştır; 2) elde edilen kümeleme kalitesini değerlendirmek için kullanılan metrik ve kriterler yeterli düzeyde değildir; ve 3) yapay arı koloni algoritması (artificial bee colony (ABC)) [11] önemli bir evrimsel hesaplama tekniği olmasına karşın, nispeten yeni bir algoritma olması nedeniyle görüntü kümeleme uygulamaları üzerine yeterli sayıda bilimsel çalışması mevcut değildir.

Günümüzdeki verilerin büyük bir kısmı küme sayısı bilgisini içermemektedir. Dolayısıyla klasik yöntemler ile bu verilerin kümeleme analizinin yapılması mümkün değildir. Veriye uygun küme sayısının elle belirlenmesi de ciddi zaman kaybına neden olmakta ve zorlu bir süreçtir. Verinin karmaşıklığı ve hacmi arttıkça küme sayısının elle belirlenebilmesi imkânsız hale gelmektedir. Bu sebeplerden ötürü, mevcut veri için uygun küme sayısının bulunması son zamanlarda önemli ve araştırmacılar tarafından ilgi gören bir konu haline gelmiştir. Araştırmacılar tarafından problemin çözümüne yönelik birçok yöntem önerilmiştir. Ancak, bu çalışmaların bir kısmı performans analizini sentetik veriler üzerinde gerçekleştirmekte, bir kısmı da optimal veya optimum küme sayısının elde edilmesinde yeterli başarı sağlayamamaktadır. Optimal küme sayısını elde etmede başarı sağlanamamasının en önemli sebeplerinden biri, küme sayısı arama sürecinde yerel optimum küme sayısına takılma problemleridir. Küresel araştırma özellikleri sebebiyle evrimsel hesaplama (evolutionary computation) tabanlı algoritmalar da verilerin optimal küme sayısının bulunması amacıyla kullanılmıştır. Bu yöntemlerin arasında genetik algoritmalar [12] ve diferansiyel gelişim algoritması [13] önemli bir yer tutarken, başarılı bir evrimsel hesaplama yöntemi olan ABC'nin bu alanda sadece bir tek uygulaması [14] mevcuttur. ABC'nin küme sayısının belirlenmesi problemi üzerindeki performansı henüz tam olarak incelenmiş ve araştırılmış değildir.

Bu tezin genel amacı, ABC tabanlı görüntü kümeleme yöntem ve uygulamalarının geliştirilmesidir. ABC [11], bal arılarının zeki yiyecek arama davranış prensiplerinin

modellenmesi sonucu ortaya konmuş bir algoritma olup, evrimsel hesaplama disiplininin sürü zekası temelli algoritmalar grubundandır. ABC'nin bu tezin konusu olan problemlerin çözümünde araç olarak seçilmesinin temel sebepleri olarak aşağıdakiler söylenebilir [15]:

1. Literatüre tanıtıldığı günden itibaren farklı alanlarda değişik problemlerin çözümü için başarılı bir şekilde kullanılmıştır.
2. Çözümlerin farkına dayalı çözüm geliştirme prensibi kullandığından dolayı yakınsama hızı oldukça iyidir.
3. Küresel ve yerel araştırma arasında iyi bir denge söz konusudur. Bu nedenle araştırmanın yerele takılma ihtimali nispeten düşüktür.
4. Kullanıcı tarafından değerleri ayarlanması gereken kontrol parametrelerinin sayısı diğer algoritmalara göre daha azdır.
5. Basit olması nedeniyle uygulanabilirliği ve anlaşılabilirliği oldukça kolaydır.

Tezin amacına ulaşmak için gerçekleştirilmek istenen hedefler şunlardır:

1. Geniş bir literatür taraması yapılarak mevcut yöntemlerin analiz edilmesi ve bu yöntemleri ortak özelliklerine göre sınıflarına ayırarak yeni bir taksonominin ortaya konulması.
2. Görüntü segmentasyonu için ABC kümeleme tabanlı yöntemlerin ve uygulamalarının geliştirilmesi:
    a. ABC tabanlı görüntü segmentasyon metodolojisi ile beyin MRI görüntülerindeki tümörün saptanması,
    b. ABC ile kullanılmak üzere yeni bir görüntü kümeleme kriterinin geliştirilmesi,
    c. ABC kümeleme tabanlı renk kuantalama yönteminin ve uygulamasının gerçekleştirilmesi.
3. Görüntü ve veri setlerinde en uygun küme sayısının saptanmasına yönelik olarak ABC'nin performansının araştırılması ve PSO algoritmasındaki küresel en iyi çözümün kullanılmasından esinlenerek yeni bir sürekli ABC modelinin geliştirilmesi.
4. Görüntü ve veri setlerindeki en uygun küme sayısının belirlenmesine yönelik olarak:

a. Genetik operatörlerin işleyişinden esinlenilerek yeni bir ikili (binary) ABC modelinin geliştirilmesi,
b. En çok bilinen ikili ABC modelinin analizi yapılarak eksikliklerinin saptanması ve bu modelin eksiklerinin giderilerek problemin çözümüne yönelik daha güçlü bir modelin önerilmesi.

Tezin Organizasyonu

Tezin genel organizasyonu şu şekildedir:

Bölüm 1, konu ile ilgili gerekli altyapıyı vermekte ve geniş bir literatür özeti sunmaktadır. İlk olarak ele alınan problemlerin genel tanımları yapılmakta ve problemlerin çözümüne yönelik yöntemler analiz edilerek yöntemlerin avantaj ve eksiklikleri saptanmaktadır. Son olarak ilgili yöntemler ortak özelliklerine göre kategorilerine ayrılarak yeni bir sınıflandırma şeması ortaya çıkarılmaktadır.

Bölüm 2, evrimsel hesaplama temellerini ve evrimsel hesaplama tekniklerinin genel işleyişini özetlemekte ve bu tezde deneysel çalışmalarda kullanılan evrimsel hesaplama tabanlı ikili ve numerik modelleri açıklamaktadır.

Bölüm 3, ABC tabanlı kümeleme yöntemlerini ve onların görüntüler üzerindeki uygulamalarını ele almaktadır. Bölüm temel olarak üç kısımdan oluşmaktadır: 1) ABC tabanlı kümeleme ile beyin MRI görüntülerindeki tümörlü bölgelerin tespit edilmesi, 2) yeni bir görüntü kümeleme kriterinin dizayn edilerek ABC ile uygulanması ve 3) ABC kümeleme tabanlı renk kuantalama yöntemi geliştirilerek RGB ve L*a*b* renk uzaylarında uygulanması.

Bölüm 4, sürekli ABC tabanlı küme sayısını otomatik saptayan kümeleme yönteminin geliştirilmesi ve uygulamasını sunmaktadır. Öncelikle vektörel araştırma ve küresel en iyi kaynağın ABC'nin komşu üretme mekanizmasına entegrasyonu ile tasarlanan sürekli ABC modeli açıklamaktadır. Daha sonra, geliştirilen sürekli ABC modeli ile küme sayısını otomatik saptayan kümeleme yöntemi genel hatlarıyla verilmekte ve geliştirilen yöntemin literatürdeki bilinen bazı sürekli modellerle birlikte görüntü ve veri setleri üzerindeki performans analizi yapılmaktadır.

Bölüm 5, ikili ABC ve K-means tabanlı küme sayısını otomatik saptayan hibrit kümeleme yöntemlerinin geliştirilmesi ve uygulamasını içermektedir. İlk olarak geliştirilen ikili ABC modelleri sunulmakta ve onların küme sayısını saptamaya yönelik uygulamaları açıklanmaktadır. Önerilen modellerin literatürdeki bilinen ikili modeller ile karşılaştırmalı analizi yapılarak bölüm sona ermektedir.

Bölüm 6, tez çalışmasında elde edilen sonuçları vurgulamakta ve değerlendirmelere yer vererek ileride yapılabilecek çalışmalar ile ilgili önerileri sunmaktadır.

1. BÖLÜM

KÜMELEME PROBLEMİ VE ÇÖZÜMÜ İÇİN ÖNERİLEN YÖNTEMLER

Bu bölümde tez çalışmasında kullanılan problemlerin detayları verilmektedir. İlk olarak kümeleme analizi tanımlanmakta ve farklı kümeleme konsept ve yöntemleri tartışılmaktadır. Ardından kümelemenin görüntü segmentasyon ve renk kuantalama alanlarındaki uygulamaları incelenmektedir. Son olarak da küme sayısının belirlenmesi üzerine detaylı bir literatür taraması ve bu problemin çözümüne yönelik yöntemlerin sınıflandırmasını içeren bir taksonomi ortaya konulmaktadır.

1.1. Kümeleme Analizi

Kümeleme yöntemleri veri elemanlarını belirli sayıdaki kümelere (gruplara) ayırır. Kümeler benzer elemanların bir arada olduğu ve farklı elemanların birbirinden ayrı olarak bulunduğu topluluklardır. Reel uzayda bir küme, grup içindeki herhangi iki nokta arasındaki uzaklığın, grup içindeki ve dışındaki iki nokta arasındaki uzaklıktan daha küçük olduğu bir araya toplanmış noktalar yığınıdır. Yüksek boyutlu uzayda küme tanımı ise, noktaların yoğun olarak bulunduğu bölgelerin düşük yoğunluklu bölgelerden ayrılmasıdır. Yine bir başka tanımda küme, içsel homojenlik (internal homogenity) ve dışsal ayrıklık (external separation) ilkelerini bir arada bulunduran bütündür [16]. Kısacası herkes tarafından kabul gören genel bir küme tanımı yapmak pek mümkün değildir [17].

Kümelemenin matematiksel tanımı: N tane eleman (instance/pattern) ve d tane öznitelikten (feature) oluşan $N \times d$'lik bir Z verisi ($Z=\{z_1,\ldots,z_p,\ldots,z_N\}$, $z_p=\{z_{p1},z_{p2},\ldots,z_{pd}\}$) verilmiş olsun. Bir veri elemanı (z_p), verinin bir noktasını veya tek bir objesini temsil eder. Öznitelikler (z_{pd}), bir elemanı bireysel olarak nitelendiren parçalardır. Z verisini

bölünmeli bir kümeleme yöntemi ile K tane kümeye ($\{C_1,C_2,...C_K\}$, $K \leq N$) ayırmak için gerekli şartlar şunlardır:

- Her birey bir kümeye atanmak zorundadır, $\bigcup_{k=1}^{K} C_k = Z$
- Her kümenin en az bir tane bireyi olmalıdır, $C_k \neq \emptyset$, $k=1,2,...K$
- Her birey yalnızca bir kümeye atabilir, $C_k \cap C_j = \emptyset$, $k,j=1,...,K$ *ve* $k \neq j$ (Not: katı bölünmeli kümeleme için geçerli)

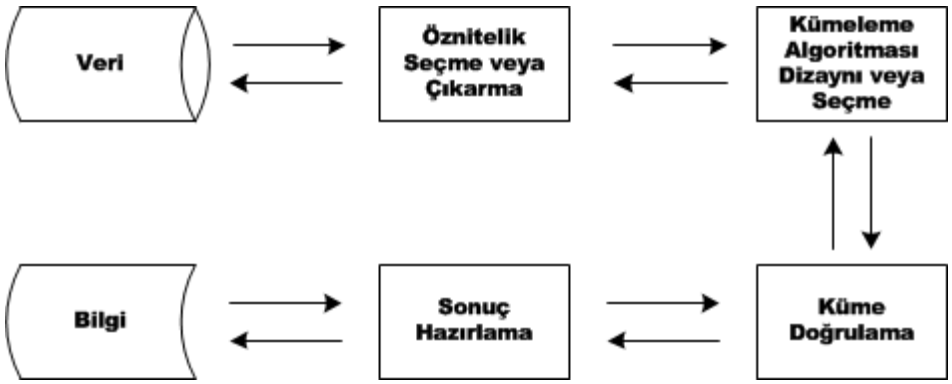
Kümeleme yöntemleri küme veya gruplarla ilgili sayı, hacim, yoğunluk ve konum gibi bilgileri veya kısaca sınıf bilgisi (class labels) içermeyen 'etiketsiz' ('unlabelled') verilerle ilgilenir. Bundan ötürü kümeleme, makine öğrenmesi tekniklerinin danışmasız öğrenme (unsupervised learning) bölümünde değerlendirilir ve danışmalı öğrenme (supervised learning) kapsamına dahil olan sınıflandırmadan (classification) ayrılır. Danışmasız öğrenme yapısı nedeniyle kümeleme veri analizinde zor işlemlerden biri olarak da kabul edilir. Kümeleme analizinin temel adımları şu şekildedir [18, 19]:

- Öznitelik seçme veya çıkarma (feature selection or extraction): Öznitelik seçme bir veriye ait öznitelik setinden belirli bir kriter esas alınarak uygun olanların seçilmesi işlemidir. Öznitelik çıkarma ise bir veriye ait öznitelik setini transformasyona uğratarak faydalı ve yeni öznitelikler elde etme işlemidir. Bu yöntemlerin kullanılması bellek gereksinimi azaltır, verinin daha anlaşılır hale gelmesini sağlar, verideki gürültünün giderilmesine yardımcı olur ve işlem süresini kısaltır. Dolayısıyla iki yöntem de kümeleme uygulamaları için önemli bir yere sahiptir.
- Kümeleme yöntemi seçimi veya dizaynı: Bu adım benzerlik ölçütünün belirlenmesi ve kümeleme yönteminin dizayn-seçim aşamasıdır. Neredeyse bütün kümeleme yöntemleri doğrudan veya dolaylı olarak veri elemanları arasındaki benzerliği ölçen bir benzerlik ölçütü ile bağlanmıştır. Elde edilecek kümeleme sonuçları kümeleme yöntemi ile doğrudan bağlantılıdır. Ancak bütün problemleri çözebilecek bir evrensel kümeleme yöntemi mevcut değildir. Bu sebepten ötürü uygun kümeleme stratejisinin seçilmesi ve dizaynı için ilgili problemin karakteristik özelliklerinin araştırılması büyük önem taşımaktadır.
- Küme doğrulama (Cluster validation): Farklı kümeleme yöntemleri aynı veri için genellikle farklı kümeleme sonuçları üretir. Aynı kümeleme algoritması

farklı parametre değerleri ile kullanıldığında da farklı kümeleme sonuçları elde edilebilir. Bu yüzden doğrulama kriterleri kullanıcılara kümeleme sonuçlarının kalitesi ile ilgili bilgi sağlaması noktasında kritik bir öneme sahiptir. Buna ek olarak veri setindeki bilinmeyen küme sayısının da ortaya çıkarılmasını sağlar. İki çeşit değerlendirme kriteri vardır: dışsal (external) indeksler ve içsel (internal) indeksler. Dışsal indeksler, verinin karakteristik yapısında görünmeyen etiketli (labelled) bilgiyi (örn. sınıf bilgisi) kullanır ve probleme uygun kümeleme yönteminin veya bir kümeleme algoritmasına ait parametre değerlerinin saptanmasını sağlar. İçsel indeksler, veride görünmeyen etiketli bilginin aksine veride görünen karakteristik bilgiyi kullanır ve kümeleme kalitesinin analiz edilmesinin yanında, bir veri setinin karakteristik yapısına uygun optimal küme sayısının saptanmasını da sağlar. Günümüz verilerinin önemli bir bölümünün 'etiketsiz' olması sebebiyle içsel indeksler dışsal indekslerden daha fazla kullanım alanı bulmaktadır.

- Sonuçların yorumlanması: Kümelenmiş verinin matematiksel, istatistiksel vb. metotlarla analiz edilerek anlamlı bilginin elde edilme ve problemin çözümünde kullanılma aşamasıdır.

Kümeleme analizinin genel işleyişi Şekil 1.1'de verilmiştir. Şekil 1.1'den de görüleceği üzere kümeleme analizi adımları arasında bir geri besleme söz konusudur. Bir diğer söylemle kümeleme analizi tek yönlü bir işlem değildir. Birçok durumda kümelemenin sağlıklı bir şekilde gerçekleştirilebilmesi için bir deneme-yanılma zincirine ihtiyaç vardır.



Şekil 1.1. Kümeleme analizinin temel akış şeması.

1.1.1. Benzerlik Ölçütleri

Kümelerin birbirine benzer veri elemanlarının toplandığı gruplar olduğu ve farklı kümelerdeki elemanların birbirine benzer olmadığı belirtilmişti. Veri elemanları arasındaki benzerliği ölçmek amacıyla bazı kriterlere ihtiyaç vardır. Bu kriterlerin genel adı da benzerlik ölçütleridir. Benzerlik ölçütleri arasında en çok bilineni uzaklık ölçütleridir. En çok kullanılan uzaklık ölçütü ise Öklid mesafesidir. Öklid mesafesinin genelleştirilmiş formu Minkowski metriği [18] olup, Denklem 2.1 ile tanımlanmıştır:

$$d^{\alpha}(z_k, z_p) = \left(\sum_{j=1}^{N} (z_{kj} - z_{pj})^{\alpha} \right)^{1/\alpha} = \left\| z_k - z_p \right\|^{\alpha} \qquad (2.1)$$

Burada α parametresi metriğin özelleştirilmesi sağlar. Örneğin, Manhattan uzaklığı Minkowski metriğinin α= 1 durumudur ve en çok kullanılan uzaklıklardan biri olan Öklid uzaklığı Minkowski metriğinin α= 2 durumudur.

Minkowski metriği büyük boyutlu verilerde hesaplama zamanının artmasına sebep olmaktadır. Minkowski metriğinin kullanımı verideki büyük ölçeğe sahip özniteliğin diğer özniteliklere baskın gelmesine de sebep olmaktadır. Bu problem özniteliklerin belirli bir aralıkta normalize edilmesiyle çözülebilmektedir [18].

Bir başka uzaklık ölçütü de Mahalanobis uzaklığıdır [18]. Mahalanobis uzaklığı iki veri elemanı arasındaki mesafeyi hesaplamak için bütün veri elemanlarının kovaryansını kullanır. Dolayısıyla diğer elemanların davranışı da işleme dâhil edilmiş olur. Mahalanobis uzaklığı matematiksel olarak aşağıda tanımlanmıştır:

$$d_M(z_k, z_p) = (z_k - z_p) \sum^{-1} (z_k - z_p)^T \qquad (2.2)$$

Burada $\sum^{-1}$ verideki elemanların kovaryans matrisidir.

Bu uzaklıkların dışında Pearson korelasyon katsayısı, cosinus uzaklığı ve nokta simetri uzaklığı da önemli benzerlik ölçütlerine örnek olarak verilebilir.

1.1.2. Kümeleme Yöntemleri

Kümeleme yöntemleri genel olarak bölünmeli ve hiyerarşik olmak üzere iki kategoride incelenir [20]. Han ve Kamber [21] bu kategorilere yoğunluk tabanlı, model tabanlı ve

bölge tabanlı kümeleme yöntemleri olmak üzere üç kategori daha ilave etmiştir. Ancak kullanım sıklığı sebebiyle bu çalışmada sadece hiyerarşik ve bölünmeli yöntemler üzerinde durulacaktır.

1.1.2.1. Hiyerarşik Kümeleme Yöntemleri

Hiyerarşik yapı alt seviyede temsil edilen bir bireyin üst seviyelerde de temsil edileceği prensibine dayanır. En üst seviyede tüm ilgili bireyler temsil edilirken, alt seviyelere doğru daha az birey temsil edilmektedir. En alt seviyeye ulaşıldığında ise bir birey temsil edilmektedir [22]. Hiyerarşi prensibine bağlı oluşturulan kümeleme yöntemleri üst ve alt seviyenin arasında bulunan katmanlardaki kümelerin birleştirilmesi veya bölünmesi esasına dayanır. Üst seviyede tüm bireyler tek bir kümede temsil edilirken, alt seviyede her birey tek başına bir küme olarak temsil edilir. Üst seviyeden alt seviyeye doğru kümeleri bölerek işlem yapan yöntemler ayrıştırıcı (divisive) hiyerarşik kümeleme yöntemleri olarak isimlendirilir. Alt seviyeden üst seviyeye doğru kümeleri birleştirerek işlem yapan yöntemler ise birleştirici (agglomerative) kümeleme yöntemleri olarak isimlendirilir. Ayrıştırma/birleştirme işlemi benzerlik metriklerine göre gerçekleştirilir ve istenilen küme yapısı elde edilene kadar devam eder. En çok bilinen benzerlik metrikleri tek bağ (single link) [23] ve tam bağdır (complete link) [5]. Tek bağ metrikler farklı kümelerin birbirine en yakın elemanları arasındaki uzaklığı temel alırken, tam bağ metrikler farklı kümelerin birbirlerine en uzak elemanları arasındaki uzaklığı temel alır. Tek bağ temelli yöntemler iki farklı küme arasındaki az sayıdaki noktayı köprü gibi görerek iki kümeyi bir küme olarak algılayabilmektedir. Tam bağ temelli yöntemler tek bağ temelli yöntemlere göre daha kompakt kümeler oluşturabilmekte ve daha faydalı hiyerarşiler kurabilmektedir. Ancak tek bağ temelli yöntemler tam bağ temelli yöntemlere göre çok amaçlıdır. Örneğin, izotropik olmayan kümelerin bulunmasında iyi performans sağlamaktadır. Hiyerarşik yöntemler genel olarak başlangıç koşullarından bağımsızdır ve küme sayısına bir giriş parametresi olarak ihtiyaç duymaz. Fakat zaman karmaşıklığı N tane elemandan oluşan bir veri için en az $O(N^2)$'dir. Dolayısıyla N değeri arttıkça zaman karmaşıklığı da artış gösterecek ve büyük boyutlu verilerde kümeleme işlemini yapmak zorlaşacaktır. Hiyerarşik yöntemlerdeki bir diğer sorun da geri besleme özelliğinin olmamasıdır. Yani önceki aşamalarda bir kümede yer alan bir elemanı daha sonraki aşamalarda güncelleme özelliğine sahip değildir. Son olarak da verinin formu ve kümelerin boyutu hakkındaki

yetersiz bilgi nedeniyle çakışmış kümeler (overlapping clusters) ortaya çıkabilmektedir. Bu alandaki en çok bilinenler, BIRCH [24], ROCK [25] ve LIMBO [26] yöntemleridir. Hiyerarşik kümeleme hakkında daha detaylı bilgiye [27, 28]'den ulaşılabilir.

1.1.2.2. Bölünmeli Kümeleme Yöntemleri

Bölünmeli kümeleme yöntemleri veri elemanlarını bir kümeden diğer kümeye atayarak belirli sayıdaki kümelere dağıtan yöntemlerdir. Bu yöntemler veri setine uygun küme sayısının bir kullanıcı tarafından belirlenmesine ihtiyaç duyar. Veri elemanlarının atama işlemi benzerlik kriterleri doğrultusunda gerçekleştirilir. Minkowski uzaklığı (Öklid mesafesi), Pearson ilişkisi ve Mahalanobis uzaklığı bölünmeli yöntemlerde en çok kullanılan benzerlik kriterlerdir. Bu yöntemler belirli bir kriteri minimize etmeye çalışır (örn. toplam hatalar karesi). Bu sebeple optimizasyon problemi olarak da ele alınabilir. Bölünmeli yöntemlerde kümeler, merkezleri (centroid) vasıtasıyla belirtilir ve birbirinden ayırt edilir. Bölünmeli kümeleme yöntemlerinin avantajları hiyerarşik kümeleme yöntemlerinin dezavantajları, dezavantajları ise hiyerarşik kümeleme yöntemlerinin avantajlarıdır [29]. Bölünmeli yöntemler hiyerarşik yöntemlere göre daha geniş kullanım alanına sahiptir.

En çok bilinen bölünmeli kümeleme yöntemlerinden biri K-means algoritmasıdır [30]. K-means algoritmasında önce kullanıcıdan bir K küme sayısı istenir ve K tane merkez rasgele üretilir. Ardından iteratif olarak veri elemanları Öklid mesafesi benzerlik ölçütüne göre kümelere dağıtılır ve küme merkezleri, bünyesinde bulunan veri elemanlarının ortalaması alınarak güncellenir. Bu prosedür durdurma kriteri sağlanıncaya kadar tekrar edilir. Durdurma kriteri, değerlendirme sayısı veya merkez pozisyonlarında güncellenmenin durması olabilir. K-means uygulanması kolay ve hızlı bir algoritma olmasına rağmen dış etkenlerden ve başlangıç koşullarından etkilenebilmektedir. Bir diğer bilinen kümeleme yöntemi de Fuzzy C-Means (FCM) [31] algoritmasıdır. FCM, K-means algoritmasının bulanık mantık teorisi temel alınarak tasarlanmış versiyonudur. K-means algoritmasından farklı olarak bir veri elemanı birden fazla mevcut kümeye üyelik katsayısı ile atanabilmekte ve üyelik katsayılarının güncellenmesi prensibiyle çalışmaktadır. FCM K-means algoritmasından birçok alanda daha iyi performans sağlamasına karşın, K-means gibi başlangıç koşullarına bağımlı ve dış etkenlerden etkilenebilen bir algoritmadır. Bu algoritmaların dışında K-medoids

[32], CLARA [33] ve CLARANS [34] da popüler bölünmeli kümeleme metotları arasında yer almaktadır.

1.1.3. Küme Doğrulama Teknikleri

Küme doğrulama, kümeleme analizinin temel adımlarından biri olup (Şekil 1.1) üç temel görev için kullanılır [2]:

- Bir algoritmanın kümeleme sonuçlarının değerlendirilerek veri setindeki en iyi bölütlemenin (partition) bulunması,
- Bir algoritmanın parametre değerlerinin saptanması,
- Veri setinin karakteristik yapısının ortaya çıkarılması (örn. belirli bir dağılıma sahip bölümlerin ayrıştırılması).

Bu görevleri yerine getirecek birçok doğrulama indeksi geliştirilmiş ve bu indeksler temel olarak dışsal ve içsel olmak üzere iki gruba ayrılmıştır. Dışsal indeksler verinin karakteristik yapısında görünmeyen 'etiketli' bilgiyi kullanır (örn. sınıf bilgisi). Dışsal indeksler aynı algoritmanın farklı parametre değerleri ile veya tamamen farklı algoritmalar ile elde edilen kümeleme sonuçlarının değerlendirilmesi ve karşılaştırılmasında kullanılır. En çok bilinen dışsal indeksler F-measure [35], Rand indeks [36], mutual information [37], purity [38] ve entropy [38] kriterleridir. Ancak günümüzde etiketli bilgiyi içeren verilerin sayısı azdır. Dolayısıyla dışsal indekslerin kullanımı fazla yaygın değildir.

Dışsal indekslerden farklı olarak içsel indeksler veri setindeki etiketli bilgiye ihtiyaç duymazlar. Dolayısıyla içsel indeksler dışsal indekslere göre daha yaygındır ve veriye uygun küme sayısının saptanmasında kullanılabilir. Genel olarak içsel indekslerin ölçümünü yaptığı iki kriter vardır: kompaktlık (compactness) ve ayrıklık (separateness). Kompaktlık küme dâhilindeki elemanların birbirine ne kadar benzer (yakın) olduğunu ifade eder. Ayrıklık farklı kümelere ait elemanların birbirinden ne kadar farklı (uzak) olduğunu ifade eder. En çok bilinen içsel indeksler Tablo 1.1'de matematiksel ifadeleriyle birlikte sunulmuştur. Küme doğrulama ile ilgili daha fazla bilgi [2]'den elde edilebilir.

Tablo 1.1. Bilinen içsel indeksler ve matematiksel formülleri.

SSW	$SSW = \sum_{k=1}^{K} \sum_{\forall z_p \in C_k} d(z_p, m_k)^2$ $d(z_p, m_k) = \|z_p - m_k\|$ Burada z_p verideki p. eleman ve m_k *kth* merkezdir
SSB	$SSB = \sum_{k=1}^{K} n_k d(m_k, \overline{M})^2$ Burada $\overline{M}$ merkezlerin ortalaması ve n_k *kth* kümedeki toplam eleman sayısıdır.
Calinski-Harabasz [39]	$CH = \frac{SSB/(K-1)}{SSW/(N-K)}$ Burada *K* küme sayısı ve *N* veri elemanı sayısıdır.
Dunn [40]	$Dunn = \frac{\min_{i=1\ldots K} \min_{j=i+1\ldots,K} dis(m_i, m_j)}{\max_{k=1\ldots K} diam(m_k)}$ $dis(m_i, m_j) = \min_{z_p \in C_i, z_q \in C_j} \|z_p - z_q\|^2$ $diam(m_k) = \max_{z_p, z_q \in C_k} \|z_p - z_q\|^2$
Davies-Bouldin [41]	$S_i = \frac{1}{n_i} \sum_{z_p \in C_i} d(z_p, m_i)^2$ $R_{ij} = \frac{S_i + S_j}{d(m_i, m_j)^2}, i \neq j$ $R_k = \max_{j=1,2,\ldots,K} R_{ij}$, where $i = 1,2,\ldots,K$ $DBI = \frac{1}{K} \sum_{k=1}^{K} R_k$
CS measure [42]	$CS = \frac{\sum_{k=1}^{K} \frac{1}{n_k} \sum_{z_p \in C_k} \max_{z_q \in C_k} d(z_p, z_q)}{\sum_{k=1}^{K} \min_{j \in K, j \neq k} d(m_k, m_j)}$
XBI [43]	$XBI = K * \frac{SSW}{SSB}$
I-Index [44]	$I = \left(\frac{1}{K} \times \frac{E_1}{E_k} \times D_k\right)^2$ $D_k = \max_{i,j=1,..K} d(m_i, m_j)^2$ E_1 sabit değerdir. $E_k = \sum_{k=1}^{K} \sum_{z_p \in C_i} d(z_p, m_k)^2$
Xie-Beni [45]	$XB = \frac{\sum_{i=1}^{K} \sum_{p=1}^{N} u_{ip} d(z_p, m_i)^2}{K \min_{t \neq s} d(m_t, m_s)^2}$ Burada u_{ip} *pth* elemanın *ith* kümeye ait üyelik değeridir.

1.2. Görüntü Segmentasyonu

Görüntü segmentasyonu birçok görüntü işleme ve bilgisayarla görü uygulamalarında kullanılmaktadır. Görüntü segmentasyonu görüntü işleme ve örüntü tanımanın bir ön

işlemi olarak da ele alınır [46]. Görüntü segmentasyonu bir resmin birbiriyle bağlantısı olmayan homojen bölgelere ayrılması işlemidir. Bölgelerin homojenliği, piksel yoğunluğu gibi resmin bazı özellikleri kullanılarak ölçülür. Görüntü segmentasyonu kural setiyle bağlantılı olarak aşağıdaki şekilde tanımlanabilir:

Bir resim I ve bir homojenlik kriteri P verilmiş olsun. I resminin K bölgeye $\{R_1, R_2, \ldots, R_k\}$ ayrılması (segmentasyonu) için şu koşullar sağlanmalıdır [29]:

- Resimdeki her piksel bir bölgeye atanmalıdır, $\bigcup_{k=1}^{K} R_k = I$
- Her piksel sadece bir bölgeye atanabilir, $R_k \cap R_j = \emptyset,\ k,j=1,\ldots,K \text{ ve } k \neq j$
- Her bölge homojenlik kriterini sağlamalıdır, $P(R_k)=\text{Doğru},\ k=1,\ldots,K$
- İki faklı bölge homojenlik kriterini sağlayamaz, $P(R_k \cup R_j)=\text{Yanlış},\ k \neq j$

Literatürde birçok görüntü segmentasyon tekniği mevcuttur. Bu yöntemler genel olarak şu kategorilere ayrılır [47]:

1. Eşikleme (thresholding): En basit görüntü segmentasyon tekniğidir. En basit haliyle resmin arka plan ve nesne olacak şekilde bir eşikleme değeri vesilesi ile ikiye ayrılmasıdır. Bir piksel (yoğunluk) değerinden büyük olanlar bir bölgeye, küçük olanlar ise diğer bölgeye olacak şekilde bölme işlemi gerçekleşir. İçerik olarak daha zengin resimler için çoklu eşikleme değerleri kullanılabilir.
2. Kenar (edge) tabanlı: Bölgelerin kenarları saptanarak segmentasyon işlemi gerçekleştirilir. Genellikle resim üzerinde bir çerçeve gezdirilerek yerel değişikliklerin saptanması prensibine dayanır.
3. Bölge büyütme (region growing): Bir tohum (seed) piksel veya piksel seti seçilir. Bu piksel setinin komşuluğundaki pikseller bir homojenlik kriterini sağlıyorsa bir araya getirilir. Bu işlem bölgeye herhangi bir piksel eklenmeyinceye kadar devam eder. Bu yöntemlerin performansı kullanılan homojenlik kriterine ve tohum piksellerin seçimine bağlıdır.
4. Kümeleme: Görüntü segmentasyonu bir kümeleme problemi olarak ele alınabilir. Pikseller veri elemanlarına (patterns, instances) ve bölgeler kümelere (clusters) karşılık gelir. Bu benzerlik, görüntü segmentasyon için

tanımlanan koşul seti ve kümeleme analizi için tanımlanan kural seti arasında da görülebilmektedir.

Takip eden bölümlerde bu tez çalışmasında kümeleme yöntemlerinin görüntü segmentasyonu üzerindeki uygulamaları üzerinde durulacaktır.

1.3. Renk Kuantalama

Renk kuantalama (color quantization) bir resimdeki renk sayısını resimde en az bozulma olacak şekilde azaltmaktır. Kuantalama ile oluşturulmuş resim orijinaline görsel olarak mümkün mertebe yakın olmalıdır. Renk kuantalama genel olarak iki aşamada gerçekleştirilir. İlk aşamada belirli sayısal büyüklükte (genellikle 8 ve 256 arasında) resme uygun renk seti tanımlanır. İkinci aşamada orijinal resimdeki her piksel (yoğunluk) değeri tanımlanan renk setindeki uygun değerle değiştirilir. Dolayısıyla kuantalama işlemi kayıplı resim sıkıştırma yöntemi olarak da tanımlanabilir.

Renk kuantalama yöntemleri segmentasyon, sıkıştırma, doku analizi, damgalama ve doku tespiti gibi birçok uygulamada önemli bir role sahiptir. Bir kuantalama yöntemi orijinal ve üretilmiş resimler arasındaki fark, yöntem kompleksliği ve canlı görsel sistemlerin (human visual systems (HVS)) karakteristik özelliklerini dikkate alması sebebiyle optimum çözüm üretemez. Bunun en temel sebebi, kuantalama işlemi gerçekleştirilirken resimdeki renk katmanlarının ve belirgin önemli detayların birlikte korunamamasıdır. Örneğin, renk setinin resimdeki piksel yoğunluğunun fazla olduğu bölgelerden seçilmesi, küçük sayıdaki piksellerin oluşturduğu önemli detayların kuantalama işlemi sırasında kaybolmasına neden olacaktır. Yine aynı şekilde, renk setinin resimdeki piksel yoğunluğunun az olduğu bölgelerden seçilmesi resmin genel yapısının bozulmasına neden olacaktır. Ciddi çaba ve zaman gerektirdiğinden dolayısıyla küresel optimum çözümün elde edilmesi pek mümkün değildir. Renk kuantalama bu özellikleri sebebiyle NP-hard problem olarak da bilinmektedir [48].

Literatürdeki kuantalama yöntemleri temel olarak bölmeli (splitting) ve kümeleme (clustering) tabanlı olmak üzere ikiye ayrılabilir. Bölmeli yöntemler, tanımlanmış kriterleri esas alarak orijinal görüntüdeki renkleri benzer renklerden oluşan farklı bölgelere ayırır. Bu işlem istenilen bölge sayısı elde edilinceye kadar devam eder. İşlemin sonunda her bölge renk seti olarak seçilir. Böylece oluşturulan renk seti her

ebeveynin iki çocuğa sahip olduğu bir ikili ağaç yapısında görünür. Bu yöntemlerin genel olarak hesaplama karmaşıklığı düşüktür. Fakat optimal bir çözüm elde etmesi zordur. Bilinen bölmeli kuantalama yöntemleri uniform kuantalama [49], median cut [50], octree [51], popularity metot [52] ve minimum varyanstır [49]. Bu yöntemlerin arasında Heckbert [49] tarafından geliştirilen median cut algoritması yaygın olarak kullanılmaktadır. Median cut algoritması histogram eşitlemeye dayalı bir algoritmadır. Bu algoritma ile kuantalama işleminden sonra resmin renk bilgisinin maksimum seviyede kalması sağlanır [49]. Algoritma dört adımdan oluşmaktadır:

1. RGB renk uzayında resimdeki tüm renkleri içeren en küçük hücrenin (bölgenin) seçilmesi;
2. Hücrenin uzun olan ekseni boyunca renklerin küçükten büyüğe sıralanması;
3. Renkleri sıraladıktan sonra hücrenin uzun ekseni boyunca ortadan iki parçaya bölünmesi;
4. Madde 3'teki işlemin istenilen hücre sayısı elde edilinceye kadar devam etmesi.

Kümeleme tabanlı yöntemler ise yüksek hesaplama karmaşıklığına sahiptir ve başlangıç koşullarından etkilenebilmektedir. Ancak bölmeli yöntemlere göre optimal çözüm bulmakta daha başarılıdır. En çok bilinenler K-means, FCM ve ikili kümelemedir. Bu yöntemlere ilişkin detaylı bilgiye Bölüm 1.1.2'den erişilebilir.

Evrimsel hesaplama yöntemleri, başlangıç koşullarının kümeleme tabanlı kuantalama yöntemleri üzerindeki olumsuz etkisinin azaltılması amacıyla bu probleme de uygulanmıştır. Freisleben ve Schrader [53] evrimsel algoritma ve K-means algoritmasını (EA&K-means) birleştirerek bir kuantalama yöntemi önermiştir. Amaç fonksiyonu olarak standart Öklid mesafesi hata fonksiyonu kullanılmıştır. Yöntemde ilk olarak renk setlerini temsil edecek başlangıç popülasyonu üretilir. Bireylerin evrimsel algoritma ile gelişiminin ardından her çevrim sonunda tüm popülasyon bireylerine K-means uygulanır. Bu şekilde başlangıç koşullarının olumsuz etkisi azaltılmaya çalışılmıştır. Önerilen yöntem median cut ve octree bölmeli kuantalama yöntemleri ile karşılaştırılmış ve önerilen yöntemin daha iyi olduğu görülmüştür. Genetik C-means algoritması (GCMA) [54], EA&K-means kuantalama yöntemine benzer bir yolu takip etmiştir. Bu yöntemde GA ve K-means algoritmaları birleştirilir ve K-means algoritması GA'nın her çevrimi sonunda popülasyon bireylerine uygulanır. GCMA, EA&K-means

yönteminden farklı olarak ortalama hata kareleri toplamını (mean square error (MSE)) amaç fonksiyonu olarak kullanır. Elde edilen sonuçlar neticesinde GCMA kuantalama yönteminin median cut ve K-means yöntemlerinden daha iyi olduğu görülmüştür. Omran ve Engelbrecht [55] PSO ve K-means algoritmalarını kullanarak GCMA ve EA&K-means yöntemlerine benzer bir kuantalama yöntemi (PSO-CIQ) önermiştir. PSO-CIQ yöntemi GCMA gibi MSE'yi amaç fonksiyonu olarak kullanır. Ancak GCMA yönteminden farklı olarak K-means her çevrim sonunda hesaplama zamanını azaltmak amacıyla olasılıklı olarak bireylere uygulanır. Sonuçlara göre, PSO-CIQ yönteminin kuantalama performansı Kohonen'in kendi kendine organize olabilen haritalar (SOM) yönteminden [56, 57] daha iyidir. Mevcut çalışmalardan da anlaşılacağı üzere renk kuantalama problemi henüz tam olarak çözülebilmiş değildir ve üzerinde araştırmalar devam etmektedir.

1.4. Küme Sayısının Belirlenmesi

Bilgisayar donanımı ve yazılımı alanında yaşanan teknolojik gelişmeler sayesinde büyük hacimli veriler rahatlıkla saklanabilmektedir. Büyük hacimli veriler faydalı bilgi elde edilebilmesi noktasında önemli bir potansiyele sahiptir. Ancak, bu verilerin birçoğu gerçek küme sayısı bilgisini içermemektedir ve bu verilerin kümelenmesi Bölüm 1.1.2'de tanıtılan kümeleme yöntemleri ile mümkün değildir. Bunun yanında bu verilere ait küme sayısı bilgisinin önceden saptanması verilerdeki karmaşıklık ve büyüklük nedeniyle pek kolay değildir. Dolayısıyla optimal küme sayısının tahmini kümeleme sürecinde önemli işlemlerden biridir.

Literatürde kümeleme ile ilgili birçok inceleme makalesi mevcuttur. Hansen ve Jaumard [16] kümeleme yaklaşımlarını matematiksel programlama üzerinden ele almıştır. Bock [58] kümeleme için olasılık modellerini incelemiştir. Baraldi ve Blonda [59, 60] fuzzy kümeleme yöntemlerinin değerlendirmesini yapmıştır. Wei ve arkadaşları [61] hızlı kümeleme algoritmalarının performansını büyük hacimli veriler üzerinde araştırmıştır. Xu ve Wunsch [62] istatistik ve bilgisayar bilimlerinde yerleşmiş kümeleme yöntemlerinin sistematik açıklamasını sunmaya çalışmıştır. Hruschka ve arkadaşları [8] evrimsel hesaplama tabanlı kümeleme yöntemlerine yoğunlaşmıştır. Mukyopadhyay ve arkadaşları [63] çok amaçlı evrimsel hesaplama tabanlı kümeleme yöntemlerini araştırmıştır. Dharmarajan ve Velmurugan [64] kümeleme yöntemlerinin farklı

alanlardaki uygulamalarını değerlendirmiştir. Fahad ve arkadaşları [65] kümeleme yöntemlerini kategorilere ayırmış ve bu kategorileri temsil eden kümeleme yöntemlerini büyük boyutlu veriler üzerinde karşılaştırmıştır. Kümeleme ile ilgili diğer inceleme makalelerine [66-69]'dan ulaşılabilir.

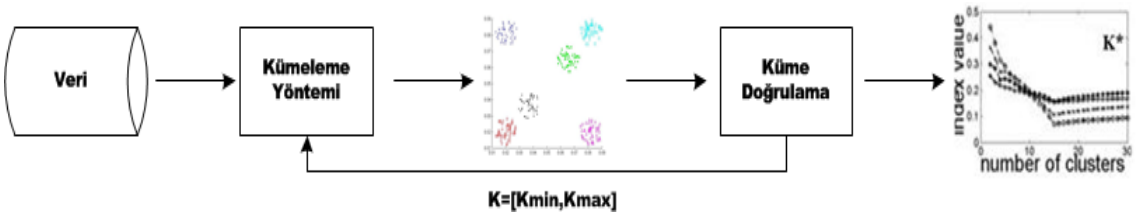
Kümeleme ile ilgili literatürde birçok araştırma ve inceleme makalesi olmasına rağmen küme sayısının tespit edilmesine odaklanan kapsamlı bir çalışma bulunmamaktadır. Erişilebilen bilgiye göre, küme sayısının tespit edilmesi konusu sadece Mirkin [70] tarafından detaylıca ele alınmıştır. Ancak bu çalışmada da konu ile ilgili birçok çalışma göz ardı edilmiş ve detaylı bir analiz yapılmamıştır. Bu sebeplerden ötürü, bu alt bölümde küme sayısının saptanması üzerine geniş bir literatür taramasına yer verilmektedir. İlgili yöntemlerin ortak özellikleri temel alınarak kapsamlı bir taksonomi de beraberinde ortaya konulmaktadır.

Küme sayısını saptayan yöntemler genel olarak üç kategoride ele alınmaktadır: geleneksel yöntemler, birleştirme/bölme tabanlı yöntemler ve evrimsel hesaplama tabanlı yöntemler.

1.4.1. Geleneksel Yöntemler

Bir veri seti için seçilen bir kümeleme algoritmasının küme sayısı K hariç tüm parametre değerleri belirlenmiş olsun. Optimal küme (K^*) sayısını belirlemek amacıyla yürütülen geleneksel yöntemin temel adımları Şekil 1.2'de gösterilmiş ve aşağıda açıklanmıştır [71]:

1. Bir içsel doğrulama indeksi ve $[K_{min},K_{max}]$ ile tanımlı küme sayısı aralığını belirle;
2. K_{min} değerini K parametresine ata;
3. Kümeleme yöntemini (örn. K-means) K küme sayısı ile çalıştır;
4. Elde edilen kümeleme sonuçlarının doğrulama indeks değerini hesapla;
5. K değerini 1 artır ve Adım 3 ve 4'ü $K= K_{max}$ oluncaya kadar tekrarla;
6. K değerleri x ekseninde ve K değerlerine karşılık gelen doğrulama indeks değerleri y ekseninde olacak biçimde bir değerlendirme grafiği oluştur.
7. Oluşturulan grafikteki belirli bir kurala göre saptanan en iyi sonucu K^* küme sayısı olarak belirle.



Şekil 1.2. Geleneksel yöntem ile küme sayısı belirleme.

Değerlendirme grafiğinden doğru küme sayısının elde edilmesi dönüm noktasının (knee point) saptanması ile ilgilidir. *[K_{min},K_{max}]* tanım aralığında bir dönüm noktası minimum, maksimum veya grafikteki önemli değişikliğin olduğu noktalar olabilir. İndeks değerlerinde önemli değişimin olduğu bir dönüm noktası küme sayısı olarak tanımlanır. Ancak, yerel minimum ve maksimum noktalar dönüm noktasının saptanmasında problemlere yol açabilmektedir ve önemli değişimin olduğu noktaları saptamak kolay bir iş değildir [71].

Miligan ve Cooper [72] 30 içsel indeksi kapsayan karşılaştırmalı bir çalışma sunmuştur. Uygunlanan 30 indeks arasında Calinski-Harabasz indeksi [39] küme sayısının saptanmasında en iyi performansı sağlamıştır. Değerlendirme grafiğinden küme sayısını elde etmek için grafiğin maksimum ve minimum olduğu noktalar veya iki nokta arasındaki maksimum farkın sağlandığı noktalar kullanılmıştır.

Dimitriadou ve arkadaşları [73] 15 doğrulama indeksini ikili veriler üzerinde test etmiştir. Elde edilen sonuçlara göre, Ratkowsky-Lance indeksi [74] en iyi performansı elde etmiştir. Onu Xu [75], Scott-Symons [76], Calinski-Harabasz [39] ve *C* [77] indeksleri takip etmiştir. Ayrıca değerlendirme grafiğinde sadece maksimum ve minimum noktaların küme sayısını belirlemek amacıyla kullanılmasının birçok durumda yanlış sonuçlara sebep olabileceği ileri sürülmüştür. Dolayısıyla maksimum ve minimum noktaları seçmenin yanı sıra aşağıdaki farklara dayalı kurallar uygulanmıştır:

1. *x*-ekseninde soldan sağa doğru grafikteki maksimum artışın sağlandığı nokta Denklem 2.3 ile bulunur:

$$\max_k \left(index\_val_k - index\_val_{k-1}\right) \tag{2.3}$$

Burada k K_{min}'*den* K_{max}'a doğru artan küme sayısı değerini ve $index\_val_k$, *k* küme sayısı için indeks değerini ifade eder.

2. *x*-ekseninde sağdan sola doğru grafikteki maksimum azalışın sağlandığı nokta Denklem 2.4 ile hesaplanır:

$$\max_k \left(index\_val_k - index\_val_{k+1} \right) \tag{2.4}$$

3. İkinci farkların en küçük değeri Denklem 2.5 ile belirlenir:

$$\max_k \left(\left(index\_val_{k+1} - index\_val_k \right) + \left(index\_val_k - index\_val_{k-1} \right) \right) \tag{2.5}$$

Yukarıda tanımlanan kural setlerinden hangisinin seçileceğine karar vermek için verilen bir veriye üç kural da uygulanır. Ardından üç kural sonucunda elde edilen küme sayısı sonuçlarının ortalaması alınır. Üç kuraldaki küme sayısı sonucunun hangisi ortalamaya yakınsa o kuralın belirlediği küme sayısı optimal küme sayısı olarak kabul edilir.

Gap statistics [78] ilgili çalışmalardaki [72, 73] gibi direkt değerlendirme grafiğinde dönüm noktası bulmaya çalışmaz. Hatta veri hiçbir anlamlı küme içermiyorsa ve içinde gürültülü elemanlar barındırıyorsa, gap statistics o veriyi temsil edecek örnek bir veri üretir. Örnek verinin küme içi uzaklıklar toplamının (W_k) logaritması daha sonra gözlemlenen veri ile karşılaştırılarak küme sayısının saptanması sağlanır. Yan ve Ye [79] ortalama küme içi uzaklıklar toplamının logaritmasını (*Wk'*) kullanarak gap statistics yönteminin ağırlıklandırılmış versiyonunu önermiştir. Sonuçlar hem standart hem de ağırlıklandırılmış gap statistics yöntemlerinin temel gerçek dünya problemlerinde (örn. Iris, Breast Cancer) doğru küme sayısını saptayamadığını göstermiştir. Mohajer ve arkadaşları [80] W_k and W_k' kriterlerini logaritmalarını almadan uygulamışlardır.

Salvador ve Chan [81] değerlendirme grafiğindeki dönüm noktasını saptamak amacıyla 'L' metodunu önermiştir. 'L' metodu, değerlendirme grafiğine en uygun olacak şekilde grafiğin sağ ve sol tarafından çizilmesi sonucu oluşan kesim noktasını dönüm noktası olarak alır. Deneysel sonuçlar 'L' metodunun gap statistics yönteminden daha iyi sonuç verdiğini ve gap statistics yönteminin 7 verinin sadece 1'inde doğru küme sayısını bulabildiğini göstermiştir. Ancak önerilen metot sadece sentetik veriler üzerinde test edilmiştir.

Jump metodu [82] Gauss dağılım modeline bağlı bir küme içi uzaklık ölçüsü ile bir eğri oluşturur. Oluşturulan eğrinin verinin boyutuna bağlı negatif kuvveti alınarak dönüşümü

yapılır. Dönüştürülen eğride en yüksek sıçramaya (farka) sahip nokta küme sayısı olarak atanır. Ancak jump metodunun hesaplama karmaşıklığı yüksektir ve oluşturulan dağılım eğrisi başlangıç koşullarına bağlıdır.

Zhao ve arkadaşları [83] Denklem 2.5'in değerlendirme grafiğini tam olarak ele alamadığını ve grafikte çok sayıda yerel değişikliklerin olduğu durumlarda yerele takılma problemlerine yol açtığını öne sürmüştür. Bu problemleri gidermek amacıyla Denklem 2.5 açı tabanlı forma dönüştürülmüş ve Denklem 2.6 ile ifade edilmiştir. Yöntemin sadece BIC indeks [84] için dizayn edilmesi ve indeks değerleri arasındaki açı farkının yanlış sonuçlar üretebilmesi bu yöntemin handikapları olarak ortaya çıkmaktadır. BIC indeksi için bir diğer dönüm noktası saptama metodu olan DiffBIC [85], BIC değerlerinin [K_{min},K_{max}] arasına normalize edilmesi prensibine dayanır. Ancak DiffBIC'in performansı [K_{min},K_{max}] değerlerine bağlıdır.

$$\max_k \begin{pmatrix} \arctan(1/|\, index\_val_{k+1} - index\_val_k \,|) \\ +\arctan(1/|\, index\_val_k - index\_val_{k-1} \,|) \end{pmatrix} \tag{2.6}$$

Fujita ve arkadaşları [86] Silhouette metodunu [87] genişleterek eğim istatistik (slope statistics) metodunu önermiştir. Önerilen yöntemin performansı gap statistics, jump ve prediction strength [88] yöntemleri ile test edilmiştir. Önerilen yöntem Iris gibi temel bilinen problemde doğru küme sayısını saptayamamıştır. Dolayısıyla etkili bir yöntem olduğunu söylemek güçtür.

Peng ve arkadaşları [89] küme sayısının saptanmasını bir çoklu kriter karar verme problemi (multi criteria decision making (MCDM)) olarak modellemiştir. MCDM tabanlı yöntemler, doğrulama indekslerini ağırlıklandırılmış ve sıralanmış bir şekilde kullanmıştır. MCDM yöntemlerinin performansı gerçek dünya problemleri üzerinde Dunn, DB ve Hubert gibi birçok doğrulama indeksiyle karşılaştırılmıştır. Elde edilen sonuçlar neticesinde MCDM yöntemleri diğer doğrulama indekslerinden daha iyi performans sağlamıştır. Fakat bu yöntemler de Iris, E-coli ve Glass gibi önemli verilerde doğru küme sayısını saptayamamıştır.

Pedersen ve Kulkarni [90] küme sayısını otomatik saptamak amacıyla 'SenseClusters' adında ücretsiz bir yazılım geliştirmiştir. Bu yazılımda dört yöntem kullanılmıştır. Bu yöntemlerden üçü Hartigan [91], Mojena [92] ve Dice Coefficent [93] temelleri üzerine

kurulmuş olup değerlendirme grafiğinde dönüm noktası bulmaya çalışan yöntemlerdir. Son yöntem ise gap statistics yönteminin geliştirilmiş bir versiyonudur. Bu versiyon küme içi uzaklıklar toplamı (W_k) yerine herhangi bir kriteri kullanma izni vermektedir. Sonuçlara göre gap statistics metodunun geliştirilmiş versiyonu diğer yöntemlerden daha iyi performans sağlamaktadır. Bunun en önemli sebebi, bu yöntemin diğer yöntemler gibi değerlendirme grafiğinde direkt dönüm noktası bulmaya çalışmamasıdır.

Charrad ve arkadaşları [94] 30 doğrulama indeksinden oluşan 'NbClust' adında bir yazılım paketi geliştirmiştir. Dönüm noktası, uygulanan doğrulama indeksine göre maksimum indeks değeri, minimum indeks değeri, noktalar arasındaki maksimum fark ve eşikleme gibi yöntemler kullanılarak saptanır. NbClust paketinde her bir doğrulama indeksi için bir dönüm noktası belirlenir ve toplamda bir veri için (30 doğrulama indeksine karşılık) 30 tane küme sayısı sonucu elde edilir. Bu sonuçlar arasında en çok tekrar edilen küme sayısı optimal küme sayısı olarak atanır. Ancak 30 doğrulama indeksinin hesaplanmasının hesaplama zamanı açısından ciddi bir yük getireceği de bir gerçektir.

Zhang ve arkadaşları [95] ağırlıklandırılmış network tabanlı fuzzy doğrulama indeksi (WGLI) önermiştir. WGLI indeksinin değerlendirme grafiğindeki maksimum değeri küme sayısı olarak belirlenir. WGLI indeksi DBI, Xie-Beni ve PE gibi bazı indekslerle altı sentetik ve dokuz gerçek dünya verileri üzerinde karşılaştırılmıştır.

Küme sayısının belirlenmesi için örneklem tabanlı teknikler de kullanılmıştır [96]:

1. Veride rasgele örneklem yapmak [97].
2. Veriyi eğitim ve test olarak ikiye ayırmak [98].
3. Veride bazı elemanları çoğaltarak diğer elemanlar ile yer değiştirmek (Bootstrapping) [99, 100].
4. Veriye rasgele gürültü eklemek [101].

En çok bilinen örneklem tabanlı yöntem the prediction strength metodudur [88]. Prediction strength metodu danışmalı (supervised) bir bakış açısı ile küme sayısını saptamaya çalışır.

1.4.2. Birleştirme/Bölme Tabanlı Yöntemler

Bu alt bölümde K-means, beklenti büyütme (expectation-maximization) ve istatistiksel kriterler etrafına kurulmuş birleştirme/bölme tabanlı yöntemler sunulacaktır.

Ball ve Hall [102] ISODATA (Iterative Self-Organizing Data Analysis Technique) programını geliştirmiştir. ISODATA, K-means algoritmasında olduğu gibi bir verinin bireylerini iteratif olarak bir kümeden diğer kümeye atar. ISODATA büyük kümeleri küçük kümelere böler ve merkezleri tanımlanmış eşik değerinden daha fazla birbirine yakın olan kümeleri birleştirir. Basit bir yöntem olmasına rağmen toplam altı tane belirlenmesi gereken kontrol parametresi içermektedir.

DYNOC (Dynamic Optimal Cluster Seeking) [103] ISODATA ile benzer karakteristik özelliklere sahiptir. ISODATA'da olduğu gibi benzer kümelerin birleştirilmesi ve büyük kümelerin bölünmesi üzerine kurulmuştur. Ama DYNOC da birçok parametre değeri içermektedir. Huang [104] K-means ve hiyerarşik yöntemleri birleştirerek ISODATA'ya alternatif bir yöntem olan SYNERACT'ı önermiştir. ISODATA ile karşılaştırıldığında SYNERACT zaman karmaşıklığını önemli ölçüde azaltmıştır. Ancak SYNERACT yönteminin de performansı parametre değerlerine bağlıdır.

SNOB [105] bölme, birleştirme, değiştirme vb. taktikler kullanarak veri elemanlarını kümelere atar. İşlemin reddedilmesi veya kabul edilmesine Minimum Length [Encoding] (MML) [105] değerine bakılarak karar verilir. Kanungo ve arkadaşları [106] çoklu bant görüntü segmentasyonu için Minimum Description Length (MDL) tabanlı bir yöntem önermiştir. Bir diğer MDL tabanlı yöntem de Bischof ve arkadaşları [107] tarafından önerilmiştir. Önerilen yöntem çok sayıda küme ile başlar ve bu kümeleri tanımlı uzunlukta bir azalma oluncaya kadar ortadan kaldırır.

Maksimum varyans kümeleme (MVC) metodu [108] küme içi uzaklığı minimize ederek optimal küme sayısını bulmaya çalışır. Kriterin minimizasyonundan önce varyans katsayısının tanımlanması gerekir. MVC işleme çok sayıda küme ile başlar ve tanımlanan kritere bağlı olarak kümeleri böler veya birleştirir.

K-means'in genişletilmiş bir versiyonu olan X-means [109] kümeleri bölme işlemi için BIC kriterini kullanır. İlk olarak alt ve üst sınırlar [K_{min},K_{max}] tanımlanır ve başlangıç

küme sayısı *K*, K_{min} olarak atanır. Akabinde K-means algoritması ile *K* tane küme elde edilir. Bir sonraki adımda K-means ile elde edilen her (ebeveyn) küme BIC kriterine bağlı olarak bölünerek iki (çocuk) küme oluşturulur. Eğer çocuk kümelerin BIC değeri ebeveyn kümeden daha iyiyse ebeveyn kümeler yerine çocuk kümeler temsil edilir. Son olarak yeni kümeleri de kapsayacak şekilde küme seti güncellenir. Bu prosedür *K değeri* K_{max} değerine veya metot en iyi kümeleme yapısına ulaşıncaya kadar devam eder. X-means, Gauss normal dağılımlı kümelerin saptanmasında iyi performans vermektedir. Shahbaba ve Beheshti [110] X-means yöntemini BIC yerine MNDL [111] kriterini kullanarak uygulamıştır. Elde edilen sonuçlar neticesinde MNDL tabanlı X-means yönteminin standart X-means yönteminden daha iyi performans gösterdiği görülmüştür.

Bir diğer K-means'in genişletilmiş versiyonu G-means [112] kümenin Gauss dağılıma sahip olup olmadığını test etmek için Anderson-Darling istatistiğini kullanır. Öncelikle X-means'te olduğu gibi [K_{min},K_{max}] sınırları tanımlanır ve başlangıç küme sayısı *K*, K_{min} olarak atanır. K-means algoritması ile *K* tane küme elde edilir. Akabinde Gauss dağılımına uymayan her ebeveyn küme bölünerek iki yeni çocuk küme oluşturulur. Gauss dağılımına uyan kümeler ise bir sonraki jenerasyon için saklanır. Bu prosedür bölünerek küme ortaya çıkması duruncaya kadar devam eder. G-means ayrık ve küresel olmayan kümelerin saptanmasında iyi performans sağlar. G-means ayrıca işlemcilerin SIMD yönergelerinde kullanılmak üzere de modifiye edilmiştir [113]. G-means X-means'e göre performansı daha yüksektir. Ancak G-means de X-means gibi Gauss dağılımına uymayan kümelerin saptanmasında doğru çalışmayabilir.

PG-means [114] Gauss karışım model ve beklenti büyütme yöntemine dayanarak bir verideki Gauss dağılımlı kümeleri bulmaya çalışır. PG-means küçük sayıdaki *K* kümesiyle başlar (*K* genellikle 1 seçilir) ve *K* her çevrimde 1 artırılarak veriye uygun kümeler elde edilinceye kadar işlem devam eder. Her bir çevrimde hem veri hem de eğitilmiş model bir boyutlu modele dönüştürülür. Akabinde Kolmogorov-Smirnov (KS) testi modele uygulanarak modelin içerdiği *K* sayıdaki kümenin iyi bir dağılım sağlayıp sağlamadığı test edilir. Eğer sağlamışsa o modelin içerdiği *K* kabul edilir. Aksi takdirde *K* bir artırılarak prosedür tekrarlanır. PG-means küçük boyutlu ve çakışan (overlapping) problemlerde X-means ve G-means metotlarından daha iyi çalışır. Ancak, PG-means de sadece Gauss dağılımlı kümeleri araştırır ve sadece numerik özniteliklerle başa çıkabilir.

Vatsavai ve arkadaşları [115] G-means ve X-means metotlarının eksikliklerini azaltarak coğrafi mekânsal (geospatial) görüntü kümeleme için GX-means metodunu önermiştir. G-means'in gelişmiş bir versiyonu olan GX-means, hem X-means hem G-means'in iyi özelliklerini kullanmaya çalışır. İlk olarak Anderson-Darling istatistiği yerine yüksek boyutlu verilere uygun olan Shapiro-Wilk istatistiği kullanır. Bir diğer yenilik ise, BIC değerinin iyileşmesi halinde X-means metodundaki gibi bölünür ve bu işlemin ardından mevcut küme setinde kümelerin birbirine yakın olanlarını tespit etmek amacıyla KL ıraksama testi uygulanır. Eğer birbirine yakın kümeler mevcutsa birleştirilir.

Büyük hacimli problemlerin üstesinden gelmek amacıyla geliştirilen S-means [116], X-means, G-means ve PG-means metotlarının aksine veriyi Gauss karışım modeli olarak ele almaz. S-means ilk olarak küçük sayıda K kümesi ile başlar (örn. K=1). K sayıda küme random bir şekilde üretilir veya veriden seçilir. İkinci adımda her bir veri elemanından tüm kümelere olan radyal temelli fonksiyon (RBF) kernel değerleri hesaplanır ve veri elemanları en yüksek RBF kernel değerine sahip olduğu kümeye belirli bir eşik değerden yüksek olmak şartıyla atanır. Aksi takdirde bir kümeye atanamayan veri elemanı için yeni bir küme oluşturulur. Son adımda her küme merkezi kümenin bünyesinde bulunan veri elemanlarının ortalaması alınarak güncellenir ve boş bir küme ortaya çıkması durumunda da o küme silinir. Bu prosedür yeni bir küme oluşmayıncaya veya küme merkezleri güncellenmeyinceye kadar tekrarlanır. S-means'in performansı başlangıç koşullarına ve atama için belirlenen eşik değerine bağlıdır.

Kısıtlanmış doğrulama (constrained validation (CV)) kümeleme metodu [117] küme içindeki homojenliği temel alır. CV kümeleme metodu bir küme ile başlar ve kümelerin homojen olup olmadıklarını homojenlik testi ile kontrol eder. Eğer bütün kümeler homojenlik prensibine uyuyorsa, metot mevcut kümeleri döndürür ve çalışmayı durdurur. Aksi takdirde homojen olmayan kümelerden küme içindeki uzaklık kriteri değerleri en iyi olanlar bölünmek için seçilir. Bu prosedür maksimum küme sayısına erişinceye kadar veya bütün homojen kümeler bulununcaya kadar devam eder. CV kümeleme metodunda kümenin homojen olup olmadığını belirleyen eşik değerin kümeleme performansında önemli bir etkisi vardır.

Tang ve arkadaşları [117] S-means, CV kümeleme ve G-means metotlarını LIGO zaman serisi üzerinde test etmiştir. Sonuçlar CV kümeleme metodunun doğru küme sayısını bulduğunu ve X-means ve G-means yöntemlerinden daha iyi performans sağladığını göstermiştir. Sonuçlar ayrıca S-means'in performansının kullanıcı tarafından tanımlanan eşik değerine çok bağlı olduğunu ve G-means'in benzer homojen kümeleri bir araya toplamada başarısız olduğunu da göstermiştir.

Dip-means [118] bir verideki kümeleri tespit etmek için Dip-dist istatistiği kullanır. Ardından her bir küme için unimodality skoru hesaplanır. Eğer bir küme için elde edilen değer tanımlanmış eşik değerden büyükse o küme 2-means ile iki parçaya ayrılır. Bu prosedür küme sayısında değişiklik olmayıncaya kadar tekrarlanır. Deneysel sonuçlar neticesinde Dip-means'in X-means, G-means ve PG-means'ten hem sentetik hem de gerçek dünya verilerinde daha iyi performans gösterdiği görülmüştür. Ancak Dip-means'in performansı önemlilik derecesi ve eşik değeri olmak üzere iki parametreye bağlıdır.

MLBG metodu [119] *K* tanımlanmış kümeyle başlar ve en yüksek küme içi uzaklığa sahip kümeyi iki kümeye bölerek devam eder. Bir kümeyi bölmek için kontrol noktalarının LBG metodu ile tanımlanması gerekmektedir. Ancak MLBG'nin performansını ölçecek herhangi bir karşılaştırmalı çalışma yapılmamıştır.

Zhang ve arkadaşları [120] aglomeratif FCM ve komşu paylaşım şemasının (neighbors sharing scheme) birleştirilmiş bir versiyonunu (NSS-AKmeans) önermiştir. Komşu paylaşım şeması, veri elemanları arasındaki yoğunluğu ölçmeyi sağlayan aglomerasyon enerjisi ve yerel yoğunluk bağlantısını ölçmeyi sağlayan komşu ölçeklendirme faktörü (neighbor scaling factor) temellerine dayanır. NSS-AKmeans yönteminde önce komşu paylaşım şeması ile küme merkezleri elde edilir. Elde edilen küme merkezlerinin benzer olanları aglomeratif FCM ile birleştirilerek optimal küme sayısı belirlenir. Ancak NSS-AKmeans yönteminin literatürdeki diğer yöntemler ile karşılaştırılmaması ve sadece sentetik veriler üzerinde denenmesi sebebiyle performansının sağlıklı bir şekilde değerlendirilmesi mümkün değildir.

X-means, G-means ve benzeri yöntemlerin uzaktan algılama uygulamaları için çok uygun olmaması sebebiyle Koonsanit ve arkadaşları [121] parametresiz K-means yöntemi önermiştir. K-means'in geliştirilmesi şu temel adımlarla sağlanmıştır: 1)

resimdeki bölgelerin eş-oluşum matrisi ile otomatik belirlenmesi ve 2) maxima fonksiyonu ile eş-oluşum matrisindeki dönüm noktalarının sayılması. Önerilen yöntemin X-means ve ISODATA'dan daha iyi olduğu görülmüştür.

He ve arkadaşları [122] K-means, Gauss karışım model vb. bir kümeleme algoritmasına ihtiyaç duymayan yöntem geliştirmiştir. Bu yöntem temel olarak özdeğerlerdeki (eigenvalues) boşluğun saptanması prensibine bağlı olarak çalışır. Gupta ve arkadaşları [123] beklenti büyütme ve Gauss dağılıma bağlı bir yöntem önermiştir. Ancak bu yöntem sadece Gauss dağılıma sahip kümeleri saptayabilmektedir.

Shahbaba ve Beheshti [124] küme içi uzaklığın minimizasyonuna bağlı bir yöntem (MACE-means) önermiştir. MACE-means yönteminin etkinliği X-means, G-means ve PG-means gibi birçok yöntemle birçok gerçek dünya problemi üzerinde karşılaştırılarak ortaya konulmuştur.

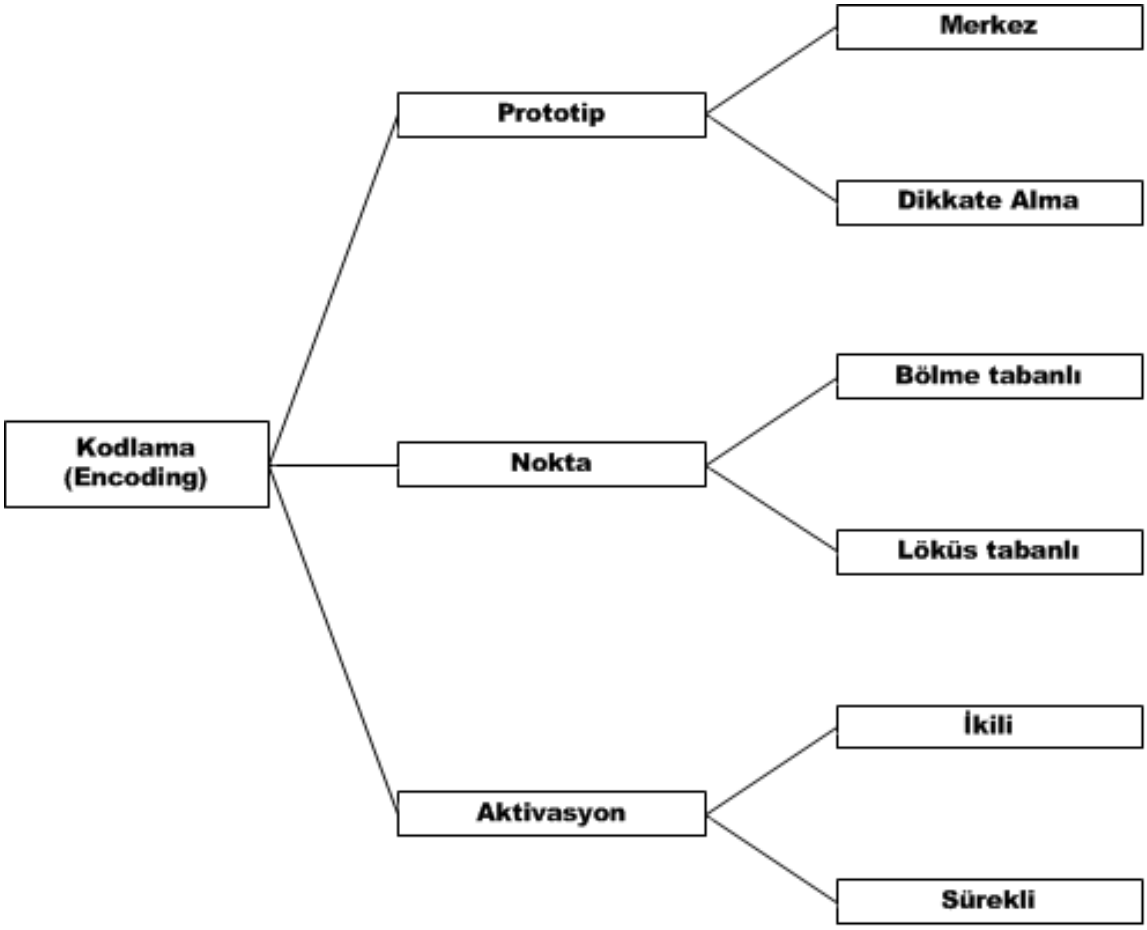
Liang ve arkadaşları [125] Renyi entropi ve tamamlayıcı entropiyi birlikte entegre ederek hem sürekli hem de ayrık özniteliği bünyesinde barındıran veriler için bir mekanizma önermiştir. Bu mekanizma yüksek sayıdaki kümeyle başlar, iteratif olarak en kötü kümeyi k-propotip algoritması ile bularak elimine eder ve açıkta kalan veri elemanlarını geriye kalan kümelere dağıtır. Küme doğrulama için de kategori yarar fonksiyonuna [126] bağlı bir indeks geliştirilmiştir. Ancak bu mekanizma literatürdeki hiçbir yöntem ile karşılaştırılmamıştır.

1.4.3. Evrimsel Hesaplama Tabanlı Yöntemler

Evrimsel hesaplama teknikleri, kabul edilebilir zamanda yaklaşık optimal çözümler sağlaması sebebiyle NP-hard problemler için etkili bir yol olarak kabul edilmektedir [62]. Kümeleme de bir çeşit NP-hard problem olarak kabul edilmesi nedeniyle evrimsel hesaplama teknikleri kümeleme problemlerini çözmek için sıklıkla uygulanmıştır. Optimal veya optimum küme sayısının belirlenmesi optimizasyon problemi olarak değerlendirilirse, doğrulama indeksi bir evrimsel hesaplama tekniği tarafından minimize veya maksimize edilen amaç fonksiyonunu ifade etmektedir.

Bir evrimsel tekniğin kümeleme problemlerine uygulanabilmesi için, öncelikli olarak kodlama (encoding) birimleri vasıtasıyla belirli sayıdaki olası çözümlerin

popülasyondaki bireylerde (örn. yiyecek kaynaklarında) temsil edilmesi gerekmektedir. Mukhopadhyay ve arkadaşları [63] kodlama birimlerini prototip ve nokta (point) tabanlı olmak üzere ikiye ayırmıştır. Prototip tabanlı kodlama biriminde bir birey, küme merkezleri ve temsilci nesneleri (medoids) temsil eder. Nokta tabanlı encoding şemasında bir birey veri elemanlarının sınıf veya küme etiketlerini kodlar. Ancak bu tip sınıflandırma, küme merkezleri ve aktivasyon kodlarının beraber bulunduğu aktivasyon tabanlı kodlama birimlerini içermez. Bu nedenle birey temsili için yeni bir genişletilmiş sınıflandırma önerilmiş ve Şekil 1.3'te sunulmuştur.



Şekil 1.3. Kodlama birimleri sınıflandırma ağacı.

Şekil 1.3'teki kodlama birimleri aşağıda izah edilmiştir:

1. Merkez (centroid) tabanlı kodlama: Bireyler d boyutlu bir uzaydaki küme merkezlerinin reel pozisyon değerlerini içerir. d boyutlu uzayda K küme sayısı için birey uzunluğu $K \times d$ kadardır. Küme sayısının belirlendiği uygulamalarda

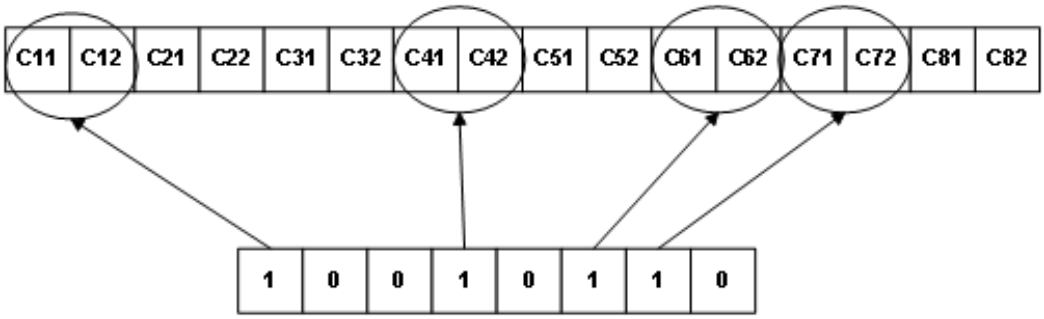
başlangıç popülasyonundaki her bir bireyin uzunluğu K_{min} ve K_{max} arasında rasgele üretilen K değerine bağlı olarak değişir. Bireylerin uzunlukları bu temsil biriminde genellikle fazla uzun olmamaktadır. Dolayısıyla bireyler üzerinde uygulanan evrimsel operatörlerin hesaplama performansı kabul edilebilir seviyede kalmaktadır. Ancak, bu temsil biriminde bireylerin farklı uzunluklarda kodlanması sebebiyle standart evrimsel hesaplama tekniği ile bu temsil birimini birlikte kullanmak mümkün değildir. Bu temsile yönelik özel transformasyon operatörlerinin tasarımına ihtiyaç vardır.

2. Dikkate alma (don't care) tabanlı kodlama: Bireyler küme pozisyonları ve dikkate alma sembolü # ile temsil edilir. D boyutlu ve N elemanlı bir veri için rasgele K tane üretilmiş küme merkezi bireydeki pozisyonlara rasgele dağıtılır ve geri kalan pozisyonlar # ile temsil edilir. Bir bireyin K=4, D=16 ve d=2 için pozisyonlardaki merkez ve # sembollerinin dağılımını gösteren örnek Şekil 1.4'te verilmiştir. Bu kodlama birimi merkez tabanlı kodlama birimine göre fazla yaygın değildir.

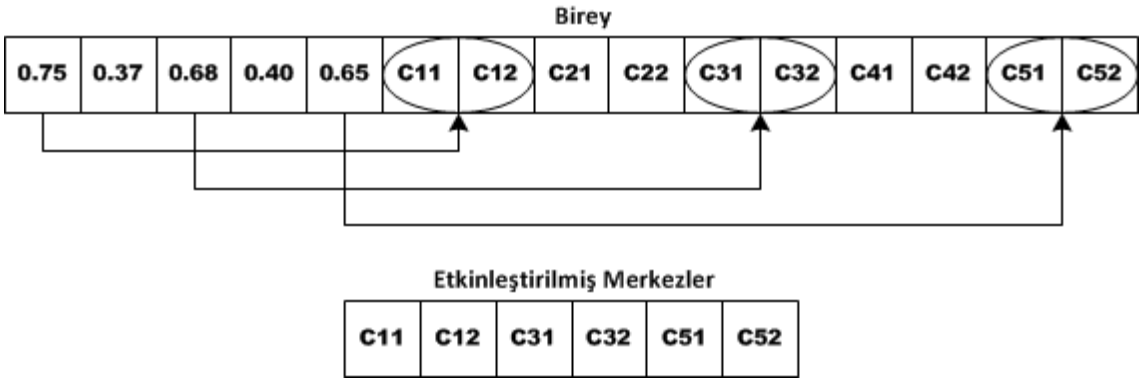
C11	C12	C21	C22	#	#	C31	C32	#	#	#	#	C41	C42	#	#

Şekil 1.4. Dikkate alma tabanlı temsile bir örnek.

3. Bölme (partition) tabanlı kodlama: Bireyin uzunluğu verideki eleman sayısına eşittir ve her pozisyon ilgili veri elemanının ait olduğu küme etiketini ifade eder. Bu kodlama birimi rahatlıkla kategorik ve sürekli verilere uygulanabilir. Ayrıca başlangıç koşullarına da bağlı değildir. Ancak büyük boyutlu verilerde hesaplama karmaşıklığını artırmaktadır. Bununla beraber bu birimde kodlanan bireylerin gelişimi için özel operatörlere ihtiyaç vardır.
4. Löküs (locus) tabanlı kodlama: Her birey N elemanlı veri için N boyutlu bir vektördür ve her pozisyon 1'den N'e kadar tamsayı değerler alır. Bir i. pozisyonun j. değeri, i ve j veri elemanları arasında bir bağ olduğunu ifade etmektedir. Bu sayede veri elemanları aynı kümeye atanır ve bir grafik oluşturulur. Bu kodlama biriminde küme sayısı otomatik olarak bireylerin içinde saklanır. Ancak bu temsil şeması büyük boyutlu verilerde yüksek hesaplama karmaşıklığı gerektirmektedir.



Şekil 1.5. İkili aktivasyon tabanlı temsile bir örnek.



Şekil 1.6. Sürekli aktivasyon tabanlı temsile bir örnek.

5. İkili (binary) aktivasyon tabanlı kodlama: Her birey, dışarıda tanımlanmış bir küme setinin 0 ve 1'lerden oluşan aktivasyon kodlarını temsil eder. Dolayısıyla bir bireyin uzunluğu küme setindeki merkez sayısı kadardır. K_{max}=8 ve d=2 için küme merkezlerinin nasıl etkinleştirileceğine dair bir örnek Şekil 1.5'te sunulmuştur. Ancak bu kodlama biriminin performansı tanımlanmış küme setine bağlıdır; çünkü bireylerin gelişimi esnasında küme merkezlerinin güncellenme şansı yoktur.
6. Sürekli (continuous) aktivasyon tabanlı kodlama: Bireyler d boyutlu küme merkezleri ve K_{max} tanımlı küme sayısı için $K_{max}+K_{max}\times d$ boyutlu vektörlerdir. İlk K_{max} tane pozisyon 0 ve 1 aralığındaki aktivasyon kodlarını ve geri kalan pozisyonlar d boyutlu K_{max} tane küme merkezini temsil eder. K_{max}=5 ve n=2 için küme merkezlerinin nasıl etkinleştirileceğine dair örnek Şekil 1.6'da sunulmuştur. Bu şema herhangi bir evrimsel hesaplama tekniği ile kullanılabilir ve birçok doğrulama indeksi ile doğru şekilde çalışabilir. Zaman karmaşıklığı d ve K_{max} değerlerine bağlıdır.

Popülasyon bireylerini temsil edecek bir kodlama biriminin belirlenmesinden sonra kodlama birimine uygun evrimsel hesaplama tekniğinin seçilmesi ve araştırma operatörlerinin dizayn edilmesi gereklidir. Son olarak da kullanılan veya dizayn edilen evrimsel hesaplama tekniğine uygun bir optimize edilecek doğrulama indeksinin seçilmesi gereklidir. Doğrulama indeksleri ile ilgili detaylı bilgi Bölüm 1.1.3'te sunulmuştur.

Evrimsel hesaplama tabanlı yöntemler tek amaçlı ve çok amaçlı olmak üzere iki sınıfta incelenmiş ve literatürde olduğu gibi otomatik kümeleme yöntemleri olarak nitelendirilmiştir.

1.4.3.1. Tek Amaçlı Yöntemler

Bandyopadhyay ve Maulik [127] variable string length genetik algoritmasını (VGA) [128] kullanarak DBI, Dunn, genelleştirilmiş Dunn ve *I*-index doğrulama indekslerini içeren karşılaştırmalı bir çalışma sunmuştur. Sonuçlar *I*-index doğrulama indeksinin optimal küme sayısının saptanmasında diğelerinden daha iyi olduğunu göstermiştir. VGA otomatik kümelemede merkez tabanlı kodlama kullanılmış ve çaprazlama iki pozisyonla sınırlandırılmıştır. Xie-Beni indeksi amaç fonksiyonu olarak kullanılarak VGA otomatik kümeleme yönteminin fuzzy versiyonu (FVGA) [129] da önerilmiş ve uzaktan algılama resimlerinin kümelenmesinde kullanılmıştır. Bir diğer çalışmada Bandyopadhyay ve Maulik [12] dikkate alma kodlama birimi ve DBI indeksini GA ile entegre ederek bir otomatik kümeleme yöntemi (GCUK) önermiştir. GCUK yönteminde tek nokta çaprazlama yapılır, mutasyon VGA'daki gibi uygulanır ve en iyi birey sonraki jenerasyonlarda temsil edilmek üzere saklanır.

Hruschka ve Ebecken [130] standart genetik algoritma (GA) [131] yapısının gen anlatım (gene-expression) verisine uygun olmadığını öne sürmüştür. Bundan dolayı Silhoutte indeksi ve bölme tabanlı kodlama biriminin farklı bir çeşidini kullanarak GA tabanlı bir otomatik kümeleme yöntemi (CGA) [132] önermiştir. N tane elemandan oluşan veri için her bir birey N tane verinin küme etiketini ve son pozisyonda da toplam küme sayısını içerir. Örneğin, 10 eleman ve 4 küme '1132223344 4' formunda temsil edilebilir. Çaprazlama için G1 ve G2 olarak tanımlı iki birey rasgele seçilir. Ardından G1'den rasgele seçilen pozisyonlar (genlerin) G2'ye kopyalanır. Kopyalama işleminin dışında kalan pozisyonlar '0' olarak atanır ve '0' olarak belirlenen pozisyonlar en

yakınlarındaki kümelere atanır. Bu şekilde çocuk birey üretilir. Aynı işlem G2'den G1'e kopyalama ile tekrarlanır. Mutasyon için iki yaklaşım kullanılmıştır. İlk yaklaşımda ikiden fazla kümeye sahip bireyden rasgele seçilen bir küme silinir ve açığa çıkan elemanlar en yakın kümelere atanır. Diğer yaklaşımda bir küme, merkezine yakın bulunan elemanlar ile uzak bulunan elemanların yer alacağı iki ayrı kümeye bölünür. Seleksiyon için rulet tekerleği uygulanır ve en iyi birey bir sonraki jenerasyonda temsil edilir.

Hruschka ve arkadaşları [133] CGA'yı iyileştirerek yeni modeller önermiştir: 1) CGA-II: K-means algoritmasının CGA'ya entegrasyonu, 2) CGA-III: İyileştirilmiş amaç fonksiyonun CGA'ya entegrasyonu ve 3) EAC: CGA'dan çaprazlama operatörünün çıkarılması ve çok yönlü mutasyon şemalarının CGA'ya entegrasyonu. Alves ve arkadaşları [134] CGA'nın performansını şu yollarla iyileştirmiştir: 1) mutasyon için seleksiyon mekanizmasının dizaynı, 2) belli bir çevrimden sonra farklı bir mutasyon kullanımı, 3) Rand indeksinin amaç fonksiyonu olarak kullanılması. Sonuçlara göre EAC'nin iyileştirilmiş versiyonları standart EAC'den daha iyi performans göstermiştir. Ancak yöntemler gerçek dünya problemleri üzerinde test edilmemiştir. Buna ek olarak, kullanılan bölme tabanlı kodlama birimi sebebiyle büyük hacimli verilerde bu yöntemlerin hesaplama karmaşıklığı yüksektir.

Paralel GA bir sistemin çalışma zamanını kısaltabilir ve erken yakınsamayı engeller. Fakat paralel GA'nın yerel arama özelliğinde bazı problemler vardır. Paralel GA'nın yerel arama özelliğini geliştirmek amacıyla K-means ve paralel GA'nin birleştirilmiş bir versiyonu (PGAK) [135] önerilmiştir. PGAK otomatik kümeleme için merkez tabanlı kodlama ve küme içi uzaklığı kullanır. PGAK'da çaprazlama ise bir bireyden bir bölümü rasgele çıkarıp başka bir bireye eklemekle gerçekleştirilir. Ancak PGAK diğer mevcut otomatik kümeleme yöntemleriyle karşılaştırılmamıştır.

Lai [136] hiyerarşik GA (HGA) [137], DBI indeks ve sürekli aktivasyon tabanlı kodlama birimini birleştirerek bir otomatik kümeleme yöntemi önermiştir. Birey çiftlerinin aktivasyon değerleri arasında eşiklemeye bağlı olarak tekdüze (uniform) çaprazlama uygulanır. Çaprazlama uygulanan bireylerin aktivasyon değerlerinin temsil ettiği merkezlerde de bu işlem gerçekleşir. Fakat önerilen yöntem temel bir veri olan

Iris'te doğru küme sayısını bulamamış ve yöntemi de içine alan herhangi bir karşılaştırmalı çalışma sunulmamıştır.

Lin ve arkadaşları [138] farklı bir kodlama birimi ile GA tabanlı bir kümeleme yöntemi önermiştir. Yöntemde her bireyin uzunluğu veri uzunluğuna eşittir ve küme merkezleri veriden direkt olarak seçilir. Oluşturulan tablo yardımıyla veri elemanları arasındaki mesafeler ezberlenir ve bütün işlem boyunca sadece bir kez hesaplanır. Ancak önerilen yöntemin hesaplama karmaşıklığı, kullanılan kodlama birimi sebebiyle artabilmektedir.

Liu ve arkadaşları [139] DBI indeks ve merkez tabanlı kodlama birimini kullanarak GA tabanlı bir otomatik kümeleme yöntemi (AGCUK) geliştirmiştir. Bu yöntemde farklı mutasyon ve seleksiyon operatörleri kullanılarak popülasyon bireyleri arasında çeşitliliğin sağlanması amaçlanmıştır. AGCUK yönteminin etkinliği Lai [136], Lin [138] ve GCUK [12] yöntemleri ile karşılaştırılarak ortaya konulmuştur. Ancak AGCUK'un hesaplama karmaşıklığı Lin ve arkadaşlarının yönteminden daha yüksektir ve deneyler gerçek dünya verileri üzerinde gerçekleştirilmemiştir.

Sheng ve arkadaşları [140] DBI, Dunn, genelleştirilmiş Dunn, CH, *I*-index ve Silhoutte indekslerinin ağırlıklandırılmış toplamlarını yeni bir indeks (WSVF) olarak sunmuştur. WSVF indeksinin optimizasyonu hibrit niching GA (HNGA) ve merkez tabanlı kodlama birimi ile sağlanmıştır. HNGA'da niching metot K-means ile hibridize edilerek hesaplama zamanı kısaltılmıştır. Önerilen yöntem bazı sentetik ve gerçek dünya verileri üzerinde CGA ve VGA ile karşılaştırılmıştır. Ancak önerilen yöntem Iris gibi temel bir veride bile doğru küme sayısını bulamamıştır. Leon ve arkadaşları [141] danışmasız hücre kümeleme (unsupervised niche clustering) algoritması ve hibrit adaptif evrimsel algoritma tabanlı bir GA kümeleme yöntemi (ECSAGO) önermiştir. ECSAGO bilinenlerden farklı bir kodlama birimi kullanmaktadır.

Doğrusal, dairesel ve çok köşeli gibi birçok farklı geometrik şekillerle (kümelerle) ilgilenen uzaklık ölçüleri mevcuttur. Ancak mevcut ölçüler simetrik şekilleri saptamakta yetersiz kalabilmektedir. Bundan dolayı Bandyopadhyay ve Saha [142] nokta simetrik uzaklık tabanlı doğrulama indeksi (Sym-index) [143] ve VGA algoritmasını kullanarak bir otomatik kümeleme yöntemi (VGAPS) geliştirmiştir. Nokta simetri uzaklığının hesaplama kompleksliği azaltmak için Kd-ağaç tabanlı komşu arama mekanizması uygulanmıştır. VGAPS yönteminin etkinliği HNGA ve GCUK yöntemleri ile

karşılaştırılarak sentetik ve gerçek dünya verileri üzerinde gösterilmiştir. Sym-index uzaktan algılama resimlerini kümelemek için de kullanılmıştır [144]. Bir diğer çalışmada Sym-index'in fuzzy versiyonu (FSym-index) geliştirilmiş ve VGA algoritması ile optimize edilmiştir [145].

He ve Tan [146] merkez tabanlı kodlama birimi ve CH indeksini kullanarak iki aşamalı bir genetik kümeleme yöntemi (TGCA) önermiştir. TCGA yönteminde seleksiyon ve çaprazlama olasılıkları iki aşamalı seleksiyon ve çaprazlama operatörleri ile dinamik olarak belirlenmekte ve başlangıçtaki bireylerin üretilmesi de farklı bir öznitelik bölme metodu ile yapılmaktadır. Çaprazlama sadece aynı pozisyon sayısına sahip bireyler arasında sınırlı tutulmuştur. Bazı sentetik ve gerçek dünya verileri üzerinde elde edilen sonuçlar neticesinde TCGA yönteminin hiyerarşik K-means, genetik K-means ve otomatik spektral kümeleme yöntemlerden daha iyi olduğu görülmüştür.

Agustin Blas ve arkadaşları [147] gruplama tabanlı GA (GGA) [148, 149] ile otomatik kümeleme yöntemi geliştirmiştir. GGA yönteminde bölme tabanlı kodlama birimi ve DBI indeks kullanılmıştır. Çaprazlama üretilmiş bireylere göre düzenlemiş ve mutasyon kümelerin birleştirilmesi ve bölünmesiyle gerçekleştirilmiştir. Alt gruplar arasında paralel olarak yer değiştirmeyi sağlayan bir model de GGA'nın performansını arttırmak için geliştirilmiştir. Ancak GGA Iris, Wine vb. bilinen gerçek dünya problemleri üzerinde test edilmemiş ve kullanıldığı temsil birimi büyük hacimli verilerde zaman karmaşıklığını artırmaktadır. Bir diğer GGA otomatik kümeleme yöntemi [150] de DBI indeksinin fuzzy versiyonu kullanılarak ve kodlama biriminin buna uygun olarak üyelik matrisi ve grup elemanlarını kapsayacak şekilde entegrasyonu sağlanarak geliştirilmiştir. Diğer GA tabanlı otomatik kümeleme yöntemlerine [151-156]'dan erişilebilir.

Omran ve arkadaşları [157] ikili aktivasyon tabanlı kodlama birimi ve VI indeksi kullanarak ikili parçacık sürü optimizasyon (PSO) tabanlı kümeleme yöntemi (DCPSO) geliştirmiştir. DCPSO'da merkezler veriden rasgele seçilerek bir merkez seti (*S*) oluşturulur. 0 ve 1 değerlerinden oluşan popülasyon bireyleri *S* setindeki merkezlerin aktivasyon değerlerini ifade etmektedir. Öncelikle BPSO maksimum çevrim sayısı sağlanıncaya kadar koşulur. Elde edilen en iyi parçacığı temsil eden merkez alt setine (*Sbest*) K-means uygulanarak başlangıç koşullarının sebep olabileceği olumsuz etkiler

giderilmeye çalışılır. Bu işlemin akabinde *S* setinin ilgili pozisyonlarına *Sbest* setindeki güncellenmiş merkezler yerleştirilir. Bu prosedür güncellenmiş *S* seti ile durdurma kriteri sağlanınca kadar tekrarlanır. Elde edilen en iyi popülasyon bireyini temsil eden alt merkez seti optimal çözüm olarak kabul edilir. DCPSO yöntemi SNOB ve özdüzenleyici haritalar (self organizing maps) [158] yöntemlerinden daha iyi sonuç elde etmiştir.

Das ve arkadaşları [159] sürekli aktivasyon tabanlı kodlama birimi ve Xie-Beni indeksi kullanarak multi-elitist PSO (MEPSO) tabanlı otomatik kümeleme yöntemini önermiştir. MEPSO kümeleme yöntemi FVGA yönteminden hem optimal küme sayısının elde edilmesi hem de hesaplama karmaşıklığı açısından daha iyi sonuç vermiştir. Ancak yöntem az sayıda problem üzerinde test edilmiştir.

Ouadfel ve arkadaşları [160] mutasyon operatörleriyle geliştirilmiş bir PSO tabanlı otomatik kümeleme yöntemi (ACMPSO) önermiştir. ACMPSO'da sürekli aktivasyon tabanlı kodlama birimi ve CH indeksi kullanılmıştır. Elde edilen sonuçlar neticesinde ACMPSO yönteminin VGA ve DCPSO yöntemlerinden daha iyi sonuç verdiği görülmüştür. Kuo ve arkadaşları [161] PSO ve GA algoritmalarını paralelleştirerek (DCPG) küme sayısını saptamaya çalışmıştır. Genel olarak PSO ve GA birbirinden bağımsız eş zamanlı çalıştırılarak ortak popülasyon bireyleri geliştirilmekte ve her çevrim sonunda iki algoritmanın da elde ettiği en iyi bireyler bir sonraki jenerasyona aktarılmaktadır. DCPG yönteminde ikili aktivasyon tabanlı kodlama birimi ve küme içi uzaklık kullanılmıştır. Sonuçlara göre, DCPG kümeleme yöntemi ACMPSO ve DCPSO yöntemlerinden daha iyi performans göstermiştir. Kao ve Lee [162], CPSO [163] ve K-means algoritmalarını birleştirerek bir otomatik kümeleme yöntemi (KCPSO) önermiştir. Kuo ve Zulvia [164] sürekli aktivasyon tabanlı kodlama birimi ve VI indeksi kullanarak bir PSO tabanlı otomatik kümeleme yöntemi geliştirmiştir. Geliştirilen yöntem DCPSO, DCPG ve DCGA yöntemleriyle Iris, Wine, Glass ve Aggreation gibi bilinen veriler üzerinde karşılaştırılmıştır.

Das ve arkadaşları [13, 159, 165, 166] diferansiyel gelişim (differential evolution (DE)) tabanlı otomatik kümeleme yöntemleri (ACDE ve fuzzy versiyonu AFDE) önermiştir. ACDE ve AFDE yöntemlerinde *F* faktörü ve çaprazlama oranı adaptif olarak belirlenmektedir. Popülasyon bireylerinin temsili sürekli aktivasyon tabanlı kodlama

birimi ile sağlanmaktadır. Amaç fonksiyonu olarak DBI ve CH indeksler ACDE yönteminde kullanılmıştır. AFDE yönteminde ise Xie-Beni indeksi amaç fonksiyonu olarak kullanılmıştır. ACDE yönteminin başarısı DCPSO, standart DE ve GCUK yöntemleriyle karşılaştırılarak ortaya konmuştur.

Das ve arkadaşları [167] Öklid uzaklığını kullanan birçok yöntemin yüksek spektral ve lineer ayrılabilen verilerde iyi sonuçlara erişemediğini ve çakışan kümelerin olduğu ortamlarda performansının düştüğünü ileri sürmüştür. Bu problemin üstesinden gelmek amacıyla Das ve arkadaşları [168, 169] kernel uzaklık metriğini kullanarak XB indeksini modifiye etmiştir. Kernel tabanlı XB fonksiyonun minimizasyonu komşu mutasyon stratejine bağlı çalışan DE algoritması (DEGL) ile sağlanmıştır. Das ve arkadaşları [170] CS indeksi de kernel uzaklık metriği ile modifiye etmiştir. Kernel tabanlı CS indeksi [171]'de de kullanılmıştır.

Maulik ve Saha [172] küresel ve yerel en iyi bireylerle geliştirilen DE modelini (MoDEAFC) uzaktan algılama resimlerinin otomatik kümelenmesinde kullanmıştır. MoDEAFC yönteminin etkinliği AFDE ve FVGA yöntemleri ile karşılaştırılarak ortaya konmuştur. Saha ve arkadaşları [173] MoDEAFC modeline benzer bir DE modelini (IDEAFC) gen analizi için uygulamıştır. IDEAFC, ModeAFC'den farklı olarak SVM sınıflandırıcı kullanır. Rui ve arkadaşları [174] DE ve PSO'nun hibrit versiyonunu (DEPSO) kullanarak CH, DBI, Dunn, CS, *I*-index ve Silhouette indekslerinin içinde olduğu toplam sekiz indeksin karşılaştırmalı bir çalışmasını ortaya koymuştur. Sonuçlar Silhouette indeksin en iyi performansı elde ettiğini göstermiştir.

Su ve arkadaşları [14] merkez tabanlı kodlama birimi ve genelleştirilmiş bir mutasyon sistemi kullanarak modifiye yapay arı koloni (ABC) tabanlı otomatik kümeleme yöntemi (VABC-FCM) geliştirmiştir. VABC'de kullanılan mutasyon sistemi üç çeşit operatör içermektedir: 1) seçilen bireyin uzunluğunun küresel en iyi bireyin uzunluğundan uzun olması durumunda bireyden rasgele seçilen pozisyonun silinmesi, 2) seçilen bireyin uzunluğunun küresel en iyi bireyin uzunluğundan kısa olması durumunda veri setinden rasgele seçilen merkezin bireye eklenmesi ve 3) seçilen bireyin uzunluğu küresel en iyi bireyin uzunluğuna eşitse değişiklik yapılmadan bireyin sonraki jenerasyona eklenmesi. VABC-FCM yönteminin performansı VGA, PSO ve EA yöntemleri ile karşılaştırılarak değerlendirilmiş ve elde edilen sonuçlar neticesinde

VABC-FCM en iyi performansı sağlamıştır. Ancak deneysel çalışmalarda kullanılan veriler çoğunlukla iki veya üç küme içermektedir. Büyük küme sayısı içeren veriler performans analizinde kullanılmamıştır.

Pan ve Cheng [175] ikili aktivasyon kodlama birimi ve PBM indeksini [176] kullanarak gelişimsel tabu arama kümeleme yöntemini (ETSA) önermiştir. Kumar ve arkadaşları [177] merkez aktivasyonu için sabit bir eşikleme değerinin bütün veri tiplerinde kullanılmasının uygun olmadığı ve eşikleme değerinin verinin tipine göre belirlenmesi gerektiğini belirtmiştir. Bundan dolayı verilen verinin varyansı eşikleme değeri olarak belirlenmiştir.

Liu ve arkadaşları [178] sürekli aktivasyon tabanlı kodlama biriminin uzunluğunu kısaltmak için üç tane adaptif kodlama birimi önermiştir. İlk birimde merkezler ve ilgili aktivasyon kodları bireyde beraber temsil edilir; fakat merkezler, aktivasyon kodları üzerinden hesaplanır. İkinci şemada aktivasyon kodları merkezlerin ilk pozisyonlarındaki değerler ile temsil edilir. Son şemada bireydeki ilk pozisyondan itibaren $[L\times(K_{max}\text{-}2)+2]$ kadar merkez etkinleştirilir; burada L [0,1] arasında random bir sayıdır. Kashan ve arkadaşları [179] grup evrimsel stratejiler (GES) [180] tabanlı bir fuzzy otomatik kümeleme yöntemi (FuzzyGES) önermiştir. FuzzGES yönteminde Xie-Beni indeks ve bölme tabanlı kodlama biriminin farklı bir çeşidi kullanılmıştır. Ancak bölme tabanlı kodlama birimi büyük hacimli verilerde yüksek zaman karmaşıklığına sebep olmaktadır. Alia ve arkadaşları [181] harmoni arama tabanlı bir fuzzy ototmatik kümeleme yöntemi (DCHS) önermiştir. DCHS, dikkate alma tabanlı kodlama birimini ve PBMF indeksini kullanır. Fakat DCHS optimal küme sayısını bulmada istikrarlı bir yöntem değildir. Das ve arkadaşları [167] bakteri gelişim algoritmasını optimal küme sayısının tespitinde kullanmıştır. Liu ve arkadaşları [182] yapay bağışıklık algoritması tabanlı bir otomatik kümeleme yöntemi (DLSIAC) geliştirmiştir. DLSIAC, dinamik yerel araştırma operatörleriyle desteklenmiştir. DLSIAC yönteminde sürekli aktivasyon tabanlı kodlama birimi ve PBM indeksi kullanılmıştır. DLSIAC yönteminin performansı çeşitli veri setleri üzerinde TGCA ve MoDEAFC gibi bilinen yöntemlerle karşılaştırılarak değerlendirilmiştir. Yerçekimsel arama algoritması ve bilind naked mole-rats algoritması [177, 183] gibi yeni nesil evrimsel hesaplama teknikleri de küme sayısının saptanması amacıyla uygulanmaya başlanmıştır.

1.4.3.2. Çok Amaçlı Yöntemler

Küme sayısının saptanması ile ilgilenen birçok çalışma tek amaçlıdır. Ancak tek bir amaç bazı gerçek dünya verilerinde yeterli gelmeyebilmektedir. Bu sebepten ötürü araştırmacılar birden fazla doğrulama indeksini eş zamanlı kullanarak küme sayısı saptanması problemini çok amaçlı optimizasyon problem olarak ele almıştır. Handl ve Knowles [184] bölge seçme tabanlı çok amaçlı optimizasyon algoritması (PESA-II) [185] ve löküs tabanlı kodlama birimine bağlı bir otomatik kümeleme yöntemi (MOCK) önermiştir. MOCK, küme varyansı ve küme bağlantısı olmak üzere iki amaç fonksiyonu içerir. Popülasyonu oluşturmak için minimum örten ağaç (minimum spanning tree) ve işlem sonucunda elde edilen domine olmayan (non-dominated) çözümlerden bir çözüm seçmek için gap statistics kullanılır. Önerilen yöntem tek amaçlı yöntemlerden daha iyi performans sağlamasına karşın çakışan kümelerin saptanmasında doğru çalışmayabilmektedir. Bunun yanında löküs tabanlı kodlama birimi büyük hacimli verilerde yüksek hesaplama karmaşıklığına sebep olmaktadır.

Bandyophadhyay ve arkadaşları [186] merkez tabanlı kodlama birimi ve çok amaçlı bir GA (NSGA-II) [187] algoritması kullanarak uzaktan algılama resimleri için çok amaçlı genetik kümeleme yöntemi önermiştir. Je ve Xie-Beni indeksleri eş zamanlı olarak NSGA-II algoritması ile optimize edilmiştir. Bireylerin temsili ise merkez tabanlı kodlama birimi ile sağlanmıştır. Sonuçlara göre, önerilen çok amaçlı yöntem standart GA ve FCM yöntemlerinden daha iyi performans göstermiştir. Mukhopadhyay ve Maulik [188] NSGA-II tabanlı bu yöntemi SVM sınıflandırıcı ile birleştirerek uzaktan algılama resimlerindeki yüksek belirginliğe sahip noktaları elde etmiştir. Je ve Xie-Beni indeksleri eş zamanlı olarak hibrit çok amaçlı GA ve DE algoritması (GADE) ile de optimize edilmiştir [189]. GADE yönteminde bireylerin temsili için sürekli aktivasyon tabanlı kodlama birimi ve domine olmayan çözümlerden bir çözüm seçmek için gap statistics yöntemi kullanılmıştır. Dutta ve arkadaşları [190] küme içi ve kümeler arası uzaklık kriterleri ve merkez tabanlı kodlama birimini kullanılarak çok amaçlı genetik algoritma (MOGA) kümeleme yöntemi önermiştir. Ancak MOGA kümeleme yöntemi diğer mevcut yöntemlerle karşılaştırılmamıştır.

Suresh ve arkadaşları [191, 192] standart Je kriterinin küme sayısı belirtilmeden minimize edilmesinin küme sayısını artırma eğilimi gösterdiğini öne sürmüştür. Bundan

dolayı sınırlamalı Je kriterini önermiştir. Sınırlamalı Je ve Xie-Beni indeksleri eş zamanlı olarak çok amaçlı DE algoritmaları MODE ve DEMO [193] ile optimize edilmiştir. Bireylerin temsili için sürekli aktivasyon tabanlı kodlama birimi ve domine olmayan çözümlerden bir çözüm seçmek için gap statistics kullanılmıştır. Sonuçlara göre, MODE ve DEMO yöntemleri MOCK ve NSGA-II yöntemlerinden daha iyi performans göstermiştir. Bir diğer yöntem variable-length real jumping genes GA (VRJGA) [194] tabanlı olup küme içi entropi ve kümeler arası uzaklığı eş zamanlı olarak minimize etmeye çalışır.

Saha ve Bandyopadhyay [195] nokta simetri uzaklığı tabanlı çok amaçlı kümeleme yöntemi (MOPS) önermiştir. Bu yöntemde çok amaçlı ısıl işlem algoritması (AMOSA) [196], bütün simetri tabanlı uzaklıklar toplamı ve küme dâhilindeki Öklid uzaklıklar toplamı olmak üzere iki amacı eş zamanlı olarak optimize eder. Birey temsili merkez tabanlı kodlama birimi ile sağlanır ve domine olmayan çözümlerden bir çözüm seçimi Minkowski skor ile yapılır. Veri elemanlarının en yakın kümelere simetri uzaklığı kullanılarak atanması vb. diğer adımlar GAPS [143] yönteminde olduğu gibi uygulanır. Elde edilen sonuçlar neticesinde MOPS kümeleme yöntemi MOCK, GAPS ve GAK-means [197] yöntemlerinden daha iyi performans sağlamıştır. Saha ve Bandyopadhyay [198] iki farklı nokta simetri tabanlı kriteri eş zamanlı kullanarak bir AMOSA tabanlı çoklu amaç kümeleme yöntemi (SSym-AMOSA) geliştirmiştir. Ancak deneysel çalışmalar için sadece 2 veya 3 kümeden oluşan veriler seçilmiştir. Bir diğer AMOSA tabanlı kümeleme yöntemi (VAMOSA) Sym-index ve Xie-Beni indeksleri eş zamanlı olarak optimize eder [199]. Xie-Beni ve Sym-index indeksleriyle oluşturulan çoklu amaç kriteri beyin MRI görüntülerinin segmentasyonu için de kullanılmıştır [200]. Ancak VAMOSA simetrik olmayan kümeleri saptayamamaktadır. VAMOSA, MOPS ve SSym-AMOSA yöntemlerinden farklı olarak GenClustMOO [201] kümeleri alt kümelere böler ve bu alt kümeler bireylerde temsil edilir. Bir diğer farklılık da GenClustMOO yöntemi Sym-index, I-index ve Con-index [202] olmak üzere üç amaç fonksiyonunu eş zamanlı olarak optimize etmeye çalışır.

Wikaisuksakul [203] bir NSGA-II tabanlı otomatik kümeleme yöntemi (FCM-NSGA) önermiştir. Önerilen yöntemde veri elemanları FCM ile kümelere dağıtılır. Alınan sonuçlara göre FCM-NSGA yöntemi VAMOSA, MOCK ve VGAPS yöntemlerinden

daha iyi performans sağlamıştır. Ancak bu yöntemin test edildiği veri setleri en fazla 3 tane küme içermektedir.

Yukarıda bahsedilen çalışmalarda optimize edilecek doğrulama indeksleri önceden belirlenmiştir. Ancak bu durumda her veride istenen sonuç elde edilemez. Bu sebepten ötürü Mukhopadhyay ve arkadaşları [204] uygulama esnasında kullanıcıdan komut alan ve bu komutlara göre uygun doğrulama indekslerini elde etmek için adaptif olarak öğrenen interaktif birçok amaçlı kümeleme yöntemi geliştirmiştir.

Yanfei ve arkadaşları [205] MODE algoritmasının uzaktan algılama resimlerinin segmentasyonunda direkt uygulanmasının mümkün olmadığını öne sürmüştür. Bu yüzden iki aşamalı sistem önermiştir: optimizasyon katmanı ve kümeleme katmanı. Optimizasyon katmanında araştırma uzayı daraltılarak optimal küme sayısı elde edilmeye çalışılır. Kümeleme katmanında elde edilen en iyi birey amaç fonksiyonları eş zamanlı uygulanarak geliştirilmeye çalışılır ve aynı zamanda *F* ve *CR* parametreleri de adaptif olarak belirlenir.

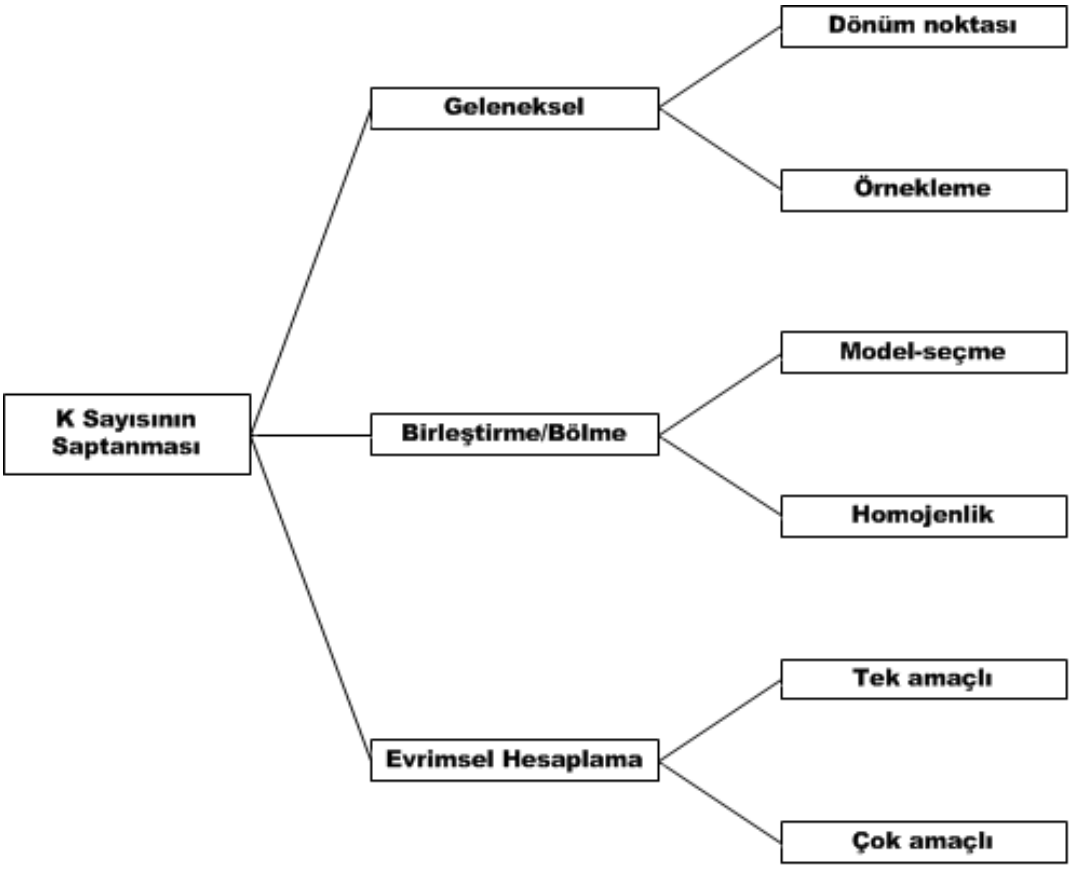
1.4.4. Özet ve Taksonomi

Bu alt bölümde temel amaç, küme sayısının belirlenmesi üzerine bir taksonomi ortaya koymak ve bu problemle ilgili bahsi geçen yöntemlerin genel bir özetini sunmaktır. Şekil 1.7'de görüleceği üzere küme sayısını belirleyen yöntemler temel olarak üç kategoriye ayrılmıştır: geleneksel, bölme/birleştirme ve evrimsel hesaplama. Geleneksel yöntemler temel olarak iki yol takip etmektedir. İlk yolda, bir kümeleme algoritması ve doğrulama indeksi kullanılarak Şekil 1.2'deki gibi değerlendirme grafiği oluşturulur ve ardından grafikteki dönüm noktası bulunur. Değerlendirme grafiğinden dönüm noktasını saptamak için noktalar arası maksimum fark, ikinci farkların minimum değeri ve L-metot gibi birçok yöntem önerilmiştir. Uygulanacak yöntem değerlendirme grafiğinin oluşturulduğu doğrulama indeksine göre seçilir. Diğer yol ise, verilerde örneklem yaparak küme sayısını saptamaya çalışan yöntemlerdir. Ancak bu yöntemler uygulanabilirlik açısından daha zordur. Dolayısıyla ilk yol olan değerlendirme grafiğini oluşturup dönüm noktasının saptanması, örneklem tabanlı yöntemlerden daha yaygındır.

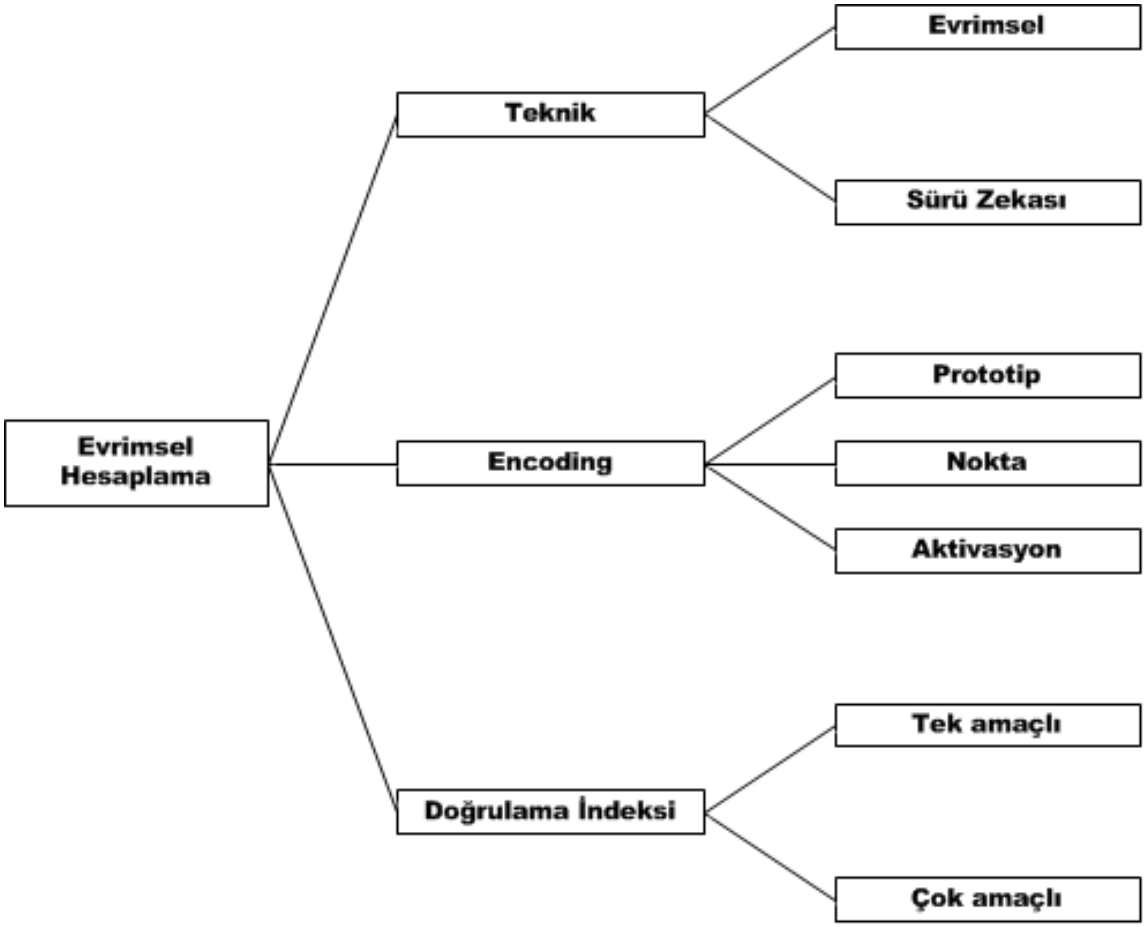
K-means, beklenti büyütme ve karışım modeller temel alınarak oluşturulmuş birleştirme/bölme tabanlı yöntemler, bünyesinde tanımlanmış kriterlere göre kümeleri

böler veya birleştirir. Kriter cinsinden birleştirme/bölme tabanlı yöntemler model-seçmeli ve homojenlik olmak üzere iki grupta incelenebilir. Model-seçmeli yöntemler (örn. X-means, G-means), kümeleri bölme veya birleştirme işlemini BIC, MML ve MDL gibi istatistiksel veya olasılık kriterini temel alarak gerçekleştirir. Homojenlik tabanlı yöntemler (örn. ISODATA, SYNERACT) ise küme içi uzaklık, kümeler arası uzaklık ve küme içi varyans gibi kriterleri temel alır. Homojenlik kriterleri model-seçmeli kriterlerden daha basit olmasına rağmen, model-seçmeli kriterler homojenlik kriterleri gibi kullanıcı tarafından önceden tanımlanmış eşik değerler içermediğinden dolayı daha yaygındır.

Evrimsel hesaplama tabanlı yöntemler tek amaçlı ve çok amaçlı olmak üzere iyi kategoride incelenir. Çok amaçlı yöntemler tek amaçlı yöntemlerin bazı eksikliklerini giderebilmek amacıyla geliştirilmiştir. Karakteristik özelliklerine göre evrimsel hesaplama tabanlı yöntemlerini birey temsili, teknik ve kullanılan doğrulama indeksi olmak üzere üç ayrı kategoride de değerlendirmek mümkündür. İlgili taksonomi Şekil 1.8'de sunulmuştur.



Şekil 1.7. Küme sayısını belirleyen yöntemlerin sınıflandırma ağacı.



Şekil 1.8. Evrimsel hesaplama tabanlı yöntemlerin sınıflandırma ağacı.

Tek ve çok amaçlı evrimsel tabanlı teknikler kullanılan temsil, indeks ve yönteme göre sırasıyla Tablolar 1.2 ve 1.3'te gösterilmiştir. Tablo1.2'ye göre en çok kullanılan tek amaçlı evrimsel hesaplama tekniği GA'dır. Onu sırasıyla DE ve PSO algoritmaları takip etmektedir. Evrimsel algoritmalar daha uzun bir geçmişe sahip olması sebebiyle küme sayısının saptanması problemine sürü zekası temelli algoritmalardan daha fazla uygulanmıştır. Doğrulama indekslerini ele aldığımızda yumuşak (fuzzy) indeksi olarak Xie-Beni ve katı (crisp) indeks olarak DBI ve CH indekslerinin diğerlerine nazaran daha fazla tercih edildiği görülmektedir. Birey temsili olarak merkez ve sürekli aktivasyon tabanlı kodlama birimleri en çok tercih edilenlerdir. Merkez tabanlı kodlama birimi özel gelişim öperatörleri gerektirmesi sebebiyle genellikle GA modelleri ile kullanılmıştır. Diğer taraftan sürekli aktivasyon tabanlı kodlama birimi genellikle DE ve PSO tarafından tercih edilmiştir.

Tablo 1.3'e göre en çok kullanılan çok amaçlı evrimsel hesaplama teknikleri AMOSA ve NSGA-II'dir. Çok amaçlı tekniklerde en çok tercih edilen birey temsili merkez tabanlı kodlama birimidir. Çok amaçlı kriterler olarak da Je ve Xie-Beni indeksleri araştırmacılar tarafından sıklıkla kullanılmıştır. Bunun yanında, Xie-Beni indeksi nokta simetri tabanlı indekslerle de senkronize edilerek daha kompakt ve birbirinden ayrılmış kümeler elde edilmeye çalışılmıştır.

1.5. Sonuçlar

Bu bölüm tez kapsamında ele alınacak problemler ile ilgili kapsamlı bir literatür taraması sunmuştur. Bununla beraber literatürdeki çalışmaların eksiklikleri de ele alınarak tez çalışmasının motivasyonu ortaya konulmaya çalışılmıştır. Bunun sonucunda evrimsel hesaplama tekniklerinin kümeleme problemlerinin çözümünde önemli bir potansiyeli olduğu görülmüştür. Ancak mevcut çalışmalardan görüleceği üzere, ABC önemli bir evrimsel hesaplama tekniği olmasına rağmen araştırmacılar tarafından tezde dikkate alınan problemlerin çözümünde çok fazla kullanılmamıştır. Dolayısıyla ABC'nin görüntü kümeleme problemlerinin çözümünde kullanımı henüz tam olarak araştırılmış değildir. Bu tez çalışmasını yönlendirecek temel eksiklikler aşağıdaki gibi özetlenebilir:

1. Birkaç çalışma dışında ABC kümeleme tabanlı görüntü segmentasyon yönteminin mevcut olmaması,
2. Amaç fonksiyonu olarak kullanılan literatürdeki mevcut görüntü kümeleme kritelerinin ayarlanması gereken parametreler içermesi ve kompaktlık-ayrıklık prensiplerini tam sağlayamaması,
3. Mevcut kuantalama yöntemlerinin bazı dezavantajlarının olması ve ABC'nin renk kuantalama problemine bugüne kadar uygulanmaması,
4. Standart ABC'nin verideki küme sayısını belirleme problemi üzerindeki performans analizinin ve probleme uygun bir sürekli ABC modelinin tasarımının henüz gerçekleştirilmemiş olması,
5. Küme sayısını belirlemek amacıyla variable string length tabanlı ABC modeli dışında ABC ile bir otomatik kümeleme yönteminin geliştirilmemiş olması.

Tablo 1.2. Tek amaçlı otomatik kümeleme yöntemleri.

Metot/Yöntem	Kodlama	Amaç Fonk.	Evrimsel Hesaplama Tekniği
VGA-clustering[127]	Merkez tabanlı	I-index	Variable string length GA
FVGA [206]	Merkez tabanlı	Xie-Beni	Variable string length GA
GCUK [12]	Dikkate alma tabanlı	DBI	Variable string length GA
CGA [132]	Merkez tabanlı	SIL	GA
CGA-II-III,EAC [133]	Merkez tabanlı		GA
PGAK [135]	Merkez tabanlı	Küme içi uzaklık	Paralel GA & K-means
HGA [136]	Sürekli aktiv. tabanlı	DBI	Hiyerarşik GA
Lin et al. [138]	Bölme tabanlı	DBI	Kuralsal GA
AGCUK [139]	Merkez tabanlı	DBI	Div&Abs tabanlı GA
HNGA [140]	Merkez tabanlı	WSVF	Niching GA & K-means
ECSAGO [141]	Binary+Real	UNC	UNC & HAEA
VGAPS [142]	Merkez tabanlı	Sym-index	Variable string length GA
Fuzzy-VGAPS [145]	Merkez tabanlı	FSym-index	Variable string length GA
TGCA [146]	Merkez tabanlı	CH	İki aşamalı GA
GGA [147]	Bölme tabanlı	DBI	Gruplama tabanlı GA
GenClust [155]	Merkez tabanlı	COSEC	GA & K-Means
ACGA [154]	Sürekli aktiv. tabanlı	CH	Geliştirilmiş operatörler ile GA
DNNM-clustering [153]	Merkez tabanlı	Density based Js	Niching metot
DCPSO [157]	İkili aktiv. tabanlı	VI index	BPSO
MEPSO [159]	Sürekli aktiv. tabanlı	Xie-Beni	Multi-elisit PSO
ACMPSO [160]	Sürekli aktiv. tabanlı	CH	Mutation tabanlı PSO
DCPG [161]	İkili aktiv. tabanlı	Küme içi uzaklık	Paralel GA&PSO
ACPSO [164]	Sürekli aktiv. tabanlı	VI index	PSO
ACDE [13]	Sürekli aktiv. tabanlı	DBI & CH	Adaptif DE
AFDE [165]	Sürekli aktiv. tabanlı	Xie-Beni	Adaptif DE
KFNDE [168, 169]	Sürekli aktiv. tabanlı	Kernel Xie-Beni	Komşuluk tabanlı DE
Kernel-MEPSO [170]	Sürekli aktiv. tabanlı	Kernel CS	Multi-elitist PSO
AKC-BCO [171]	Sürekli aktiv. tabanlı	Kernel CS	ABC
MoDEAFC [172]	İkili aktivasyon tabanlı	I-index	Gbest&Lbest güdümlü DE
IDEAFC [173]	İkili aktivasyon tabanlı	Xie-Beni	Gbest&Lbest güdümlü DE+SVM
DEPSO [174]	Sürekli aktiv. tabanlı	CH-DBI-Dunn-CS-I-index-SIL	Hibrit DE&PSO
VABC-FCM [14]	Merkez tabanlı	Xie-Beni, PBM	Variable string length ABC
ETSA [175]	İkili aktivasyon tabanlı	PBM	Tabu arama algoritması
EA's [178]	Adaptif sürekli aktiv. tabanlı	V_{Izz}	Evrimsel algoritma
ACGSA [177]	Sürekli aktiv. tabanlı	Scatter variation	Yerçekimsel arama algoritması
FuzzyGES [179]	Bölme tabanlı	Xie-Beni	Grup evrimsel stratejiler
ABBEA [167]	Sürekli aktiv. tabanlı	Ort. ikili uzaklık	Bakteri yiyecek arama algoritması
DCHS [181]	Dikkate alma tabanlı	PBMF-index	Harmoni arama algoritması
DLSIAC [182]	Sürekli aktiv. tabanlı	PBM	Yerel arama tabanlı yapay bağışıklık algoritması
BKRM-clustering [183]	Merkez tabanlı	Küme içi uzaklık	Blink, naked mole-rats algoritması

Tablo 1.3. Çok amaçlı otomatik kümeleme yöntemleri.

Metot/Yöntem	Kodlama	Amaç Fonk.	Evrimsel Hesaplama Tekniği
MOCK [184]	Löküs tabanlı	Dev & Conn	PESA-II
Multi-obj. GA [186]	Merkez tabanlı	Xie-Beni & Je	NSGA-II
Multi-obj. GA [188]	Merkez tabanlı	Xie-Beni & Je	NSGA-II+SVM
GADE [189]	Sürekli aktiv. tabanlı	Xie-Beni & Je	Hibrit GA & DE
Multi-obj. GA [190]	Merkez tabanlı	Küme içi uzaklık & Kümeler arası uzaklık	MOGA
Multi-obj. DE [191, 192]	Sürekli aktiv. tabanlı	Xie-Beni & Penalized Je	MODE, DEMO
VRJGA [194]	Merkez tabanlı	Küme içi uzaklık & Kümeler arası uzaklık	Genişletilmiş JGGA
MOPS [195]	Merkez tabanlı	TotalSym & Toplam Öklid Mesafesi	AMOSA
Ssym-AMOSA [198]	Merkez tabanlı	SsymAvg & SymDiff	AMOSA
VAMOSA [199]	Merkez tabanlı	Xie-Beni & Sym-index	AMOSA
MCMOClust [200]	Merkez tabanlı	Xie-Beni & Sym-index	AMOSA
GenClutMOO [201]	Merkez tabanlı	Sym-index & I-index & Con-index	AMOSA
FCM-NSGA [203]	Merkez tabanlı	Je & overlap-sep.	NSGA-II
IMOC-Auto [204]	Merkez tabanlı	DBI, Xie-Beni, Je, PBM, SIL	NSGA-II
AFCDME [205]	Yarı adaptif aktivasyon tabanlı	Xie-Beni & Je	Modifiye MODE

2. BÖLÜM

EVRİMSEL HESAPLAMA TEKNİKLERİ

Bu bölüm evrimsel hesaplamanın temellerini ve tez kapsamındaki deneysel çalışmalarda kullanılan algoritmaların genel işleyişini içermektedir. Mevcut algoritmalar içinde yapay arı koloni ve ona bağlı olarak geliştirilen modeller ayrıntılı olarak işlenmektedir.

2.1. Evrimsel Hesaplama

Son 20-30 yıldır yapay zekanin bir alt dalı olarak kabul edilen evrimsel hesaplama (evolutionary computation) araştırmacılar tarafından oldukça ilgi görmekte ve bu alandaki tekniklerin sayısı hızla artmaktadır [132]. Evrimsel hesaplama biyolojik gelişim prensiplerine dayalı olarak ortaya çıkmıştır. Evrimsel hesaplama teknikleri, evrimsel algoritmalar (evolutionary algorithms) ve sürü zekası temelli algoritmalar (swarm intelligence algorithms) olmak üzere iki temel kategoride incelenmektedir.

2.1.1. Evrimsel Algoritmalar

Evrimsel algoritmalar biyolojik gelişim prensiplerine dayalı ve kalıtım olayını simule eden popülasyon tabanlı skolastik araştırma metotlarıdır.

Genel olarak bütün evrimsel algoritmaların çalışma prensibi şu şekildedir: 1) her bir birey probleme uygun potansiyel bir çözümü temsil eder ve bir popülasyon birden fazla bireyden oluşur; 2) her bireyin kalitesi bir amaç fonksiyonu kullanılarak ölçülür ve daha uygun bireyler seçmek suretiyle yeni popülasyon oluşturulur; 3) popülasyonu yenilemek amacıyla bazı operatörler vasıtasıyla yeni bireyler üretilir. Bu prosedür durdurma kriteri sağlanıncaya kadar tekrarlanır. Bir evrimsel algoritmanın genel çalışma prensibi Şekil 2.1'de sunulmuştur.

Popülasyonu üret
Popülasyon bireylerinin kalitesini hesapla
Repeat
Önceki popülasyondaki bireylere seleksiyon uygulayarak yeni popülasyon oluştur
Popülasyon bireylerini mutasyon ve çaprazlama ile değiştir
Değiştirilmiş popülasyon bireylerinin kalitesini değerlendir
Until durdurma kriteri sağlanıncaya kadar

Şekil 2.1. Evrimsel algoritmanın temel adımları.

Temel iki transformasyon (evrimsel operatör) şunlardır:

- **Mutasyon:** Bir bireyde ufak değişiklik yapılarak yeni bir birey oluşturulmasıdır. Bu değişiklik ikili gösterime sahip bireylerde seçilen bir pozisyonun değerinin tersine çevrilmesi ve numerik (continuous) gösterime sahip bireylerde seçilen bir pozisyonun değerine ekleme (çıkarma) yapılmasıyla gerçekleştirilebilir. Mutasyon işlemini uygulamaktaki temel amaç popülasyona genetik çeşitlilik eklenerek yerel minimuma takılmanın engellenmesidir [207].

- **Çaprazlama (Crossover):** Çaprazlama iki veya daha fazla bireyin bilgisinin birleştirilmesiyle yeni bireyler oluşturma işlemidir. Bu şekilde araştırma uzayında yeni alanlar keşfedilir [208]. Bir diğer söylemle global araştırma sağlanır.

Yaygın kullanılan beş temel evrimsel algoritma vardır:

- Genetik algoritmalar [209]: Genetik sistemleri simule ederek geliştirilen ilk algoritmalardan biri olan genetik algoritmalar doğal seleksiyon prensibine dayanır. Problem çözümünü temsil eden popülasyon bireyleri sabit uzunluktaki bit parçacıkları (bitstrings) ile temsil edilir ve popülasyon bireylerine genetik operatörler uygulanarak optimal çözüm elde edilmeye çalışılır. Gradyen yöntemler ile kıyaslandığında, genetik algoritmanın yerel yakınsama problemleri daha azdır. Ancak hesaplama karmaşıklığı daha yüksektir.
- Genetik programlama [210]: Bir problemi çözmek için en uygun bilgisayar programını araştırır. Bireylerin temsili ağaç formundadır. Genetik programlama gibi genotipik gelişimi esas alır.

- Evrimsel stratejiler [211]: Evrimsel stratejiler birçok açıdan genetik algoritmalar ile benzer yaklaşım gösterir. Ancak evrimsel stratejiler genotipik ve fenotipik gelişimi esas almakla beraber fenotipik gelişime daha fazla meyillidir. Temel olarak reel tanımlı süreklilik gösteren problemlerin çözümünde kullanılır. Operatörlerde yapılacak değişiklik ile ayrık problemlerin çözümünde de kullanılması mümkündür. Seleksiyon, mutasyon ve çaprazlama operatörlerinin hepsini kullanır.
- Evrimsel programlama [212]: Genellikle reel uzayda tanımlı süreklilik gösteren fonksiyonları optimize etmek için kullanılır. Evrimsel programlama seleksiyon ve mutasyon operatörlerini kullanırken çaprazlama operatörünü kullanmaz. Fenotipik gelişimi esas alır.
- Diferansiyel gelişim [213]: Reel uzayda tanımlı problemlerin çözümünde kullanılır. Mutasyon, çaprazlama ve seleksiyon operatörlerinin hepsi kullanılır. Uygulanabilirliği kolay, basit ve güçlü bir algoritmadır.

2.1.2. Sürü Zekası Temelli Algoritmalar

Sürü zekası, canlı bireylerin birbirleri ve çevreleri ile olan etkileşimleri sonucu ortaya çıkan davranışların araştırılması ve modellenmesi ile ilgilenir. Kuş ve balık sürülerinin birlikte hareket etmeleri, bal arılarının yiyecek arama eğilimleri ve karıncaların feremon maddesi aracılığıyla haberleşmeleri sürü zekası disiplinin yoğunlaştığı sosyal davranış modellerine örnek olarak verilebilir. Bir sürünün sürü zekası kapsamında değerlendirilebilmesi için şu özelliklere sahip olması gerekmektedir [11]:

a) İş bölümü (Division of Labour): Bir sürüdeki bireylerin dışardan bir yönlendirme olmadan sahip oldukları yetenekleri göz önüne alarak iş bölümü yapması beklenir. Arı türünün davranış modeli incelendiğinde, yeteneklerine bağlı olarak bir kısmı yiyecek arama işlemini yapar, bir kısmı yiyecek kaynağına yoğunlaşır ve bir kısmı da yiyecek kaynakları hakkında bilgi sağlar.

b) Kendi başına organize olabilme (Self Organization): Kendi başına organize olabilme, bir sürüdeki bireyin diğer bireylerle etkileşim ve iletişiminden elde ettikleri tecrübeyi temel alarak hareket etmesidir. Bu işleviyle sürünün tamamını etkiler. Bir bireyin diğer bireyler ile etkileşimi komşuluk prensibi üzerinden gerçekleşir. Kendi başına organize olabilmenin dört temel prensibi vardır:

- Pozitif geri besleme (positive feedback): İyiye veya iyi olana teşvik vardır. Sürü içerisinde yapılan bir işlemin başarılı olması, o işlemi gerçekleştiren birey sayısında artışı da beraberinde getirir. Örneğin, feremon madde miktarının fazla olduğu yollardaki karınca sayısının fazla olması bir pozitif geri beslemedir.
- Negatif geri besleme (negative feedback): Talep görmeyenden veya başarısız olandan uzaklaşma eğilimidir. Sürü bireyleri amaca uygun sonuç elde edemedikleri işlemlerden uzaklaşmaya başlar. Feremon madde miktarının az olduğu yollardaki karınca sayısının daha az olması negatif geri beslemeye örnektir.
- Rasgelelik-salınım (fluctuation): Bireylerin sürü içindeki davranışlarının keskin sınırlar ile belirlenmemesi veya bireyin alacağı kararlarda bir tolerans bandının veya rasgelelik olması durumudur. Örneğin, karıncaların her zaman feremon maddesinin fazla olduğu hatlara yönelmemesi, bazen de içgüdüsel olarak farklı hatlara yönelmesi söz konusudur. Yani karıncaların hareketlerinde bir rasgelelik de söz konusudur.
- Çoklu etkileşim (multiple interaction): Çoklu etkileşim, sürüdeki bireylerin birbirlerinden etkilenerek zeki davranışların ortaya çıkmasını sağlar. Çoklu etkileşim kendi başına organize olan bireylerin en önemli özelliğidir.

En yaygın kullanılan iki sürü zekası temelli algoritma karınca koloni optimizasyon (ant colony optimization (ACO)) [214] ve parçacık sürü optimizasyon (particle swarm optimization (PSO)) [215] algoritmalarıdır. ACO, karıncaların yiyecek arama davranışlarını simule ederken, PSO kuş ve balık gibi birlikte hareket eden toplulukların davranışlarını simule etmektedir. Genellikle ACO ayrık problemlerin çözümünde kullanılırken, PSO nümerik problemlerin çözümünde kullanılmaktadır. Diğer oldukça popüler olan sürü zekası temelli algoritma ise arıların zeki yiyecek arama davranışlarından ilham alınarak geliştirilen yapay arı koloni (artificial bee colony (ABC)) [216] algoritmasıdır. ABC algoritmasının dışında ateşböceği (firefly) [217], bakteri yiyecek arama (bacterial foraging) [218] vb. sürü zekası temelli algoritmalar da mevcuttur.

2.2. Parçacık Sürü Optimizasyon

2.2.1. Standart PSO Algoritması

PSO algoritması 1995'te Eberhart ve Kennedy [215] tarafından önerilmiş sürü zekası temelli algoritmalarından biridir. PSO temel olarak kuş ve balık sürülerinin sosyal ve bireysel davranışlarının modellenmesi sonucu ortaya çıkmıştır. PSO uygulanabilirliği kolay ve ortaya çıktığı yıldan itibaren biyoloji, tıp ve bilgisayar grafiklerine kadar birçok alanda kullanılmıştır. PSO sisteminde bir parçacık (particle) problemin çözümüne uygun olası çözümü ifade eder. Parçacığın performansı da amaç fonksiyonu ile ölçülür.

Bir parçacığın pozisyonu daha önce ziyaret ettiği en iyi pozisyon bilgileri (kendi elde ettiği tecrübeler) ve komşuluğunda bulunan en iyi parçacığın pozisyon bilgilerinden (komşu parçacıkların tecrübeleri) etkilenerek değişir. Eğer bir parçacık sürünün tamamına komşu ise, parçacığın en iyi pozisyona sahip komşusu küresel (global) en iyi parçacıktır. Bir parçacığın kendi elde ettiği tecrübeler sonucundaki yerel (local) en iyi pozisyonu Denklem 2.1 ile belirlenir. Bir sürüdeki küresel en iyi parçacık ise yerel en iyi parçacıkların arasından Denklem 2.2 ile belirlenir.

$$lbest_i(t+1) = \begin{cases} lbest_i(t), & f(x_i(t+1)) \geq f(lbest_i(t)) \\ x_i(t+1), & f(x_i(t+1)) < f(lbest_i(t)) \end{cases} \tag{2.1}$$

Burada $lbest_i(t+1)$, $t+1$ zamanındaki i. parçacığın yerel en iyi pozisyonunu ve $x_i(t+1)$, $t+1$ zamanındaki parçacığın pozisyonunu ifade eder.

$$gbest(t) = \min\{lbest_1(t), lbest_2(t), ..., lbest_s(t)\} \tag{2.2}$$

Burada $gbest(t)$ t anındaki küresel en iyi parçacığın pozisyonunu ve s sürünün büyüklüğünü ifade eder.

Yerel ve küresel en iyi parçacık pozisyonlarının belirlenmesinin ardından hız ve yer değiştirmeye bağlı olarak pozisyon bilgilerinin güncellenmesi sırasıyla Denklem 2.3 ve 2.4 ile gerçekleştirilir.

$$v_{ij}(t+1) = w_{init} v_{ij}(t) + c_1 r_{1j}(lbest_{ij}(t) - x_{ij}(t)) + c_2 r_{2j}(0,1)(gbest_{ij}(t) - x_{ij}(t)) \tag{2.3}$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{2.4}$$

Burada $v_i(t+1)$, $t+1$ anındaki i. parçacığın hız vektörünü, c_1-c_2 bilişsel ve sosyal sabitleri, r_{1j}-r_{2j} 0 ve 1 arasında rasgele üretilmiş sayıları ve w_{init} hız vektörü için ağırlıklandırma katsayısını ifade etmektedir.

PSO algoritmasının kaba kodu Şekil 2.2'de sunulmuştur. PSO, GA'ya göre genel olarak uygulanması daha kolay bir algoritma, işlemci hesaplama karmaşıklığı daha düşük ve optimal çözümler elde etmede daha başarılıdır [29]. Ancak PSO'da yerel yakınsama problemleri mevcuttur. [219]'a göre, PSO evrimsel algoritmalardan daha hızlı iyi çözümler bulmasına karşın belirli bir çevrim sayısından sonra genellikle çözümleri iyileştirememektedir. Parçacıklar arasındaki hızlı bilgi akışı benzer parçacıkların ortaya çıkmasına ve yerel yakınsama riskinin artmasına yol açmaktadır. PSO'nun yerel yakınsama problemlerini çözmek amacıyla araştırmacılar tarafından modifikasyon ve hibritleme yöntemlerine sıklıkla başvurulmuştur [220]. Bir diğer problem de PSO'daki kontrol parametresi sayısının fazla olmasıdır. Bütün problemlerde kullanılabilen tek bir parametre değeri yoktur. Probleme bağlı olarak parametre değerlerinin belirlenmesi gerekmekte ve parametre değerinin problem çözümünde önemli etkisi vardır. Örneğin, w_{init} değerinin artırılması parçacıkların hızının artmasına ve bu da daha fazla global araştırma ve daha az yerel araştırmaya sebep olmaktadır.

```
begin
    foreach parcacik i do
        Rasgele x_i uret;
        Rasgele v_i uret;
        Set lbest_i = x_i;
    end
    for cycle ← 1 to MCN do
        foreach parcacik i do
            parcacik i'nin (f((x_i)) kalitesini hesapla;
            Denklem (2.1) ile lbest_i guncelle;
            Denklem (2.2) ile gbest_i guncelle;
            foreach pozisyon j do
                Denklem (2.3) ile v_ij guncelle;
            end
            Denklem (2.4) ile parcacik pozisyonu guncelle;
        end
    end
end
```

Şekil 2.2. Temel PSO algoritmasının kaba kodu.

2.2.2. İkili PSO Modelleri

PSO'nun ayrık problemlere uygulanması amacıyla Kennedy ve Eberhart [221] ikili PSO (BPSO) algoritmasını önermiştir. BPSO vektörel değerlerin sigmoid fonksiyon ile ikili uzaya transformasyonu esasına dayanır. PSO algoritmasındaki problemler BPSO algoritması için de geçerlidir. Lojik operatörlerden esinlenerek geliştirilen XOR-PSO algoritması [222] ünite yüklenme probleminin çözümünde kullanılmış ve BPSO'dan daha iyi performans sağlamıştır. Pampara ve Engelbrecht [223] açı modülasyon tabanlı ikili PSO (AMPSO) algoritması geliştirmiştir. AMPSO'da dörtlü sürekli uzay ile ikili ayrık uzay arasındaki transformasyon açı modülasyonuna bağlı fonksiyon ile sağlanmıştır. Ancak transformasyon ikili ayrık uzayda global araştırmaya yeteri kadar imkan vermemektedir. Nezambadi-pour ve arkadaşları [224] BPSO'da kullanılan sigmoid fonksiyonundan daha farklı bir sigmoid fonksiyon kullanmıştır. Ancak bu fonksiyon da yerele takılma problemini çözebilmiş değildir. Khanesar ve arkadaşları [225] w_{init} parametresinin hız vektörü ve parçacık güncelleme mekanizmasında negatif etkileri olduğunu tespit etmiş ve parçacıkları güncellemek amacıyla farklı bir hız mekanizması önermiştir (NBPSO). Ancak önerilen mekanizma uygulanabilirlik açısından kompleks bir yapı içermektedir. Yun-Won ve arkadaşları [226] kuantum hesaplamanın prensiplerini BPSO algoritmasına uygulamıştır (QBPSO). QBPSO'da parçacık pozisyonlarının 0 veya 1 değeri alması $|\alpha|^2$ veya $|\beta|^2$ üzerinden belirlenir ($|\alpha|^2+|\beta|^2=1$). Bir diğer söylemle BPSO'daki vektör parçacığı kuantum hesaplama ile değiştirilir. QBPSO'da c_1-c_2 sabitlerine ve w_{init} ağırlıklandırma katsayısına gerek yoktur. Onların yerini açı rotasyonu parametresi alır.

Bu tez çalışmasında BPSO, QBPSO ve NBPSO modelleri deneysel çalışma ve karşılaştırmalarda kullanılacaktır.

2.3. Yapay Arı Koloni Algoritması

2.3.1. Standart ABC Algoritması

Kovanda bir arada yaşayan bal arıları arasında iş bölümü ve organize hareket edebilme özelliği vardır [11]. Bu kabiliyetleri sebebiyle zeki sürü kapsamında değerlendirilmiş ve 2000'li yıllardan itibaren bal arılarının davranışları üzerinde çalışmalar yoğunlaşmıştır. Bu çalışmalar neticesinde arılar algoritması (bees algorithm) [227], sanal arı algoritması

(virtual bee algorithm) [228], arı sistemi (bee system) [229], arı koloni optimizasyon (bee colony optimization) [230] ve yapay arı koloni algoritması (artificial bee colony) [231] gibi modeller ortaya çıkmıştır. Bahsi geçen modeller içinde uygulanabilirliği kolay ve yerele takılma probleminin diğerlerine nispeten az olması sebebiyle en çok kabul gören algoritmalardan biri yapay arı kolonisi (ABC) algoritmasıdır [216]. ABC algoritması, Karaboğa tarafından arıların zeki yiyecek kaynağı arama davranışları esas alınarak çok parametreli numerik problemlerin çözümünü bulmak amacıyla geliştirilmiştir. ABC algoritmasının PSO, GA ve ES algoritmalarından daha iyi performans sergilediği gösterilmiştir [231].

ABC algoritmasında kaşif (scout), görevli (employed) ve gözcü (onlooker) olmak üzere üç farklı tür arı vardır. ABC'de her görevli arı bir yiyecek kaynağından sorumludur ve görevli arıların sayısı gözcü arıların sayısına eşittir. Dolayısıyla yiyecek kaynağı sayısı görevli ve gözcü arıların toplamının yarısı kadardır. Yiyecek kaynaklarının yerleri probleme ait olası çözümleri ve kaynakların nektar miktarları da o kaynaklara karşılık gelen çözümlerin kalitesini temsil eder. Görevli arılar görevli oldukları kaynakların tükenmesi ile kaşif arılara dönüşür ve kaşif arılar da tükenen kaynak yerine yeni kaynak arar. ABC modelinin genel akış şeması Şekil 2.3'te verilmiş olup modele ilişkin temel süreç aşağıdaki gibidir [11]:

1. Kaşif arılar çevrede rasgele yiyecek aramaya başlar.
2. Kaşif arılar yiyecek kaynaklarını bulduktan sonra görevli arı haline gelir ve ilgili kaynaklardan kovana nektar taşımaya başlar. Kovana nektar taşıyan görevli arılar ya ilgili kaynaklara geri döner veya kaynakla ilgili bilgiyi dans alanında bulunan gözcü arılarla paylaşır. Eğer ilgili kaynak tükenmişse görevli arı tekrar kaşif arı haline gelir ve yeni kaynak arama işlemine başlar.
3. Kovanda bekleyen gözcü arılar yiyeceğin kalitesi ve dans frekansı ile orantılı olarak bir kaynağı tercih ederler.

Modele ilişkin olarak ABC algoritması ilk olarak yiyecek kaynakları ve bu yiyecek kaynaklarına uygun yerlerin üretilmesi ile işleme başlar. Yiyecek kaynağının üretimi aşağıdaki denklem ile gerçekleştirilir;

$$x_{ij} = x_j^{min} + rand(0,1)\left(x_j^{max} - x_j^{min}\right) \tag{2.5}$$

Burada x_{ij} *i.* kaynağın *j.* pozisyonunu, $i=\{1,2,...,SN\}$, $j=\{1,2,...D\}$, *SN* popülasyon büyüklüğünü ve *D* toplam pozisyon sayısını ifade etmektedir; x_j^{max} ve x_j^{min} pozisyonların alt-üst limitlerini belirtir ve *rand(0,1)* 0 ile 1 arasında rasgele üretilmiş bir sayıdır.

Şekil 2.3. ABC'nin temel akış diyagramı.

Başlangıç yiyecek kaynaklarının üretilmesinin ardından görevli arılar ilgili kaynaklara gönderilir. İşçi arılar sorumlu oldukları yiyecek kaynaklarının komşuluğunda Denklem 2.6 ile yeni kaynakları belirler. Denklem 2.6'da görüleceği üzere rasgele seçilmiş tek pozisyon üzerinden işlem gerçekleşir. Bulunan yiyecek kaynağının kalitesi hâlihazırda bulunan yiyecek kaynağının kalitesinden daha iyi ise yeni yiyecek kaynağı hafızaya alınır, eski yiyecek kaynağı hafızadan silinir. Aksi takdirde mevcut kaynağın deneme sayaci bir artırılır ve araştırma işlemine devam edilir.

$$v_{ij} = x_{ij} + \theta_{ij}\left(x_{ij} - x_{kj}\right) \tag{2.6}$$

Burada x_{ij} *i.* kaynağın *j.* pozisyonunu (*j* rasgele seçilmiş) ve x_{kj} i. kaynağın komşuluğunda rasgele seçilmiş *k.* kaynağın *j.* pozisyonunu ve θ_{ij} ise -1 ve 1 arasında üretilmiş rasgele bir sayıdır.

Kaynaklarda araştırma işleminin tamamlanmasının ardından görevli arılar yiyecek kaynakları ile ilgili bilgileri (pozisyon ve kalite bilgisi) gözcü arılar ile paylaşır. Gözcü arılar görevli arılardan aldıkları bilgileri değerlendirir ve kaynakların nektar miktarlarına bağlı olarak Denklem 2.7 ve 2.8 ile hesaplanan olasılık değerleri üretilir. Eğer bir *i.* kaynak için üretilen olasılık değeri (p_i) 0 ve 1 arasında üretilen rasgele sayıdan küçükse, o kaynak gözcü arı tarafından araştırma işlemine tabi tutulur. Denklem 2.7'den anlaşılacağı üzere kalitesi yüksek yiyecek kaynakları daha fazla seçilme eğilimindedir. Bir pozitif geri besleme söz konusudur. Yine anlaşılacağı üzere iyi kaynakların seçilebilmesi de bir rasgelelik prensibine bağlanmıştır ve her zaman iyi kaynakların seçilmesi garanti edilmez. Bu da algoritmanın rasgelelik-salınım prensibinden bir örnek olarak verilebilir. Araştırma işlemi görevli arılarda olduğu gibi Denklem 2.6 ile gerçekleştirilir. Bulunan yeni kaynağın nektar miktarı fazla olması durumunda yeni kaynak mevcut kaynağın yerine araştırmaya alınır. Aksi takdirde mevcut kaynağın deneme sayacı bir artırılır ve araştırma işlemine devam edilir. Sonuç olarak kaynak seçiminde görevli ve gözcü arılar aç gözlü yaklaşım (greedy selection) sergiler.

$$p_i = \frac{fitness_i}{\sum_{i=1}^{SN} fitness_i} \tag{2.7}$$

$$\text{fitness}_i = \begin{cases} \dfrac{1}{1+fit_i} & fit_i \geq 0 \\ 1+abs(fit_i) & fit_i < 0 \end{cases} \tag{2.8}$$

Burada fit_i i. kaynağın nektar miktarını (amaç fonksiyon değerini) ifade etmektedir. Bir kaynağın uygunluk değeri arttıkça o kaynağın seçilme olasılığının artması söz konusudur.

begin
- SN, MCN ve $limit$ parametrelerini belirle;
- **foreach** *kasif ari i* **do**
  - Denklem (2.5) ile rasgele x_i kaynagini uret;
  - x_i kaynaginin kalitesini ($f(x_i)$) degerlendir;
- **end**
- **for** $cycle \leftarrow 1$ **to** MCN **do**
  - **foreach** *gorevli ari i* **do**
    - x_i kaynaginin komsulugunda Denklem (2.6) ile yeni kaynak (v_i) bul;
    - v_i kaynaginin kalitesini ($f(v_i)$) degerlendir;
    - **if** $(f(v_i)) > (f(x_i))$ **then**
      - x_i kaynaginin yerine v_i kaynagini arastirmaya al;
      - $denemesayac_i = 0$;
    - **else**
      - x_i kaynagini arastirmaya devam et;
      - $denemesayac_i = denemesayac_i + 1$;
    - **end**
  - **end**
  - Kaynaklarin olasilik degerlerini (p_i) Denklem (2.7) ile hesapla;
  - **foreach** *gozcu ari i* **do**
    - Kaynaklarin olasilik degerlerine bagli olarak bir kaynak (x_i) sec;
    - x_i kaynaginin komsulugunda Denklem (2.6) ile yeni kaynak (v_i) bul;
    - v_i kaynaginin kalitesini ($f(v_i)$) degerlendir;
    - **if** $(f(v_i)) > (f(x_i))$ **then**
      - x_i kaynaginin yerine v_i kaynagini arastirmaya al;
      - $denemesayac_i = 0$;
    - **else**
      - x_i kaynagini arastirmaya devam et;
      - $denemesayac_i = denemesayac_i + 1$;
    - **end**
  - **end**
  - **if** *bir kaynagin deneme sayaci* $>$ *limit* **then**
    - Kasif ari Denklem (2.5) ile yeni bir kaynak belirler;
  - **end**
  - En iyi kaynagi belirle;
- **end**

end

Şekil 2.4. Temel ABC algoritmasinin kaba kodu.

Her çevrimin sonunda deneme sayaçlarına bakılır. Deneme sayacı, tanımlanan "limit" eşik değerinin üzerinde olan kaynak ilgili görevli arı tarafından terkedilir ve görevli arı kaşif arı haline gelerek Denklem 2.1 ile yeni kaynak bulur. ABC algoritmasının kaba kodu Şekil 2.4'te sunulmuştur.

ABC algoritmasının performansını artırmaya yönelik araştırmacılar tarafından bazı modifikasyon ve hibrit modeller de geliştirilmiştir. Akay ve Karaboga [232] Denklem 2.6'nın tek pozisyondaki güncelleme işlemini olasılıklı olarak birden fazla pozisyonda gerçekleşecek şekilde (MRABC) yeniden dizayn etmiştir. MRABC'de, bir kaynağın her pozisyonu için [0,1] aralığında üretilen rasgele pozisyon kullanıcı tarafından tanımlı *MR* parametresinden küçükse o pozisyon Denklem 2.6 ile güncellenir. PSO'daki küresel en iyi çözüm Denklem 2.6'da temsil yoluyla yeni ABC modelleri de türetilmiştir. Örneğin, ekonomik emisyon dağıtım problemini çözmek amacıyla Denklem 2.6'da küresel en iyi çözüm bilgisinden (GABC) faydalanılmıştır [233]. MRABC ve GABC modelleri deneysel çalışmaların bir bölümünde kullanılmak üzere bu tez çalışmasına dâhil edilmiştir.

2.3.2. İkili ABC Modelleri

Standart ABC algoritması numerik ve süreklilik arz eden problemlerin çözümü için önerilmiş olup, yapısı sebebiyle ayrık ve ikili problemlere direkt uygulanması mümkün değildir. Bu sebepten ötürü ABC algoritmasının üzerinde bazı modifikasyonların yapılması gerekmektedir. Son 2-3 yılda bu alandaki çalışmalar yoğunlaşmıştır. Pampara ve Engelbrecht [234] açı modülasyon tekniğini kullanarak reel uzay ile ikili uzay arasında transformasyon sağlamıştır (AMABC). Kashan ve arkadaşları [235] ikili vektörler arasındaki benzerlik kriterini kullanarak bir ayrık ABC modeli (DisABC) önermiştir. Önerilen model kapasite kısıtsız tesis yeri seçimi problemi üzerinde test edilmiş ve modelin standart BPSO algoritmasından daha iyi sonuç verdiği görülmüştür. Kiran ve Gündüz [236] lojik operatörlere bağlı bir ikili ABC modeli (XOR-ABC) geliştirmiştir. Önerilen XOR-ABC modeli, DisABC ve lojik operatörleri temel alarak oluşturulan XOR-PSO'dan daha iyi performans sağlamıştır. Wei ve arkadaşları [237] sırt çantası probleminin çözümü için bir ikili model geliştirmiştir.

Bu tez çalışmasında AMABC ve DisABC modellerine yaklaşımlarındaki orijinallik ve ikili ABC'nin ilk örnekleri olmaları sebebiyle deneysel çalışmalarda da yer verilmiştir.

Ayrıca DisABC tez çalışması kapsamında geliştirilen bir yöntemin temeli olma niteliği de taşımaktadır. Dolayısıyla bu modellere ait gerekli açıklama ve detaylar takip eden alt bölümlerde sunulmuştur.

2.3.2.1. Açı Modülasyona Dayalı ABC Modeli

Açı modülasyon (angle modulation) tabanlı ABC [234], bit vektörü üretmek amacıyla bir trigonometrik fonksiyon kullanır. Bu trigonometrik fonksiyon sinyal işlemede önemli bir yere sahip olan açı modülasyona bağlı bir teknikten türetilmiş ve Denklem 2.9 ile tanımlanmıştır:

$$g(x) = \sin(2\pi(x-a)\times b \times \cos(2\pi(x-a)\times c) + d \tag{2.9}$$

Burada x bir aralıktaki giriş değerini, a fonksiyon g'nin genliğini, b sinüs fonksiyonun frekansını veya periyodunu, c cosinüs fonksiyonun frekansını veya periyodunu ve d fonksiyon g'nin vektörel ötelemesini ifade eder.

Bu fonksiyonu kullanmaktaki temel amaç dört boyutlu sürekli uzaydan D boyutlu ikili uzaya dönüşümü sağlamaktır.

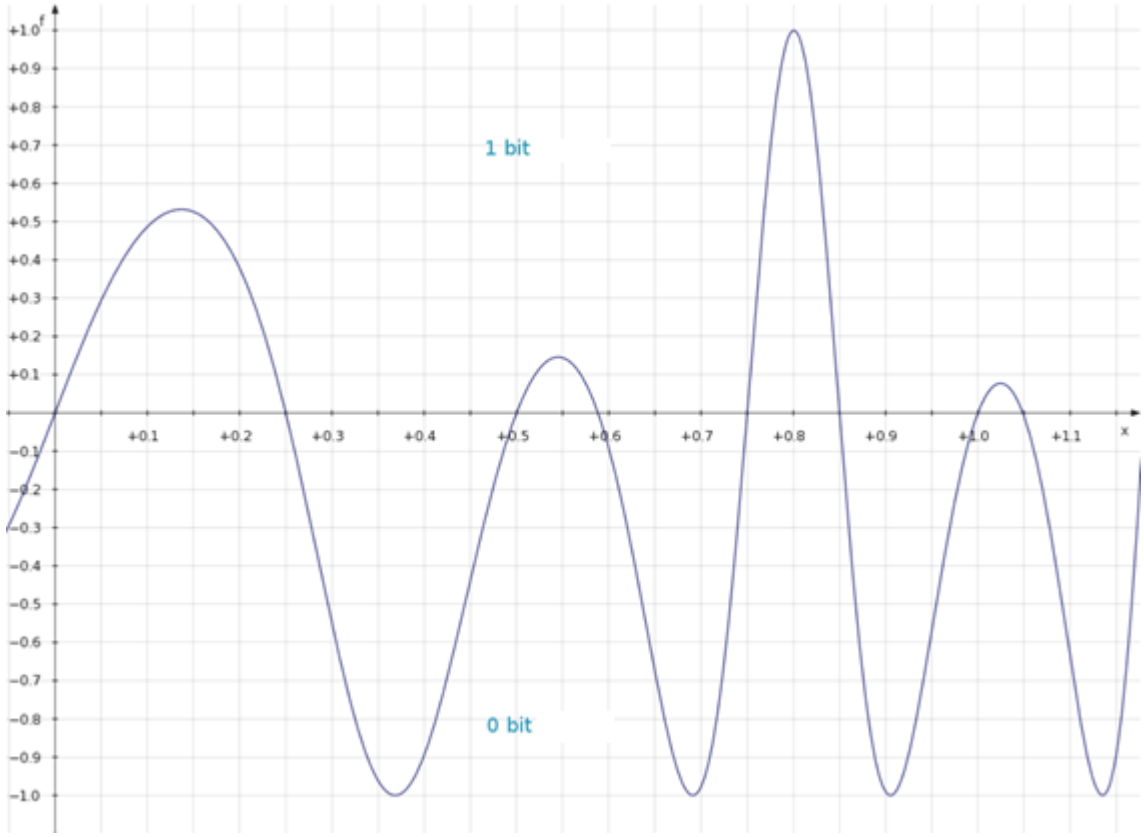
$$g : \Re^4 \rightarrow \{0,1\}^D \tag{2.10}$$

AMABC'de her kaynak ($x_i=(a_i,b_i,c_i,d_i)$) dört boyutlu sürekli uzayda tanımlanmıştır. Bir kaynağı ikili vektöre dönüştürmek için kaynağın *[a,b,c,d]* değerleri g fonksiyonunda yerlerine yerleştirilir. Ardından tanımlanmış bir aralıkta g fonksiyonuna D tane örneklem yaptırılarak Denklem 2.11 ile D boyutlu ikili vektör oluşturulur. İkili vektörün oluşturulmasının ardından kaynağın kalitesi hesaplanır.

$$bit_{ij} = \begin{cases} 1 & if \quad g(x_i, sample_j) \geq 0 \\ 0 & Otherwise \end{cases} \tag{2.11}$$

Burada $sample_j$ j. biti üretmek için belirli bir aralıkta tanımlanmış bir değerdir.

Örnek: Örneklemler (değerler) [0.0, 1.1] aralığında ve soldan sağa x ekseninde 0.1 artacak şekilde tanımlı olsun. Daha önceden belirlenmiş *a=0, b=1, c=1* ve *d=0* parametrelerine göre oluşan dağılım grafiği ve ikili vektör aşağıdaki şekildedir.



Şekil 2.5. *g* fonksiyonun örneklenmesi sonucu oluşan grafik [238].

Tablo 2.1. Örneklemler sonucu oluşan ikili vektör [238].

Örneklem	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1
Bit	1	1	1	0	0	1	0	0	1	0	1	0

AMABC, 4 boyutlu temsil sayesinde hesaplama zamanı performansı iyi bir algoritmadır. Ancak reel uzaydan ikili uzaya transformasyonda küresel araştırma özelliğini yitirebilmekte ve olası çözümleri geniş uzayda taramakta sıkıntılar yaşayabilmektedir.

2.3.2.2. Benzerlik Esasına Dayalı ABC Modeli

İkili vektörler arasındaki benzerlik/farklılık ölçmeyi sağlayan kriterler sınıflandırma, kümeleme ve resim betimleme gibi birçok uygulamada hayati önem taşımaktadır [239]. Yaklaşık bir asırdır ikili vektörler arasındaki benzerlik ve farklılığı ölçmek amaçlı birçok çeşitli kriter geliştirilmiştir. En çok bilinen ve kullanılan benzerlik kriterlerinden biri olan Jaccard katsayısı [240] şu şekilde tanımlanır:

$$Similarity(X_i, X_k) = \frac{M_{11}}{M_{11} + M_{10} + M_{01}} \tag{2.12}$$

Burada $X_i=(x_{i1},...x_{id},...,x_{iD})$ ve $X_k=(x_{k1},...x_{kd},...,x_{kD})$ ikili D boyutlu vektörler, M_{11} değeri $x_{id}= x_{kd}=1$ şartını sağlayan toplam bit sayısı, M_{10} değeri $x_{id}=1$ ve $x_{kd}=0$ şartını sağlayan toplam bit sayısı ve M_{01} değeri $x_{id}=0$ ve $x_{kd}=1$ şartını sağlayan toplam bit sayısıdır.

X_i ve X_k arasındaki farklılık da Jaccard katsayısı cinsinden şu şekilde ifade edilir:

$$Dissimilarity(X_i, X_k) = 1 - Similarity(X_i, X_k) = 1 - \frac{M_{11}}{M_{11} + M_{10} + M_{01}} \tag{2.13}$$

DisABC [235] ikili vektörler arasındaki Jaccard katsayısı esas alınarak tasarlanmıştır. DisABC'de Denklem 2.6 öncelikle Denklem 2.14 formunda yazılır. Denklem 2.14'te (X_i, X_k) ve (V_i, X_i) kaynak vektör çiftleri arasındaki farka dayalı işlem '-' Denklem 2.13'teki farklılık esasına göre tasarlanarak Denklem 2.15 formatına dönüştürülür.

$$v_{ij} - x_{ij} = \theta_{ij}\left(x_{ij} - x_{kj}\right) \tag{2.14}$$

$$Dissimilarity(V_i, X_i) \approx \ddot{Q} \times Dissimilarity(X_i, X_k) \tag{2.15}$$

Burada Q pozitif indirgeme faktörü olup her çevrimde değeri Denklem 2.16 ile belirlenir.

$$Q = Q_{\max} - \left(\frac{Q_{\max} - Q_{\min}}{MCN} \right) \ cevrim \tag{2.16}$$

Burada Q_{max} ve Q_{min} indirgeme faktörünün alt ve üst seviyelerini, MCN maksimum çevrim sayısını ve *cevrim* ise o andaki çevrim sayısını ifade eder.

Denklem 2.15'ten anlaşılacağı üzere (V_i, X_i) vektör çiftleri arasındaki farklılık (*Dissimilarity*(V_i, X_i)), $Q\times$ *Dissimilarity*(X_i, X_k)'nin sonucuna mümkün mertebe yakın olmalıdır. Bu bilgiden yola çıkarak hem V_i hem X_i vektör pozisyonlarında 1 olan toplam bit sayısı (M11); V_i vektör pozisyonunda 1, X_i vektör pozisyonunda 0 olan toplam bit sayısı (M10); ve V_i vektör pozisyonunda 0, X_i vektör pozisyonunda 1 olan toplam bit sayısı (M01) matematiksel programlama ile elde edilir:

$$\min\left\{1-\frac{M_{11}}{M_{11}+M_{10}+M_{01}}-\mathrm{Q}\times\mathrm{Dissimilarity}(\mathrm{X_i},\mathrm{X_k})\right\} \tag{2.17.1}$$

$$M_{11}+M_{01}=m_1 \tag{2.17.2}$$

$$M_{10}\le m_0 \tag{2.17.3}$$

$$M_{11},M_{10},M_{01}>0 \, ve tamsayı \tag{2.17.4}$$

Burada m_1, X_i vektöründeki toplam 1 sayısı ve m_0, X_i vektöründeki toplam 0 sayısıdır. Denklem 2.17.2, $(m_1+1)(m_0+1)$ tane M_{11}, M_{10} ve M_{01} kombinasyonunun olduğunu ifade eder. Bu kombinasyonlardan Denklem 2.17.1'i minimum yapan ilk M_{11} M_{10} ve M_{01} kombinasyonu seçilir.

Bulunan M_{11}, M_{10} ve M_{01} kombinasyonu X_i ve V_i arasındaki benzerlik değerleridir. Bu işlemin akabinde V_i vektörünün üretim aşaması başlar. V_i $1\times D$'lik sıfır vektörü olarak tanımlanır ve aşağıdaki seçim mekanizmalardan biri rassal olarak çalıştırılır:

1. Rasgele seçim: X_i vektöründe 1 değeri taşıyan M11 tane biti rasgele seç ve V_i vektöründe ilgili bitlerin değerini 1'e çevir. Ardından X_i vektöründe 0 değeri taşıyan M10 tane biti seç ve V_i vektöründe ilgili bitlerin değerini 1'e çevir.
2. Aç gözlü seçim: Hem global en iyi vektör (*Gbest*) hem de X_i vektöründeki değeri 1 olan M11 tane biti seç ve V_i vektöründe ilgili bitlerin değerini 1'e çevir. Bazı durumlarda hem *Gbest* hem de X_i vektöründe M11 tane 1 değeri taşıyan bit olmayabilir. Eğer seçilen bit sayısı M11'den az ise, kalan bitleri X_i vektöründen 1 değeri taşıyanların arasından seç ve V_i vektöründe ilgili bitlerin değerini 1'e çevir. Ardından X_i vektöründe değeri 0, *Gbest* vektördeki değeri 1 olan M10 tane bit seç ve V_i vektöründe ilgili bitlerin değerini 1'e çevir. Eğer seçilen bit sayısı M10'dan az ise, kalan bitleri X_i vektöründe değeri 0 olanların arasından seç ve V_i vektöründe ilgili bitlerin değerini 1'e çevir.

DisABC'de Denklem 2.5 ikili kaynak üretmeye uygun olmadığı için ikili kaynak üretmek için Denklem 2.18 kullanılır:

$$x_{ij}=\begin{cases}0, & U(0,1)<Th\\ 1, & U(0,1)\ge Th\end{cases} \tag{2.18}$$

```
begin
    SN, MCN ve limit parametrelerini belirle;
    foreach kasif ari i do
        Denklem (2.18) ile rasgele x_i kaynagini uret;
        x_i kaynaginin kalitesini (f(x_i)) degerlendir;
    end
    for cycle ← 1 to MCN do
        foreach gorevli ari i do
            x_i kaynaginin komsulugunda x_k kaynagini sec;
            Denklem (2.13) ile Dissimilarity(x_i, x_k) hesapla;
            Denklem (2.17) ile M_11, M_10 ve M_01 degerlerini bul;
            if rand(0,1) < 0.5 then
                v_i = rasgelesecim(M_11, M_10, M_01, X_i)
            else
                v_i = acgozlusecim(M_11, M_10, M_01, X_i, Gbest);
            end
            v_i kaynaginin kalitesini degerlendir;
            if (f(v_i)) > (f(x_i)) then
                x_i kaynaginin yerine v_i kaynagini arastirmaya al;
                cezapuani_i = 0;
            else
                x_i kaynagini arastirmaya devam et;
                cezapuani_i = cezapuani_i + 1;
            end
        end
        Kaynaklarin olasilik degerlerini (p_i) Denklem (2.7) ile hesapla;
        foreach gozcu ari i do
            Kaynaklarin olasilik degerlerine bagli olarak bir kaynak (x_i) sec;
            x_i kaynaginin komsulugunda x_k kaynagini sec;
            Denklem (2.13) ile Dissimilarity(x_i, x_k) hesapla;
            Denklem (2.17) ile M_11, M_10 ve M_01 degerlerini bul;
            if rand(0,1) < 0.5 then
                v_i = rasgelesecim(M_11, M_10, M_01, X_i)
            else
                v_i = acgozlusecim(M_11, M_10, M_01, X_i, Gbest);
            end
            v_i kaynaginin kalitesini degerlendir;
            if (f(v_i)) > (f(x_i)) then
                x_i kaynaginin yerine v_i kaynagini arastirmaya al;
                cezapuani_i = 0;
            else
                x_i kaynagini arastirmaya devam et;
                cezapuani_i = cezapuani_i + 1;
            end
        end
        if Bir kaynagin ceza puani > limit then
            Kasif ari Denklem (2.18) ile yeni bir kaynak belirler;
        end
        En iyi kaynagi belirle;
    end
end
```

Şekil 2.6. DisABC algoritmasının kaba kodu.

Burada U(0,1) uniform rasgele olarak üretilmiş 0 ve 1 arasında sayıdır ve *Th* kullanıcı tarafından belirlenen eşik değerdir. *Th* genellikle 0.5 veya 0.75 olarak alınır. DisABC'nin kaba kodu Şekil 2.6'da verilmiştir.

2.4. Sonuçlar

Bu bölüm evrimsel hesaplama teknikleri hakkında kısaca bilgilendirme amacı taşımaktadır. Bu tekniklerden problemlerin çözümünde kullanılan ABC algoritması daha ayrıntılı olarak tanıtılmıştır. Bunun yanında deneysel çalışma ve karşılaştırmalarda sonuçları kullanılacak olan modeller de tanıtılmıştır.

3. BÖLÜM

KÜME SAYISINI PARAMETRE OLARAK ALAN ABC TABANLI GÖRÜNTÜ KÜMELEME YÖNTEMLERİ VE UYGULAMALARI

Görüntü segmentasyonu ve renk kuantalama işlemlerinde en çok kullanılan yöntemlerden biri de kümelemedir. K-means, FCM vb. yöntemler bu işlemlerde önemli bir yere sahiptir. Ancak bu yöntemler başlangıç koşullarından (örn. başlangıç küme seti) ve dış etkenlerden (örn. gürültü) olumsuz etkilenebilmektedir. Bu da yerele takılma problemini ortaya çıkarmaktadır. Bu tür problemi aşmak amacıyla evrimsel hesaplama teknikleri de kullanılmaktadır. Ancak bu alandaki çalışmaların çoğu PSO ve GA tabanlı olup ABC ile ilgili literatürde çok fazla çalışma mevcut değildir. Hatta ABC kümeleme tabanlı renk kuantalama yöntemi araştırmacılar tarafından henüz ele alınmamıştır. Sonuç olarak ABC'nin görüntü segmentasyonu ve renk kuantalama üzerindeki etkisi henüz tam anlamıyla araştırılmış değildir.

Bu bölümün temel amacı ABC kümeleme tabanlı görüntü segmentasyon ve renk kuantalama yöntemlerinin geliştirilmesidir. Bu amaca ulaşmak için beyin MRI görüntülerinde kullanılmak üzere ABC tabanlı tümor segmentasyon metodolojisi, evrimsel teknikler ile kullanılmak üzere yeni bir görüntü kümeleme kriteri (amaç fonksiyonu) ve K-means algoritmasının ABC ile entegrasyonu sağlanarak ABC tabanlı bir kuantalama yöntemi geliştirilmiştir. Bu bölümde temel olarak aşağıdaki incelemeler yapılmıştır:

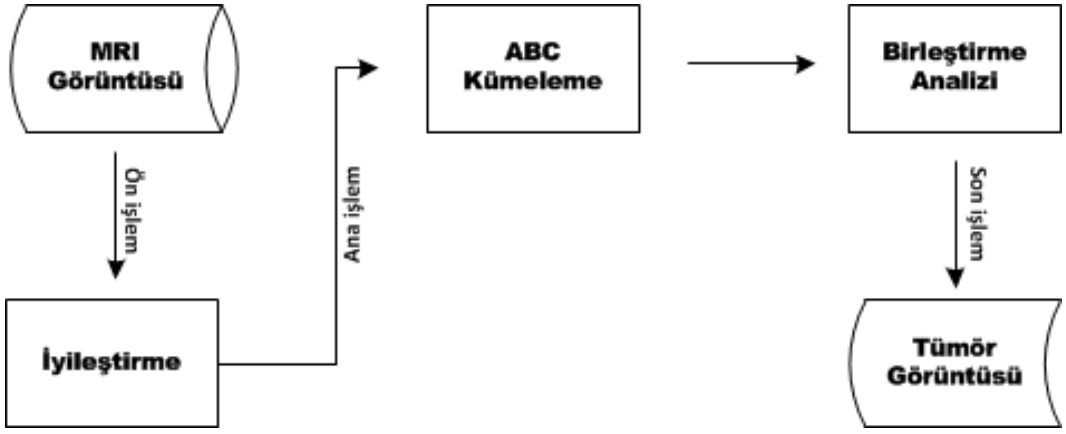
- ABC tabanlı tümör segmentasyon metodolojisinin bilinen iki klasik kümeleme tabanlı segmentasyon metodolojilerine karşı performans analizi,
- Geliştirilen görüntü kümeleme kriterinin mevcut kriterlere ve klasik algoritmalara karşı ABC ile birlikte üç evrimsel hesaplama tekniği üzerinden değerlendirilmesi,

- ABC tabanlı kuantalama yönteminin iki evrimsel hesaplama, iki kümeleme ve bir bölmeli tabanlı kuantalama yöntemlerine karşı performans analizi.

Bu bölümde ilk olarak ABC tabanlı segmentasyon metodolojisi sunulmakta ve ilgili deneysel çalışmalara yer verilmektedir. Daha sonra yeni bir görüntü kümeleme kriteri önerilmekte ve üç temel evrimsel hesaplama tekniği ile bu kriter test edilmektedir. Son olarak ABC tabanlı renk kuantalama yöntemi açıklanmakta ve ilgili yöntemler ile karşılaştırılıp genel değerlendirmesi yapılmaktadır.

3.1. ABC Tabanlı Tümör Segmentasyonu

MRI görüntülerinden beyin tümörünün saptanması ön işlem, ana işlem ve son işlem olmak üzere üç aşamada ele alınmış ve genel akış diyagramı Şekil 3.1'de gösterilmiştir.



Şekil 3.1. ABC tabanlı tümör segmentasyon metodolojisi.

Ön işlem aşaması, MRI görüntüsünün iyileştirilmesini ve görüntüde mevcut olan gürültülerin elimine edilerek ana işlem için hazırlanmasını sağlar. Böylece ana işlem sırasında görüntülerde oluşacak bozulmaların engellenmesi amaçlanır. Görüntüleri iyileştirmede en çok bilinen yollarından birisi filtrelemedir. Filtreleme, bölgesel görüntü işleme olarak da ele alınmaktadır. Griton resimler için filtrelemenin üç klasik uygulaması söz konusudur [241]: griton düzgünleştirme (grey level smooting), griton farklılıklarını vurgulama (emphasizing grey level differences) ve griton geçişlerini keskinleştirmek (sharpening grey level steps). Bu ön işlemde gerekli olan griton düzgünleştirme kullanılarak gürültülerin elimine edilmesidir. Temel olarak normal ortalama, ağırlıklı ortalama, min operatörü, max operatörü, orta-değer (median) operatörü ve k-enyakın-komşu operatörleri griton düzgünleştirme filtrelerinin en çok

bilinenleridir. Bu operatörler arasından resimdeki yüksek ve alçak griton piksellerini griton bölgeleri ayıran ve griton basamaklarını yassılaştırmadan elimine eden orta-değer operatörü [241] ön işlem aşaması için seçilmiştir. Orta-değer operatörde maske içerisindeki gritonlar değerlerine göre sıralanır. Bu sıralamada ortadaki piksel değeri sonuç resmin ilgilenilen pikseli için kullanılır. Orta-değer operatörünün hesaplama zamanı sıralama işlemi sebebiyle yüksektir.

Ana işlem aşaması, ön aşama esnasında iyileştirilmiş ve gürültüden arındırılmış MRI görüntüsünün kümeleme işlemini içerir. Bu işlem için ABC tabanlı bir kümeleme yöntemi geliştirilmiştir [242]. Kümeleme konseptine göre, ABC'de her bir kaynak (çözüm) K tane küme merkezini temsil eder: $x_i=\{m_{i1},\ldots,m_{ik},\ldots,m_{iK}\}$, burada m_{ik} i. kaynağın k. küme merkezidir. Bir kaynağın kalitesi Denklem 3.1 ile ölçülür:

$$fit_1(x_i, Z) = w_1 d_{\max}(Z, x_i) + w_2\left(z_{\max} - d_{\min}(Z, x_i)\right) + w_3 Je_i \tag{3.1}$$

Burada z_{max} orijinal resimdeki maksimum piksel değerini, Z veriyi (resim piksel matrisini), w_1, w_2 ve w_3 kullanıcı tarafından tanımlanmış ağırlıklandırma parametrelerini, $d_{max}(Z,x_i)$ küme dâhilindeki uzaklığı, $d_{min}(Z,x_i)$ kümeler arası uzaklığı ve Je_i toplam kuantalama hatasını ifade eder. $d_{max}(Z,x_i)$, $d_{min}(Z,x_i)$ ve Je_i bileşenleri aşağıda tanımlanmıştır:

$$d_{max}(Z, x_i) = max\left\{\sum_{\forall z_p \in C_{i,k}} \frac{d(z_p, m_{i,k})}{n_{i,k}}\right\} \tag{3.2}$$

$$d_{min}(Z, x_i) = min\{d(m_{i,j}, m_{i,k})\}, j \neq k \tag{3.3}$$

$$Je_i = \frac{\sum_{k=1}^{K}\sum_{\forall z_p \in C_k} d(z_p, m_{i,k})/n_k}{K} \tag{3.4}$$

Burada $n_{i,k}$ $C_{i,k}$ kümesindeki toplam eleman (piksel) sayısını, $d(z_p,m_{i,k})$ z_p ve $m_{i,k}$ arasındaki Öklid mesafesini ve K toplam küme sayısını ifade etmektedir.

Denklem 3.1 eş zamanlı olarak $d_{max}(Z,x_i)$ ve Je_i kriterlerini minimize ve $d_{min}(Z,x_i)$ kriterini maksimize etmeye çalışır. Denklem 3.1'in küçük değerlere ulaşmasıyla kompakt ve birbirinden ayrılmış kümelerin oluşması beklenir. Denklem 3.1 çok amaçlı optimizasyon problemi olarak da ele alınabilir. Ancak bizim amacımız ABC'nin

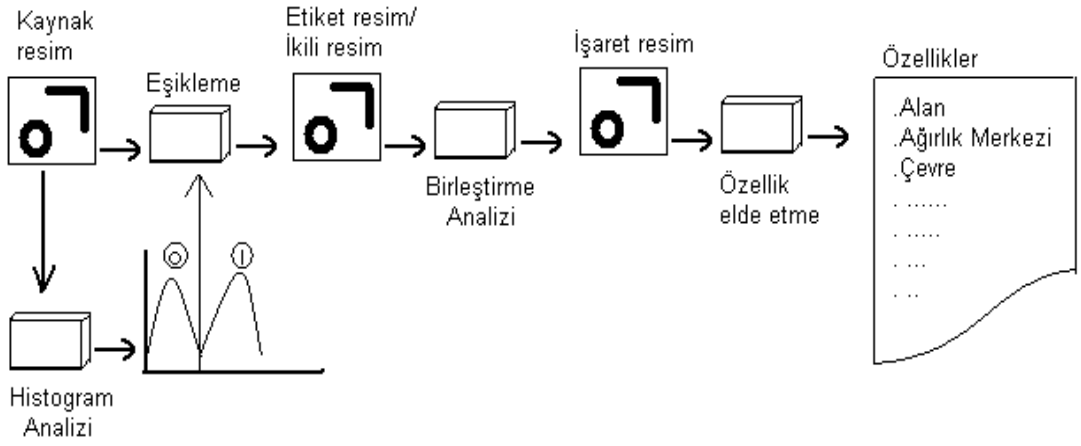
problem üzerinde uygulanabilirliğini artırmak olduğundan çok amaçlı optimizasyon üzerinde durulmamıştır.

ABC tabanlı kümeleme yöntemi Şekil 3.2'de sunulmuştur. Şekil 3.2'den görüleceği üzere bir çözümün amaç fonksiyon değerini hesaplamak için öncelikle veri elemanlarının ilgili çözümün merkezlerine Öklid mesafesi temel alınarak atanması gereklidir. Bu işlemin hesaplama zamanını griton resimlerde biraz daha hızlandırmak mümkündür. Önce ilgili resim verisinin histogramı çıkarılır. Herbir piksel (yoğunluk) değeri için gerekli hesaplamalar o pikselin frekansı ile çarpılarak gerçekleştirilir.

```
begin
    K, SN, MCN ve limit parametrelerini belirle;
    foreach kasif ari i do
        x_i kaynagi icin Z verisinden K tane rasgele merkez sec;
    end
    for cycle ← 1 to MCN do
        foreach gorevli ari i do
            x_i kaynaginin komsulugunda Denklem (2.6) ile yeni kaynak (v_i) bul;
            foreach veri elemani z_p do
                for k = 1 to K do
                    Denklem (1.1) ile d(z_p, m_{i,k}) hesapla;
                end
                d(z_p, m_{i,k})'yi minimum yapan C_{i,k} kumesine z_p'yi ata;
            end
            v_i kaynaginin kalitesini (f(v_i, Z)) Denklem (3.1) ile degerlendir;
            x_i ve v_i arasinda ac gozlu yaklasim uygula;
        end
        Kaynaklarin olasilik degerlerini (p_i) Denklem (2.7) ile hesapla;
        foreach gozcu ari i do
            Kaynaklarin olasilik degerlerine bagli olarak bir kaynak (x_i) sec;
            x_i kaynaginin komsulugunda Denklem (2.6) ile yeni kaynak (v_i) bul;
            foreach veri elemani z_p do
                for k = 1 to K do
                    Denklem (1.1) ile d(z_p, m_{i,k}) hesapla;
                end
                d(z_p, m_{i,k})'yi minimum yapan C_{i,k} kumesine z_p'yi ata;
            end
            v_i kaynaginin kalitesini (f(v_i, Z)) Denklem (3.1) ile degerlendir;
            x_i ve v_i arasinda ac gozlu yaklasim uygula;
        end
        if bir kaynagin ceza puani > limit then
            Denklem (2.5) ile yeni bir kaynak belirle;
        end
        En iyi kaynagi belirle;
    end
end
```

Şekil 3.2. ABC tabanlı kümeleme yönteminin kaba kodu.

Son aşamada, ABC tabanlı kümeleme ile bölgelere ayrılan MRI görüntüsüne bölge-yönelimli birleştirme analizi (connectivity analysis) uygulanarak tümör bölgesi saptanır. Öncelikle kümelenmiş MRI görüntüsünün histogramı değerlendirilerek bir eşik değer belirlenir ve bu eşik değere göre eşikleme (thresholding) yapılır. Eşikleme işleminin ardından ikili görüntünün sol üst köşesinden başlanarak 4'lük veya 8'lik komşuluğunda bulunan aynı (yoğunluk) değerine sahip pikseller bir araya toplanır. Bu topluluğa bir etiket değeri atanır. Böylece etiketlenmiş yeni bir resim ortaya çıkmış olur. Resimdeki etiketli bölgeler (örn. tümör) sahip oldukları alan, ağırlık merkezi veya uzunluk gibi özellikler vasıtasıyla da tanımlanabilir ve ayırt edilebilir. Birleştirme analizinin genel işleyiş şeması Şekil 3.3'te verilmiştir.



Şekil 3.3. Birleştirme analizinin genel işleyişi [47].

3.1.1. Segmentasyonun Deneysel Dizaynı

Deneysel çalışmalarda bir hastaya ait 256×256 piksel ve 8 bitlik toplam 9 tane griton beyin MRI görüntüsü kullanılmıştır. Bu görüntüler hasta beyninin kesit kesit çekilmesiyle oluşturulmuş ve sırasıyla {io70, io80, … , io150} olarak isimlendirilmiştir. Önerilen ABC tabanlı tümör segmentasyon metodolojisinin değerlendirme ve analizi iki klasik kümeleme algoritması (K-means ve FCM) ile karşılaştırılarak yapılmıştır.

ABC segmentasyon metodolojisinde her birey K tane veri setinden rasgele seçilmiş küme merkezleri ile temsil edilmektedir. Dolayısıyla K değeri aynı zamanda popülasyonun boyutunu ifade etmektedir. K değeri, io70, io80 ve io90 görüntülerinde 3 ve diğer görüntülerde 4 olarak karakteristik özelliklerine göre belirlenmiştir. ABC algoritmasında popülasyon büyüklüğü 30, limit değeri 75 ve maksimum çevrim sayısı

250 olarak belirlenmiştir. Bu değerlerin belirlenmesine deneysel denemeler sonucunda karar verilmiştir. K-means ve FCM algoritmalarında maksimum çevrim sayısı 7500 (30 popülasyon büyüklüğü × 250 çevrim sayısı) olarak alınmıştır. FCM algoritmasında μ katsayısı en çok tercih edilen 2 değeri alınmıştır. Amaç fonksiyonundaki w_1, w_2 ve w_3 ağırlıklandırma değerleri sırasıyla 0.4, 0.2 ve 0.4 olarak belirlenmiştir.

ABC ile kümelemenin akabinde 30 birey arasından en iyi bireyi temsil eden *K* tane küme merkezi alınır ve ilgili kriter ve doğrulama indeksleri hesaplamalarında kullanılarak algoritmanın kümeleme performansı belirlenir. K-means ve FCM algoritmalarının çalıştırılmasından tek bir *K* tane küme merkez seti elde edilir ve bu merkezler yine ilgili kriter ve doğrulama indeksleri hesaplamalarında kullanılarak K-means ve FCM algoritmalarının kümeleme performansları belirlenir. K-means algoritmasının boş bir küme üretmesi durumunda veriye uygun yeni bir başlangıç küme seti oluşturularak problem çözülmüştür.

Segmentasyon metodolojilerine ait sonuçların kalitelerinin değerlendirilmesi için amaç fonksiyonunun bileşenleri olan kompaktlık-ayrıklık (J_e, d_{max} ve d_{min}) kriterleri ve doğrulama indeksleri (CS ve DBI) kullanılmıştır. CS doğrulama indeksi küme içi dağılımın minimize edilmesi ve kümeler arası ayrımın da maksimize edilmesi prensibine dayanır. DBI indeksi ise küme içi dağılımın toplamının kümeler arası dağılıma oranıdır. CS ve DBI indekslerinin matematiksel formülleri Tablo 1.1'de sunulmuştur.

3.1.2. Segmentasyonun Deneysel Sonuçları

Segmentasyon sonuçları, Tablo 3.1'de birbirinden bağımsız 30 koşma üzerinden ortalama, standart sapma ve istatistiksel semboller ile sunulmuştur. Standart sapma değerleri parantez içinde ve en iyi değerler kalın yazı tipinde gösterilmiştir. Tablolarda yer alan '+' ('-') sembolü, ABC tabanlı metodolojinin K-means veya FCM'den Wilcoxon Rank Sum Test istatistiğine göre daha iyi (daha kötü) olduğunu göstermektedir. '=' sembolü, ABC tabanlı metodoloji ile ilgili metodoloji arasında Wilcoxon Rank Sum Test istatistiğine göre farklılık olmadığını ifade etmektedir. Sonuçlar kompaktlık-ayrıklık kriterleri (J_e, d_{max} ve d_{min}) ve doğrulama indeksleri (DBI ve CS) üzerinden değerlendirilmiştir. Görsel olarak da metodolojiler arasındaki farklılıklar Şekiller 3.4-3.6 aralığında ortaya konulmaya çalışılmıştır.

Tablo 3.1'deki sonuçlara göre ABC kümeleme tabanlı metodoloji io70 ve io120 dışında dmax kompaktlık kriterinde K-means ve FCM'den daha iyi ortalama ve standart sapma değerleri elde etmiştir. Örneğin, io130 görüntüsünde ABC 17.526 değeri elde ederken, K-means ve FCM sırasıyla 19.058 ve 18.696 değerlerini elde etmiştir. Bu sonuçlar istatistiksel test sonuçları ile de desteklenmiştir. dmin ayrıklık kriterinde de ABC'nin performansı diğerlerine göre kıyaslanmayacak derecede üst düzeydedir. Örneğin, io80 görüntüsünde ABC dmin uzaklığını ortalama olarak 94.243 bulurken, K-means ve FCM sırasıyla 72.245 ve 73.980 bulmuştur. Bir başka görüntü io90'da K-means ve FCM dmin uzaklığını sırasıyla 77.362 ve 79.598 olarak elde ederken, ABC 96.970 elde etmiştir. Bütün görüntü setlerinde dmin kriteri için ABC ve diğer yöntemler arasında bu derece yüksek farklar görmek mümkündür. Buradan K-means ve FCM kümelerin çakışmasına neden olabildiği ve ABC kümeleme tabanlı metodoloji kümeler arası uzaklığı sağlamada oldukça başarılı olduğu rahatlıkla söylenebilir. Algoritmaların kümeleme performanslarını doğrulama indeksleri DBI ve CS üzerinden değerlendirildiğinde de ABC'nin bariz üstünlüğü göze çarpmaktadır. Örneğin, io90 görüntüsünde ABC tabanlı metodoloji CS indeks için 0.6028 ortalama değeri elde ederken, K-means ve FCM sırasıyla 0.9741 ve 0.9091 değerlerini elde etmiştir. Hemen hemen bütün görüntü setleri için ABC ve diğer yöntemlerin kümeleme kaliteleri arasında benzer farklar mevcuttur. Dolayısıyla ABC algoritması K-means ve FCM algoritmalarından daha kaliteli kümeler oluşturabilmektedir. İstatistiksel test sonuçları da bunu destekler niteliktedir. Je dışındaki bütün kriterlerde ABC Wilcoxon Rank Sum istatistiğine göre diğer yöntemlere nispeten anlamlı sonuçlar elde etmiştir. ABC sadece Je hatasında K-means ve FCM algoritmalarından daha kötü sonuç vermiştir. Ancak bu durum tek başına bir anlam ifade etmemektedir. Sonuç olarak kümeleme kriterleri ve doğrulama indekslerini bir bütün olarak değerlendirdiğimizde ABC kümeleme metodolojisi, Kmeans ve FCM algoritmalarından daha iyi kümeleme performansı sağlamaktadır.

Görsel olarak ABC, K-means ve FCM sonuçları her bir görüntü (io70-io150) için soldan sağa orijinal, kümelenmiş ve sonuç (tümör) formatında sırasıyla Şekiller 3.4, 3.5 ve 3.6'da sunulmuştur. ABC genel olarak başarılı sayılabilecek bir segmentasyon işlemi gerçekleştirmiştir. K-means ve FCM algoritmalarında ise yer yer başarısız segmentasyon sonuçları göze çarpmaktadır. io70, io80, io110 ve io120 görüntülerinde

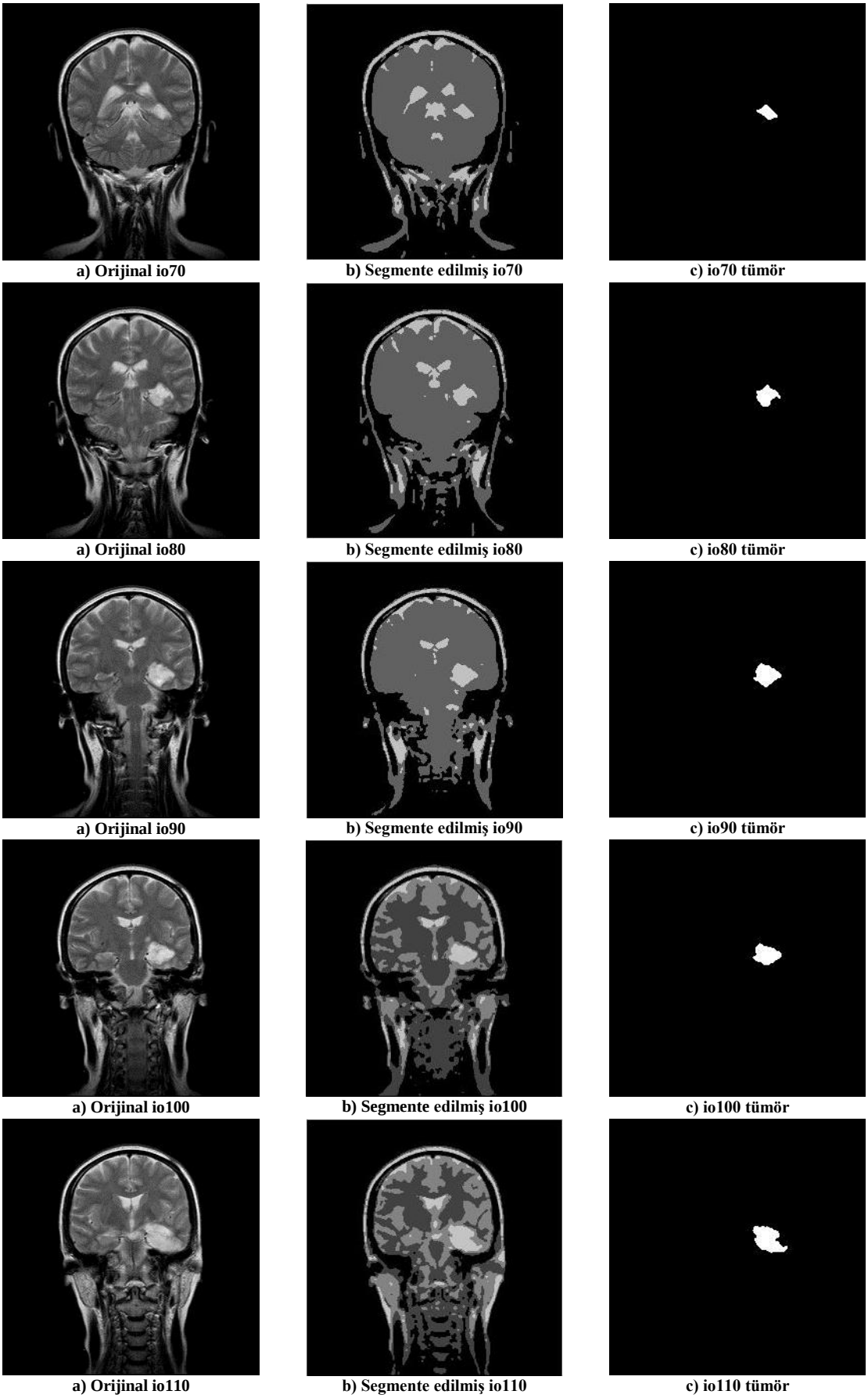
bu durum net olarak görülmektedir. Sonuç olarak görsel sonuçlar Tablo 3.1'deki sayısal sonuçları desteklemektedir.

Tablo 3.1. MRI görüntülerinin sayısal kümeleme sonuçları.

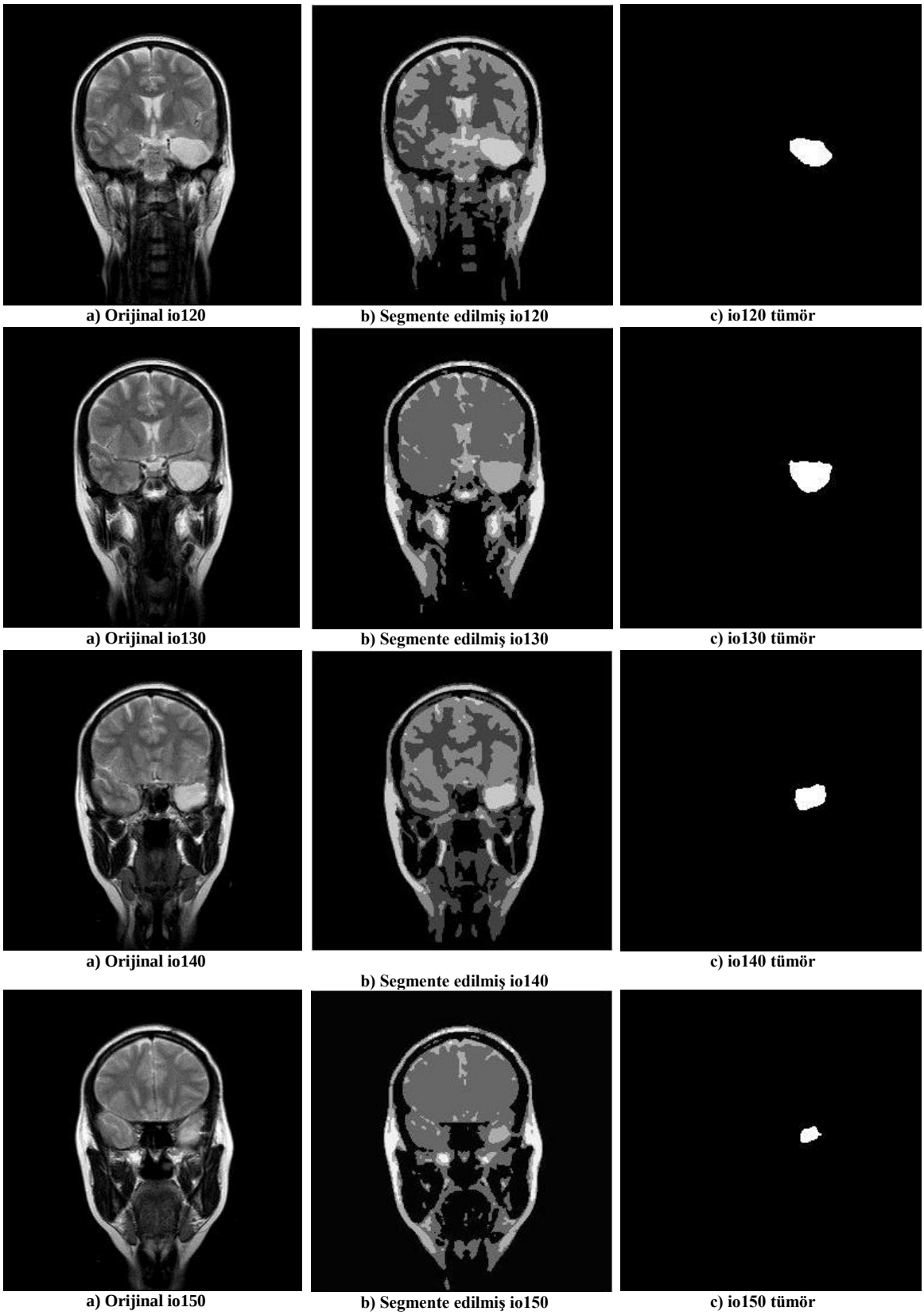
		ABC	K-means	FCM
io70	**Je**	14.051 (0.198)	13.291 (5.4E-15) '-'	**13.153** (4.1E-07) '-'
	Dmax	22.445 (0.260)	**21.277** (7.2E-27) '-'	21.498 (1.2E-06) '-'
	Dmin	**92.821** (0.871)	72.166 (2.8E-14) '+'	72.137 (7.4E-06) '+'
	DBI	**0.1062** (0.001)	0.1379 (0.000) '+'	0.1401 (2.5E-08) '+'
	CS	**0.6127** (0.006)	0.8078 (0.007) '+'	0.8131 (5.6E-16) '+'
io80	**Je**	13.669 (0.123)	13.575 (3.6E-15) '-'	**13.374** (4.3E-07) '-'
	Dmax	**20.835** (0.177)	22.312 (1.0E-14) '+'	22.103 (1.1E-06) '+'
	Dmin	**94.241** (0.583)	72.375 (1.4E-14) '+'	73.980 (1.0E-08) '+'
	DBI	**0.0992** (0.001)	0.1420 (5.6E-17) '+'	0.1357 (3.2E-08) '+'
	CS	**0.5899** (0.003)	0.8343 (0.011) '+'	0.8083 (2.2E-16) '+'
io90	**Je**	**14.008** (0.172)	14.509 (0.012) '+'	14.263 (5.1E-07) '='
	Dmax	**19.906** (0.250)	23.840 (0.070) '+'	23.367 (1.1E-06) '+'
	Dmin	**96.970** (0.750)	77.362 (0.168) '+'	79.598 (1.3E-05) '+'
	DBI	**0.1006** (0.001)	0.1396 (0.001) '+'	0.1310 (3.6E-08) '+'
	CS	**0.5150** (0.001)	0.7030 (1.1E-16) '+'	0.6746 (2.2E-16) '+'
io100	**Je**	**12.299** (0.386)	12.426 (0.168) '='	12.526 (6.0E-06) '+'
	Dmax	**17.070** (0.588)	22.246 (1.708) '+'	22.243 (9.7E-06) '+'
	Dmin	**64.191** (1.907)	45.017 (0.908) '+'	44.473 (3.9E-05) '+'
	DBI	**0.1467** (0.001)	0.1688 (0.004) '+'	0.1754 (1.5E-07) '+'
	CS	**0.6028** (0.024)	0.9741 (0.036) '+'	0.9091 (5.5E-16) '+'

Tablo 3.1. **Devamı.**

io110	**Je**	12.544 (0.639)	**12.118** (0.085) ‘-’	12.120 (2.3E-06) ‘-’
	Dmax	**16.828** (1.009)	19.847 (1.489) ‘+’	19.253 (1.5E-06) ‘+’
	Dmin	**61.962** (3.498)	45.473 (1.125) ‘+’	49.847 (3.3E-05) ‘+’
	DBI	**0.1574** (0.001)	0.1651 (0.009) ‘+’	0.1626 (1.3E-07) ‘+’
	CS	**0.6999** (0.012)	1.0248 (0.034) ‘+’	0.8987 (4.5E-16) ‘+’
io120	**Je**	13.219 (0.427)	12.224 (0.034) ‘-’	**12.173** (1.6E-06) ‘-’
	Dmax	18.196 (0.550)	**17.262** (0.114) ‘-’	17.283 (2.9E-06) ‘-’
	Dmin	**65.055** (2.181)	47.761 (0.268) ‘+’	48.065 (0.001) ‘+’
	DBI	**0.1610** (0.002)	0.1704 (0.001) ‘+’	0.1695 (5.5E-07) ‘+’
	CS	**0.6358** (0.006)	0.7121 (0.019) ‘+’	0.6877 (3.3E-16) ‘+’
io130	**Je**	13.877 (0.298)	**12.374** (0.175) ‘-’	12.377 (2.5E-06) ‘-’
	Dmax	**17.526** (0.572)	19.058 (0.706) ‘+’	18.696 (1.3E-06) ‘+’
	Dmin	**65.328** (1.588)	46.396 (3.976) ‘+’	48.860 (6.5E-05) ‘+’
	DBI	**0.1550** (0.002)	0.1479 (0.011) ‘-’	0.1479 (4.3E-08) ‘-’
	CS	**0.5596** (0.004)	0.8030 (0.011) ‘+’	0.7647 (3.3E-16) ‘+’
io140	**Je**	13.603 (0.088)	12.771 (0.007) ‘-’	**12.701** (2.1E-06) ‘-’
	Dmax	**17.637** (0.128)	21.288 (0.040) ‘+’	20.528 (2.0E-06) ‘+’
	Dmin	**65.509** (0.563)	47.496 (0.403) ‘+’	50.699 (5.4E-05) ‘+’
	DBI	**0.1616** (0.002)	0.1441 (4E-004) ‘-’	0.1439 (2.1E-008) ‘-’
	CS	**0.6684** (0.002)	0.7746 (0.001) ‘+’	0.7368 (3.3E-16) ‘+’
io150	**Je**	15.791 (0.524)	13.479 (0.015) ‘-’	**13.331** (7.4E-07) ‘-’
	Dmax	**20.646** (0.813)	24.317 (0.142) ‘+’	23.204 (7.1E-07) ‘+’
	Dmin	**73.710** (2.110)	45.065 (0.176) ‘+’	46.414 (1.1E-05) ‘+’
	DBI	**0.1679** (0.002)	0.1410 (0.0016) ‘-’	0.1346 (8.8E-009) ‘-’
	CS	**0.5294** (0.019)	0.7294 (0.004) ‘+’	0.6942 (3.3E-16) ‘+’



Şekil 3.4. ABC tabanlı segmentasyon (io70-io110) sonuçları.



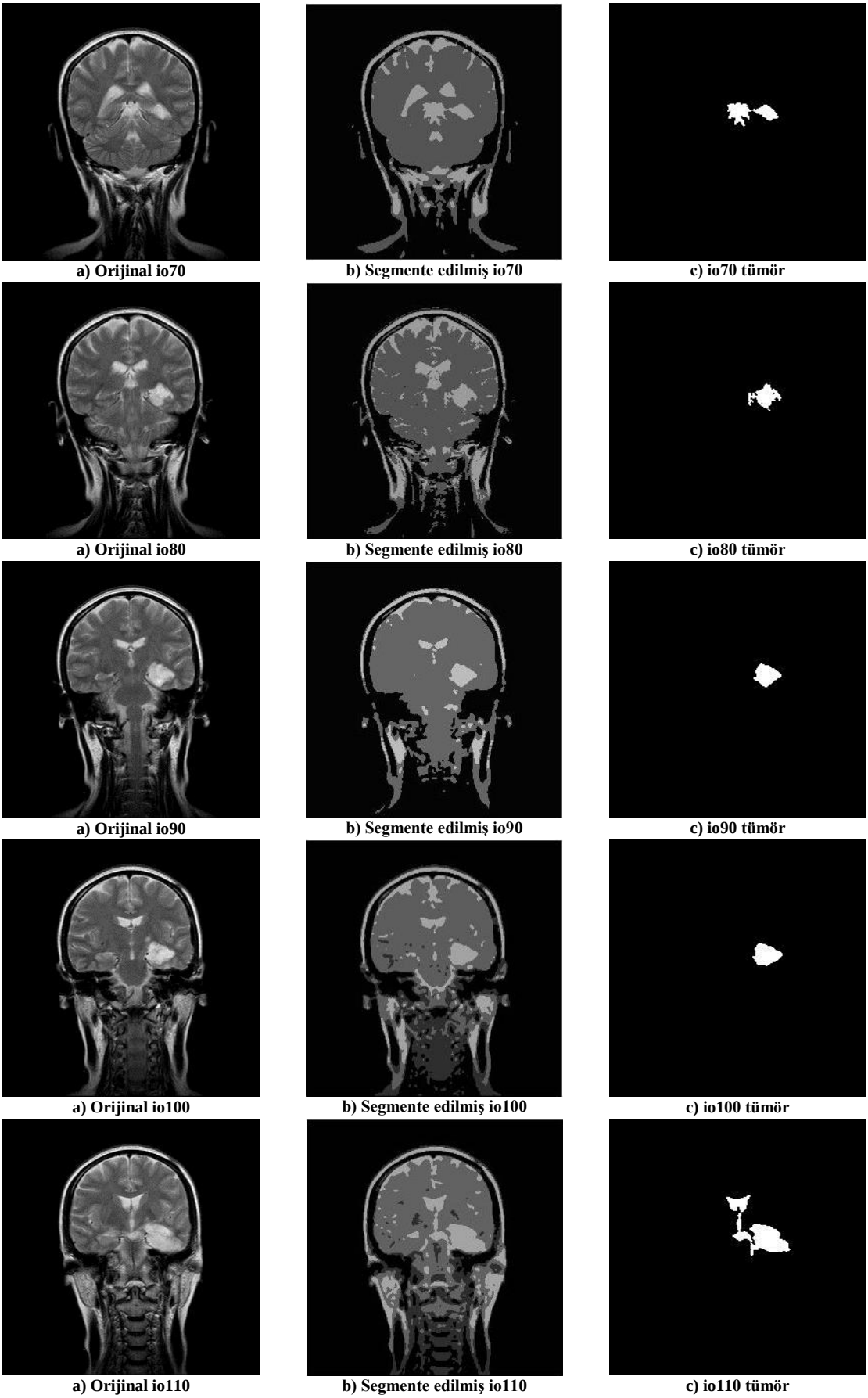
a) Orijinal io120 **b) Segmente edilmiş io120** **c) io120 tümör**

a) Orijinal io130 **b) Segmente edilmiş io130** **c) io130 tümör**

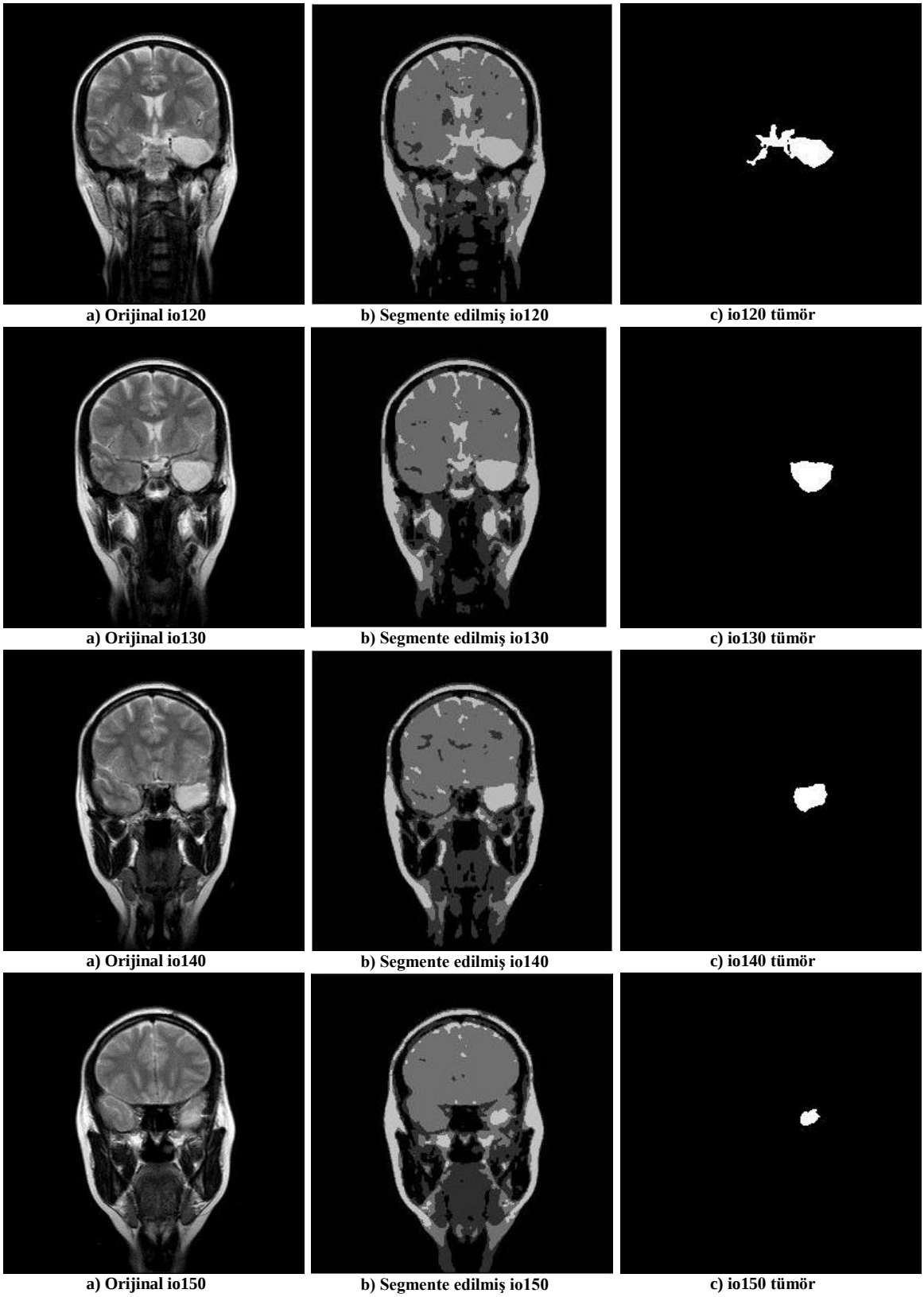
a) Orijinal io140 **b) Segmente edilmiş io140** **c) io140 tümör**

a) Orijinal io150 **b) Segmente edilmiş io150** **c) io150 tümör**

Şekil 3.4. **Devamı (io110-io150).**



Şekil 3.5. K-means tabanlı segmentasyon (io70-io110) sonuçları.



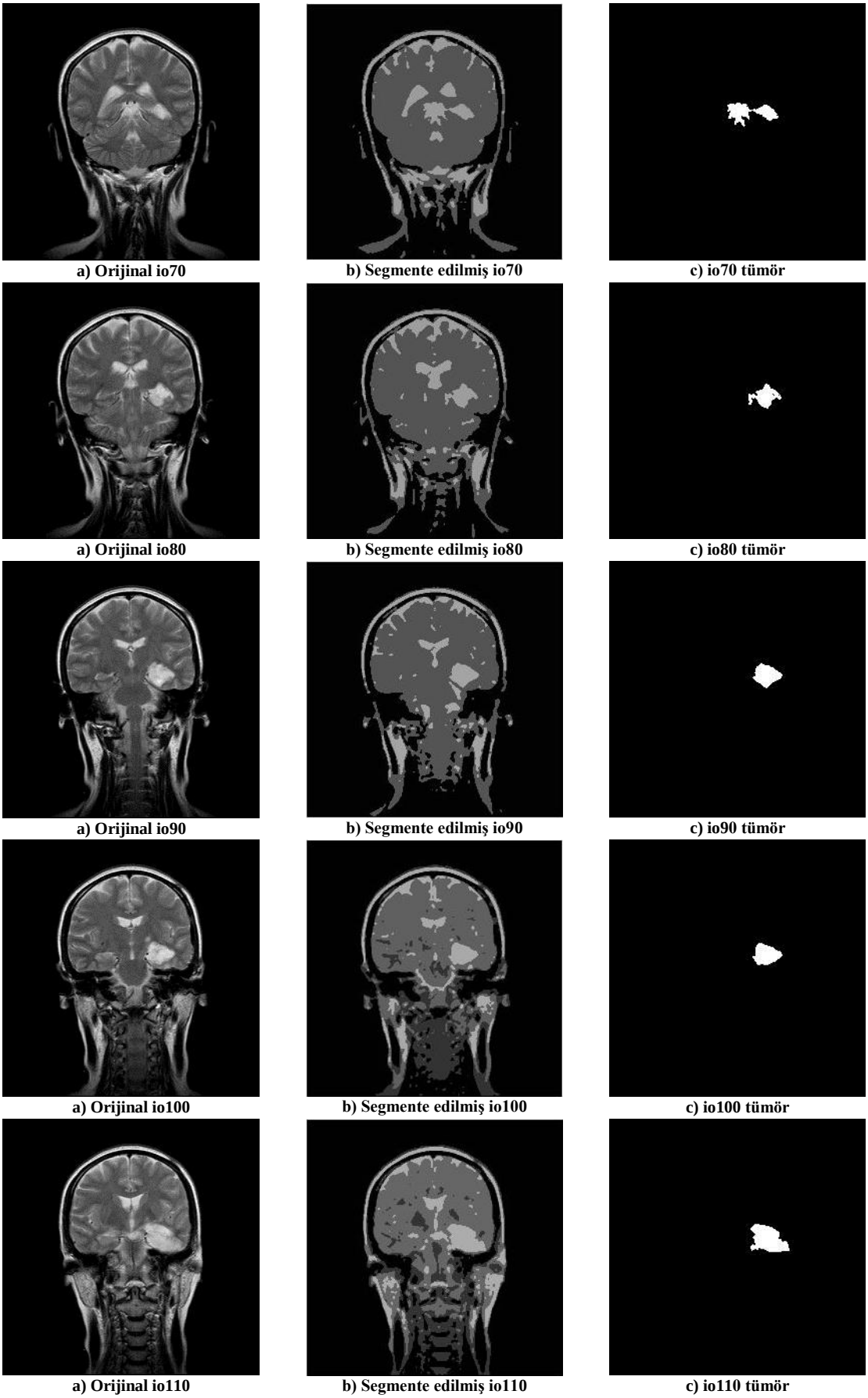
a) Orijinal io120 **b) Segmente edilmiş io120** **c) io120 tümör**

a) Orijinal io130 **b) Segmente edilmiş io130** **c) io130 tümör**

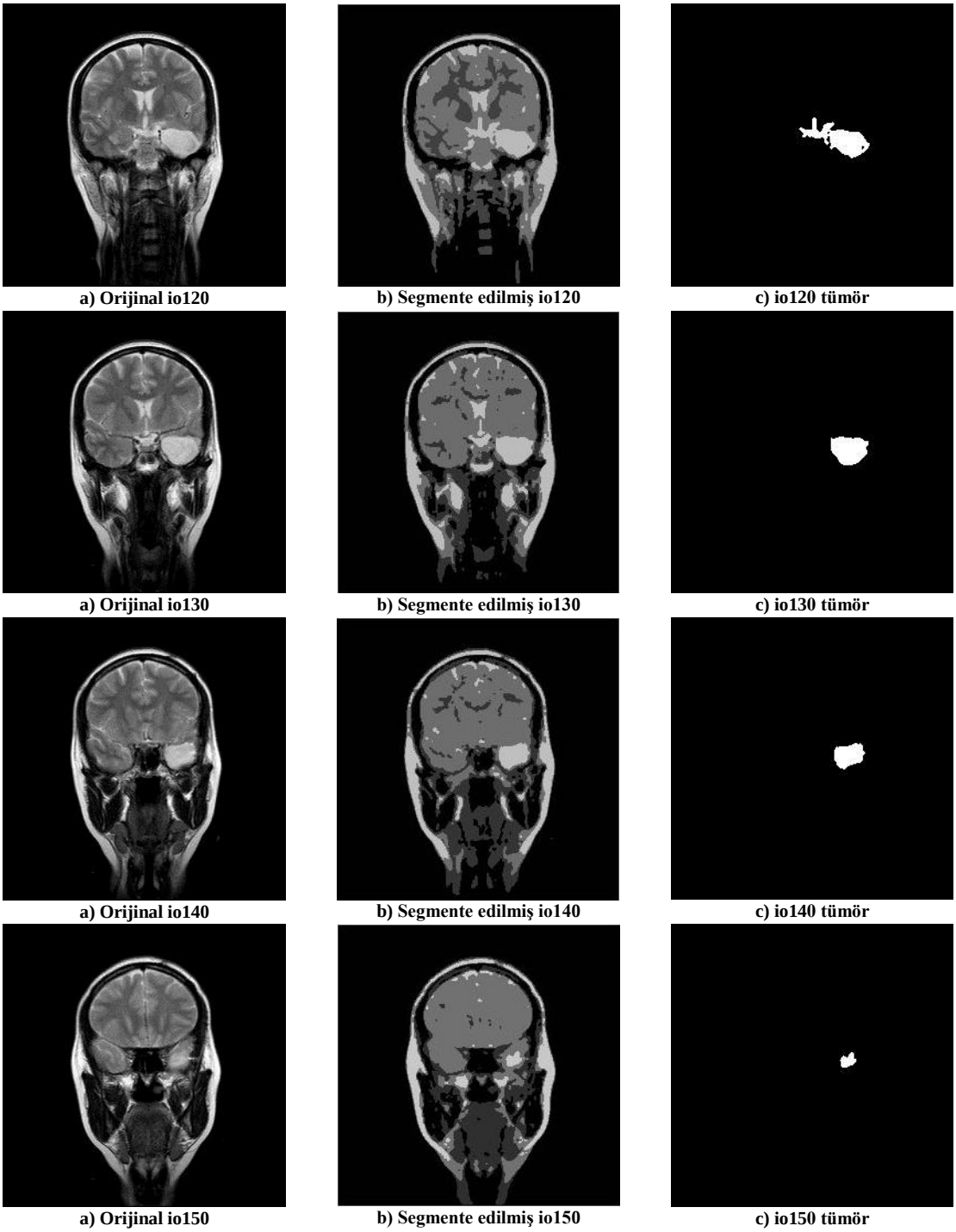
a) Orijinal io140 **b) Segmente edilmiş io140** **c) io140 tümör**

a) Orijinal io150 **b) Segmente edilmiş io150** **c) io150 tümör**

Şekil 3.5. **Devamı (io120-io150).**



Şekil 3.6. FCM tabanlı segmentasyon (io70-io110) sonuçları.



a) Orijinal io120 b) Segmente edilmiş io120 c) io120 tümör

a) Orijinal io130 b) Segmente edilmiş io130 c) io130 tümör

a) Orijinal io140 b) Segmente edilmiş io140 c) io140 tümör

a) Orijinal io150 b) Segmente edilmiş io150 c) io150 tümör

Şekil 3.6. **Devamı (io120-io150).**

3.2. Yeni Önerilen Kriter ile ABC Tabanlı Görüntü Kümeleme

Bu kısımda yeni bir görüntü kümeleme kriteri (amaç fonksiyonu) geliştirilmiş ve mevcut amaç fonksiyonları ile ABC, PSO ve GA algoritmaları üzerinden karşılaştırılarak etkinliği ortaya konulmuştur.

Evrimsel hesaplama teknikleri ile görüntü kümelemede en çok kullanılan amaç fonksiyonlarından biri Denklem 3.1'de verilmiştir. Ancak Denklem 3.1'de ağırlıklandırma parametrelerinin görüntüye uygun olarak belirlenmesi gerekmektedir. Bunun yanında bu amaç fonksiyonu Bölüm 3.1 ve [242, 243]'den görüleceği üzere Je hatasını minimize etmede çok iyi performans sağlayamamaktadır. Denklem 3.1'in parametre sorunu ilgili bileşenlerin Denklem 3.5 formatına dönüşümü ile ortadan kaldırılmıştır [29].

$$fit_2(x_i, Z) = \frac{d_{\max}(Z, x_i) + Je_i}{d_{\min}(Z, x_i)} \tag{3.5}$$

Her ne kadar Denklem 3.5 parametre gerektiren bir amaç fonksiyonu olmasa da Denklem 3.5'in performansı Denklem 3.1'e çok yakındır ve Je hatası minimize etmede problemler söz konusu olabilmektedir [29, 244, 245]. Daha güçlü bir amaç fonksiyonuna ihtiyaç vardır. Buradan yola çıkarak amaç fonksiyonu (Pfit) olarak kullanılmak üzere yeni bir görüntü kümeleme kriteri geliştirilmiştir [246]. Pfit diğerlerinden farklı olarak ortalama hata kareleri toplamı ve kuantalama hatasını beraber etkili bir biçimde kullanmaktadır:

$$Pfit(x_i, Z) = Je_i * \frac{d_{max}(Z, x_i)}{d_{\min}(Z, x_i)} * (d_{max}(Z, x_i) + z_{max} - d_{\min}(Z, x_i) + MSE_i) \tag{3.6}$$

Burada MSE_i ortalama hata kareler toplamını ifade eder ve Denklem 3.7 ile tanımlanır.

$$MSE = \frac{1}{N} \sum_{k=1}^{K} \sum_{\forall z_p \in C_k} d(z_p, m_k)^2 \tag{3.7}$$

Denklem 3.6'nın ABC ile görüntü kümelemeye uygulanması Şekil 3.2'deki uygulamanın aynısıdır. Sadece Denklem 3.1'in yerini Denklem 3.6 almaktadır.

3.2.1. Kümeleme Deneysel Dizaynı

Karşılaştırmalı çalışmalarda 256×256 piksellik Morro Bay, 380×261 piksellik House, 468×444 piksellik MRI, 512×512 piksellik Lena ve Airplane, ve Berkeley segmentasyon görüntü setinden 481×321 piksellik Ship ve Bird görüntüleri kullanılmıştır. İlgili görüntüler Şekiller 3.7.a-3.13.a'da sunulmuştur. Görüntüler ihtiva ettikleri bölge çeşitliliği, farklılık ve yoğunluk gibi kriterler temel alınarak seçilmiştir.

ABC, PSO ve GA algoritmaları üzerinden *Pfit* kümeleme fonksiyonun karşılaştırılması için mevcut fit1 ve fit2 fonksiyonlarının yanında CS indeksi de amaç fonksiyonu olarak kullanılmıştır:

$$CSfit(x_i, Z) = CS = \frac{\sum_{k=1}^{K} \frac{1}{n_k} \sum_{z_p \in C_k} \max_{z_q \in C_k} d(z_p, z_q)}{\sum_{k=1}^{K} \min_{j \in K, j \neq k} d(m_k, m_j)} \tag{3.8}$$

ABC, PSO ve GA algoritmalarının tümünde popülasyon büyüklüğü 30 ve maksimum çevrim sayısı 250 olarak belirlenmiştir. Dolayısıyla K-means ve FCM algoritmalarında maksimum değerlendirme sayısı Bölüm 3.1'de olduğu gibi 7500 (30 popülasyon büyüklüğü × 250 çevrim sayısı) olarak alınmıştır. ABC algoritmasında limit değeri Bölüm 3.1'deki gibi 75 olarak alınmıştır. PSO'nun parametre değerleri, ilgili çalışmadaki [29] gibi $w_{init}=0.72$, $w_{end}=0.4$, $c_1=c_2=1.49$ ve $V_{max}=255$ olarak belirlenmiştir. GA algoritmasında çaprazlama ve mutasyon oranları sırasıyla en çok kullanılan değerler 0.8 ve 0.2 alınmıştır. FCM algoritmasında μ katsayısı en çok tercih edilen değer olarak bilinen 2 alınmıştır. fit1 fonksiyonundaki w_1, w_2 ve w_3 ağırlıklandırma değerleri sırasıyla 0.4, 0.2 ve 0.4 olarak belirlenmiştir. Tüm görüntü verileri için *K* küme sayısı 5 olarak belirlenmiştir. Tüm görüntü verilerinde aynı küme sayısının belirlenmesindeki temel amaç elde edilecek sayısal kümeleme sonuçlarının belirli aralıkta sınırladırılması ve dolayısıyla daha kolay değerlendirilmesidir.

Kümeleme sonuçlarının değerlendirilmesinde, önceki çalışmada olduğu gibi J_e, d_{max} ve d_{min} bileşenleri ve DBI indeksi kullanılmıştır. Ancak, CS indeksi bu çalışmada amaç fonksiyonu olarak kullanılması sebebiyle kümeleme sonuçlarını değerlendirme işleminde kullanılmamıştır. CS indeksi yerine XBI indeksi kümeleme kalitesini değerlendirmek amacıyla bu çalışmaya dahil edilmiştir. XBI indeksi, küme içi farklar karesi toplamının (SSW) ve küme ortalamalarının genel ortalamadan farklarının karelerinin kümelerdeki eleman sayısı ile çarpımlarının toplamına (SSB) oranını temel alır. XBI indeksin matematiksel formülü Tablo 1.1'de sunulmuştur.

3.2.2. Kümeleme Deneysel Sonuçları

Deneysel sonuçlar dört kategoride ele alınmıştır: 1) ABC, PSO ve GA algoritmalarının elde edilen amaç fonksiyon değerlerine göre karşılaştırılması; 2) amaç fonksiyonlarının (fit1, fit2, Pfit ve CSfit) elde ettiği kümeleme sonuçlarının kompaktlık-ayrıklık (Je, dmax ve dmin) kriterleri üzerinden değerlendirilmesi; 3) amaç fonksiyonların ürettiği

kümelerin kalitesinin DBI ve XBI doğrulama indeksleri üzerinden değerlendirilmesi; ve 4) ABC, PSO ve GA algoritmalarının Pfit amaç fonksiyonu sonuçlarının klasik kümeleme yöntemleri ile karşılaştırılması.

Elde edilen sonuçlar birbirinden bağımsız 30 koşma üzerinden ortalama ve standart sapma değerleri ile tablolarda sunulmuştur. Tablolardaki en iyi değerler kalın fontta ve standart sapma değerleri parantez içinde gösterilmiştir. Deneyler 2.4 Ghz çift çekirdekli işlemci ve 4x8 GB RAM'e sahip bilgisayarda koşulmuştur. Koşmaların neticesinde algoritmaların hesaplama zamanlarının birbirine çok yakın olduğu sonucuna varılmıştır.

Tablo 3.2. Amaç fonksiyon sonuçları.

		Pfit	fit1	fit2	CSfit
Airplane	**GA**	924.596 (133.1)	47.328 (1.741)	0.570 (0.059)	0.520 (0.021)
	PSO	**777.767 (8.676)**	46.175 (0.273)	0.500 (0.013)	0.501 (0.003)
	ABC	790.618 (8.789)	**46.159 (0.079)**	**0.498 (0.005)**	**0.499 (0.001)**
House	**GA**	1192.324 (212.8)	51.732 (0.993)	0.545 (0.053)	0.619 (0.020)
	PSO	1086.34 (137.14)	51.116 (0.841)	0.505 (0.047)	0.606 (0.003)
	ABC	**1070.08 (11.840)**	**50.795 (0.155)**	**0.490 (0.008)**	**0.605 (0.001)**
Lena	**GA**	1025.065 (100.0)	50.666 (0.616)	0.544 (0.044)	0.596 (0.020)
	PSO	**899.365 (11.464)**	50.087 (0.236)	0.503 (0.011)	**0.577 (0.001)**
	ABC	919.307 (13.987)	**50.065 (0.088)**	**0.503 (0.006)**	0.578 (0.001)
Morro Bay	**GA**	998.267 (130.4)	50.741 (0.522)	0.483 (0.023)	0.619 (0.020)
	PSO	898.866 (64.64)	50.168 (0.186)	0.458 (0.008)	0.605 (0.003)
	ABC	**886.052 (9.825)**	**50.097 (0.104)**	**0.454 (0.006)**	**0.604 (0.001)**
MRI	**GA**	595.781 (61.69)	40.929 (0.466)	0.490 (0.055)	0.525 (0.041)
	PSO	523.751 (16.78)	**40.448** (0.096)	0.448 (0.005)	0.496 (0.009)
	ABC	**523.145 (7.282)**	40.490 **(0.073)**	**0.446 (0.003)**	**0.490 (0.001)**
Bird	**GA**	714.839 (45.81)	52.089 (1.065)	0.603 (0.092)	0.596 (0.011)
	PSO	**667.983** (9.110)	**50.768 (0.170)**	0.488 (0.011)	**0.523 (0.001)**
	ABC	673.370 **(7.785)**	50.789 (0.058)	**0.487 (0.005)**	0.525 **(0.001)**
Ship	**GA**	1332.13 (202.39)	51.675 (0.562)	0.531 (0.029)	**0.561** (0.004)
	PSO	1202.15 (24.536)	51.074 (0.277)	0.506 (0.016)	0.585 **(0.001)**
	ABC	**1192.57 (16.602)**	**51.064 (0.104)**	**0.505 (0.007)**	0.586 **(0.001)**

3.2.2.1. Amaç Fonksiyon Minimizasyonu

ABC, PSO ve GA algoritmalarının her bir amaç fonksiyon için elde ettiği minimum değerler ortalama ve standart sapma üzerinden Tablo 3.2'de verilmiştir. Tablo 3.2'ye göre ABC ve PSO tüm amaç fonksiyonlarının minimize edilmesinde GA'ya göre önemli ölçüde daha iyi performans sağlamıştır. GA sadece CSfit amaç fonksiyonu ile

Ship görüntüsünün kümelenmesinde en iyi değeri elde etmiştir. ABC ve PSO algoritmalarının performansları fit1, fit2 ve CSfit fonksiyonlarında genellikle birbirine çok yakındır. Örneğin, Airplane görüntüsünde fit1 amaç fonksiyonu için ABC yöntemi 46.159 ortalama değerini elde ederken, PSO yöntemi 46.175 ortalama değerini elde etmiştir. Pfit amaç fonksiyonu için ABC ve PSO yöntemleri arasındaki fark biraz daha fazladır. Örneğin, Morro Bay görüntüsünde ABC 886.052 değeri elde ederken, PSO 898.866 değeri elde etmiştir. ABC genellikle PSO'ya göre daha iyi ortalama değerler elde etmiştir. Bununla beraber ABC'nin elde ettiği değerlerin standart sapmaları hemen hemen bütün durumlarda PSO'dan daha iyidir. Dolayısıyla ABC, PSO'ya göre amaç fonksiyon minimizasyonunda daha kararlı bir algoritmadır.

3.2.2.2. Amaç Fonksiyonlarının Karşılaştırılması

Amaç fonksiyonlarına ait Je, dmax, dmin, DBI ve XBI değerleri ABC, PSO ve GA algoritmaları üzerinden Tablo 3.3'te sunulmuştur. Tablo 3.3'te bulunan '+' ('-') sembolü, ele alınan değerlendirme kriterinde önerilen Pfit fonksiyonunun karşılaştırılan diğer fonksiyonundan daha iyi (daha kötü) olduğunu göstermektedir. '=' sembolü, ele alınan değerlendirme kriterinde Pfit ile ilgili fonksiyonlar arasında farklılık olmadığını ifade etmektedir.

Amaç fonksiyonları kümeleme kriterleri üzerinden değerlendirildiğinde, fitCS fonksiyonunun diğer fonksiyonlara nazaran kümeleme peformansı oldukça düşük seviyededir. Gerek kompaktlık kriterleri (Je ve dmax) gerekse doğrulama indeks (XBI ve DBI) değerleri oldukça kötüdür. Örneğin, House görüntüsünde FitCS dmax değerini 26.477 olarak bulurken, fit1, fit2 ve Pfit amaç fonksiyonları dmax değerini sırasıyla 11.942, 11.963 ve 11.427 olarak bulmuştur. Sadece ayrıklık (dmin) kriterinde diğerlerine nispeten iyi sonuçlar sağlamıştır. Ancak dmin değerinin iyi olması tek başına bir anlam ifade etmemektedir. fit1 ve onun parametresiz versiyonu fit2 fonksiyonlarına bakıldığında, fit1 ve fit2 ile elde edilen bütün kümeleme kriterleri değerleri birbirine çok yakındır. Örneğin, Lena görüntüsünde fit1 fonksiyonunun elde ettiği dmax ve DBI değerleri 11.120 ve 0.1584 iken, fit2 fonksiyonunun dmax ve DBI kriterleri için elde ettiği değerler sırasıyla 11.096 ve 0.1586'dır. Dolayısıyla birinin diğerine açık bir üstünlüğü söz konusu değildir. Önerilen Pfit fonksiyonu diğer fonksiyonlara göre Je, dmax, DBI ve XBI kriterlerinde hem ortalama hem de standart

sapma değerleri olarak daha iyi sonuç sağlamıştır. Hatta Je, dmax ve XBI sonuçlarının tamamına yakınında ve DBI sonuçlarının büyük bir çoğunluğunda, Pfit fonksiyonunun diğer fonksiyonlardan üstünlüğü Wilcoxon Rank Sum Test istatistiğine göre de bütün evrimsel hesaplama tekniklerinde rahatlıkla görülmektedir. Pfit fonksiyonu sadece ayrıklık (dmin) kriterinde diğerlerine nispeten iyi sonuç sağlamamaktadır. Ancak diğer kriterlerde özellikle doğrulama indekslerindeki başarısı sebebiyle bu durum gözardı edilebilir. Tüm yapılan analizler neticesinde Pfit fonksiyonunun iyi ayrıştırılmış ve kompakt kümeler elde etmede dmin kriterindeki dezavantajına rağmen en uygun amaç fonksiyonu olduğu görülmüştür.

3.2.2.3. Evrimsel Tekniklerin Karşılaştırılması

Tablo 3.3'te görüleceği üzere Je ve dmax kriterlerini minimize etme açısından ABC tüm amaç fonksiyonlarında genellikle en iyi performansı sağlamıştır. Her ne kadar PSO ve GA algoritmaları dmin kriterini maksimize etmede ABC'den daha iyi olsalar da ABC algoritması kümeleme kalitesini teyit eden DBI ve XBI indekslerinde ve kompaktlık kriterlerinde daha iyi performans sergilemektedir. Örneğin, House görüntüsü için fit2 fonksiyonunda ABC'nin DBI ve XBI kalite indekslerinde elde ettiği değerler sırasıyla 0.1468 ve 0.2992 iken, PSO ve GA algoritmalarının elde ettiği DBI değerleri sırasıyla 0.1610, 0.170 ve XBI değerleri sırasıyla 0.3226 ve 0.3520'dir. Ayrıca, ABC standart sapma değerleri olarak da PSO ve GA'dan hemen hemen bütün koşullarda daha iyi sonuçlar elde etmiştir. Dolayısıyla kararlılık açısından da ABC algoritmasının PSO ve GA algoritmalarından daha üstün olduğu söylenebilir. PSO ve GA algoritmaları birbirleri ile karşılaştırıldığında ise, PSO GA'dan hem ortalama hem de standart sapma değerleri olarak daha iyi sonuçlar sağlamaktadır. Sonuç olarak kullanılan evrimsel hesaplama teknikleri arasında ABC amaç fonksiyon minimizasyonu ve kümeleme kalitesi sağlamada bütün amaç fonksiyonlarında (önerilen Pfit dahil) en uygun algoritmadır.

3.2.2.4. Klasik Yöntemler ile Karşılaştırma

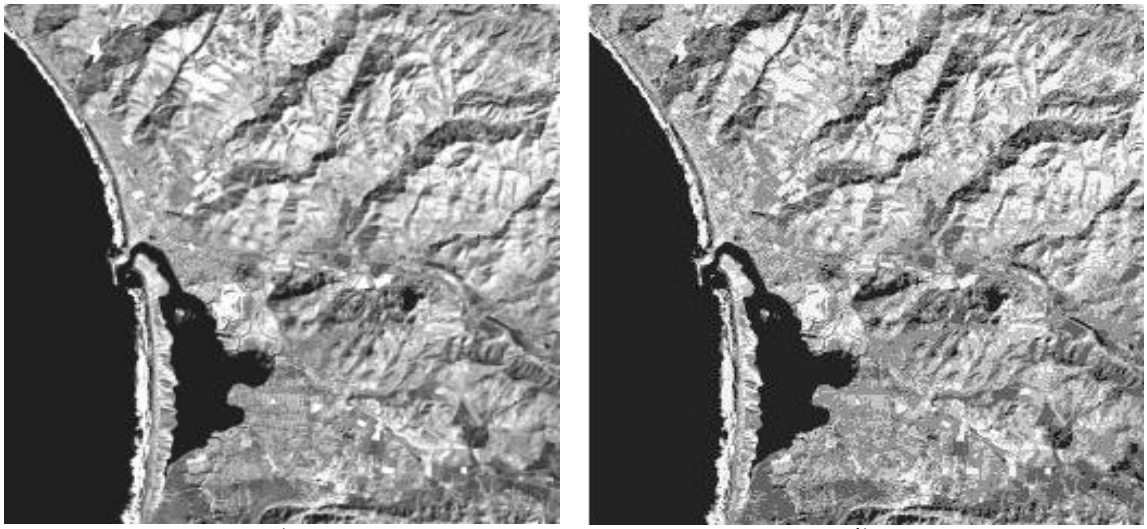
Önerilen Pfit tabanlı ve klasik kümeleme yöntemlerinin sonuçları ortalama ve standart sapma değerleri üzerinden Tablo 3.4'te sunulmuştur. Elde edilen sonuçlara göre Pfit tabanlı yöntemler klasik yöntemlerden dmax ve dmin kümeleme kriterlerinde genel olarak daha iyi performans sağlamıştır. Özellikle dmin kriterinde Pfit tabanlı yöntemler

klasik yöntemlerden çok iyi sonuçlar almıştır. Örneğin, MRI görüntüsünde Pfit tabanlı ABC, PSO ve GA yöntemleri sırasıyla 41.483, 41.578 ve 40.369 ortalama dmin değerleri elde ederken, K-means ve FCM yöntemleri bu kriter için ortalama değer olarak 21.698 ve 21.152 elde etmiştir. Tüm görüntü setlerinde dmin kriteri için Pfit tabanlı yöntemler ve klasik yöntemler arasında bu denli büyük farklar söz konusudur. Dolayısıyla kümeleri birbirinden ayırmada klasik yöntemlerin iyi sonuçlar elde edemediği ve Pfit tabanlı yöntemlerin daha başarılı olduğu açık bir şekilde görülmektedir. Je kriterini minimize etmede Pfit tabanlı ABC yöntemi Airplane, MRI, Morro Bay ve Bird görüntü setlerinde en iyi sonucu elde etmiştir. Dolayısıyla fit1 ve fit2'nin Je minimizasyonunda yaşadığı problemler Pfit ile önemli ölçüde çözülmüştür. Kümeleme doğrulama indekslerinde ise Pfit tabanlı yöntemler DBI indeksinde daha iyi sonuçlar elde etmiştir. Ancak XBI indeksinde klasik yöntemler daha iyi sonuçlar elde etmiştir. Bunun sebebi klasik yöntemlerin direkt olarak küme içi uzaklığın minimizasyonunu temel alması ve XBI indeksinin de küme içi uzaklıklar toplamı bileşenine sahip olması olabilir. Elde edilen kümeleme sonuçları genel olarak değerlendirildiğinde, Pfit tabanlı yöntemler klasik yöntemlerden daha iyi kümeler oluşturmuş ve Je kriterinin minimize edilmesi ile ilgili problemleri büyük oranda çözmüştür.

Pfit tabanlı ABC yöntemi ile görsel kümeleme sonuçları Şekiller 3.7.b-3.13.b'de sunulmuştur. Görsel verilere bakıldığında da önerilen Pfit fonksiyonunun makul kümeleme sonuçlarına ulaştığını söylemek mümkündür.

a) b)

Şekil 3.7. a) Orijinal ve b) ABC-Pfit ile kümelenmiş Airplane.

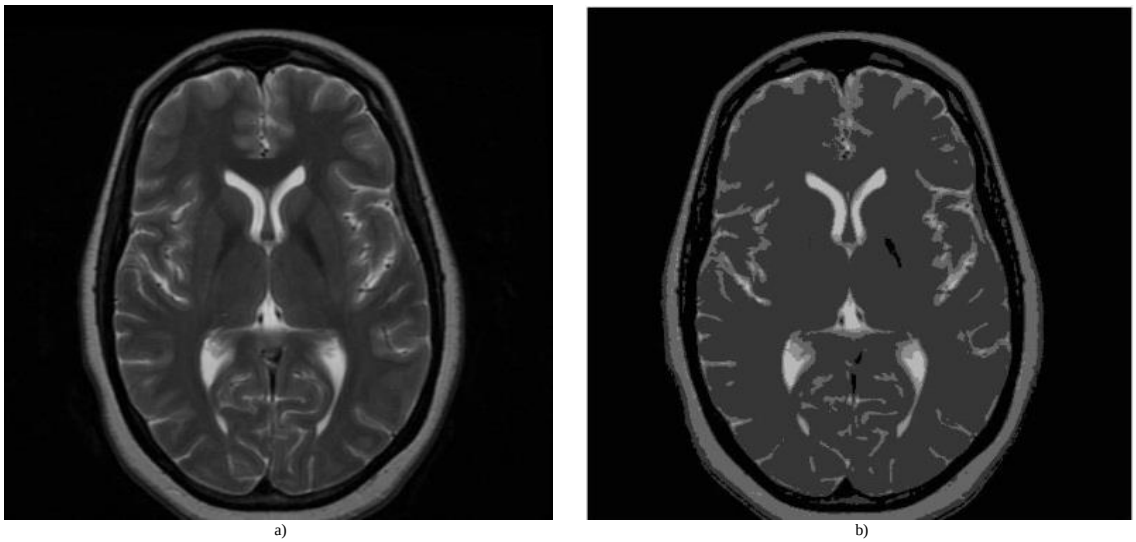


a) b)

Şekil 3.8. a) Orijinal ve b) ABC-Pfit ile kümelenmiş Morro Bay.

a) b)

Şekil 3.9. a) Orijinal ve b) ABC-Pfit ile kümelenmiş House.



a) b)

Şekil 3.10. a) Orijinal ve b) ABC-Pfit ile kümelenmiş MRI.

a) b)

Şekil 3.11. a) Orijinal ve b) ABC-Pfit ile kümelenmiş Lena.

a) b)

Şekil 3.12. a) Orijinal ve b) ABC-Pfit ile kümelenmiş Bird.

Şekil 3.13. a) Orijinal ve b) ABC-Pfit ile kümelenmiş Ship.

Tablo 3.3. Kümeleme değerlendirme kriterleri sonuçları.

	Fits.	ABC					PSO					GA				
		Je	**Dmax**	**Dmin**	**DBI**	**XBI**	**Je**	**Dmax**	**Dmin**	**DBI**	**XBI**	**Je**	**Dmax**	**Dmin**	**DBI**	**XBI**
Airplane	**FitCS**	19.701 (0.154) '+'	36.982 (0.912) '+'	49.602 (1.487) '-'	0.2401 (0.0117) '+'	0.4951 (0.0223) '+'	19.760 (0.296) '+'	34.768 (1.903) '+'	51.883 (3.199) '-'	0.2276 (0.0069) '+'	0.4699 (0.0503) '+'	21.109 (1.687) '+'	36.644 (6.005) '+'	49.180 (9.033) '-'	0.3254 (0.1452) '+'	0.8844 (0.4591) '+'
	fit1	9.889 (0.204) '+'	11.016 (0.438) '+'	42.013 (1.279) '-'	0.1590 (0.0021) '+'	0.2826 (0.0340) '+'	9.913 (0.356) '+'	11.212 (0.493) '+'	42.373 (1.286) '-'	0.1592 (0.0045) '+'	0.2792 (0.0377) '+'	10.484 (1.017) '+'	12.051 (1.363) '+'	40.689 (2.739) '='	0,1739 (0.0222) '+'	0.3101 (0.0624) '+'
	fit2	9.934 (0.362) '+'	11.077 (0.572) '+'	42.105 (1.651) '-'	0.1586 (0.0023) '+'	0.2828 (0.0349) '+'	10.081 (0.338) '+'	11.513 (0.471) '+'	43.191 (1.349) '-'	0.1597 (0.0036) '+'	0.2770 (0.0512) '+'	11.336 (1.446) '+'	12.948 (2.081) '+'	42.546 (3.400) '-'	0,1799 (0.0227) '+'	0.3287 (0.0627) '+'
	Pfit	9.710 (0.046)	10.734 (0.322)	40.531 (1.234)	0.1569 (0.0008)	0.2533 (0.0111)	9.723 (0.169)	10.835 (0.361)	41.329 (1.202)	0.1571 (0.0012)	0.2566 (0.0153)	10.083 (0.502)	11.435 (0.659)	39.697 (2.302)	0,166 (0.0126)	0.2535 (0.0236)
House	**FitCS**	18.288 (0.146) '+'	26.447 (1.173) '+'	56.087 (2.214) '-'	0.1973 (0.0047) '+'	0.4808 (0.0118) '+'	18.094 (0.377) '+'	27.471 (2.092) '+'	54.377 (4.998) '-'	0.1943 (0.0048) '+'	0.5464 (0.0966) '+'	18.081 (0.979) '+'	28.084 (4.547) '+'	49.575 (7.969) '-'	0.2174 (0.0389) '+'	0.5912 (0.2031) '+'
	fit1	11.347 (0.291) '+'	11.942 (0.416) '+'	47.601 (1.353) '-'	0.1470 (0.0033) '+'	0.2998 (0.0086) '+'	11.604 (0.776) '+'	12.436 (0.976) '+'	47.498 (2.709) '-'	0.1636 (0.0298) '+'	0.3179 (0.0429) '+'	11.636 (0.783) '+'	13.024 (0.887) '+'	45.661 (4.351) '-'	0.1780 (0.0039) '+'	0.3322 (0.0553) '+'
	fit2	11.319 (0.306) '+'	11.963 (0.397) '+'	47.494 (1.377) '-'	0.1468 (0.0029) '+'	0.2992 (0.0089) '+'	11.842 (0.905) '+'	12.714 (1.391) '+'	48.659 (1.829) '-'	0.1610 (0.0260) '+'	0.3226 (0.0514) '+'	12.410 (1.140) '+'	13.919 (1.722) '+'	48.349 (3.231) '-'	0.171 (0.0028) '+'	0.3520 (0.0478) '+'
	Pfit	10.818 (0.109)	11.427 (0.099)	44.263 (1.611)	0.1452 (0.0027)	0.2759 (0.0083)	10.821 (0.148)	11.483 (0.468)	44.680 (1.986)	0.1506 (0.0229)	0.2831 (0.0279)	11.021 (0.365)	11.950 (0.738)	43.647 (3.136)	0.1570 (0.0272)	0.3161 (0.0625)
Lena	**FitCS**	18.251 (0.118) '+'	26.551 (0.797) '+'	59.892 (1.223) '-'	0.2059 (0.0034) '+'	0.5023 (0.0046) '+'	18.221 (0.106) '+'	26.471 (0.771) '+'	60.429 (1.183) '-'	0.2046 (0.0034) '+'	0.5003 (0.0046) '+'	18.879 (1.579) '+'	30.369 (10.31) '+'	52.387 (9.332) '-'	0.2831 (0.2452) '+'	0.6689 (0.2935) '+'
	fit1	10.209 (0.384) '+'	11.120 (0.472) '+'	42.333 (1.598) '-'	0.1584 (0.0042) '+'	0.2474 (0.0112) '+'	10.608 (0.462) '+'	11.496 (0.609) '+'	43.773 (1.135) '-'	0.1628 (0.0049) '+'	0.2589 (0.0192) '+'	10.809 (0.973) '+'	11.915 (1.341) '+'	42.117 (2.372) '-'	0.1660 (0.0121) '='	0.2852 (0.0386) '+'
	fit2	10.139 (0.303) '+'	11.096 (0.362) '+'	41.987 (1.199) '-'	0.1586 (0.0058) '+'	0.2416 (0.0083) '+'	10.537 (0.533) '+'	11.605 (0.707) '+'	44.003 (1.599) '-'	0.1625 (0.0057) '+'	0.2621 (0.0174) '+'	11.241 (0.916) '+'	12.392 (1.285) '+'	43.893 (3.124) '-'	0.1690 (0.0094) '+'	0.3062 (0.0490) '+'
	Pfit	9.736 (0.108)	10.676 (0.133)	39.955 (0.967)	0.1548 (0.0044)	0.2297 (0.0047)	9.793 (0.198)	10.691 (0.224)	40.564 (0.503)	0.1554 (0.0045)	0.2361 (0.0083)	10.211 (0.489)	11.046 (0.583)	40.223 (1.945)	0.1629 (0.0073)	0.2585 (0.0287)
Morro Bay	**FitCS**	17.611 (0.063) '+'	31.967 (0.075) '+'	57.413 (0.724) '-'	0.1797 (0.0016) '+'	0.3268 (0.0013) '+'	17.659 (0.031) '+'	31.981 (0.001) '+'	58.332 (0.347) '-'	0.1804 (0.0009) '+'	0.3264 (0.0007) '+'	17.287 (0.942) '+'	30.256 (3.296) '+'	50.074 (7.947) '-'	0.2071 (0.0508) '+'	0.3478 (0.0535) '+'
	fit1	10.192 (0.479) '+'	12.285 (0.766) '+'	49.463 (1.828) '-'	0.1370 (0.0042) '='	0.1626 (0.0112) '+'	10.479 (0.455) '+'	13.196 (0.915) '+'	51.509 (2.282) '-'	0.1422 (0.0055) '+'	0.1839 (0.0128) '+'	10.949 (1.061) '+'	13.551 (1.007) '+'	50.294 (2.974) '-'	0.1468 (0.0098) '+'	0.1853 (0.0171) '+'
	fit2	10.010 (0.319) '+'	12.084 (0.754) '+'	48.688 (1.718) '-'	0.1361 (0.0039) '='	0.1566 (0.0057) '+'	10.623 (0.653) '+'	13.298 (0.929) '+'	52.196 (2.512) '-'	0.1423 (0.0057) '+'	0.1759 (0.0150) '+'	11.214 (1.010) '+'	13.824 (0.983) '+'	51.808 (2.566) '-'	0.1480 (0.0088) '+'	0.1822 (0.0193) '+'
	Pfit	9.696 (0.137)	11.710 (0.167)	46.955 (0.466)	0.1352 (0.0013)	0.1536 (0.0019)	9.763 (0.248)	12.090 (0.873)	47.972 (0.999)	0.1360 (0.0047)	0.1559 (0.0093)	10.004 (0.285)	12.451 (1.173)	47.132 (1.733)	0.1412 (0.0100)	0.1673 (0.0207)

Tablo 3.3. **Devamı.**

	Fits.	ABC					PSO					GA				
		Je	**Dmax**	**Dmin**	**DBI**	**XBI**	**Je**	**Dmax**	**Dmin**	**DBI**	**XBI**	**Je**	**Dmax**	**Dmin**	**DBI**	**XBI**
MRI	**FitCS**	18.094 (0.029) '+'	47.993 (0.038) '+'	52.174 (0.378) '-'	0.1980 (0.001) '+'	0.3120 (0.002) '+'	18.163 (0.460) '+'	47.774 (2.035) '+'	50.344 (2.980) '-'	0.2009 (0.0087) '+'	0.3178 (0.0126) '+'	18.464 (1.572) '+'	48.753 (9.238) '+'	46.221 (10.07) '+'	0.2513 (0.1187) '+'	0.3912 (0.1324) '+'
	fit1	8.469 (0.255) '+'	11.476 (0.395) '+'	44.438 (1.336) '-'	0.1322 (0.0019) '-'	0.2653 (0.0069) '='	8.645 (0.361) '+'	11.651 (0.469) '+'	45.352 (1.596) '-'	0.1316 (0.0029) '-'	0.2668 (0.0065) '='	8.799 (0.473) '+'	11.621 (0.704) '+'	43.196 (2.753) '+'	0.1410 (0.0101) '='	0.2712 (0.0139) '='
	fit2	8.331 (0.303) '+'	11.243 (0.511) '+'	43.509 (1.846) '-'	0.1321 (0.0018) '-'	0.2654 (0.0060) '='	8.607 (0.317) '+'	11.644 (0.492) '+'	45.165 (1.430) '-'	0.1329 (0.0037) '='	0.2667 (0.0058) '='	9.026 (0.903) '+'	12.174 (1.781) '+'	43.316 (2.549) '+'	0.1465 (0.0175) '='	0.2715 (0.0382) '='
	Pfit	8.049 (0.102)	10.601 (0.373)	41.483 (1.194)	0.1338 (0.0024)	0.2648 (0.0051)	8.106 (0.175)	10.693 (0.282)	41.578 (0.922)	0.1327 (0.0014)	0.2661 (0.0096)	8.484 (0.327)	11.323 (0.619)	40.369 (2.763)	0.1442 (0.0096)	0.2685 (0.0386)
Bird	**FitCS**	18.752 (0.099) '+'	27.119 (0.409) '+'	62.944 (0.626) '-'	0.2213 (0.0010) '+'	0.3351 (0.0103) '+'	18.685 (0.038) '+'	27.105 (0.228) '+'	63.612 (0.443) '-'	0.2199 (0.0004) '+'	0.3363 (0.0036) '+'	17.763 (0.651) '+'	27.917 (5.161) '+'	51.688 (6.563) '-'	0.2133 (0.0379) '+'	0.5284 (0.1378) '+'
	fit1	8.944 (0.282) '+'	10.325 (0.263) '+'	39.596 (0.960) '-'	0.1554 (0.0026) '='	0.1349 (0.0136) '+'	9.375 (0.520) '+'	10.555 (0.436) '+'	41.018 (1.311) '-'	0.1574 (0.0057) '='	0.1514 (0.0277) '+'	10.920 (0.878) '+'	12.371 (1.607) '+'	41.134 (2.240) '-'	0.1799 (0.0191) '+'	0.2074 (0.0352) '+'
	fit2	8.892 (0.248) '+'	10.201 (0.327) '+'	39.142 (1.158) '-'	0.1569 (0.0036) '='	0.1327 (0.0059) '+'	9.488 (0.631) '+'	10.662 (0.563) '+'	41.251 (1.798) '-'	0.1582 (0.0055) '='	0.1517 (0.0240) '+'	11.667 (1.776) '+'	13.811 (3.085) '+'	42.231 (3.890) '-'	0.1912 (0.0310) '+'	0.2425 (0.0765) '+'
	Pfit	8.699 (0.053)	9.900 (0.158)	38.024 (0.766)	0.1558 (0.0025)	0.1281 (0.0025)	8.744 (0.139)	10.026 (0.254)	38.930 (0.863)	0.1558 (0.0022)	0.1286 (0.0036)	9.158 (0.718)	10.521 (0.990)	37.543 (2.878)	0.1655 (0.0173)	0.1486 (0.0314)
Ship	**FitCS**	17.781 (0.227) '+'	31.301 (2.218) '+'	51.320 (3.747) '-'	0.1987 (0.0101) '+'	0.5423 (0.0400) '+'	17.982 (0.371) '+'	29.626 (3.668) '+'	54.618 (6.064) '-'	0.2011 (0.0102) '+'	0.5318 (0.0384) '+'	17.731 (0.795) '+'	28.026 (5.214) '+'	50.584 (6.959) '-'	0.2180 (0.0511) '+'	0.5299 (0.1454) '+'
	fit1	11.149 (0.245) '+'	13.362 (0.660) '+'	48.699 (1.988) '-'	0.1537 (0.0041) '+'	0.2935 (0.0101) '+'	11.515 (0.399) '+'	13.931 (0.555) '+'	50.672 (1.999) '-'	0.1575 (0.0038) '='	0.3024 (0.0129) '+'	11.678 (0.950) '+'	13.251 (1.007) '='	46.487 (4.318) '-'	0.1612 (0.0142) '='	0.2991 (0.0311) '='
	fit2	11.201 (0.389) '+'	13.449 (0.961) '+'	48.836 (2.677) '-'	0.1542 (0.0055) '+'	0.2971 (0.0138) '+'	11.778 (0.864) '+'	14.117 (0.924) '+'	51.247 (2.515) '-'	0.1578 (0.0059) '='	0.3089 (0.0183) '+'	12.175 (1.093) '+'	14.104 (1.487) '+'	49.500 (4.406) '-'	0.1639 (0.0118) '+'	0.3180 (0.0332) '+'
	Pfit	10.895 (0.103)	12.276 (0.302)	44.716 (1.914)	0.1504 (0.0023)	0.2781 (0.0100)	10.872 (0.098)	12.079 (0.272)	41.917 (2.750)	0.1552 (0.0052)	0.2699 (0.0141)	11.277 (0.701)	13.082 (1.274)	45.798 (4.553)	0.1594 (0.0089)	0.2871 (0.0294)

Tablo 3.4. Pfit tabanlı ve klasik yöntemlerin kümeleme sonuçları.

		ABC *(Pfit)*	PSO *(Pfit)*	GA *(Pfit)*	K-means	FCM
Airplane	**Je**	**9.710** (0.046)	9.723 (0.169)	10.083 (0.502)	9.854 (0.121)	9.813 (2.6E-06)
	Dmax	**10.734** (0.322)	10.835 (0.361)	11.435 (0.659)	15.434 (0.577)	15.782 (2.040)
	Dmin	40.531 (1.234)	**41.329** (1.202)	39.697 (2.302)	21.167 (3.573)	18.043 (8.2E-06)
	DBI	**0.1569** (0.0008)	0.1571 (0.0012)	0.1660 (0.0126)	0.2330 (0.001)	0.2390 (1.2E-07)
	XBI	0.2533 (0.0111)	0.2566 (0.0153)	0.2535 (0.0236)	0.2048 (0.001)	**0.2008 (1.6E-07)**
House	**Je**	10.818 (0.109)	10.821 (0.148)	11.021 (0.365)	10.330 (0.061)	**10.248 (3E-08)**
	Dmax	**11.427** (0.099)	11.483 (0.468)	11.950 (0.738)	12.545 (0.783)	11.774 (0.030)
	Dmin	44.263 (1.611)	**44.680** (1.986)	43.647 (3.136)	31.833 (1.053)	31.477 (6.469)
	DBI	**0.1452** (0.0027)	0.1506 (0.0229)	0.157 (0.0272)	0.168 (0.001)	0.1649 (1.9E-09)
	XBI	0.2759 (0.0083)	0.2831 (0.0279)	0.3161 (0.0625)	0.2454 (0.000)	**0.2425 (1.22E-09)**
Lena	**Je**	9.736 (0.108)	9.793 (0.198)	10.211 (0.489)	9.430 (0.036)	**9.375 (3.6E-07)**
	Dmax	**10.676** (0.133)	10.691 (0.224)	11.046 (0.583)	11.712 (0.261)	11.588 (0.364)
	Dmin	39.955 (0.967)	**40.564** (0.503)	40.223 (1.945)	34.833 (0.461)	34.909 (4.3E-06)
	DBI	0.1548 (0.0044)	0.1554 (0.0045)	0.1629 (0.0073)	**0.1540** (0.001)	0.1545 **(7.3E-09)**
	XBI	0.2297 (0.0047)	0.2361 (0.0083)	0.2585 (0.0287)	0.2266 (5.9E-5)	**0.2242 (5.8E-09)**
Morro Bay	**Je**	**9.696** (0.137)	9.763 (0.248)	10.004 (0.285)	9.998 (0.076)	9.860 (3.6E-08)
	Dmax	**11.710** (0.167)	12.09 (0.873)	12.451 (1.173)	14.032 (1.263)	11.734 (3.432)
	Dmin	46.955 (0.466)	**47.972** (0.999)	47.132 (1.733)	41.565 (3.863)	41.386 (6.3E-06)
	DBI	**0.1352** (0.0013)	0.1360 (0.0047)	0.1412 (0.0100)	0.1480 (0.005)	0.1561 (8.3E-09)
	XBI	**0.1536** (0.0019)	0.1559 (0.0093)	0.1673 (0.0207)	0.1666 (0.0071)	0.1603 (2.6E-08)
MRI	**Je**	**8.049** (0.102)	8.106 (0.175)	8.484 (0.327)	8.142 (0.056)	8.212 (1.6E-07)
	Dmax	**10.601** (0.373)	10.693 (0.282)	11.323 (0.619)	17.198 (0.280)	16.032 (0.022)
	Dmin	41.483 (1.194)	**41.578** (0.922)	40.369 (2.763)	21.698 (0.492)	21.152 (3.0E-06)
	DBI	**0.1338** (0.0024)	0.1327 (0.0014)	0.1442 (0.0096)	0.1540 (0.001)	0.1606 (2.8E-08)
	XBI	0.2648 (0.0051)	0.2661 (0.0096)	0.2685 (0.0386)	0.1704 (0.0001)	**0.1687 (1.4E-08)**
Bird	**Je**	**8.699** (0.053)	8.744 (0.139)	9.158 (0.718)	8.989 (0.244)	8.777 (5.2E-08)
	Dmax	**9.900** (0.158)	10.026 (0.254)	10.521 (0.990)	12.759 (1.470)	11.560 (0.524)
	Dmin	38.024 (0.766)	**38.930** (0.863)	37.543 (2.878)	19.838 (7.271)	25.873 (1.5E-05)
	DBI	**0.1558** (0.0025)	**0.1558** (0.0022)	0.1655 (0.0173)	0.2088 (0.2088)	0.1760 (2.9E-08)
	XBI	**0.1281** (0.0025)	0.1286 (0.0036)	0.1486 (0.0314)	0.1298 (0.0089)	0.1284 (2.8E-08)
Ship	**Je**	10.895 (0.103)	10.872 (0.098)	11.277 (0.701)	10.526 (0.021)	**10.305 (2.0E-07)**
	Dmax	12.276 (0.302)	**12.079** (0.272)	13.082 (1.274)	12.512 (0.232)	11.570 (0.287)
	Dmin	44.716 (1.914)	41.917 (2.750)	**45.798** (4.553)	31.350 (0.249)	30.780 (3.5E-06)
	DBI	**0.1504** (0.0023)	0.1552 (0.0052)	0.1594 (0.0089)	0.1683 (0.0012)	0.1735 (1.3E-09)
	XBI	0.2781 (0.0100)	**0.2699** (0.0141)	0.2871 (0.0294)	0.2440 (0.0002)	**0.2386 (5.9E-09)**

3.3. ABC Kümeleme Tabanlı Renk Kuantalama

Bu kısım ABC tabanlı renk kuantalama yönteminin geliştirilmesi ve uygulamasını içermektedir. Kümeleme tabanlı kuantalama yöntemlerinin bölme tabanlı kuantalama yöntemlerinden daha iyi kümeleme performansı sağlaması sebebiyle kümeleme tabanlı bir kuantalama yönteminin geliştirilmesi üzerine yoğunlaşılmıştır. Bundan dolayı geliştirilen kuantalama yöntemi kümeleme işleminin temel prensipleriyle çalışmaktadır. Kümelemeden farklı olarak kuantalama işleminde küme merkez setinin yerini renk uzayı almaktadır.

3.3.1. Geliştirilen Renk Kuantalama Yöntemi

Kuantalama işlemi sırasıyla en çok bilinen renk uzayları olan RGB ve L*a*b* (CIELAB) uzaylarında gerçekleştirilmektedir. ABC'de bir bireyin her pozisyonu RGB renk uzayı için kırmızı, yeşil ve mavi olmak üzere üç bileşenden ve L*a*b* renk uzayı için 'a*' ve 'b*' renklilik katmanları olmak üzere iki bileşenden oluşur. Renk bilgisinin tamamının 'a*' ve 'b*' katmanında bulunması sebebiyle L*a*b* uzayındaki 'L*' katmanı kuantalama işlemine dahil edilmez.

ABC'de başlangıç popülasyonu orijinal resimdeki renk tonlarından rasgele seçilerek oluşturulur ve her bir birey bir başlangıç renk uzayını temsil eder. Bireyin pozisyon sayısı (boyutu), verinin bölüneceği tanımlanmış renk uzayının büyüklüğünü ifade eder. RGB formatında bir birey $x_i=\{x_{i1},x_{i2},\ldots,x_{iK}\}$ ile temsil edilir; burada x_i popülasyondaki *i.* birey ve $x_{iK}=[R_{ik},G_{ik},B_{ik}]$ *i.* bireyin *K.* renk tonunu ifade eden vektördür. L*a*b* renk uzayı formatında *i.* bireyin *K.* renk tonu $x_{iK}=[a_{ik},b_{ik}]$ iki boyutlu vektör ile ifade edilir.

RGB ve L*a*b* renk uzaylarında ABC kuantalama mekanizmasının işleyişi (CQ-ABC) [247] aynıdır. Öncelikle başlangıç popülasyonu üretilir. Bu işlemin ardından CQ-ABC ile popülasyon bireyleri geliştirilmeye çalışılır. Her bireye kullanıcı tarafından tanımlanmış p_{KM} olasılığı ile K-means algoritması uygulanarak başlangıç koşullarının sebep olabileceği olumsuz etkiler azaltılmaya çalışılır. Bir popülasyon bireyinin kalitesini (amaç fonksiyon değerini) hesaplamak için görüntüdeki her bir piksel değeri, renk uzayındaki kendine en yakın renk tonuna atanır ve bu atama işleminden sonra bireyin MSE değeri Denklem 3.9 ile hesaplanarak kalitesi belirlenir. Bu prosedür maksimum çevrim sayısı sağlanıncaya kadar devam eder. İşlemin tamamlanmasının

ardından en iyi birey (çözüm) görüntünün renk (tonu) katmanı olarak atanır. CQ-ABC'nin genel işleyişi Şekil 3.14'te sunulmuştur.

```
begin
    SN, K, MCN ve limit parametrelerini belirle;
    foreach kasif ari i do
        x_i icin goruntuden K tane rasgele renk tonu sec;
        x_i kaynaginin kalitesini (f(x_i)) Denklem (3.9) ile degerlendir;
    end
    for cycle ← 1 to MCN do
        foreach gorevli ari i do
            if p_KM < rand(0,1) then
                x_i'ye az sayida cevrim sayisi icin K-means uygula;
            end
            x_i kaynaginin komsulugunda Denklem (2.6) ile yeni kaynak (v_i) bul;
            foreach veri elemani z_p do
                for k = 1 to K do
                    d^2(z_p, C_{i,k}) = (R_p - R'_{i,k})^2 + (G_p - G'_{i,k})^2 + (B_p - B'_{i,k})^2 ;
                end
                d(z_p, C_{i,k})'yi minimum yapan C_{i,k} renk tonuna z_p'yi ata;
            end
            v_i kaynaginin kalitesini (f(v_i)) Denklem (3.9) ile degerlendir;
            x_i ve v_i arasnda ac gozlu yaklasim uygula;
        end
        Kaynaklarin olasilik degerlerini (p_i) Denklem (2.7) ile hesapla;
        foreach gozcu ari i do
            Kaynaklarin olasilik degerlerine bagli olarak bir kaynak (x_i) sec;
            if p_KM < rand(0,1) then
                x_i'ye az sayida cevrim sayisi icin K-means uygula;
            end
            x_i kaynaginin komsulugunda Denklem (2.6) ile yeni kaynak (v_i) bul;
            foreach veri elemani z_p do
                for k = 1 to K do
                    d^2(z_p, C_{i,k}) = (R_p - R'_{i,k})^2 + (G_p - G'_{i,k})^2 + (B_p - B'_{i,k})^2 ;
                end
                d(z_p, C_{i,k})'yi minimum yapan C_{i,k} renk tonuna z_p'yi ata;
            end
            v_i kaynaginin kalitesini (f(v_i)) Denklem (3.9) ile degerlendir;
            x_i ve v_i arasnda ac gozlu yaklasim uygula;
        end
        if bir kaynagin deneme sayaci > limit then
            Kasif ari goruntuden K tane rasgele renk tonu secer;
        end
        En iyi kaynagi belirle;
    end
end
```

Şekil 3.14. CQ-ABC kuantalama yönteminin kaba kodu.

$$f(x_i) = \begin{cases} MSE_{RGB}(x_i) & if\ renk\ uzayı\ RGB \\ MSE_{l*a*b}(x_i) & if\ renk\ uzayı\ Lab \end{cases} \quad (3.9)$$

Burada MSE_{RGB} ve MSE_{L*a*b} Denklem 3.10 ve 3.11 ile tanımlanmıştır.

$$MSE_{RGB}(x_i) = \frac{1}{N}\sum_{k=1}^{K}\sum_{\forall z_p \in C_k} (R_p - R'_{ik})^2 + (G_p - G'_{ik})^2 + (B_p - B'_{ik})^2 \quad (3.10)$$

Burada $z_p=[R_p,G_p,B_p]$ görüntüdeki p. piksel (pattern) değerini, $C_k=[R'_{ik},G'_{ik},B'_{ik}]$ i. bireyin k. renk tonunu, K bireyin sahip olduğu renk uzayının büyüklüğünü ve N görüntüdeki toplam piksel sayısını ifade etmektedir.

$$MSE_{L*a*b}(x_i) = \frac{1}{N_p}\sum_{k=1}^{K}\sum_{\forall z_p \in C_k} (a_p - a'_k)^2 + (b_p - b'_k)^2 \quad (3.11)$$

Burada $z_p=[a_p,b_p]$ görüntüdeki p. piksel (pattern) değerini ve $C_k=[a'_{ik},b'_{ik}]$ i. bireyin k. renk tonunu ifade etmektedir.

3.3.2. Kuantalama Deneysel Dizaynı

CQ-ABC kuantalama yöntemi (K) 16, 32, 64 ve 128 büyüklüğündeki renk uzaylarına indirgenmek üzere ilgili kuantalama çalışmalarında [54, 234] kullanılan Airplane (Şekil 3.15.a), Lena (Şekil 3.16.a), Mandril (Şekil 3.17.a) ve Pepper (Şekil 3.18.a) görüntülerine uygulanmıştır. Yöntemin analizi CQ-PSO, K-means, FCM ve minimum varyans yöntemleri ile RGB ve L*a*b* renk uzaylarında karşılaştırılarak yapılmıştır. Buna ek olarak, genetik algoritma ile tasarlanan GCMA [54] yönteminin RGB renk uzayında elde ettiği en iyi değerlere de deneysel çalışmalarda yer verilmiştir. RGB ve L*a*b* renk uzaylarında yöntemlerin değerlendirilmesi MSE ve tepe sinyali-kuantalama gürültü oranı (PSQNR) [248] üzerinden gerçekleştirilmiştir. PSQNR orijinal görüntü ile (kuantalanmış) çıkış görüntüsü arasındaki bozulmayı ifade etmektedir:

$$PSQNR = \log_{10}\frac{{Max_{im}}^2}{MSE} = 20\log_{10} Max_{im} - 10\log_{10} MSE \quad (3.12)$$

Burada Max_{im} orijinal görüntüdeki maksimum piksel (yoğunluk) değerini ifade etmektedir. Griton uzayında Max_{im} 255 alınır. RGB uzayında Max_{im} kırmızı, yeşil ve mavi tonlarının maksimum değeri (genellikle 255) alınır. L*a*b* uzayında ise Max_{im}

belirlemede üç farklı yol vardır: 1) farklı bir renk uzayına çevirmek, 2) a* ve b* bileşenlerinin maksimumlarını toplamı, ve 3) a* ve b* bileşenlerinden maksimum olanın seçilmesi. Bu çalışmada son yol tercih edilmiştir.

CQ-ABC ve CQ-PSO yöntemlerinde popülasyon büyüklüğü 20 ve maksimum çevrim sayısı 50 olarak belirlenmiştir. Bu değerler [55] nolu referanstaki değerlerin aynısıdır. p_{KM} olasılığı 0.1 olarak alınmış ve CQ-ABC ve CQ-PSO'da bireylere uygulanan K-means 50 çevrim ile sınırlandırılmıştır. CQ-ABC'de limit değeri 50 olarak belirlenmiştir. CQ-PSO'daki parametreler ilgili çalışmadaki [55] gibi $w_{init}=0.72$, $w_{end}=0.4$, $c_1=c_2=1.49$ ve $V_{max}=255$ olarak seçilmiştir. K-means ve FCM tabanlı kuantalama yöntemlerinde maksimum değerlendirme sayısı 1000 olarak alınmıştır. FCM'de μ katsayısı 1.1 olarak belirlenmiştir.

3.3.3. Kuantalama Deneysel Sonuçları

Elde edilen sonuçlar birbirinden bağımsız 30 koşma için ortalama ve standart sapma değerleri üzerinden tablolarda sunulmuştur. Tablolardaki en iyi değerler kalın fontta ve standart sapma değerleri parantez içinde gösterilmiştir. Tablolarda yer alan '+' ('-') sembolü, CQ-ABC'nin ilgili yöntemden Wilcoxon Rank Sum Test istatistiğine göre daha iyi (daha kötü) olduğunu göstermektedir. '=' sembolü, CQ-ABC ve ilgili yöntem arasında Wilcoxon Rank Sum Test istatistiğine göre farklılık olmadığını ifade etmektedir. Deneysel sonuçlar RGB ve L*a*b* renk uzayları olmak üzere iki kategoride ele alınmıştır.

3.3.3.1. RGB Uzayı Sonuçları

RGB uzayındaki MSE ve PSQNR sonuçları Tablolar 3.5 ve 3.6'da sunulmuştur. Elde edilen sonuçlar neticesinde GCMA'nın kuantalama performansının diğer yöntemlere kıyasla oldukça düşük olduğu görülmüştür. Örneğin, GCMA Lena ve Mandril verisini 16 renge indirgediğinde elde ettiği en iyi MSE değerleri sırasıyla 332 ve 606'dır. Bu durumun sebeplerinden birisi, GA'nın gen yapısındaki boyut sayısının artmasıyla genetik operatörlerin etkinliğinin azalması olarak görülebilir. Bir diğer sebebi de GCMA yönteminde renk tonunun gen yapılarında (pozisyonlarda) vektörel olarak temsilinin mümkün olmamasıdır. Diğer algoritmaların performansı değerlendirildiğinde ise, önerilen CQ-ABC yöntemi 80 durumdan 4 durum dışında en iyi ortalama ve

standart sapma değerlerini elde etmiştir. Bu üstünlük istatistiksel anlamlılık testi sonuçları ile de desteklenmiştir. Sadece 2 durumda (Airplane 128 ve Lena 128) CQ-ABC yöntemi FCM'ye karşı istatistiksel test sonuçlarına göre daha kötü sonuç elde etmiştir. CQ-PSO ve FCM yöntemlerinin sonuçları karşılaştırıldığında ise, CQ-PSO'nun Mandril ve Pepper görüntülerinde FCM'den daha iyi olduğu, Airplane ve Lena görüntülerinde FCM'nin CQ-PSO'dan daha iyi olduğu görülmektedir. Bu yöntemleri K-means kuantalama yöntemi takip etmektedir. Bölmeli tabanlı bir kuantalama yöntemi olan minimum varyans ise kuantalama performansı olarak diğerlerine nazaran yetersiz kalmıştır. Örneğin, Lena görüntüsünün 16 renge indirgenmiş formunda CQ-ABC, CQ-PSO, K-means ve FCM yöntemlerinin elde ettikleri ortalama MSE değerleri sırasıyla 159.468, 161.329, 160.891 ve 164.954 iken, minimum varyans yönteminin elde ettiği MSE değeri 176.605'te kalmıştır. Sonuçların neredeyse tamamında minimum varyans ve bahsi geçen yöntemler arasında benzeri farklar mevcuttur. Sonuç olarak önerilen CQ-ABC'nin RGB renk uzayında kuantalama işleminde en uygun yöntem olduğu ortaya çıkmıştır. CQ-ABC'nin görsel sonuçları Şekiller 3.15-3.18'de sunulmuştur.

3.3.3.2. L*a*b* Uzayı Sonuçları

L*a*b* uzayına ait sonuçlar Tablolar 3.7 ve 3.8'de sunulmuştur. RGB uzayındaki deneysel çalışmalardan farklı olarak minimum varyans metodunun L*a*b* uzayında uygulaması bulunamadığı için bu kısımdaki deneysel çalışmalarda yer verilmemiştir. Elde edilen sonuçlara göre önerilen CQ-ABC yöntemi 5 durum dışındaki bütün durumlarda en iyi ortalama ve standart sapma değerlerini elde etmiştir. CQ-ABC'nin en iyi performansı elde edemediği 5 durumda ise FCM yöntemi daha iyi performans sergilemiştir. İstatistiksel test sonuçları da CQ-ABC yönteminin etkinliğini destekler niteliktedir. Sadece birkaç durumda (Airplane 64, Mandril 32&64&128, Pepper 64) FCM yöntemi Wilcoxon Rank Sum Test istatistiğine göre CQ-ABC'den daha iyi sonuçlar elde etmiştir. Diğer yöntemlerin performansları değerlerildiğinde ise, CQ-PSO ve FCM yöntemlerinin birkaç durum dışında birbirlerine çok yakın olduğu görülmektedir. Örneğin, Airplane görüntüsünün 32 renge indirgenmiş formunda CQ-PSO yönteminin elde ettiği ortalama MSE değeri 4.88 iken, FCM yönteminin elde ettiği ortalama MSE değeri 4.91'dir. Dolayısıyla birinin diğerinden daha iyi olduğunu söyleyebilmek mümkün değildir. Bu yöntemleri performans olarak K-means

kuantalama yöntemi izlemektedir. Sonuç olarak önerilen CQ-ABC yöntemi hem RGB hem de L*a*b* uzayında kuantalama işleminde en başarılı yöntem olmuştur.

Tablo 3.5. RGB renk uzayı MSE sonuçları.

Görüntü	**K**	**CQ-ABC**	**CQ-PSO**	**FCM**	**K Means**	**Minimum Varyans**	**GCMA [54]**
Airplane	16	**124.100 (3.687)**	126.208 (3.968) '+'	124.297 (7.789) '='	153.824 (6.412) '+'	138.3569 '+'	199 '+'
	32	**62.028 (0.095)**	62.374 (0.480) '+'	62.063 (0.177) '+'	62.303 (0.496) '+'	67.6027 '+'	96 '+'
	64	37.128 (0.587)	37.342 (0.641) '='	**37.101 (0.231) '='**	38.146 (0.433) '+'	41.9033 '+'	54 '+'
	128	25.018 **(0.350)**	25.159 (1.140) '='	**23.637** (0.434) '-'	26.406 (0.358) '+'	26.587 '+'	N/A
Lena	16	**159.468 (0.944)**	161.329 (2.697) '+'	160.891 (0.978) '+'	164.954 (4.576) '+'	176.605 '+'	332 '+'
	32	**83.047 (0.230)**	83.860 (0.847) '+'	83.101 (0.454) '='	84.632 (1.050) '+'	94.562 '+'	179 '+'
	64	47.224 (0.186)	47.413 (0.487) '+'	**47.166 (0.164) '='**	48.330 (0.898) '+'	52.920 '+'	113 '+'
	128	28.405 (0.127)	28.532 (0.248) '+'	**27.996 (0.069) '-'**	28.988 (0.314) '+'	31.742 '+'	N/A
Mandril	16	**508.500 (0.149)**	509.795 (2.019) '+'	511.246 (6.201) '+'	512.060 (5.094) '+'	571.894 '+'	606 '+'
	32	**283.781 (0.418)**	286.053 (2.266) '+'	287.137 (2.917) '+'	287.817 (3.268) '+'	326.267 '+'	348 '+'
	64	**162.412 (0.326)**	163.317 (1.131) '+'	166.828 (1.635) '+'	164.579 (1.441) '+'	184.968 '+'	213 '+'
	128	**97.844 (0.177)**	98.448 (0.582) '+'	98.686 (0.279) '+'	98.642 (0.659) '+'	113.225 '+'	N/A
Pepper	16	**356.538 (0.005)**	357.333 (1.024) '+'	358.681 (2.388) '+'	358.676 (2.715) '+'	390.544 '+'	471 '+'
	32	**203.169 (0.608)**	205.048 (2.054) '+'	206.742 (2.447) '+'	205.338 (1.756) '+'	217.383 '+'	263 '+'
	64	**118.177 (0.388)**	118.519 (0.714) '+'	120.185 (0.969) '+'	119.846 (1.527) '+'	130.721 '+'	148 '+'
	128	**69.883 (0.371)**	70.436 (0.651) '+'	72. 312 (0.543) '+'	71.374 (1.102) '+'	79.192 '+'	N/A

Tablo 3.6. RGB renk uzayı PSQNR sonuçları.

Görüntü	K	CQ-ABC	CQ-PSO	FCM	K Means	Minimum Varyans	GCMA [54]
Airplane	16	**27.194 (0.128)**	27.122 (0.137) '+'	27.193 (0.252) '='	26.264 (0.193) '+'	26.720 '+'	25.142 '+'
	32	**30.204 (0.006)**	30.181 (0.033) '+'	30.202 (0.012) '+'	30.186 (0.034) '+'	29.831 '+'	28.308 '+'
	64	32.434 (0.068)	32.409 (0.074) '='	**32.437 (0.027) '='**	32.316 (0.049) '+'	31.908 '+'	30.806 '+'
	128	34.148 (0.061)	34.128 (0.196) '+'	**34.396 (0.080) '='**	33.914 (0.059) '+'	33.884 '+'	N/A
Lena	16	**26.104 (0.026)**	26.054 (0.071) '+'	26.066 (0.026) '+'	25.959 (0.120) '+'	25.661 '+'	22.919 '+'
	32	**28.937 (0.012)**	28.895 (0.043) '+'	28.935 (0.024) '='	28.856 (0.054) '+'	28.374 '+'	25.602 '+'
	64	31.389 (0.017)	31.372 (0.044) '+'	**31.394 (0.015) '='**	31.289 (0.080) '+'	30.895 '+'	27.600 '+'
	128	33.596 (0.019)	33.578 (0.038) '+'	**33.660 (0.011) '-'**	33.509 (0.047) '+'	33.114 '+'	N/A
Mandril	16	**21.068 (0.001)**	21.057 (0.017) '+'	21.045 (0.052) '+'	21.038 (0.043) '+'	20.558 '+'	20.306 '+'
	32	**23.601 (0.006)**	23.566 (0.034) '+'	23.550 (0.044) '+'	23.540 (0.049) '+'	22.995 '+'	22.715 '+'
	64	**26.025 (0.009)**	26.001 (0.030) '+'	25.908 (0.042) '+'	25.967 (0.038) '+'	25.460 '+'	24.847 '+'
	128	**28.225 (0.008)**	28.199 (0.026) '+'	28.188 (0.012) '+'	28.190 (0.029) '+'	27.591 '+'	N/A
Pepper	16	**22.610 (0.000)**	22.600 (0.012) '+'	22.584 (0.028) '+'	22.583 (0.032) '+'	22.214 '+'	21.400 '+'
	32	**25.052 (0.013)**	25.012 (0.043) '+'	24.977 (0.051) '+'	25.006 (0.037) '+'	24.758 '+'	23.931 '+'
	64	**27.405 (0.014)**	27.393 (0.026) '+'	27.332 (0.035) '+'	27.345 (0.055) '+'	26.967 '+'	26.428 '+'
	128	**29.687 (0.023)**	29.653 (0.040) '+'	29.539 (0.033) '+'	29.596 (0.067) '+'	29.144 '+'	N/A

Tablo 3.7. L*a*b* renk uzayı MSE sonuçları.

Görüntü	K	CQ-ABC	CQ-PSO	FCM	K Means
Airplane	16	**9.45 (0.09)**	9.57 (0.25) '+'	12.37 (1.16) '+'	11.05 (0.75) '+'
	32	**4.83 (0.06)**	4.88 (0.09) '+'	4.91 (0.16) '+'	5.98 (0.47) '+'
	64	2.57 **(0.04)**	2.59 (0.07) '='	**2.48** (0.07) '-'	3.84 (0.43) '+'
	128	**1.28 (0.02)**	1.33 (0.04) '+'	1.30 **(0.02) '+'**	2.43 (0.33) '+'
Lena	16	**6.23 (0.05)**	6.43 (0.21) '+'	6.28 (0.07) '+'	6.57 (0.30) '+'
	32	**3.38 (0.04)**	3.44 (0.09) '+'	3.51 (0.15) '+'	3.58 (0.16) '+'
	64	**1.82 (0.02)**	1.87 (0.05) '+'	2.04 (0.08) '+'	1.92 (0.06) '+'
	128	**0.94 (0.01)**	0.99 (0.03) '+'	1.14 (0.05) '+'	1.00 (0.03) '+'
Mandril	16	**25.34 (0.06)**	25.50 (0.30) '+'	25.51 (0.15) '+'	25.67 (0.37) '+'
	32	12.75 **(0.07)**	12.86 (0.24) '+'	**12.67** (0.09) '-'	13.35 (0.42) '+'
	64	6.91 (0.07)	7.14 (0.18) '+'	**6.63 (0.03) '-'**	7.32 (0.25) '+'
	128	3.71 (0.05)	3.90 (0.22) '+'	**3.50 (0.03) '-'**	3.90 (0.13) '+'
Pepper	16	**24.93 (0.04)**	25.13 (0.18) '+'	25.07 (0.19) '+'	25.60 (0.59) '+'
	32	**12.24 (0.09)**	12.44 (0.27) '+'	12.25 (0.16) '='	12.79 (0.32) '+'
	64	6.27 (0.06)	6.42 (0.21) '+'	**6.09 (0.03) '-'**	6.74 (0.26) '+'
	128	**3.27 (0.04)**	3.38 (0.16) '+'	3.37 (0.05) '+'	3.45 (0.10) '+'

3.4. Sonuçlar

Bu bölümde ABC kümeleme tabanlı görüntü segmentasyon ve kuantalama yöntemlerinin geliştirilmesi ve uygulaması gerçekleştirilmiştir. İlk olarak ABC tabanlı bir tümör segmentasyon metodolojisi geliştirilmiştir. Geliştirilen metodoloji hem K-means hem de FCM tabanlı segmentasyon metodolojilerinden görsel ve sayısal olarak daha iyi sonuçlar sağlamıştır. İkinci olarak mevcut görüntü kümeleme amaç fonksiyonlarının eksiklikleri saptanarak yeni bir görüntü kümeleme fonksiyonu önerilmiştir. Önerilen fonksiyon mevcut fonksiyonlardan kümeleme kalitesi olarak daha iyi sonuç sağlamıştır. Ayrıca önerilen fonksiyonun en iyi performansa ABC algoritması ile ulaştığı görülmüştür. Son olarak da ABC ile kümeleme tabanlı bir renk kuantalama yöntemi geliştirilmiştir. Geliştirilen ABC tabanlı yöntem gerek PSO ve GA tabanlı

yöntemlerden ve gerekse klasik yöntemlerden RGB ve L*a*b* uzaylarında daha iyi performans sağlamıştır. Yine geliştirilen bu kuantalama yöntemi, ABC kullanılarak geliştirilen ilk kuantalama yöntemi olma niteliği taşımaktadır.

Bu bölümde geliştirilen bütün yöntemlerde küme sayısının veya renk sayısının kullanıcı tarafından önceden belirlenmesi gerekmektedir. Ancak her veride optimal küme sayısının bilinebilmesi mümkün değildir. Bu sebepten dolayı takip eden bölümlerde verideki küme sayısını otomatik belirleyen ABC tabanlı kümeleme yöntemlerinin geliştirilmesi üzerine yoğunlaşılacaktır.

Tablo 3.8. L*a*b* renk uzayı PSQNR sonuçları.

Görüntü	**K**	**CQ-ABC**	**CQ-PSO**	**FCM**	**K Means**
Airplane	16	**36.812 (0.043)**	36.761 (0.111) '+'	35.666 (0.457) '+'	36.141 (0.294) '+'
	32	**39.726 (0.053)**	39.684 (0.088) '+'	39.658 (0.143) '+'	38.811 (0.318) '+'
	64	**42.474 (0.073)**	42.441 (0.111) '='	42.631 (0.121) '-'	40.750 (0.471) '+'
	128	**45.482** (0.074)	45.318 (0.147) '+'	45.440 **(0.064) '+'**	42.756 (0.662) '+'
Lena	16	**36.915 (0.038)**	36.777 (0.142) '+'	36.882 (0.050) '+'	36.685 (0.194) '+'
	32	**39.571 (0.047)**	39.501 (0.119) '+'	39.419 (0.179) '+'	39.319 (0.188) '+'
	64	**42.249 (0.045)**	42.145 (0.120) '+'	41.761 (0.172) '+'	42.021 (0.146) '+'
	128	**45.120 (0.049)**	44.892 (0.114) '+'	44.858 (0.126) '+'	44.267 (0.199) '+'
Mandril	16	**31.982 (0.009)**	31.955 (0.051) '+'	31.926 (0.062) '+'	31.954 (0.025) '+'
	32	34.964 **(0.024)**	34.930 (0.079) '+'	**34.991** (0.031) '-'	34.766 (0.135) '+'
	64	37.623 (0.043)	37.484 (0.109) '+'	**37.803 (0.020) '-'**	37.377 (0.145) '+'
	128	40.325 (0.058)	40.119 (0.245) '+'	**40.578 (0.044) '-'**	40.114 (0.146) '+'
Pepper	16	**31.654 (0.007)**	31.618 (0.031) '+'	31.628 (0.033) '+'	31.538 (0.099) '+'
	32	**34.742 (0.031)**	34.673 (0.093) '+'	34.740 (0.055) '='	34.554 (0.109) '+'
	64	37.647 (0.042)	37.543 (0.144) '+'	**37.774 (0.022) '-'**	37.336 (0.170) '+'
	128	**40.474 (0.065)**	40.337 (0.199) '+'	40.239 (0.126) '+'	40.344 (0.058) '+'

a) Orijinal resim

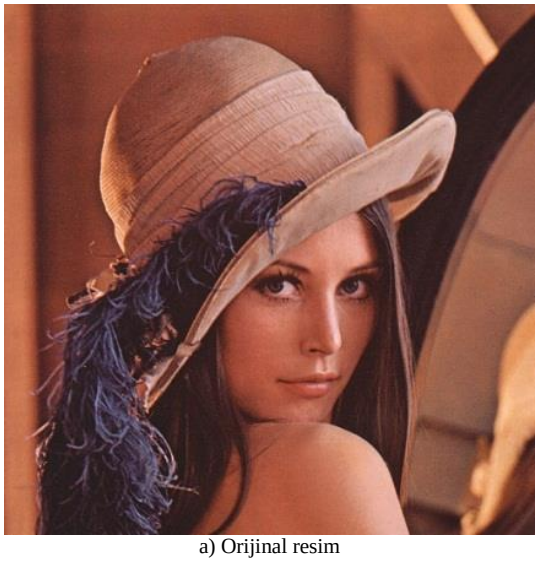
b) 16 renk

c) 32 renk

d) 64 renk

f) 128 renk

Şekil 3.15. Airplane görüntüsüne ait CQ-ABC görsel sonuçları.

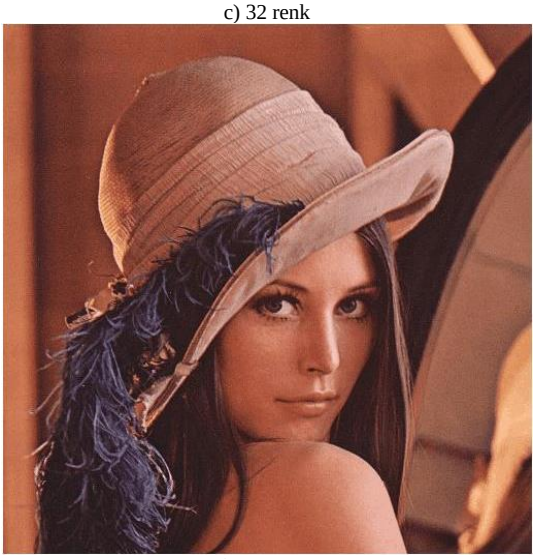


a) Orijinal resim

b) 16 renk

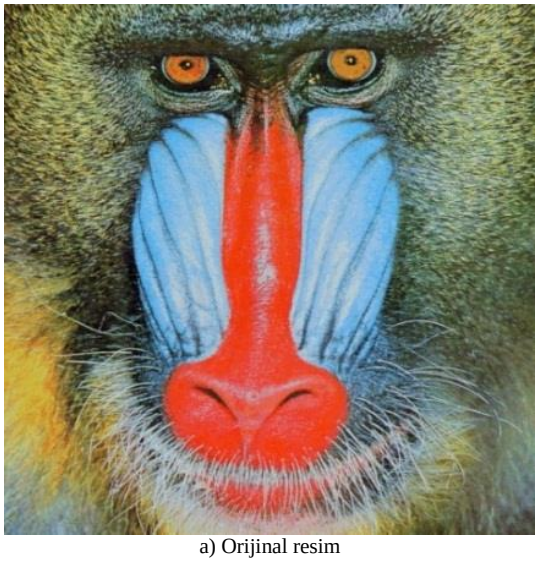
c) 32 renk

d) 64 renk

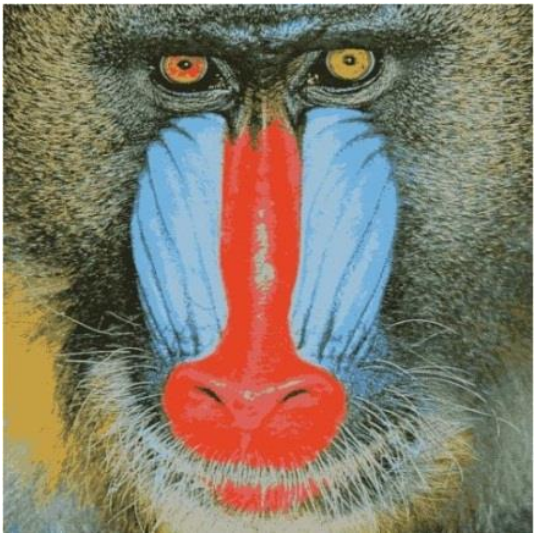


f) 128 renk

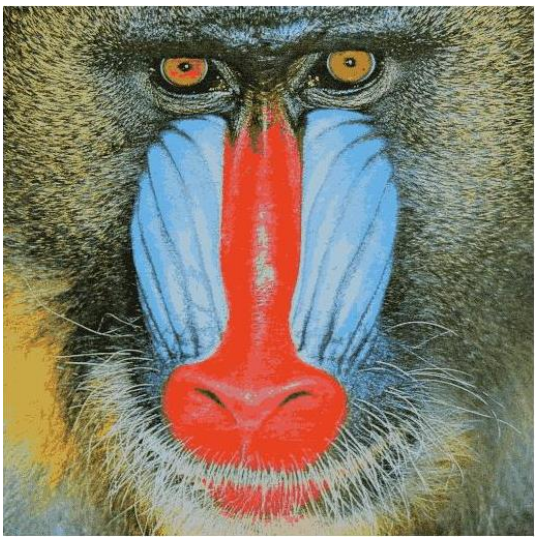
Şekil 3.16. Lena görüntüsüne ait CQ-ABC görsel sonuçları.



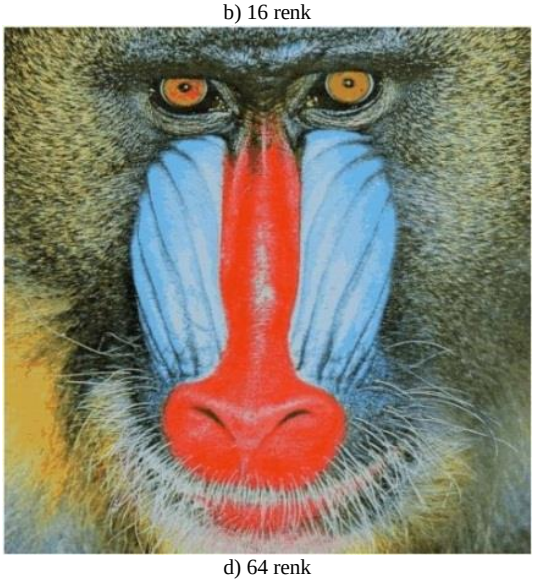
a) Orijinal resim



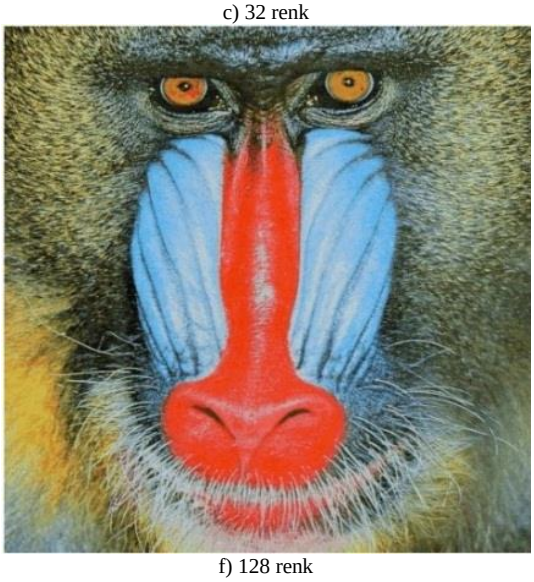
b) 16 renk



c) 32 renk

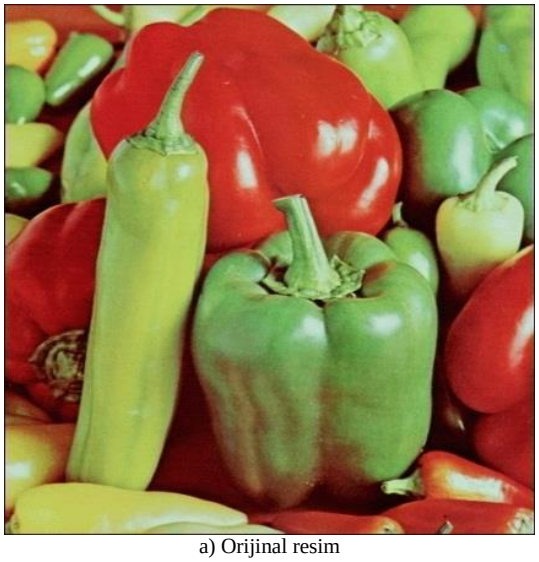


d) 64 renk



f) 128 renk

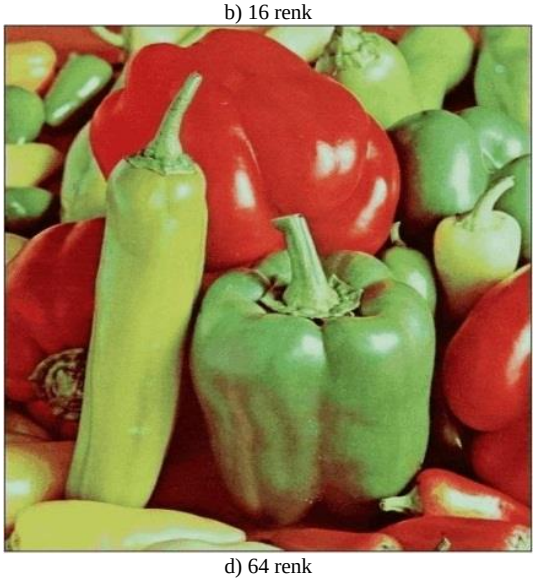
Şekil 3.17. Mandril görüntüsüne ait CQ-ABC görsel sonuçları.



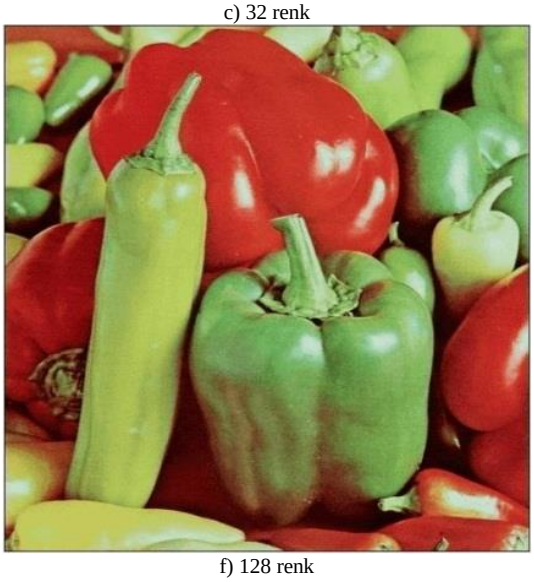
a) Orijinal resim

b) 16 renk

c) 32 renk



d) 64 renk



f) 128 renk

Şekil 3.18. Pepper görüntüsüne ait CQ-ABC görsel sonuçları.

4. BÖLÜM

KÜRESEL EN İYİ ABC ALGORİTMASI İLE OTOMATİK KÜMELEME YÖNTEMİ

Günümüz verilerinin büyük bir çoğunluğu ve özellikle görüntü setleri küme sayısı bilgisini içermemektedir. Dolayısıyla küme sayısını bir kontrol parametresi olarak kullanan yöntemler ile bu verilerde kümeleme analizinin yapılması mümkün değildir. Bir verideki optimum veya optimal küme sayısını önceden saptamak da zorlu bir süreçtir. Bu nedenle verinin karakteristik özelliklerini analiz ederek verideki küme sayısını otomatik belirleyen kümeleme yöntemlerine ihtiyaç duyulmaktadır. Her ne kadar küme sayısını otomatik saptayan yöntemler mevcut olsa da bu yöntemlerde gerçek dünya verileri üzerinde doğru çalışamama, ilgilenilen küme sayısının artması ile performanslarında azalma ve çakışan kümeleri saptayamama gibi dezavantajları mevcuttur. Bununla ilgili detaylı bilgi Bölüm 1'de verilmiştir. Ayrıca literatürden edinilen bilgiye göre standart ABC algoritmasının küme sayısını otomatik olarak saptamaya yönelik bir kullanımı veya buna dönük bir performans analizi de mevcut değildir.

Bu bölümün temel amacı ABC tabanlı bir otomatik kümeleme yöntemi geliştirmektir. Bu amaca ulaşmak için, ABC algoritmasının küresel-yerel araştırma kabiliyeti geliştirilmiş ve vektörel araştırma yapabilmesi sağlanmıştır. Geliştirilen küresel en iyi ABC yöntemi MRABC, GABC, ABC ve PSO yöntemleri ile en çok bilinen veri ve görüntü setleri üzerinde test edilmiştir. Temel olarak aşağıda belirtilen analiz ve incelemeler yapılmıştır:

- Geliştirilen yöntemin diğer yöntemlerle doğrulama indeksi üzerinden karşılaştırılması,

- Geliştirilen yöntemin diğer yöntemlerle optimal küme sayısını bulabilme kabiliyetleri üzerinden karıştırılması.

Bu bölümde ilk olarak küresel en iyi ABC (Gbest-ABC) algoritması sunulmakta ve Gbest-ABC ile otomatik kümeleme yöntemi anlatılmaktadır. Daha sonra ilgili yöntemin deneysel dizayn ve çalışmalarına yer verilmektedir. Son olarak da çalışmanın genel değerlendirmesi yapıl maktadır.

4.1. Küresel En İyi ABC Algoritması

Metasezgisel algoritmaların genel olarak başarılı çözümler üretebilmesi için çözüm uzayında küresel ve yerel araştırmanın dengeli bir şekilde yürütülmesi gerekli olduğu bilinen bir gerçektir. Çözüm uzayında yerel araştırmaya ağırlık verildiğinde, çözümlerin birbirine aşırı benzemesine ve erken yakınsama problemlerine neden olmaktadır. Çözüm uzayında küresel araştırmaya ağırlık verildiği takdirde ise, çözümlerin birbirinden aşırı uzaklaşmasına ve dolayısıyla optimal çözümlerin elde edilmesinde problemler ortaya çıkmaktadır. ABC algoritmasında görevli ve kaşif arılar küresel araştırmayı ve gözcü arılar yerel araştırmayı sağlamaktadır. Ayrıca, gözcü arıların iyi nektar miktarına sahip kaynakları yüksek olasılıkla seçmesi yerel-küresel araştırma dengesini sağlayan bir başka faktördür. Dolayısıyla algoritma yapısı üzerinde yapılacak geliştirmelerin bu dengeyi bozmadan yapılması zorunludur.

ABC'nin kompleks problemlerde küresel en iyi çözümlere yakınsama hızını arttırmak amacıyla küresel en iyi çözüm araştırmacılar tarafından yeni çözüm üretme mekanizmalarında kullanılmıştır. Gao ve arkadaşları [249], "DE/best/1" ve "DE/best/2" stratejilerini ABC'de Denklem 2.6 yerine kullanmıştır. Bir diğer çalışmada Zhu ve Kwong [250] Denklem 2.6'nın sağ tarafına dikkate alınan X_i kaynağı ile küresel en iyi kaynağın farkını ilave etmiştir. Küresel en iyi çözümü kullanan benzer bir ABC modeli ekonomik emisyon dağıtım probleminin çözümünde kullanılmıştır [233]. Tuba ve arkadaşları [251] küresel en iyi çözümü, PSO'daki hız ve yer değiştirme vektörlerini ABC'ye entegre ederek kullanmıştır. Abro ve Mohamad-Saleh [252] tek bir küresel iyi çözüm yerine birden fazla iyi çözümleri kullanarak yeni bir ABC modeli önermiştir. ABC'nin araştırma mekanizmasında küresel en iyi çözümü kullanan diğer çalışmalar [253-255]'te bulunabilir. Küresel en iyi çözümü kullanarak geliştirilen ABC modellerinin probleme bağlı olarak performansı artmış olsa da, bu modellerin sadece

küresel en iyi çözüme endeksli bir araştırma yürüttüğü görülmüştür. Bu durum standart ABC algoritmasındaki küresel-yerel araştırma dengesini bozabilmekte ve yerel yakınsama problemlerine neden olabilmektedir.

Bu çalışmada küresel-yerel araştırma dengesi korunacak şekilde PSO'nun sahip olduğu bazı özelliklerin ABC algoritmasına entegre edilerek daha etkin bir küresel en iyi çözüm tabanlı ABC modeli (Gbest-ABC) ortaya çıkarılmıştır [256]. İlk olarak bir pozisyonda yapılan değişim tüm pozisyonlara yayılarak vektörel değişim yapılması öngörülmüştür. Buradaki temel amaç, görevli arıların vektörel değişimle biraz daha geniş bir bölgede arama yapabilmesidir. İşçi arıların vektörel olarak yeni kaynaklar bulabilmesi Denklem 2.5 yerine Denklem 4.1 ile sağlanır:

$$v_i = x_i + \theta_i.(\mathrm{x_i} - \mathrm{x_k}) \tag{4.1}$$

Burada θ_i 0 ve 1 arasında rasgele üretilen sayılardan oluşan bir vektördür.

Bir diğer yapılan değişiklik gözcü arıların en iyi kaynak bilgisinden faydalanarak vektörel arama yapmasıdır. Buradaki temel amaç, gözcü arıların yerel araştırma özelliklerini biraz daha etkinleştirmek ve iyi olan kaynak bilgisinden faydalanarak daha iyi kaynak bulabilme ihtimalini artırmaktır. Bunun gerçekleştirilmesi Denklem 2.5 yerine Denklem 4.2 entegre edilerek sağlanmıştır:

$$v_i = x_i + \theta_i.(\mathrm{x_i} - \mathrm{Gbest}) \tag{4.2}$$

Burada *Gbest* en iyi kaynağı temsil etmektedir.

Küresel en iyi çözüm bilgisinden faydalanma, sadece gözcü arılarla sınırlı tutularak algoritmanın küresel araştırma yeteneği korunmuştur. Ayrıca gözcü arıların iyi çözümler komşuluğunda arama yapmaya meyilli olduğu da göz önüne alındığında, en iyi çözüm bilgisinden faydalanarak yeni kaynak arama görevinin sadece gözcü arılarla sınırlı kalmasının doğru bir karar olduğu görülmektedir. Geliştirilen küresel en iyi ABC modeli bu özellikleri ile küresel en iyi çözüm kullanılarak geliştirilen literatürdeki mevcut ABC modellerinden ayrılmaktadır.

4.2. Küresel En İyi ABC ile Otomatik Kümeleme Yöntemi

Küme sayısının saptanması içsel doğrulama indeksleri ile gerçekleştirilir. Bir kümeleme doğrulama indeksi iki temel koşulu sağlamalıdır: küme dahilindeki bireylerin birbirleri

ile maksimum düzeyde benzerlik göstermeleri (compactness) ve kümeler arası benzerliğin minimum düzeyde olması (separatness). Turi [257] geliştirmiş olduğu VI indeksinin DBI ve Dunn indekslerinden daha iyi olduğunu öne sürmüştür. Bu nedenle bu çalışmada VI indeksi amaç fonksiyonu olarak seçilmiştir.

VI indeksi iki modülden oluşmaktadır:

a) Küme içi uzaklık (Intra-Cluster Separation): Küme dahilindeki elemanların merkeze olan uzaklıklarının ortalamasıdır ve Denklem 4.3 ile ifade edilir. Küme dahilindeki elemanların birbirlerine yakın (benzer) olması beklendiğinden dolayı küme içi uzaklık modülünün minimize edilmesi gereklidir.

$$intra = \frac{1}{N_p} \sum_{k=1}^{K} \sum_{z_p \in C_k} \left\| z_p - m_k \right\|^2 \tag{4.3}$$

Burada z_p verideki *p.* veri elemanı, N_p verideki toplam eleman sayısını, *K* toplam küme sayısını ve m_k *k.* kümenin (C_k) merkezini ifade etmektedir.

b) Kümeler arası uzaklık (Inter-Cluster Separation): Kümelerin merkezleri arasındaki uzaklığın minimumudur ve Denklem 4.4 ile ifade edilir. Kümelerin birbirlerinden ayrılması için bu uzaklığın maksimize edilmesi beklenir.

$$inter = \min\{\|m_k - m_{kk}\|^2\}, k \neq kk \tag{4.4}$$

Bu iki bileşenin birleşimi ile VI indeksi ortaya çıkmıştır. VI indeksi Denklem 4.5 ile ifade edilir:

$$VI = (c \times N(\mu, \sigma) + 1) \times \frac{intra}{inter} \tag{4.5}$$

Burada *c* kullanıcı tarafından tanımlı bir sabit ve $N(\mu,\sigma)$ katsayısı *K* küme sayısı, μ ortalama ve σ standart sapma ile hesaplanan normal dağılımı ifade etmektedir:

$$N(\mu, \sigma) = \frac{1}{\sqrt{2\Pi\sigma^2}} e^{[-\frac{(K-\mu)^2}{2\sigma^2}]} \tag{4.6}$$

Burada μ değeri 5 ve 6 kümeli verilerin kümelenmesi ile ilgilenildiği durumlarda 2, 5'ten az küme içeren veriler için 1 veya 0 ve 5'ten fazla küme içeren veriler için 3 veya 4 alınır [161]. σ ise genellikle 1 olarak seçilir.

Gbest-ABC algoritmasında popülasyon birey (kaynak) temsili için sürekli aktivasyon tabanlı kodlama kullanılmıştır. Sürekli aktivasyon tabanlı kodlama ile ilgili detaylı bilgi Bölüm 1.4.3'te sunulmuştur. Bireydeki aktivasyon kodlarının etkinleştirilmesi sonucu seçilen merkez sayısının ikiden az olması durumunda, bireyin aktivasyon değerleri tekrar üretilir. Dolayısıyla VI indeksi hesaplamasında oluşacak problemleri engellemek için bir verinin sahip olabileceği merkez sayısı en az 2 olarak belirlenmiştir. Bir verinin sahip olacağı maksimum küme sayısı (K_{max}), kullanıcı tarafından veriye uygun olarak belirlenir. Bireydeki merkezlerin seçilmesinin ardından veri elemanları, merkezlere olan Öklid uzaklığına bağlı olarak kendilerine en yakın kümelere atanır. Bir kümeye herhangi bir veri elemanı atanamaması durumunda bireyin sahip olduğu merkezler veriye uygun olarak yenilenir ve atama işlemi tekrarlanır. Atama işleminin ardından Denklem 4.5 ile VI indeks değeri hesaplanır. Hesaplanan VI indeks değeri kaynağın kalitesini ifade etmektedir. Bireylerdeki merkezlerin güncellenmesi, merkez pozisyonlarında rasgele seçilmiş bir öznitelikte vektörel olarak gerçekleşir. Tüm öznitelikler üzerinde gerçekleşen merkez güncelleme merkezlerin yapısının bozabilmektedir. Gbest-ABC tabanlı otomatik kümeleme yönteminin kaba kodu Şekil 4.1'de sunulmuştur.

Örnek: Gbest-ABC modeli ile otomatik kümeleme uygulamasının daha iyi anlaşılması için burada bir örneğe yer verilmiştir. Tek boyutlu ve 50 elemanlı bir Z verisi için K_{max}=8 olsun. Bir X_i kaynağı aşağıdaki şekilde verilmiş olsun:

0.67	0.03	0.09	0.13	0.51	0.89	0.04	0.90	122	98	173	31	39	21	174	33
Aktivasyon Kodları								Küme Merkezleri							

X_i kaynağından sorumlu görevli arının da araştırma için popülasyondan rasgele seçtiğin X_k kaynağı da aşağıdaki gibi verilmiş olsun:

0.39	0.88	0.26	0.72	0.77	0.33	0.74	0.60	151	91	169	90	145	118	189	111

Denklem 4.1 ile X_i ve X_k arasındaki etkileşim sonucu ortaya çıkan V_i kaynağı aşağıda gibi elde edilmiş olsun (Not: Merkez değerleri tamsayıya yuvarlanmıştır):

0.69	0.72	0.02	0.12	0.36	0.51	0.01	0.91	141	102	174	45	105	34	174	92

Buradan V_i kaynağındaki aktivasyon kodlarının etkinleştirilmesi sonucu ortaya çıkan merkez seti M_i=[141,102,34,92] olur. Bu set kullanılarak V_i kaynağının kalitesi Denklem 4.5 ile hesaplanır.

Maksimum çevrim sayısı sonucu elde edilen *Gbest* kaynağı da aşağıdaki gibi olsun:

0.36	0.91	0.33	0.10	0.71	0.69	0.51	0.61	126	75	34	110	227	140	204	54

Buradan Gbest aktivasyon setinin etkinleştirilmesi sonucu ortaya çıkan merkez seti M_{gbest}=[75,227,140,204,54] ve optimal küme sayısı 5 olur.

begin
- K_{max}, SN, MCN ve *limit* parametrelerini belirle;
- **foreach** *kasif ari i* **do**
  - x_i icin K_{max} tane rasgele 0 ve 1 arasinda aktivasyon degeri uret ve Z verisinden K_{max} tane merkez sec;
- **end**
- **for** *cycle* $\leftarrow 1$ **to** MCN **do**
  - **foreach** *gorevli ari i* **do**
    - x_i'nin komsulugunda Denklem (4.1) ile yeni kaynak (v_i) bul;
    - v_i kaynaginin merkezlerini (m_{ik}) aktivasyon degerlerine bagli olarak etkinlestir;
    - **while** *etkinlestiren merkez sayisi* $K < 2$ **do**
      - x_i kaynaginin aktivasyon degerlerini yeniden uret;
      - x_i kaynaginin merkezlerini (m_{ik}) aktivasyon degerlerine bagli olarak etkinlestir;
    - **end**
    - **for** $p = 1$ **to** N_p **do**
      - z_p'nin etkinlestirilen kume merkezlerine olan Oklid mesafelerini ($d(z_p, m_{ik})$) hesapla;
      - z_p'yi en yakin Oklid mesafesindeki kumeye ata;
    - **end**
    - $VI(v_i)$ degerini Denklem (4.5) ile hesapla;
    - x_i ve v_i arasinda ac gozlu yaklasim uygula;
  - **end**
  - **foreach** *gozcu ari i* **do**
    - Denklem (2.7) ile p_i olasiliginda bir x_i yiyecek kaynagi sec;
    - x_i'nin komsulugunda Denklem (4.2) ile yeni kaynak (v_i) bul;
    - **while** *etkinlestiren merkez sayisi* $K < 2$ **do**
      - x_i kaynaginin aktivasyon degerlerini yeniden uret;
      - x_i kaynaginin merkezlerini (m_{ik}) aktivasyon degerlerine bagli olarak etkinlestir;
    - **end**
    - **for** $p = 1$ **to** N_p **do**
      - z_p'nin etkinlestirilen kume merkezlerine olan Oklid mesafelerini ($d(z_p, m_{ik})$) hesapla;
      - z_p'yi en yakin Oklid mesafesindeki kumeye ata;
    - **end**
    - $VI(v_i)$ degerini Denklem (4.5) ile hesapla;
    - x_i ve v_i arasinda ac gozlu yaklasim uygula;
  - **end**
  - **if** $denemesayac_i > limit$ **then**
    - x_i icin K_{max} tane rasgele 0 ve 1 arasinda aktivasyon degeri uret ve Z verisinden K_{max} tane merkez sec;
  - **end**
  - *gbest* kaynagini guncelle;
- **end**

end

Şekil 4.1. Gbest-ABC otomatik kümeleme yöntemi kaba kodu.

4.3. Deneysel Dizayn

Önerilen yöntemin performans analizi, kümeleme kalitesi ve elde edilen optimal küme sayısı üzerinden yapılmıştır. Karşılaştırmalı performans analizi için standart ABC ve PSO algoritmalarının yanında, küresel en iyi çözüm mekanizmasını kullanan GABC ve ABC'nin genişletilmiş ve bilinen bir versiyonu MRABC de deneysel çalışmalara dahil edilmiştir. GABC'nin dahil edilmesindeki temel amaç Gbest-ABC'nin küresel çözüm kullanmadaki etkinliğini ortaya koymaktır.

Deneysel çalışmalar, 7 adet 8 bitlik griton görüntü ve 7 adet veri seti olmak üzere toplam 14 veri üzerinde gerçekleştirilmiştir. Seçilen görüntüler sırasıyla Tahoe (300×300 piksel), Morro Bay (256×256 piksel), MRI (300×300 piksel), Airplane (512×512 piksel), Lena (512×512 piksel), Mandril (512×512 piksel) ve Pepper (512×512 piksel) griton resimleridir. Bu görüntüler uygulamalarda en çok kullanılan ve bilinen griton resimlerdir. Ancak bu görüntüler için optimum küme sayısı bilinememektedir. Bir araştırma grubu [257] tarafından Tahoe, MRI, Airplane, Lena, Mandril ve Pepper griton resimleri için sırasıyla [3,7], [4,8], [5,7], [5,10] ve [6,10] aralıklarında optimal küme sayıları belirlenmiştir. Morro Bay, Tahoe gibi uzaktan algılama resmi olduğundan dolayı küme sayısı aralığı aynı şekilde [3,7] olarak alınmıştır. Ancak yöntemlerin küme sayısını belirlemedeki performansını ölçmek için tek bir değere ihtiyaç vardır. Bu çalışmada görüntüler için optimal küme sayısı aşağıdaki şekilde belirlenmiştir:

$$Optimal\ Küme\ Sayısı = AltSınır + 1 \tag{4.7}$$

Denklem 4.7'ye göre Tahoe ve Morro Bay için optimal küme sayısı 4, MRI için 5, Airplane, Lena ve Mandril için 6 ve Pepper için 7 olarak belirlenmiştir. Bir çalışmada [13] Pepper için optimal küme sayısının 7 olarak alınmış olması da belirlenen değerlerin tutarlılığını göstermektedir.

Küme sayısını belirlemedeki performansını daha net bir şekilde ölçmek amacıyla otomatik kümeleme yöntemleri sınıf (küme) sayısı kullanıcılar tarafından bilinen veri setlerine de uygulanmıştır. Veri setlerine ait küme sayısı bilgisi otomatik kümeleme işlemi esnasında kesinlikle kullanılmamıştır. Bu bilgiye sadece elde edilen küme sayılarının tutarlılığının değerlendirilmesi amacıyla başvurulmuştur. Seçilen veri setleri sırasıyla Wisconsin (683 birey, 9 öznitelik, 2 sınıf), Iris (150 birey, 4 öznitelik, 3 sınıf),

Wine (178 birey, 13 öznitelik, 3 sınıf), User Knowledge Modelling (493 birey, 18 öznitelik, 4 sınıf), E-coli (336 birey, 8 öznitelik, 5 sınıf) ve Dermatology (358 birey, 34 öznitelik, 6 sınıf) verileridir [258]. Veri setlerine kümeleme işleminden önce normalizasyon işlemi uygulanmıştır.

Yöntemlerin parametre değerleri yapılan denemeler ve literatürdeki bilgiler doğrultusunda saptanmıştır. Tüm ABC modellerinde popülasyon büyüklüğü Bölüm 3'te olduğu gibi 30 olarak alınmıştır. Limit parametre değeri 20 olarak belirlenmiştir. PSO algoritmasında parçacık sayısı 30, c1 ve c2 sabitleri 1.49, w_{start} değeri 0.72 ve w_{end} değeri 0.4 olarak belirlenmiştir. Görüntü kümelemede, V_{max} aktivasyon değerleri için 1 ve merkezler için 255 alınmıştır. Veri kümelemede, V_{max} hem aktivasyon hem de merkez değerleri için 1 olarak belirlenmiştir. Tüm algoritmalar için çevrim sayısı, görüntü kümelemede 500 ve veri kümelemede 250 olarak belirlenmiştir.

VI indeks için şu parametre değerleri belirlenmiştir: c değeri görüntü kümelemede 25 ve veri kümelemede 30 olarak alınmıştır [161]. Görüntü kümeleme için μ değeri, Tahoe ve Morro Bay görüntülerinde 1, Pepper görüntüsünde 3 ve kalan görüntülerde 2 olarak belirlenmiştir. Veri kümeleme için μ değeri, Knowledge veri setinde 1, Ecoli ve Dermatology veri setlerinde 2, İris ve Wine veri setlerinde 0 ve Diabet ve Wisconsin veri setlerinde 0.5 olarak belirlenmiştir. σ katsayısı tüm verilerde 1 olarak alınmıştır. K_{max} maksimum küme sayısı, tüm görüntü ve veri setlerinde 10 olarak belirlenmiştir [13].

Görüntü ve veri kümeleme için değerlendirme temel olarak VI indeks ve küme sayısı üzerinden gerçekleştirilmiştir. Bunlara ek olarak, veri kümelemede elde edilen kümeleme kalitesini değerlendirmek amacıyla dışsal doğrulama indekslerinden biri olan Rand indeks [36] de kullanılmıştır. Rand indeksi Denklem 4.8 ile ifade edilir ve [0,1] aralığında değer alır. Rand indeks değerinin 1'e yaklaşması kümeleme işleminde doğruluğun arttığını ve 1 değerini aldığında kümeleme işleminin %100 doğru bir şekilde yapıldığını göstermektedir. Veri setlerindeki etiketli bilgi otomatik kümeleme işlemi esnasında kesinlikle kullanılmamıştır. Sadece Rand indeksi hesaplamalarında etiketli bilgiye başvurulmuştur.

$$Rand = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.8}$$

Burada *TP* ve *TN* doğru pozitif ve doğru negatifler, *FP* ve *FN* ise yanlış pozitif ve yanlış negatifler olarak nitelendirilir.

4.4. Deneysel Çalışmalar

Deneysel çalışmalar görüntü kümeleme ve veri kümeleme olarak iki kategoride ele alınmıştır. Elde edilen sonuçlar birbirinden bağımsız 30 koşma için ortalama ve standart sapma değerleri üzerinden tablolarda sunulmuştur. Tablolardaki en iyi değerler kalın fontta ve standart sapma değerleri parantez içinde gösterilmiştir. Tablolar 4.1 ve 4.3'te yer alan '+' ('-') sembolü, Gbest-ABC'nin ilgili yöntemden Wilcoxon Rank Sum Test istatistiğine göre daha iyi (daha kötü) olduğunu göstermektedir. '=' sembolü, Gbest-ABC ve ilgili yöntem arasında Wilcoxon Rank Sum Test istatistiğine göre farklılık olmadığını ifade etmektedir.

4.4.1. Görüntü Kümeleme

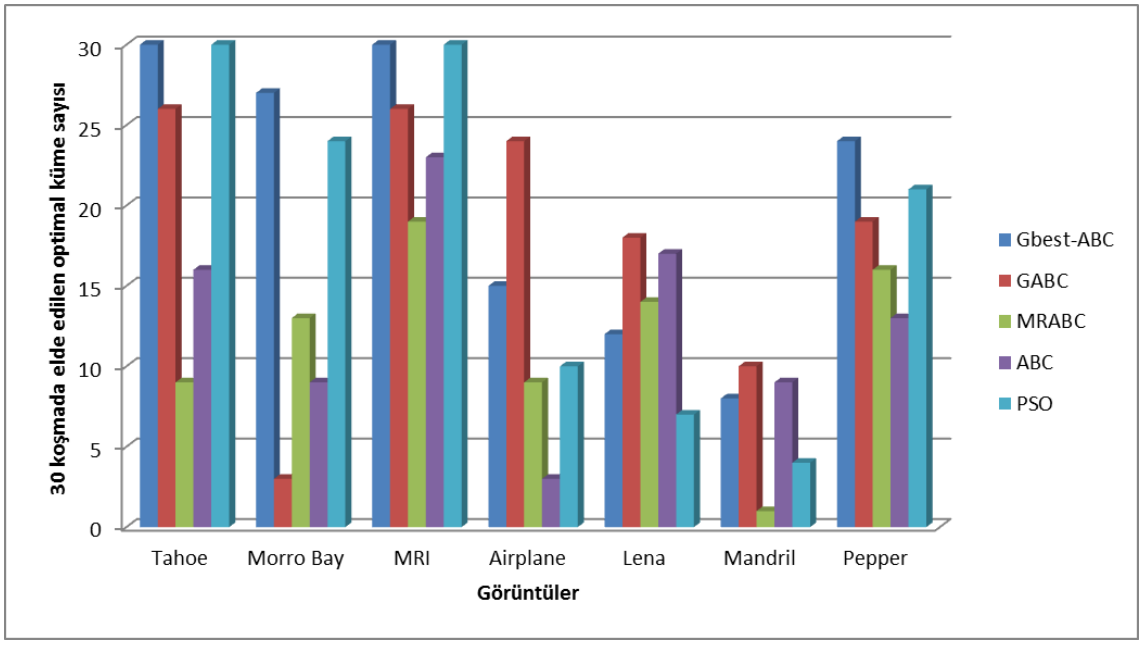
Görüntü kümeleme için doğrulama indeksi ve optimal küme sayısı sonuçları sırasıyla Tablolar 4.1 ve 4.2'de sunulmuştur. Tablo 4.1'e göre Gbest-ABC kümeleme kalitesi olarak diğer metotlardan Morro Bay, Airplane ve Pepper dışında daha iyi ortalama değerler elde etmiştir. En iyi sonucu elde edemediği görüntü setlerinde bile en iyi ikinci performansa sahiptir. Gbest-ABC'nin performansı istatistiksel test sonuçlarına göre değerlendirildiğinde, Lena dışında PSO ile benzer sonuçlar ürettiği Tablo 4.1'de görülmektedir. MRABC ile aralarında Pepper görüntüsü dışında pozitif yönde anlamlı bir farklılık söz konusudur. Kalan 14 durumun 8'inde, Gbest-ABC yöntemi ABC ve GABC'den istatistiksel olarak daha iyi sonuçlar elde etmiştir. Gbest-ABC dışındaki yöntemlerin performansı değerlendirilecek olursa, PSO en iyi ikinci performansa sahiptir ve onu GABC, ABC ve MRABC takip etmektedir. Sonuç olarak Gbest-ABC ve PSO, doğrulama indeksi minimizasyonunda mevcut ABC modellerinden genellikle daha iyi sonuçlar elde etmiştir. Bunda Gbest-ABC ve PSO'nun çözüm uzayında vektörel arama yapmalarının etkisi olduğu kanısına varılmıştır.

Tablo 4.1. Görüntü kümeleme için VI indeks sonuçları.

Görüntüler	Gbest ABC	GABC	MRABC	ABC	PSO
Tahoe	**0.0375** (7.2E-05)	0.0384 (0.0009) '+'	0.0542 (0.0036) '+'	0.0391 (0.0008) '+'	**0.0375** (2.6E-06) **'='**
Morro Bay	0.0683 (0.0027)	**0.0662** (0.0025) '-'	0.0696 (0.0021) '+'	0.0689 (0.0025) '='	0.0684 (0.0020) '='
MRI	**0.0345** (0.0006)	0.0354 (0.0009) '+'	0.0358 (0.0008) '+'	0.0360 (0.0009) '+'	0.0346 0.0007 '='
Airplane	0.0577 (0.0036)	0.0592 (0.0035) '='	0.0628 (0.0029) '+'	**0.0543** (0.0014) '-'	0.0591 (0.0032) '='
Lena	0.0784 (0.0069)	**0.0766** (0.0032) **'='**	0.0838 (0.0040) '+'	0.0787 (0.0032) '='	0.0819 (0.0071) '+'
Mandril	**0.0827** (0.0038)	0.0839 (0.0028) '+'	0.0865 (0.0036) '+'	0.0864 (0.0038) '+'	0.0828 (0.0048) '='
Pepper	**0.0849** (0.0064)	0.0875 (0.0036) '+'	0.0861 (0.0029) '='	0.1073 (0.0096) '+'	0.0871 (0.0101) '='

Tablo 4.2. Görüntü kümeleme için elde edilen küme sayıları.

Görüntüler	No	Gbest ABC	GABC	MRABC	ABC	PSO
Tahoe	4	**4 (0)**	4.033 (0.4901)	3.833 (1.2340)	3.466 (0.8995)	**4 (0)**
Morro Bay	4	**4.1 (0.3051)**	4.966 (0.8087)	4.366 (0.8087)	4.2 (1.2429)	4.2 (0.4068)
MRI	5	**5 (0)**	5.033 (0.4901)	4.433 (1.040)	4.566 (0.8584)	**5 (0)**
Airplane	6	5.6 **(0.5085)**	**5.8** (0.8050)	4.7 (1.2905)	4.066 (0.8277)	5.4 (0.5632)
Lena	6	**5.7** (0.5959)	6.413 **(0.5680)**	5.2 (1.2148)	5.6 (1.5887)	5.4 (0.6214)
Mandril	6	5.266 **(0.4497)**	**5.7** (0.9523)	4.533 (0.8603)	5.3 (1.0875)	5.2 (0.4842)
Peppers	7	**7.066 (0.4497)**	6.9 (0.6047)	6.333 (1.1244)	7.433 (1.073)	7.2 (0.7611)

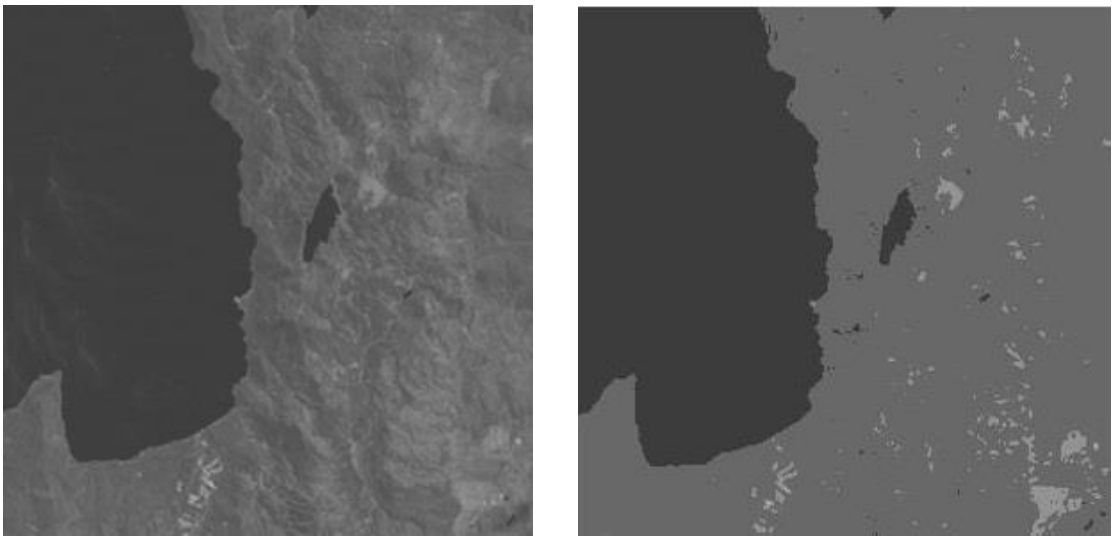


Şekil 4.2. Görüntü kümeleme için 30 koşma üzerinden toplamda kaç defa optimal küme sayısının elde edildiğini gösteren grafik.

Elde edilen ortalama küme sayılarına bakıldığında ise, Gbest-ABC yöntemi Airplane ve Mandril dışındaki görüntü setlerinde optimal küme sayısına yakınlık noktasında en iyi ortalama değerleri elde etmiştir. Airplane ve Mandril görüntülerinde ABC'nin bir başka küresel en iyi çözümü kullanan versiyonu olan GABC daha iyi değerler elde etmiştir. Standart ABC ve MRABC yöntemleri, diğer yöntemlere oranla daha kötü kümeleme sonuçları elde etmiştir. Örneğin, ABC ve MRABC yöntemleri Airplane görüntüsü için 4.06 ve 4.7 gibi düşük ortalama değerler elde etmiştir. PSO yönteminin Tahoe, Morro Bay ve MRI görüntülerinde genel olarak başarılı bir performans sağladığı görülmektedir. Ancak diğer 4 görüntüde bu başarılı performansını devam ettirememiştir.

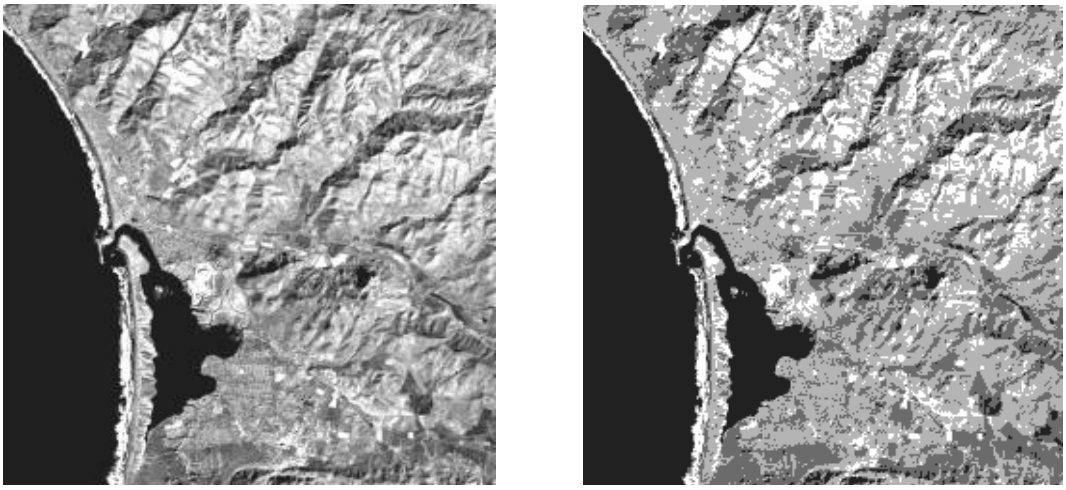
Yöntemlerin görüntü kümeleme için 30 koşma üzerinden toplamda kaç defa optimal küme sayısını elde ettiğini gösteren sütun grafiği Şekil 4.2'de sunulmuştur. Tahoe görüntüsünde 30 koşmanın hepsinde Gbest-ABC ve PSO optimal küme sayısını bularak %100'lük bir başarı sağlamıştır. GABC de Tahoe görüntüsünde Gbest-ABC ve PSO'nun ardından kabul edilebilir bir oran elde etmiştir. Ancak standart ABC ve MRABC optimal küme sayısını bulmada yetersiz kalmıştır. Bir diğer uzaktan algılama görüntüsü olan Morro Bay için Gbest-ABC 25'in üzerinde koşmada optimal küme sayısını elde ederek %87 civarı bir oranla hatırı sayılır bir performans göstermiştir.

Gbest-ABC'yi PSO %80'lik bir oranla takip etmektedir. Diğer yöntemlerin Morro Bay görüntüsünde performansı düşük bir seviyede kalmıştır. Bu görüntüde %10'luk bir optimal küme sayısı bulma oranıyla GABC'nin performansı oldukça kötüdür. MRI görüntüsünde Gbest-ABC ve PSO yöntemleri Tahoe görüntüsünde olduğu gibi optimal küme sayısını bulmada %100'lük bir performans göstermiştir. Onları sırasıyla GABC, ABC ve MRABC takip etmektedir. Airplane görüntüsünde GABC %80 ile iyi sayılabilir bir performans sağlamıştır. Ancak Airplane görüntüsünde diğer yöntemlerin performansları düşük seviyede kalmıştır. Bu yöntemlerin arasında Gbest-ABC %50'lik bir oranla diğerlerine nazaran daha iyi bir performans göstermiştir. Lena ve Mandril görüntülerinde genel olarak tüm yöntemlerin performansı düşük seviyede kalmıştır. Mandril görüntüsünde en yüksek performans bile %30'larda bulunmaktadır. Son görüntü Pepper'da ise önerilen Gbest-ABC %80'lik bir optimal küme sayısı bulma oranıyla en iyi performansı elde etmiştir. Onu %70, %63, %53 ve %43 oranlarla sırasıyla PSO, GABC, MRABC ve ABC takip etmektedir. Sonuç olarak önerilen Gbest-ABC otomatik kümeleme yöntemi görüntü kümeleme için hem kümeleme kalitesi hem de optimal küme sayısını elde etmede genel olarak diğer yöntemlerden daha iyi performans göstermiştir. Gbest-ABC ile elde edilen görsel kümeleme sonuçları Şekiller 4.3-4.9 aralığında sunulmuştur.



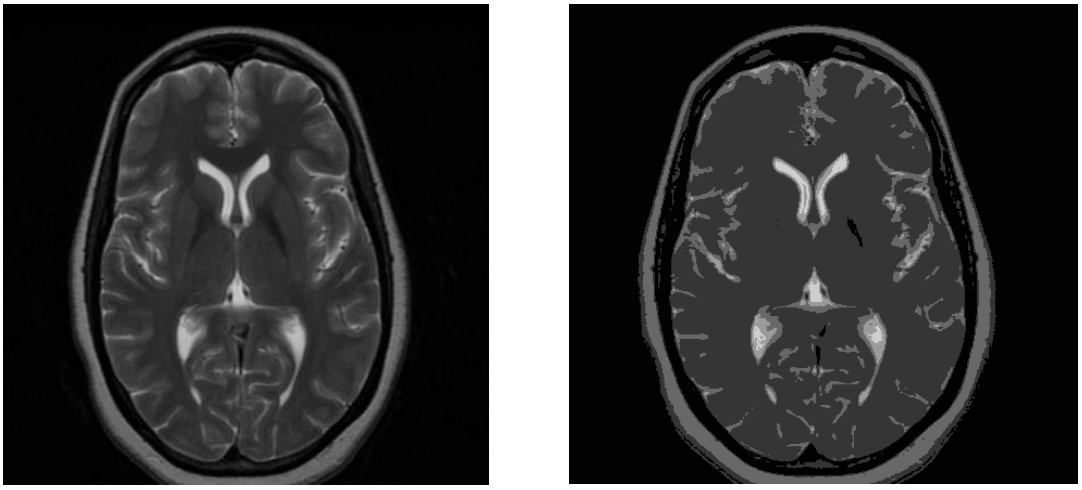
a) b)

Şekil 4.3. a) Orijinal ve b) Gbest-ABC ile kümelenmiş Tahoe görüntüsü.



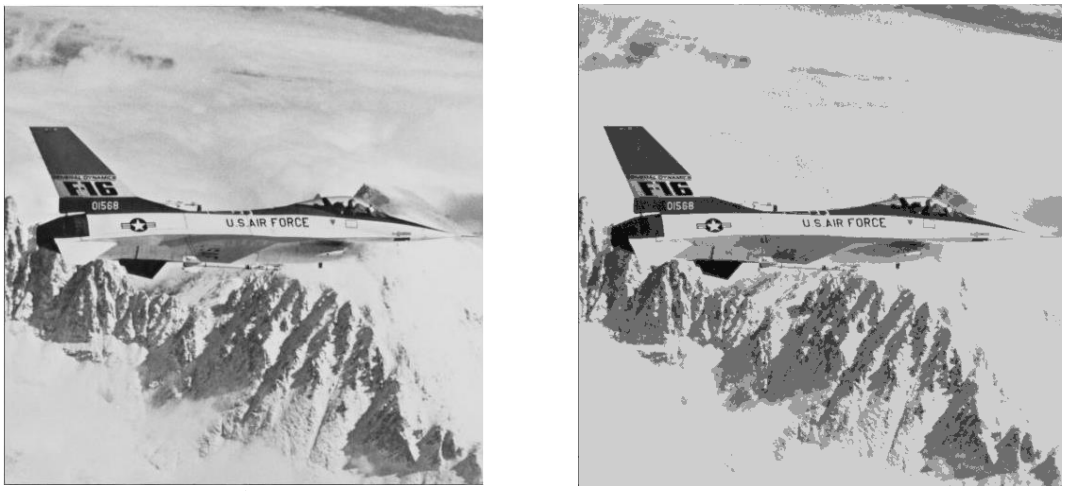
a) b)

Şekil 4.4. a) Orijinal ve b) Gbest-ABC ile kümelenmiş Morro Bay görüntüsü.



a) b)

Şekil 4.5. a) Orijinal ve b) Gbest-ABC ile kümelenmiş MRI görüntüsü.

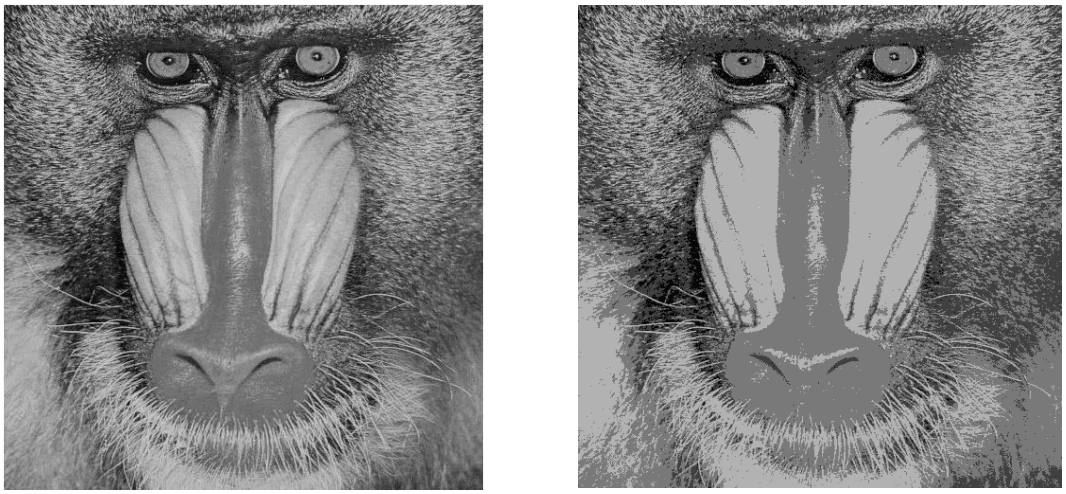


a) b)

Şekil 4.6. a) Orijinal ve b) Gbest-ABC ile kümelenmiş Airplane görüntüsü.

a) b)

Şekil 4.7. a) Orijinal ve b) Gbest-ABC ile kümelenmiş Lena görüntüsü.



a) b)

Şekil 4.8. a) Orijinal ve b) Gbest-ABC ile kümelenmiş Mandril görüntüsü.

a) b)

Şekil 4.9. a) Orijinal ve b) Gbest-ABC ile kümelenmiş Pepper görüntüsü.

4.4.2. Veri Kümeleme

Veri kümeleme için doğrulama indeks ve optimal küme sayısı sonuçları sırasıyla Tablolar 4.3 ve 4.4'te sunulmuştur. Tablo 4.3'e göre Gbest-ABC yöntemi 28 durumun 26'sında diğer yöntemlerden daha iyi VI indeks değerleri elde etmiştir. Gbest-ABC'nin diğer yöntemlerden üstünlüğü istatistiksel anlamlılık test sonuçlarında da rahatlıkla görülmektedir. Sadece Knowledge ve E-coli verilerindeki iki durumda kötü sonuç elde etmiştir. Gbest-ABC yöntemini performans olarak MRABC takip etmektedir. Bazı durumlarda Gbest-ABC ve MRABC yöntemlerinin performansı diğer yöntemlere göre en az iki kat daha iyidir. Diğer yöntemlerin performanslarına bakıldığında ise, 2 kümeli veri setlerinde PSO yöntemi ABC ve GABC'ye göre daha iyi kümeleme kalitesi sağlamıştır. 2'den fazla küme içeren veri setlerinde, ABC ve GABC yöntemleri Knowledge dışında PSO'dan daha iyi kümeleme kalitesi sağlamıştır. Standart ABC ve GABC yöntemleri arasında önemli bir farklılık yoktur. Genel olarak birbirine benzer sonuçlar üretmiştir. Sonuç olarak Gbest-ABC diğer yöntemlere göre VI indeks minimize etmede daha iyi sonuçlar elde etmiştir.

Tablo 4.3. Veri kümeleme için VI indeks sonuçları.

Veri Setleri	**Gbest ABC**	**GABC**	**MRABC**	**ABC**	**PSO**
Wisconsin	**0.0356** (0.0029)	0.1028 (0.0167) '+'	0.0373 (0.0021) '+'	0.1028 (0.0211) '+'	0.0504 (0.0074) '+'
Diabet	**0.0175** (0.0011)	0.1194 (0.0560) '+'	0.0194 (0.0032) '+'	0.1344 (0.0772) '+'	0.0284 (0.0079) '+'
Iris	**0.0871** (0.0082)	0.1283 (0.0141) '+'	0.0895 (0.0097) '+'	0.0881 (0.0074) '='	0.1269 (0.0329) '+'
Wine	**0.1664** (0.0206)	0.2861 (0.0256) '+'	0.1794 (0.0261) '+'	0.2688 (0.0286) '+'	0.3388 (0.0291) '+'
Knowledge	0.2335 (0.0134)	0.3386 (0.0248) '+'	0.2614 (0.0111) '+'	0.3254 (0.0203) '+'	**0.2187** (0.0273) '-'
E-coli	0.2334 (0.0214)	0.2377 (0.0262) '+'	**0.2090** (0.0216) '-'	0.2527 (0.0235) '+'	0.4465 (0.1055) '+'
Dermatology	**0.3078** (0.0347)	0.4197 (0.0395) '+'	0.3473 (0.0341) '+'	0.4279 (0.0398) '+'	0.4617 (0.1068) '+'

Elde edilen ortalama küme sayılarına bakıldığında ise, Gbest-ABC yönteminin Ecoli ve Dermatology veri setleri dışında en iyi performansı sağladığı Tablo 4.4'te görülmektedir. Dermatology ve E-coli veri setlerinde, GABC ve ABC yöntemleri optimum küme sayısına ortalama olarak en çok yaklaşmıştır. Ancak, diğer veri setlerinde bu yöntemlerin performansı genel olarak düşük kalmıştır. GABC'nin Iris veri setinde hiçbir zaman 3 küme sayısını bulamaması ve ABC'nin Knowledge veri setinde 4.9 gibi bir ortalama değer elde etmesi bu durumu destekler niteliktedir. Bir diğer ABC modeli MRABC ise genel olarak optimum küme sayısına yakın ortalama değerler elde etmiştir. PSO yöntemi de Dermatology veri seti dışında genel olarak optimum küme sayısına yakın ortalama değerler elde etmiştir. Sonuç olarak önerilen Gbest-ABC yönteminin elde ettiği ortalama küme sayısı değerleri, optimum küme sayılarına genel olarak en yakın değerlerdir.

Tablo 4.4. Veri kümeleme için elde edilen küme sayıları.

Veri Setleri	**No**	**Gbest ABC**	**GABC**	**MRABC**	**ABC**	**PSO**
Wisconsin	2	**2** (0)	2.066 (0.2537)	2 (0)	2.0667 (0.2537)	2 (0)
Diabet	2	**2** (0)	2.033 (0.1825)	2 (0)	2.2 (0.5509)	2 (0)
Iris	3	**3** (0)	4.133 (0.3457)	3.033 (0.1825)	3.1 (0.3051)	2.966 (0.3198)
Wine	3	**3.133** (0.3457)	3.666 (0.8022)	3.233 (0.4301)	3.8667 (0.8603)	**3.133** (0.3457)
Knowledge	4	**4.033 (0.1825)**	4.566 (0.6789)	4.233 (0.4301)	4.9 (0.8030)	4.066 (0.2537)
E-coli	5	5.433 (0.5040)	5.8 (0.6643)	5.533 (0.5074)	5.766 (0.6260)	**5.033** (0.3198)
Dermatology	6	5.833 (0.5921)	**5.933** (0.6914)	6.1 (0.8030)	5.733 (0.6397)	5.3 (0.5349)

Yöntemlerin veri kümeleme için 30 koşma üzerinden toplamda kaç defa optimum küme sayısını elde ettiğini gösteren sütun grafiği Şekil 4.10'da sunulmuştur. Wisconsin ve Diabet veri setlerinde Gbest-ABC, MRABC ve PSO optimum küme sayısını her koşmada bularak %100'lük bir başarı sağlamıştır. GABC ve ABC'nin de 2 kümeli bu veri setlerinde optimum küme sayısı bulma performansı %100 olmasa da iyi seviyede bulunmaktadır. Sonuç olarak yöntemlerin hepsi 2 kümeli veri setlerinde optimum küme sayısını elde etme noktasında iyi çalışmaktadır. 3 kümeli Iris veri seti ele alındığında

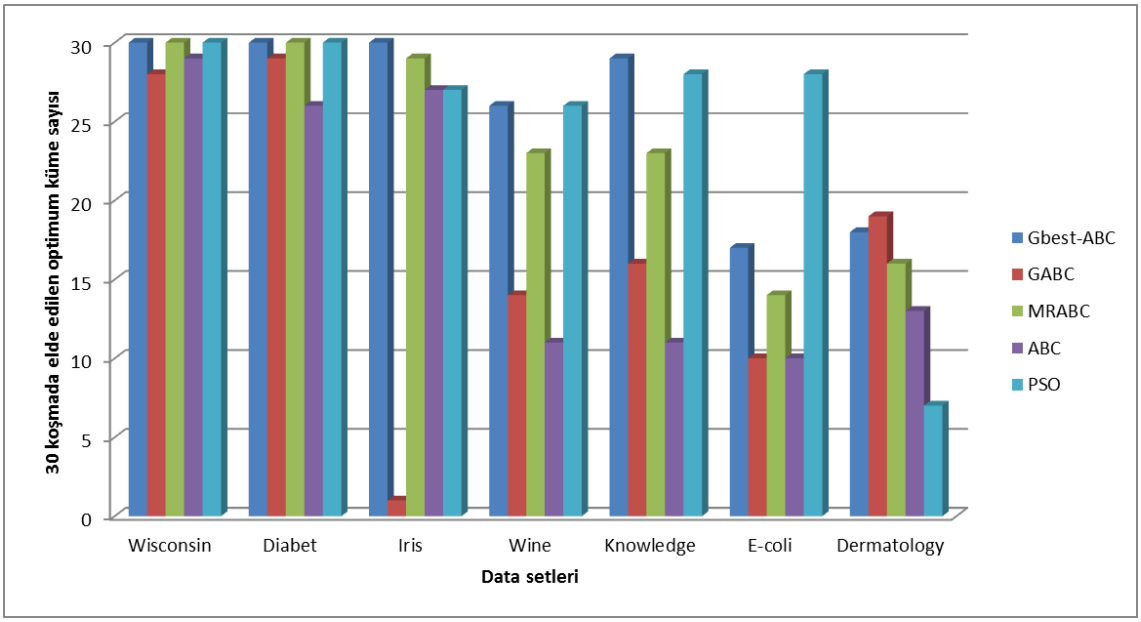
ise, Gbest-ABC yine her koşmada optimum küme sayısını bularak %100'lük bir performans sağlamıştır. GABC dışındaki diğer yöntemler de Iris veri setinde %100 olmasa da iyi bir performans ortaya koymuşlardır. Ancak GABC Iris veri setinde hiç optimum küme sayısını bulamamıştır. Bir diğer 3 kümeli Wine veri setinde Gbest-ABC ve PSO yaklaşık %87'lik bir oranla optimum küme sayısını bulmuştur. Onları yaklaşık %77'lik oranla MRABC takip etmiştir. Ancak GABC ve ABC yöntemleri Wine veri setinde düşük yüzdeyle optimum küme sayısı elde etmiştir. Sonuç olarak 3 kümeli veri setlerinde Gbest-ABC, PSO ve MRABC yötemleri kabul edilebilir bir performans göstermiştir. Ancak GABC ve ABC yöntemleri bu konuda sınıfta kalmıştır. 4 kümeli Knowledge veri setinde Gbest-ABC 30 koşmanın sadece 1'inde optimum küme sayısını elde edememiştir ve %97'lik bir oran ortaya koymuştur. Onu %93 ve %73'lük oranlarla sırasıyla PSO ve MRABC takip etmiştir. GABC ve ABC yöntemlerinin Knowledge veri setinde de performansları diğerlerine nazaran düşük seviyede kalmıştır. 5 kümeli E-coli veri setinde PSO %93'lük oranla diğer yöntemlerden önemli ölçüde iyi performans göstermiştir. Onu yaklaşık %60'lık oranla Gbest-ABC takip etmektedir. Diğer yöntemlerin performansları ise Ecoli veri setinde oldukça düşük kalmıştır. 6 kümeli Dermatology veri setinde GABC ve Gbest-ABC yöntemleri %60-%63 civarında bir oranla optimum küme sayısını bulmuştur. Geri kalan yöntemler, Dermatology veri setinde optimum küme sayısını elde etmede düşük seviyede kalmıştır. Özellikle PSO %23'lük oranla Dermatology veri setinde kötü bir performans ortaya koymuştur. Genel olarak önerilen Gbest-ABC otomatik kümeleme yöntemi, hem kümeleme kalitesi hem de optimum veya optimal küme sayısını bulmada diğer yöntemlerden daha iyi performans göstermiştir.

Veri setlerine ait Rand indeks sonuçları Tablo 4.5'te sunulmuştur. Elde edilen sonuçlara göre Diabet, Iris ve Knowledge veri setlerinde önerilen Gbest-ABC yöntemi en iyi değerlere ulaşmıştır. Wisconsin veri setinde ABC yöntemi, Wine veri setinde GABC yöntemi ve geriye kalan E-coli ve Dermatology veri setlerinde MRABC yöntemi en iyi ortalama Rand indeks değerlerini elde etmiştir. Ancak, genel olarak yöntemlerin hemen hemen hepsinin birçok durumda tutarlı ve iyi sonuçlar elde etmede başarısız olduğu görülmektedir. Elde edilen standart sapma değerleri bazı durumlarda oldukça yüksek seviyededir. Örneğin, MRABC yöntemi 0.6142 ile en iyi ortalama değere sahip olduğu E-coli veri setinde 0.1295 gibi çok yüksek bir standart sapma değerine sahiptir.

Dolayısıyla otomatik kümeleme yöntemleri VI indeks ve optimum küme sayısında gösterdikleri kabul edilebilir seviyedeki performansı Rand indeks sonuçlarında sergileyememiştir.

Tablo 4.5. Veri kümeleme için Rand indeks sonuçları.

Veri Setleri	Gbest ABC	GABC	MRABC	ABC	PSO
Wisconsin	0.6516 (0.0045)	0.7286 (0.0554)	0.6523 (0.0011)	**0.7501** (0.0706)	0.6533 (0.0208)
Diabet	**0.6523** (0)	0.6421 (0.0167)	**0.6523** (0)	0.6252 (0.0664)	0.6521 (0.0051)
İris	**0.7630** (0.0671)	0.6413 (0.0541)	0.7000 (0.0635)	0.7570 (0.1186)	0.6357 (0.1434)
Wine	0.4286 (0.0645)	**0.6710** (0.1650)	0.4893 (0.1281)	0.6438 (0.1588)	0.3734 (0.0682)
Knowledge	**0.4144** (0.0656)	0.4120 (0.0809)	0.3960 (0.0743)	0.3781 (0.0815)	0.3671 (0.0665)
E-coli	0.4862 (0.1792)	0.4801 (0.2095)	**0.6142** (0.1295)	0.5556 (0.1997)	0.4425 (0.1647)
Dermatology	0.5004 (0.1150)	0.5879 (0.1176)	**0.6051** (0.1075)	0.6046 (0.1351)	0.3138 (0.1097)



Şekil 4.10. Veri kümeleme için 30 koşma üzerinden toplamda kaç defa optimum küme sayısının elde edildiğini gösteren grafik.

4.5. Sonuçlar

Bu bölümde vektörel arama ve küresel en iyi çözüm mekanizmalarının standart ABC algoritmasına entegrasyonu ile yeni bir otomatik kümeleme yöntemi geliştirilmiştir. Geliştirilen Gbest-ABC yöntemi bilinen görüntü ve veri setlerine uygulanmıştır. Elde edilen sonuçlar neticesinde, genel olarak Gbest-ABC otomatik kümeleme yöntemi PSO, ABC, MRABC ve küresel en iyi çözüm mekanizmasını kullanan GABC'den hem kümeleme kalitesi hem de küme sayısı açısından daha iyi performans göstermiştir. Ayrıca standart ABC algoritmasının küme sayısı saptamadaki performansı ilk olarak bu çalışma ile incelenmiştir. Önerilen ABC modeli VI indeks ve optimal küme sayısını elde etmede kabul edilebilir bir performans göstermiştir. Ancak, dolaylı olarak hesaplanan Rand indeks sonuçları arzu edilen seviyede olmamıştır.

5. BÖLÜM

İKİLİ ABC VE K-MEANS'E DAYALI HİBRİT OTOMATİK KÜMELEME YÖNTEMLERİ

Küme sayısının belirlenmesini bir ikili optimizasyon problemi olarak da ele almak mümkündür. Kuncheva ve Bezdek tarafından geliştirilen prensipleri kullanan DCPSO [157] otomatik kümeleme yöntemi bu alandaki en çok bilinen yöntemlerden biridir. DCPSO yöntemi K-means ile senkronize bir şekilde çalışarak merkezleri güncellemeye çalışır. Yöntem her ne kadar başarılı sonuçlar üretse de BPSO'nun yapısından kaynaklanan yerele takılabilme dezavantajı vardır. ABC algoritmasının diğer evrimsel hesaplama tekniklerine göre yerele takılma problemi ile daha az karşılaştığı bilinen bir gerçektir. Ancak, bir ikili ABC modeliyle oluşturulan otomatik kümeleme yöntemi de henüz araştırmacılar tarafından geliştirilmemiştir. Bu bölümde ikili ABC ve K-means algoritmalarına dayalı otomatik hibrit kümeleme yöntemleri önerilecektir.

Bu bölümün amacı, küme sayısının belirlenmesini bir ikili (binary) optimizasyon problemi olarak ele almak ve ikili ABC ile bu problemi çözmektir. Bunun gerçekleştirilmesi için ilk olarak genetik operatörlerden esinlenerek yeni bir genetik ikili ABC modeli önerilmiş ve buna dayalı yeni bir otomatik kümeleme yöntemi (GBABC) tanımlanmıştır. İkinci olarak vektörler arasındaki benzerlik esasına dayalı ayrık ABC modeli olan DisABC'nin üzerinde bazı iyileştirmeler yapılmış ve bu modelin yeni versiyonu üzerine yeni bir otomatik kümeleme yöntemi (IDisABC) tanımlanmıştır. GBABC ve IDisABC yöntemlerinin performansları görüntü ve veri setleri üzerinde bilinen ikili ABC, PSO ve GA modelleri ile karşılaştırılarak değerlendirilmiştir. GBABC ve IDisABC yöntemlerinin performanslarının değerlendirme ve analizi için şu yol takip edilmiştir:

- bilinen ikili evrimsel hesaplama tabanlı yöntemlerle kümeleme kalitesi üzerinden karşılaştırılması,
- bilinen ikili evrimsel hesaplama tabanlı yöntemlerle doğru küme sayısının saptanması üzerinden değerlendirilmesi,
- birbirleri ile kümeleme kalitesi ve küme sayısı üzerinden karşılaştırılması.

Bu bölümde ilk olarak geliştirilen ikili ABC modelleri sunulmakta ve onların küme sayısı saptanmasına yönelik uygulamaları açıklanmaktadır. Daha sonra ilgili yöntemlerin deneysel dizayn ve çalışmalarına yer verilmektedir. Son olarak da çalışmaların genel bir değerlendirmesi yapılmaktadır.

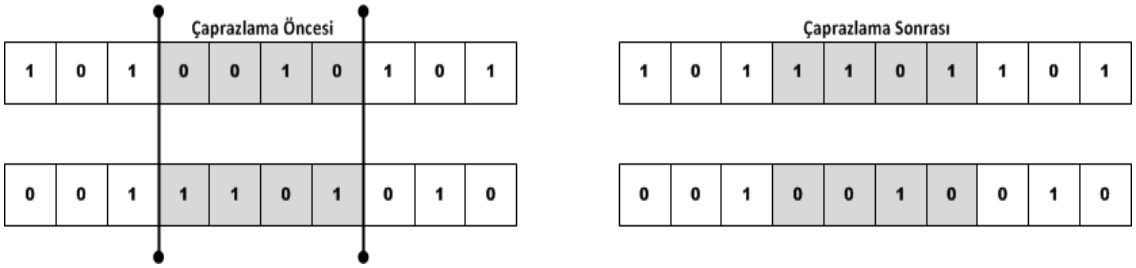
5.1. Geliştirilen İkili ABC Modelleri

5.1.1. Genetik İkili ABC Modeli

Mevcut bir optimizasyon algoritmasını geliştirmek için takip edilen iki yol vardır [259]: hibritleme (hybridization) ve modifikasyon (modification). Hibritleme, iki farklı mekanizmanın bilinçli olarak veya doğal yollarla manipüle edilerek karıştırılması işlemidir. Modifikasyon ise mevcut mekanizmadaki bazı parçaları iç veya dış etkiyle modifiye etme işlemidir. Doğadaki olayları gözlemlediğimizde de kütürel veya sosyal içeriğin kaynağı olarak tanımlanan memler (memes) bir sonraki nesle aktarılırken bazı değişikliklere maruz kalarak modifikasyona uğrayabilir veya başka memler ile birleşerek daha güçlü memleri ortaya çıkarabilir. Bu yönüyle memler sosyal gen olarak da nitelendirilir. Ancak memler soyut kavramlardır. Dilin yaşamı bu konsepte güzel bir örnektir. Toplumun ihtiyaç ve istekleri doğrultusunda diller yeni kelimeler alabilir. Aynı şekilde bazı kelimeler zamanla popülerliğini yitirip dilden uzaklaşabilir. Algoritma perspektifinden bakılacak olursa, iki ve daha fazla farklı algoritmanın doğru şekilde birleşmesi veya bir algoritmanın başka bir algoritmanın kullandığı operatörlerle değişime uğraması ile algoritmanın problem çözebilme kabiliyeti artabilir. Bu perspektife uygun olarak evrimsel sistemler içinde yerel arama tekniklerini kullanan memetik algoritmalar [260] araştırmacılar tarafından oldukça kabul görmektedir. Bundan yola çıkılarak bu çalışmada, genetik operatör ve sistemler ile dizayn edilerek memetik algoritmalara benzeyen bir genetik ikili ABC modeli (GBABC) geliştirilmiştir [261].

GBABC modelinde iki tane temel genetik operatör kullanılmıştır:

- İki noktalı çaprazlama: Çaprazlama yapılacak ikili vektörler üzerinde iki pozisyon rasgele seçilir. Pozisyonlar arasında kalan bölgeler ikili vektörler arasında değiş tokuş yapılarak iki yeni vektör üretilir. Şekil 5.1'de iki noktalı çaprazlamaya görsel bir örnek sunulmuştur.
- Değiş-tokuş (swap): İkili vektör üzerinde 1 değerine sahip olan bir pozisyon rasgele seçilir ve 0'a çevrilir. Daha sonra vektör üzerinde 0 değerine sahip olan bir pozisyon (0'a çevrilen pozisyon hariç) rasgele seçilir ve 1'e çevrilir. Şekil 5.2'de değiş-tokuş işlemine bir örnek gösterilmiştir.



Şekil 5.1. İki noktalı çaprazlamanın uygulanışı.

Şekil 5.2. Değiş-tokuş operatörünün uygulanışı.

Standart ABC'de Denklem 2.6 ile tanımlanan komuşuluk arama mekanizmasının yerini GBABC'de genetik komşuluk arama (GKA) mekanizması almaktadır. GKA mekanizmasının temel adımları şu şekildedir:

1. Üzerine yoğunlaşılan D boyutlu X_i kaynağının komşuluğunda rasgele iki tane X_k ve X_j kaynağı seç.
2. Popülasyonun haricinde sıfırlardan oluşan $1 \times D$ boyutlu bir *Zero* vektörü tanımla.
3. X_i, X_k, X_j, *Zero* ve *Gbest* kaynaklarını ebeveyn kaynaklar olarak ata.

4. Ebeveyn kaynakları bire-bir (one-to-one) ve örten (onto) ilişkisini esas alarak rasgele birbirleriyle eşleştir ve eşleştirilen kaynaklar arasında iki noktalı çaprazlama uygula.
5. Çaprazlama işleminin sonucunda ortaya çıkan çocuk kaynaklar arasında sadece sıfırlardan oluşan bir vektör mevcut olması durumunda, sıfırlardan oluşan kaynağın kullanıcı tarafından tanımlanmış *Th* sayısı kadar pozisyonunu rasgele 1'e çevir.
6. Çaprazlama sonucu üretilen çocuk kaynakların her birine değiş-tokuş operatörü uygula ve bu yolla üretilen kaynakları torun (ikinci nesil) olarak nitelendir.
7. Çocuk ve torun kaynaklardan en iyisini V_i kaynağı olarak belirle.

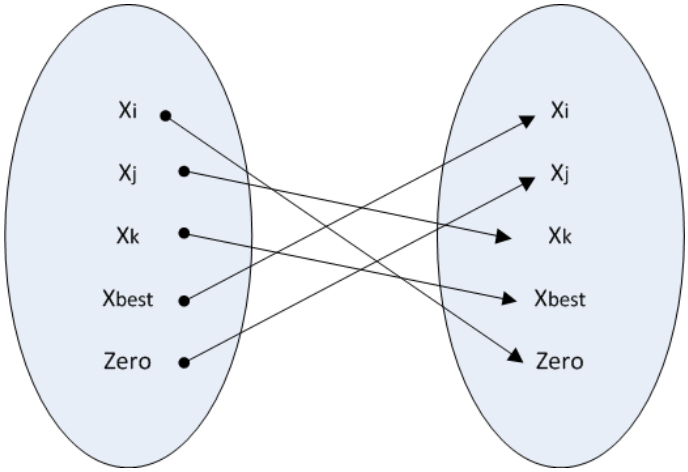
GKA'da *Zero* kaynağı tanımlanarak çocuk kaynakların oluşumunda çeşitliliğin artması amaçlanmıştır. *Th* parametresi, amaç fonksiyon hesaplanması sırasında oluşabilecek hataların önlenmesi amacıyla tanımlanmıştır. *Th* parametresinin değeri 3 olarak alınmıştır. Ancak GBABC modelinin etkili arama mekanizması sebebiyle *Th* parametresinin modelin performansı üzerinde önemli bir etkisi yoktur. Dolayısıyla Th parametresi için [1,*D*] arası herhangi bir değer belirlenmesi de mümkündür. GBABC modelinin kaba kodu Şekil 5.3'te sunulmuştur.

Örnek: GKA mekanizmasının işleyişini daha detaylı açıklamak amacıyla burada küçük bir uygulama gösterilmiştir. Üzerine yoğunlaşılan kaynak X_i=(1 0 0 1 0 1 0 1 0 1 0 1), *Gbest*=(1 1 0 1 0 0 1 1 0 0) ve X_i kaynağının seçilen komşuları X_k=(0 0 1 0 1 0 1 0 1 1) ve X_j=(1 0 1 1 1 0 1 1 0 1) olarak verilmiş olsun.

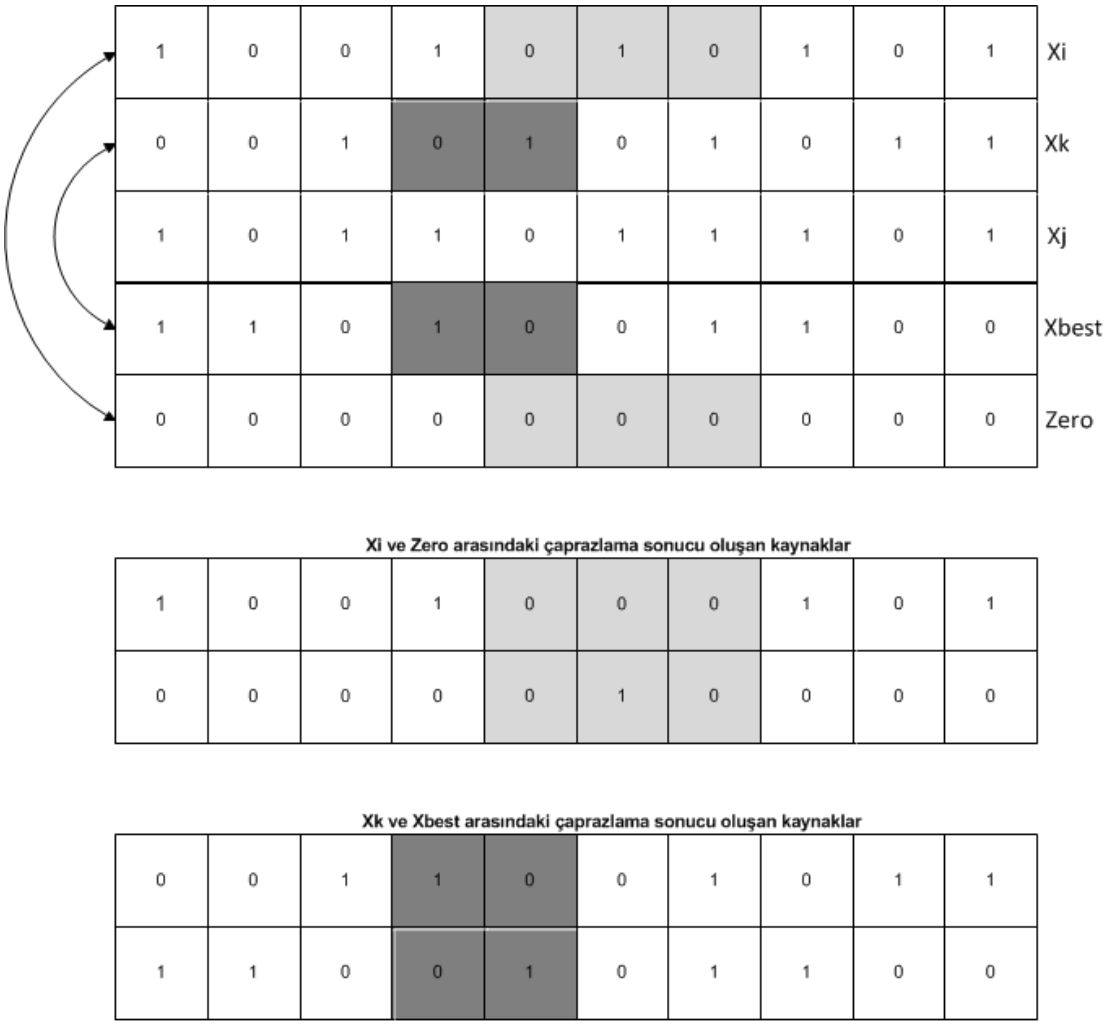
İlk olarak 1×12'lik 0'lardan oluşan bir *Zero* vektörü tanımlanır. X_i, X_k, X_j, *Gbest* ve *Zero* kaynakları ebeveyn kaynaklar olarak atanır ve bu kaynaklar arasında bire-bir ve örten esasına dayalı olarak Şekil 5.4'teki gibi eşleştirme yapılır. Eşleştirilen ebeveyn kaynaklar arasında Şekil 5.5'teki gibi iki noktalı çaprazlama uygulanarak çocuk kaynaklar üretilir. Üretilen çocuk kaynaklar Şekil 5.6'daki gibi değiş-tokuş işlemine tabi tutularak torun kaynaklar üretilir. Çocuk ve torun kaynakların kaliteleri değerlendirilerek en kaliteli kaynak V_i kaynağı olarak atanır. Böylece üzerine yoğunlaşılan bir kaynaktan türetilecek komşu kaynak için toplamda 20 tane amaç fonksiyon değerlendirmesi gerçekleştirilmiş olur. GBABC bu yönüyle literatürdeki bilinen evrimsel hesaplama tabanlı yöntemlerden ayrılır.

```
begin
    SN, MCN ve limit parametrelerini belirle;
    Baslangic populasyonunu uret;
    Gbest kaynagini belirle;
    for cycle ← 1 to MCN do
        foreach gorevli ari i do
            x_i kaynaginin komsulugunda x_k ve x_j kaynaklarini sec;
            D boyutlu 0'lardan olusan bir Zero vektor tanimla;
            x_i, x_k, x_j, Zero ve Gbest kaynaklarini ebeveyn olarak ata;
            Ebeveyn kaynaklari bire-bir ve orten prensibine gore eslestir;
            foreach ebeveyn kaynak e_z do
                e_z ile esi arasinda iki noktali caprazlama uygula;
                Ortaya cikan kaynaklari cocuk kaynaklara ata;
            end
            foreach cocuk kaynak c_z do
                c_z kaynagina degis-tokus operatoru uygula;
                Ortaya cikan kaynagi torun kaynaklara ata;
            end
            Cocuk ve torun kaynaklar arasindan en iyi olani v_i kaynagi olarak ata;
            x_i ve v_i arasinda ac gozlu yaklasim uygula;
            Cocuk ve torun kaynaklari temizle;
        end
        Kaynaklarin olasilik degerlerini (p_i) Denklem (2.7) ile hesapla;
        foreach gozcu ari i do
            Kaynaklarin olasilik degerlerine bagli olarak bir kaynak (x_i) sec;
            x_i kaynaginin komsulugunda x_k ve x_j kaynaklarini sec;
            D boyutlu 0'lardan olusan bir Zero vektor tanimla;
            x_i, x_k, x_j, Zero ve Gbest kaynaklarini ebeveyn olarak ata;
            Ebeveyn kaynaklari bire-bir ve orten prensibine gore eslestir;
            foreach ebeveyn kaynak e_z do
                e_z ile esi arasinda iki noktali caprazlama uygula;
                Ortaya cikan kaynaklari cocuk kaynaklara ata;
            end
            foreach cocuk kaynak c_z do
                c_z kaynagina degis-tokus operatoru uygula;
                Ortaya cikan kaynagi torun kaynaklara ata;
            end
            Cocuk ve torun kaynaklar arasindan en iyi olani v_i kaynagi olarak ata;
            x_i ve v_i arasinda ac gozlu yaklasim uygula;
            Cocuk ve torun kaynaklari temizle;
        end
        if Bir kaynagin ceza puani > limit then
            Kasif ari Denklem (2.18) ile yeni bir kaynak belirler;
        end
        En iyi kaynagi belirle;
    end
end
```

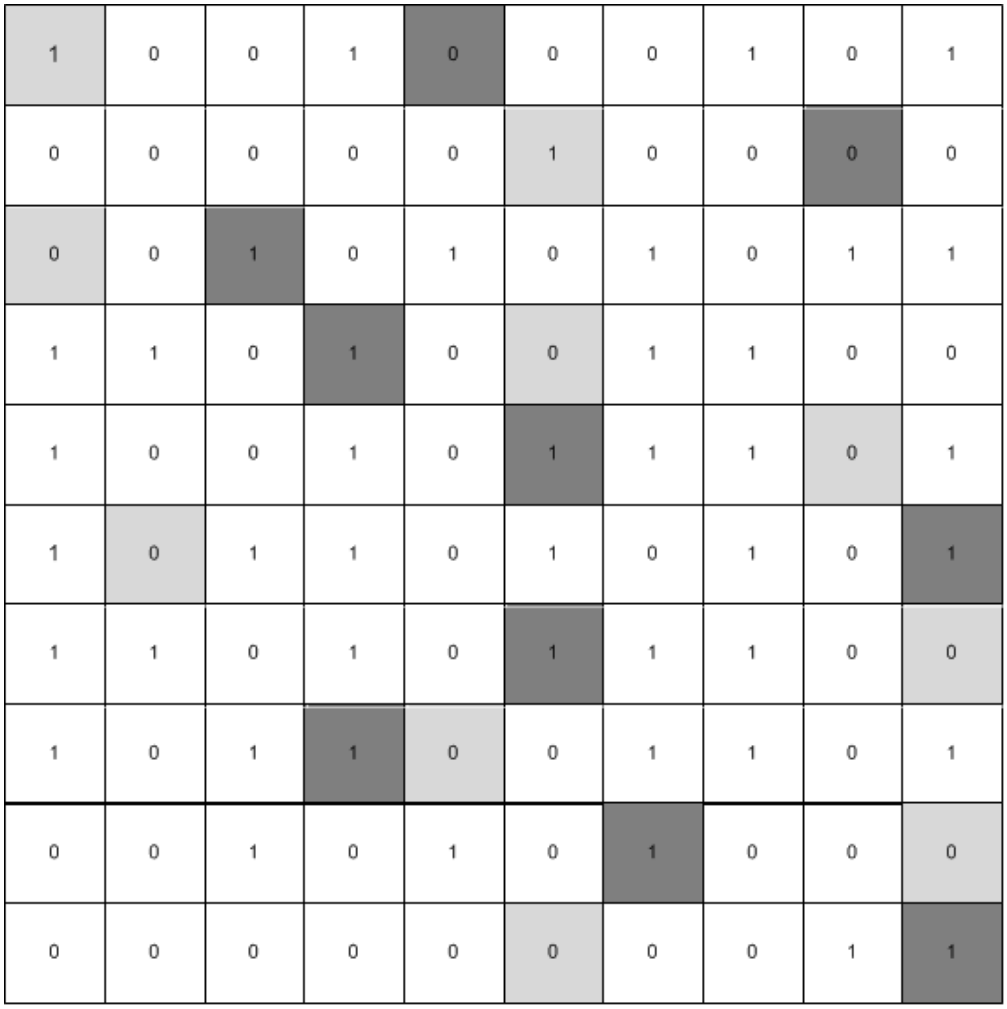
Şekil 5.3. GBABC modelinin genel kaba kodu.



Şekil 5.4. GBABC'de bire-bir ve örten eşleştirmenin uygulanışı.



Şekil 5.5. GBABC'de çaprazlama operatörünün uygulanışı.



Şekil 5.6. GBABC'de değiş-tokuş operatörünün uygulanışı.

5.1.2. İyileştirilmiş DisABC Modeli

DisABC [235] ikili vektörler arasındaki Jaccard katsayısı benzerlik/farklılık esasına dayanır. DisABC'de yeni kaynak V_i 'yi üretmek için X_i ve V_i arasındaki M11, M10 ve M01 bit sayılarının Denklem 2.17 ile tanımlanan matematiksel programlama ile saptanması gerekmektedir. Bu işlemin sonucunda birden fazla M11, M10 ve M01 kombinasyonu elde edilebilmektedir. DisABC bu kombinasyonlardan ilk elde ettiği kombinasyonu değerlendirmeye almaktadır. Diğer kombinasyonlar DisABC tarafından değerlendirmeye alınmamaktadır. Ancak değerlendirmeye alınmayan kombinasyonlardan ilk kombinasyona göre daha kaliteli bir V_i kaynağı elde etme ihtimali de mevcuttur. DisABC hakkında detaylı bilgi Bölüm 2.3.2'de sunulmuştur.

DisABC'nin yeni çözüm üretme mekanizması matematiksel programlama sonucu ortaya çıkabilecek tüm kombinasyonları kullanabilecek şekilde iyileştirilmiştir [262].

İyileştirilmiş DisABC (IDisABC) modelinin kaynak üretme mekanizmaları şu şekilde işlemektedir:

1. İyileştirilmiş rasgele seçim: V_i $1\times D$'lik sıfır vektörü olarak belirlenir. Eğer M11, M10 ve M01 için tek bir optimal kombinasyon varsa, DisABC'de olduğu gibi X_i vektöründe 1 değeri taşıyan M11 tane biti rasgele seç ve V_i vektöründe ilgili bitlerin değerini 1'e çevir. Ardından X_i vektöründe 0 değeri taşıyan M10 tane biti seç ve V_i vektöründe ilgili bitlerin değerini 1'e çevir. Eğer M11, M10 ve M01 için birden fazla optimal kombinasyon varsa, her bir kombinasyon için bu işlemin başında olduğu gibi bir V_i kaynağı üretilir. Akabinde üretilen V_i kaynakları, X_i ve *Gbest* arasında iki noktalı çaprazlama operatörü uygulanır. Çaprazlama sonucu ortaya çıkan kaynaklara da değiş-tokuş (swap) operatörü uygulanır. Rasgele seçim, çaprazlama ve değiş-tokuş operatörleri sonucu ortaya çıkan kaynaklar arasında da en iyi kaynak final komşu kaynak V_i olarak atanır.
2. İyileştirilmiş aç gözlü seçim: V_i $1\times D$'lik sıfır vektörü olarak belirlenir. Eğer M11, M10 ve M01 için tek bir optimal kombinasyon varsa, DisABC'de olduğu gibi hem global en iyi vektör (*Gbest*) hem de X_i vektöründeki değeri 1 olan M11 tane biti seç ve V_i vektöründe ilgili bitlerin değerini 1'e çevir. Bazı durumlarda hem *Gbest* hem de X_i vektöründe M11 tane 1 değeri taşıyan bit olmayabilir. Eğer seçilen bit sayısı M11'den az ise, kalan bitleri X_i vektöründen 1 değeri taşıyanların arasından seç ve V_i vektöründe ilgili bitlerin değerini 1'e çevir. Ardından X_i vektöründe değeri 0, *Gbest* vektördeki değeri 1 olan M10 tane bit seç ve V_i vektöründe ilgili bitlerin değerini 1'e çevir. Eğer seçilen bit sayısı M10'dan az ise, kalan bitleri X_i vektöründe değeri 0 olanların arasından seç ve V_i vektöründe ilgili bitlerin değerini 1'e çevir. Eğer M11, M10 ve M01 için birden fazla optimal kombinasyon varsa, her bir kombinasyon için bu işlemin başında olduğu gibi bir V_i kaynağı üretilir. Akabinde üretilen V_i kaynakları, X_i ve *Gbest* arasında iki noktalı çaprazlama operatörü uygulanır. Çaprazlama sonucu ortaya çıkan kaynaklara da değiş-tokuş (swap) operatörü uygulanır. Aç gözlü seçim, çaprazlama ve değiş-tokuş operatörleri sonucu ortaya çıkan kaynaklar arasında da en iyi kaynak final komşu kaynak V_i olarak atanır.

```
begin
    for cycle ← 1 to MCN do
        foreach gorevli ari i do
            x_i kaynaginin komsulugunda x_k kaynagini sec;
            Denklem (2.13) ile Dissimilarity(x_i, x_k) hesapla;
            Denklem (2.17) ile M_11, M_10 ve M_01 degerlerini bul;
            if rand(0,1) < 0.5 then
                if M kombinasyon sayisi == 1 then
                    v_i = rasgelesecim(M_11, M_10, M_01, X_i)
                else
                    foreach kombinasyon j do
                        v_j = rasgelesecim(M_11j, M_10j, M_01j, X_i);
                    end
                    Uretilen v_j'ler, Gbest ve X_i arasinda iki noktali caprazlama uygula;
                    Caprazlama sonucu uretilen kaynaklara degis-tokus uygula;
                    Caprazlama ve degis-tokus sonucu uretilen kaynaklardan en iyisini v_i olarak ata;
                end
            else
                if M kombinasyon sayisi == 1 then
                    v_i = acgozlusecim(M_11, M_10, M_01, X_i, Gbest)
                else
                    foreach kombinasyon j do
                        v_j = acgozlusecim(M_11j, M_10j, M_01j, X_i, Gbest);
                    end
                    Uretilen v_j'ler, Gbest ve X_i arasinda iki noktali caprazlama uygula;
                    Caprazlama sonucu uretilen kaynaklara degis-tokus uygula;
                    Caprazlama ve degis-tokus sonucu uretilen kaynaklardan en iyisini v_i olarak ata;
                end
            end
            v_i ve x_i arasinda ac gozlu yaklasim uygula;
        end
        Kaynaklarin olasilik degerlerini (p_i) Denklem (2.7) ile hesapla;
        foreach gozcu ari i do
            Kaynaklarin olasilik degerlerine bagli olarak bir kaynak (x_i) sec;
            x_i kaynaginin komsulugunda x_k kaynagini sec;
            Denklem (2.13) ile Dissimilarity(x_i, x_k) hesapla;
            Denklem (2.17) ile M_11, M_10 ve M_01 degerlerini bul;
            if rand(0,1) < 0.5 then
                if M kombinasyon sayisi == 1 then
                    v_i = rasgelesecim(M_11, M_10, M_01, X_i)
                else
                    foreach kombinasyon j do
                        v_j = rasgelesecim(M_11j, M_10j, M_01j, X_i);
                    end
                    Uretilen v_j'ler, Gbest ve X_i arasinda iki noktali caprazlama uygula;
                    Caprazlama sonucu uretilen kaynaklara degis-tokus uygula;
                    Caprazlama ve degis-tokus sonucu uretilen kaynaklardan en iyisini v_i olarak ata;
                end
            else
                if M kombinasyon sayisi == 1 then
                    v_i = acgozlusecim(M_11, M_10, M_01, X_i, Gbest)
                else
                    foreach kombinasyon j do
                        v_j = acgozlusecim(M_11j, M_10j, M_01j, X_i, Gbest);
                    end
                    Uretilen v_j'ler, Gbest ve X_i arasinda iki noktali caprazlama uygula;
                    Caprazlama sonucu uretilen kaynaklara degis-tokus uygula;
                    Caprazlama ve degis-tokus sonucu uretilen kaynaklardan en iyisini v_i olarak ata;
                end
            end
            v_i ve x_i arasinda ac gozlu yaklasim uygula;
        end
        if Bir kaynagin ceza puani > limit then
            Kasif ari Denklem (2.18) ile yeni bir kaynak belirler;
        end
    end
end
```

Şekil 5.7. IDisABC modelinin genel kaba kodu.

IDisABC'deki iyileştirilmiş rasgele ve aç gözlü seçim mekanizmalarının işleyişine rassal olarak karar verilir. Bir X_i kaynağı için [0,1] arasında üretilen rasgele sayı 0.5 değerinden küçük olması durumunda iyileştirilmiş rasgele seçim mekanizması, aksi takdirde iyileştirilmiş aç gözlü seçim mekanizması uygulanır.

Örnek: DisABC ve IDisABC arasındaki farkı göstermek için [235]'teki örneğe burada da yer verilmiştir. Kaynak üretimi için rasgele seçim mekanizması kullanılmıştır. X_i=(1011010100) ve X_k=(1000101101) vektörleri verilmiş ve Q değeri 0.7 olsun. X_i ve X_k vektörleri arasındaki M11, M10 ve M01 değerleri sırasıyla 2, 3 ve 3 olarak bulunur. Buradan Q×Dissimilarity(X_i,X_k)=$0.7\times\left(1-\frac{2}{2+3+3}\right)$= 0.525 elde edilir. X_i vektöründeki toplam 1 sayısı (m_1) 5 ve toplam 0 sayısı (m_0) 5'tir. Denklem 2.17 ile V_i ve X_i arasındaki M11, M10 ve M01 belirlenir:

$$\min\left\{1-\frac{M_{11}}{M_{11}+M_{10}+M_{01}}-0.525\right\}$$

$$M_{11}+M_{01}=5$$

$$M_{10}\le 5$$

$$M_{11},M_{10},M_{01}>0 \; ve \; tamsayı$$

Burada denklemi minimum yapan değer 0.025'tir ve bu değeri sağlayan M11, M10 ve M01 kombinasyonları şunlardır:

- M11=3, M01=2 ve M10=1;
- M11=4, M01=1 ve M10=3;
- M11=5, M01=0 ve M10=5;

DisABC, bu kombinasyonlardan ilk ulaştığı M11=3, M01=2 ve M10=1 değerini V_i kaynağı üretimi için kullanır. İlk olarak V_i 1×10'luk sıfır vektörü olarak belirlenir. Rasgele seçim mekanizmasına göre X_i vektörünün 1 olduğu pozisyonlardan M11=3 tane pozisyon {1,4,8} rasgele seçilmiş olsun. V_i vektörü ilgili pozisyonların 1'e çevrilmesi ile (1001000100) haline gelir. X_i vektörünün 0 olduğu pozisyonlardan M10=1 tane pozisyon {9} rasgele seçilmiş olsun. Vi vektörü ilgili pozisyonun 1'e çevrilmesi ile (1001000110) haline gelerek son şeklini alır.

IDisABC, üç M11, M01 ve M10 kombinasyonun her biri için rasgele seçime göre üç tane V_i kaynağı üretir:

- M11=3, M01=2 ve M10=1 için $V1_i$=(1 0 0 1 0 0 0 1 1 0)
- M11=4, M01=1 ve M10=3 için $V2_i$=(1 0 0 1 1 1 0 1 0 0)
- M11=5, M01=0 ve M10=5 için $V3_i$=(1 0 1 1 0 1 0 1 0 0)

X_i, *Gbest* ve üretilen V_i kaynaklarını sırasıyla rasgele olarak birbiriyle eşleyerek aralarında iki noktalı çaprazlama gerçekleştirilir ve çaprazlama işlemi sonucu ortaya çıkan kaynakların her birine de değiş-tokuş uygulanarak yeni kaynaklar üretilir. Çaprazlama ve değiş-tokuş işlemi ile üretilen yeni kaynakların en iyisi de final V_i kaynağı olarak atanır. IDisABC modelinin kaba kodu Şekil 5.7'de sunulmuştur.

5.1.3. Önerilen İkili ABC Modelleri ile Otomatik Kümeleme

Geliştirilen GBABC ve IDisABC ikili ABC algoritmalarının küme sayısının otomatik olarak saptanması için Kuncheva ve Bezdek [263] tarafından sunulan prensipler takip edilmiş ve K-means ile etkileşimli hibrit kümeleme yöntemleri oluşturulmuştur. Popülasyon bireylerinin temsili için ikili aktivasyon tabanlı kodlama kullanılır. İkili aktivasyon tabanlı kodlama için öncelikle kullanıcı tarafından tanımlanmış bir *M* merkez seti *Z* verisinden rasgele seçilir. Popülasyon bireyleri de bu merkez setini etkinleştirecek 0 ve 1 aktivasyon kodlarından oluşur. Bireylerin aktivasyon kodlarının üretimi Denklem 2.18 ile gerçekleştirilir. Popülasyon bireyleri IDisABC veya GBABC ile iteratif olarak geliştirilmeye çalışılır. Bireylerin kalitesi VI indeksi ile hesaplanır. Maksimum çevrim sayısı sağlanınca, araştırma sonunda bulunan en iyi birey *Gbest*'in etkinleştirdiği M_{gbest} merkez seti K-means uygulanarak güncellenir. *M* merkez setinin M_{gbest} dışında kalan merkez seti M_{kalan} ise *Z* verisinden rasgele seçilerek tekrar oluşturulur. Böylece *M* merkez seti tamamen güncellenmiş olur. İşlem, güncellenmiş *M* merkez seti ile tekrarlanır. Durdurma kriteri (*TC*) sağlandığında elde edilen M_{gbest} seti, *Z* verisindeki optimal veya optimum küme setini ifade eder. M_{gbest} setinin büyüklüğü (N_{merkez}) de *Z* verisindeki optimal veya optimum küme sayısını ifade etmektedir. İkili ABC modelleri ile otomatik kümeleme uygulamasının genel kaba kodu Şekil 5.8'de sunulmuştur.

```
begin
    SN, MCN, limit, TC ve N_c parametrelerini belirle;
    Z verisinden rasgele N_c tane merkezden olusan M setini sec;
    for t ← 1 to TC do
        Baslangic populasyonunu uret;
        for cycle ← 1 to MCN do
            foreach gorevli ari i do
                x_i kaynaginin komsulugunda kaynak arama mekanizmasi ile v_i bul;
                v_i kaynagini kullanarak M_i (M_i ⊂ M) merkez setini etkinlestir;
                foreach veri elemani z_p do
                    z_p'nin M_i setindeki merkezlere olan mesafelerini hesapla;
                    z_p'yi en yakin M_i setindeki kumeye ata;
                end
                v_i kaynaginin kalitesini (f(v_i, Z)) Denklem (4.5) ile degerlendir;
                x_i ve v_i arasinda ac gozlu yaklasim uygula;
            end
            Kaynaklarin olasilik degerlerini (p_i) Denklem (2.7) ile hesapla;
            foreach gozcu ari i do
                Kaynaklarin olasilik degerlerine bagli olarak bir kaynak (x_i) sec;
                x_i kaynaginin komsulugunda kaynak arama mekanizmasi ile v_i bul;
                v_i kaynagini kullanarak M_i (M_i ⊂ M) merkez setini etkinlestir;
                foreach veri elemani z_p do
                    z_p'nin M_i setindeki merkezlere olan mesafelerini hesapla;
                    z_p'yi en yakin M_i setindeki kumeye ata;
                end
                v_i kaynaginin kalitesini (f(v_i, Z)) Denklem (4.5) ile degerlendir;
                x_i ve v_i arasinda ac gozlu yaklasim uygula;
            end
            if Bir kaynagin ceza puani > limit then
                Kasif ari Denklem (2.18) ile yeni bir kaynak belirler;
            end
            En iyi kaynagi (Gbest) belirle;
            Gbest'i kullanarak M_gbest (M_gbest ⊂ M) setini etkinlestir;
            M setinde M_gbest setinin haricindeki M_kalan merkez setini Z verisinden rasgele sec;
            M_gbest setini K-means uygulayarak guncelle;
            M = M_gbest ∪ M_kalan
        end
    end
end
```

Şekil 5.8. İkili ABC modelleri ile otomatik kümeleme uygulaması.

Örnek: İkili ABC modeli ile otomatik kümeleme uygulamasının daha iyi anlaşılması için burada bir örneğe yer verilmiştir. Tek boyutlu ve 50 elemanlı bir *Z* verisi için N_c=10 olsun. *M* merkez seti *Z* verisinden aşağıdaki şekilde seçilmiş olduğunu varsayalım:

10	29	60	88	120	145	166	200	223	245

Bir X_i kaynağı da aşağıdaki şekilde verilmiş olsun:

0	1	0	1	1	0	1	1	0	0

Buradan X_i kaynağının etkinleştirdiği M_i seti [29,88,120,166,200] olur. Bu set kullanılarak X_i kaynağının kalitesi Denklem 4.5 ile hesaplanır.

Maksimum çevrim sayısı sonucu elde edilen *Gbest* kaynağı da aşağıdaki gibi olsun:

0	1	1	0	1	1	0	0	1	0

Buradan *Gbest* kaynağının etkinleştirdiği M_{gbest} seti [29,60,120,145,223] olur. M_{gbest} seti dışındaki M_{kalan} seti *Z* verisinden yeniden rasgele seçilir. M_{kalan} seti [15,76,177,210,251] olarak seçilmiş olsun. M_{kalan} seti, güncellenen *M* setinde ilgili (Gbest kaynağının 0 olduğu) pozisyonlara yerleştirilir:

15			76			177	210		251

M_{gbest} setine K-means algoritması uygulanır. K-means ile elde edilen güncellenmiş M_{gbest} seti de [32,61,122,144,228] şeklinde verilmiş olsun. M_{gbest} seti de güncellenen *M* setindeki boş pozisyonlara sırasıyla yerleştirilir. Böylece M_{gbest} ve M_{kalan} setinin birleşimi ile oluşan güncellenmiş *M* seti son halini almış olur:

15	32	61	76	122	144	177	210	228	251

5.2. Deneysel Dizayn

Deneysel çalışmalar, 7 adet 8 bitlik griton görüntü ve 7 adet veri seti olmak üzere toplam 14 veri üzerinde gerçekleştirilmiştir. Bölüm 4'te olduğu gibi seçilen görüntüler sırasıyla Tahoe (300×300 piksel, 4 sınıf), Morro Bay (256×256 piksel, 4 sınıf), MRI (300×300 piksel, 5 sınıf), Airplane (512×512 piksel, 6 sınıf), Lena (512×512 piksel, 6 sınıf), Mandril (512×512 piksel, 6 sınıf) ve Pepper (512×512 piksel, 7 sınıf) griton resimleridir. Görüntülerdeki optimal küme sayısının belirlenmesinde Bölüm 4'te olduğu gibi Denklem 4.7 temel alınmıştır. Çalışmalarda kullanılan veri setleri Bölüm 4'teki gibi sırasıyla Wisconsin (683 birey, 9 öznitelik, 2 sınıf), Iris (150 birey, 4 öznitelik, 3 sınıf), Wine (178 birey, 13 öznitelik, 3 sınıf), User Knowledge Modelling (493 birey, 18 öznitelik, 4 sınıf), E-coli (336 birey, 8 öznitelik, 5 sınıf) ve Dermatology (358 birey, 34 öznitelik, 6 sınıf) verileridir. Veri setlerine kümeleme işleminden önce normalizasyon işlemi uygulanmıştır.

Önerilen yöntemlerin performansları, kümeleme kalitesi ve elde edilen optimal küme sayısı üzerinden değerlendirilmiştir. Karşılaştırmalı performans analizi için DisABC, AMABC, BPSO, NBPSO, QBPSO ve GA ikili evrimsel hesaplama modelleri kullanılmıştır. Modellerin deneysel çalışmalarda kullanılmak üzere seçilmesinin gerekçeleri ve bu modeller ile ilgili detaylı bilgi Bölüm 2'de sunulmuştur.

Modellerin optimal parametre değerlerinin saptanması için literatürdeki benzer çalışmalar araştırılmıştır. Kuo ve arkadaşları [161] veri kümelemede GA ve BPSO için popülasyon büyüklüğünü 20, maksimum fonksiyon değerlendirme sayısını 10000 ve *TC* durdurma kriterini 5 olarak olarak belirlemiştir. Ancak popülasyon büyüklüğü için 20 değeri, GA için küçük bir değerdir ve *TC* için 5 değeri de zaman karmaşıklığını artmasına sebep olmaktadır. Bununla birlikte yapılan bazı denemeler neticesinde veri kümeleme için tüm ikili modellerde popülasyon büyüklüğü 30, *TC* değeri 3 ve maksimum çevrim sayısı 9000 olarak belirlenmiştir.

Görüntü kümeleme için Omran ve arkadaşları [29] BPSO ve GA'da popülasyon büyüklüğünde 100 ve 20 değerlerini kullanmış, *TC* değerini 2 olarak almış ve çevrim sayısını 50 ile sınırlandırmışlardır. Elde edilen sonuçlar neticesinde popülasyon büyüklüğünün 100 veya 20 olmasının BPSO algoritmasının performansı üzerinde büyük farklılık yaratmadığı görülmüştür. Ouadfel ve arkadaşları [160] bir başka PSO modeli için 50 popülasyon büyüklüğünü kullanmıştır. Bununla birlikte yapılan bazı denemeler neticesinde görüntü kümeleme için bu çalışmada popülasyon büyüklüğü 50, *TC* değeri 2 ve maksimum çevrim sayısı 10000 olarak belirlenmiştir.

İkili modellerin diğer parametreleri şu şekilde belirlenmiştir: BPSO için [264]'te olduğu gibi Vmax=6, c_1=2, c_2=2, w_{init}=0.9 ve w_{end}=0.4 alınmıştır. QBPSO için [226]'da olduğu gibi Q_{max}=0.05π ve Q_{min}=0.01π olarak alınmıştır. NBPSO'da [225]'te olduğu gibi w_1=0.5 ve Vmax=6 olarak belirlenmiştir. DisABC ve IDisABC'de [235]'te olduğu gibi Q_{max}=0.9 ve Q_{min}=0.5 olarak belirlenmiştir. Limit parametresi için DisABC ve AMABC ikili modellerinde 100 değeri, GBABC ve IDisABC modellerinde ise 20 değeri belirlenmiştir. GBABC ve IDisABC modellerinin bir kaynak için birden fazla fonksiyon değerlendirmesi yapması sebebiyle limit değerinin daha düşük tutulması uygun görülmüştür. GA'da çaprazlama ve mutasyon oranları sırasıyla 0.8 ve 0.2 olarak alınmıştır.

N_c maksimum küme sayısı ilgilenilen küme sayılarına göre belirlenmiştir. Veri kümelemede N_c değeri, Wisconsin ve Diabet veri setlerinde 13, Iris, Wine ve Knowledge veri setlerinde 15 ve Dermatology ve E-coli veri setlerinde 20 olarak belirlenmiştir [161]. Görüntü kümelemede N_c değeri, Tahoe ve Morro Bay görüntülerinde 15 ve diğer görüntülerde 20 olarak belirlenmiştir. VI indeks parametre değerleri Bölüm 4'te belirtildiği gibi alınmıştır.

Görüntü ve veri kümeleme için değerlendirme temel olarak VI indeks ve küme sayısı üzerinden gerçekleştirilmiştir. Bunlara ek olarak, veri kümelemede elde edilen kümeleme kalitesini daha net olarak ortaya koymak amacıyla dışsal doğrulama indekslerinden biri olan Rand indeks [36] de kullanılmıştır. Rand indeksi Denklem 4.8 ile tanımlanmıştır ve [0,1] aralığında değer almaktadır.

5.3. Deneysel Çalışmalar

Deneysel çalışmalar görüntü ve veri kümeleme olarak iki kategoride ele alınmıştır. Elde edilen sonuçlar birbirinden bağımsız 30 koşma için ortalama ve standart sapma değerleri üzerinden Tablolar 5.1, 5.3, 5.4 ve 5.6'da sunulmuştur. Tablolardaki en iyi değerler kalın fontta ve standart sapma değerleri parantez içinde gösterilmiştir. İstastiksel test sonuçları Tablolar 5.2 ve 5.5'te sunulmuştur. Tablolar 5.2 ve 5.5'te yer alan '+' ('-') sembolü, sütunda bulunan GBABC ('GB') veya IDisABC'nin ('IDis') satırda bulunan ilgili yöntemden Wilcoxon Rank Sum Test istatistiğine göre daha iyi (daha kötü) olduğunu göstermektedir. '=' sembolü, sütunda bulunan GBABC veya IDisABC ile satırda bulunan ilgili yöntem arasında Wilcoxon Rank Sum Test istatistiğine göre farklılık olmadığını ifade etmektedir.

5.3.1. Görüntü Kümeleme

5.3.1.1. Görüntüler için VI İndeks Sonuçları

Görüntü kümeleme için elde edilen VI indeks değerleri Tablo 5.1'de ve istatistiksel test sonuçları Tablo 5.2'de sunulmuştur. Tablo 5.1'e göre önerilen GBABC ve IDisABC otomatik kümeleme yöntemleri diğer yöntemlerden genellikle daha iyi ortalama ve standart sapma değerleri elde etmiştir. IDisABC yöntemi sadece MRI görüntüsünde iyi sonuç elde edememiştir. Diğer yöntemlere bakıldığında ise DisABC önerilen

yöntemlerin performans olarak hemen ardında yer almaktadır. GA otomatik kümeleme yöntemi diğer yöntemlerden daha kötü sonuçlar üretmiştir. Geriye kalan AMABC, BPSO, NBPSO ve QBPSO yöntemlerinin tam anlamıyla birinin diğerine üstün olduğunu söylemek mümkün değildir. Genel olarak BPSO'nun birkaç durumu hariç bu yöntemlerin sonuçları birbirine oldukça yakındır.

Tablo 5.1. Görüntü kümeleme için elde edilen VI indeks sonuçları.

Görüntüler	GBABC	IDisABC	DisABC	AMABC	BPSO	NBPSO	QBPSO	GA
Tahoe	0.0569 (0.006)	**0.0566** (0.005)	0.0574 (0.006)	0.0586 (0.007)	0.0644 (0.017)	0.0608 (0.008)	0.0644 (0.017)	0.0884 (0.037)
Morro Bay	0.0831 (0.010)	**0.0789** (0.009)	0.0875 (0.012)	0.0935 (0.026)	0.1053 (0.033)	0.0837 (0.013)	0.0828 (0.015)	0.1268 (0.039)
MRI	**0.0485** (0.006)	0.1090 (0.012)	0.0499 (0.006)	0.0595 (0.007)	0.1339 (0.118)	0.0561 (0.012)	0.0521 (0.006)	0.1778 (0.212)
Airplane	0.0945 (0.014)	**0.0922** (0.020)	0.0959 (0.030)	0.1043 (0.016)	0.1366 (0.118)	0.1097 (0.026)	0.1238 (0.100)	0.1517 (0.059)
Lena	**0.0982** (0.008)	0.0982 (0.011)	0.1032 (0.014)	0.1109 (0.014)	0.1126 (0.013)	0.1116 (0.018)	0.1126 (0.013)	0.1395 (0.025)
Mandril	0.1082 (0.012)	**0.1043** (0.010)	0.1045 (0.011)	0.1121 (0.015)	0.1077 (0.011)	0.1183 (0.022)	0.1023 (0.011)	0.1419 (0.032)
Pepper	**0.1069** (0.015)	0.1083 (0.014)	0.1201 (0.015)	0.1325 (0.029)	0.1323 (0.046)	0.1201 (0.022)	0.1202 (0.031)	0.1662 (0.059)

Tablo 5.2. VI indeks değerlerinin Wilcoxon Rank Sum Test istatistiğine göre değerlendirme sonuçları.

	Tahoe		MorroBay		MRI		Airplane		Lena		Mandril		Pepper	
	GB	IDis	GB	IDis	GB	IDis	GB	IDis	GB	IDis	GB	IDis	GB	IDis
DisABC	=	=	=	=	=	-	=	=	+	+	=	=	+	+
AMABC	+	+	=	+	+	-	+	+	+	+	=	+	+	+
BPSO	=	+	+	+	+	=	+	+	+	+	=	=	+	+
NBPSO	=	=	=	=	+	-	+	+	+	+	=	=	+	+
QBPSO	+	+	=	=	+	-	=	=	+	+	=	=	=	=
GA	+	+	+	+	+	+	+	+	+	+	+	+	+	+

Önerilen yöntemler ile diğer yöntemler arasındaki farklılık Wilcoxon Rank Sum Testine göre değerlendirildiğinde DisABC'nin Lena ve Pepper görüntü setleri dışında genel olarak GBABC ve IDisABC ile benzer VI indeks sonuçları ürettiği Tablo 5.2'de görülmektedir. Önerilen yöntemlerden IDisABC ile diğer yöntemler arasında Tahoe, Airplane, Lena ve Pepper görüntülerinde VI indeks sonuçlarında genel olarak anlamlı

bir farklılık söz konusudur. Ancak MRI görüntü setinde IDisABC genel olarak diğer yöntemlerden daha kötü sonuçlar üretmiştir. Önerilen diğer yöntem GBABC ise, MRI, Airplane, Lena ve Pepper görüntü setlerinde diğer yöntemlere göre genel olarak daha kaliteli sonuçlar elde etmiştir. Sonuç olarak önerilen otomatik kümeleme yöntemleri diğer yöntemlere oranla daha iyi VI indeks değerleri üretmiştir.

Önerilen yöntemler kendi aralarında karşılaştırıldığında, IDisABC otomatik kümeleme yöntemi Tahoe, Morro Bay, Airplane ve Mandril görüntülerinde GBABC'den daha iyi ortalama ve standart sapma değerleri elde ettiği Tablo 5.1'de görülmektedir. GBABC yöntemi diğer 3 görüntüde IDisABC'den daha iyi VI indeks değerleri üretmiştir. Ancak, sonuçlar genel olarak birbirine çok yakındır ve Wilcoxon Rank Sum istatistiğine göre de değerlendirildiğinde genel olarak aralarında anlamlı bir farklılık söz konusu değildir. Sadece MRI görüntü setinde IDisABC yöntemi GBABC'ye göre oldukça kötü değerler üretmiştir. Bu durum GBABC'yi IDisABC'ye oranla VI indeks minimizasyonunda bir adım öne çıkarmaktadır.

5.3.1.2. Görüntüler için Küme Sayısı Sonuçları

Görüntü kümeleme için elde edilen ortalama küme sayısı değerleri Tablo 5.3'te sunulmuştur. Tablo 5.3'e göre önerilen GBABC ve IDisABC otomatik kümeleme yöntemleri ilk görüntü setinde optimal küme sayısına ortalama olarak en çok yaklaşan yöntemler olmuştur. Morro Bay görüntüsünde de önerilen GBABC ve IDisABC yöntemleri, DisABC ve BPSO yöntemleri ile beraber optimal küme sayısına çok yaklaşmıştır. MRI görüntüsünde IDisABC başarısız bir performans ortaya koyarken, GBABC optimal küme sayısına en çok yaklaşan yöntem olmuştur. Lena ve Mandril görüntüsünde IDisABC ve GBABC yöntemleri ortalama değer olarak optimal küme sayısına yine en çok yaklaşan yöntemlerdir. Lena görüntüsünde AMABC de optimal küme sayısına çok yaklaşmıştır. Airplane ve Pepper görüntü setlerinde sırasıyla BPSO ve AMABC yöntemleri ortalama olarak optimal küme sayısına diğerlerine nazaran daha çok yaklaşmıştır. Ancak, bu yöntemlerin elde ettikleri standart sapma değerlerinin önerilen yöntemlerin elde ettiği standart sapma değerlerinden daha yüksek olduğu da dikkate alınmalıdır. Diğer yöntemlere bakıldığında ise NBPSO, QBPSO ve GA otomatik kümeleme yöntemlerinin oldukça kötü ortalama küme değerleri elde ettikleri görülmektedir. BPSO yöntemi de MRI, Lena ve Mandril görüntü setlerinde iyi sonuçlar

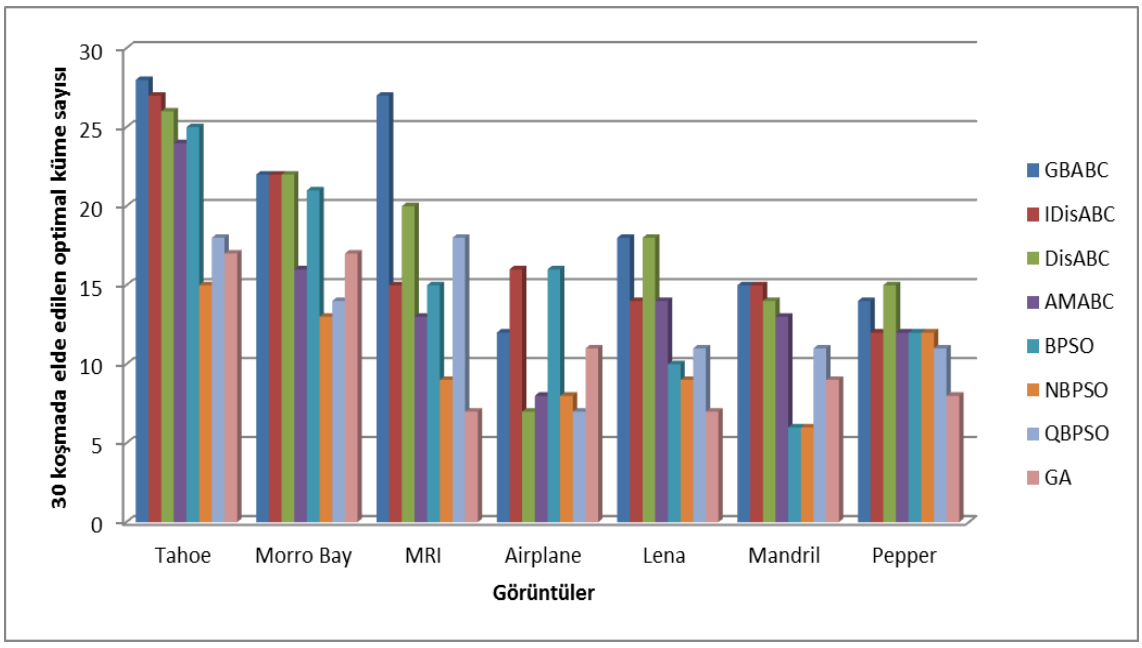
üretememiştir. Sonuç olarak optimal küme sayısına en yakın ortalama değer elde eden yöntemlerin ikili ABC modelleri olduğu görülmüştür.

Tablo 5.3. Görüntü kümeleme için elde edilen küme sayısı sonuçları.

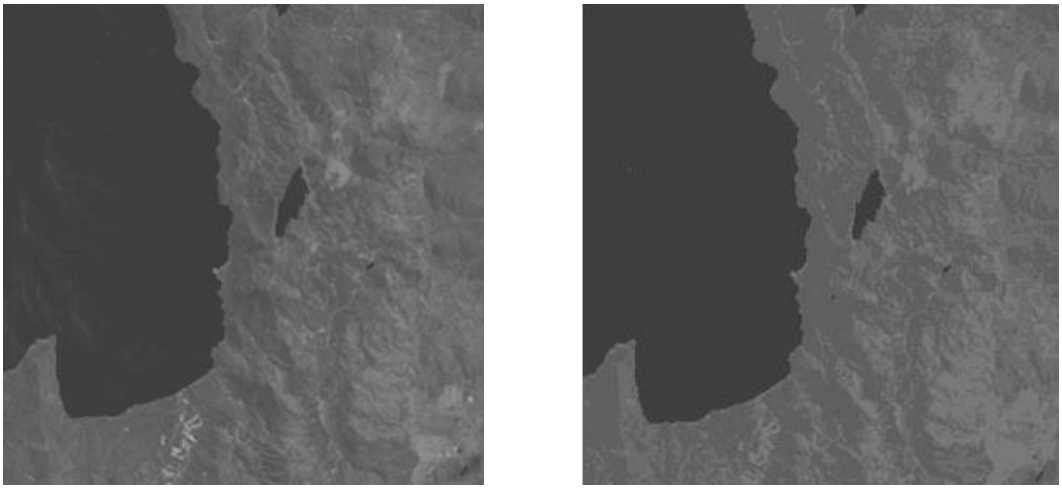
Görüntü	No	GBABC	IDisABC	DisABC	AMABC	BPSO	NBPSO	QBPSO	GA
Tahoe	4	**4.066** (0.253)	4,1 (0.305)	4.133 (0.345)	4,2 (0.406)	4,20 (0.482)	5.071 (1.585)	4.566 (0.954)	4.791 (1.178)
Morro Bay	4	**4.333** (0.479)	**4.333** (0.479)	**4.333** (0.479)	4,5 (0.572)	**4.333** (0.546)	4.733 (0.827)	4.720 (0.842)	4.621 (1.146)
MRI	5	**5.133** (0.434)	5,9 (1.155)	5,40 (0.621)	5.866 (0.973)	6.166 (1.341)	6.466 (1.413)	5.724 (1.130)	7,29 (2.123)
Airplane	6	5,7 (0.836)	5.733 (0.868)	5,5 (0.861)	5,5 (0.900)	**6.033** (0.934)	6.733 (2.033)	6.366 (1.299)	6.733 (1.680)
Lena	6	5,70 (0.595)	**5,9** (0.922)	5,666 (0.547)	**5,9** (0.922)	6.695 (1.063)	7.166 (2.134)	6,7 (1.417)	7,1 (1.516)
Mandril	6	5.933 (0.739)	**5.966** (0.905)	6.166 (1.019)	5.733 (1.014)	6,9 (1.471)	7.166 (2.214)	7.066 (1.172)	7,4 (1.652)
Pepper	7	6.633 (0.668)	6.733 (0.691)	6.566 (0.568)	**6.866** (0.766)	7.333 (1.212)	7.266 (1.387)	7.133 (1.212)	7.466 (1.105)

Yöntemlerin görüntü kümeleme için 30 koşma üzerinden toplamda kaç defa optimal küme sayısını elde ettiğini gösteren sütun grafiği Şekil 5.9'da sunulmuştur. Tahoe görüntüsünde GBABC ve IDisABC otomatik kümeleme yöntemleri 25'in üzerinde koşmada optimal küme sayısını bularak %93 ve %90'lık oranlar yakalamıştır. Onları %80 ve %86 aralığında optimal küme sayısını bulma oranlarıyla DisABC, AMABC ve BPSO yöntemleri takip etmektedir. NBPSO, QBPSO ve GA yöntemlerinde optimal küme sayısı elde etme oranları %50-%60 aralığına düşmektedir. Morro Bay görüntüsünde, GBABC, IDisABC ve DisABC yöntemleri yaklaşık %74'lük bir optimal küme sayısı bulma oranıyla ilk sırayı almıştır. Onları %70'lik oranla BPSO takip etmektedir. Diğer yöntemlerin Morro Bay görüntüsünde optimal küme sayısı elde etme oranları %50 civarı ve altına düşmüştür. MRI görüntüsünde, %90'lık optimal küme sayısı elde etme oranıyla GBABC diğer yöntemlerden çok iyi bir performans sergilemiştir. Ona en yakın performansı sergileyen DisABC ancak %66'lık bir oran yakalamıştır. QBPSO'da bu oran %60'a ve IDisABC'de %50'lere kadar düşmüştür. AMABC, NBPSO ve GA'nın MRI görüntüsünde optimal küme sayısı elde etme oranları %40'lardan başlayıp %20'lere kadar düşmüştür. Dolayısıyla MRI görüntüsü

için sadece GBABC otomatik kümeleme yöntemi iyi performans göstermiştir. Airplane görüntüsünde genel olarak optimal küme sayısı elde etme oranları düşük seviyede kalmıştır. IDisABC ve BPSO yöntemleri %53 civarı bir oran elde ederken, GBABC onları yaklaşık %43'lük bir oranla takip etmiştir. Diğer yöntemlerde bu oran %23'lere kadar düşebilmektedir. Lena görüntüsünde GBABC ve DisABC %60'lık oranla optimal küme sayını bulmuştur. Onları yaklaşık %50'lik bir oranla IDisABC takip etmiştir. Ancak diğer yöntemlerde bu oran %33'lerden %23'lere kadar gerilemektedir. Mandril görüntüsünde, %50'lik optimal küme sayısı bulma oranıyla GBABC ve IDisABC yöntemleri ilk sırayı almıştır. Onları yaklaşık %46 ve %43'lük oranlarla DisABC ve AMABC takip etmektedir. Geriye kalan yöntemlerin optimal küme sayısı bulma oranı Mandril görüntüsünde %20'lere kadar düşmektedir. Pepper görüntüsünde genel olarak tüm yöntemlerin birbirine benzer oranlar elde ettiği görülmektedir. Yaklaşık %50 ve %40 arasında bir oranla GA dışındaki tüm yöntemler optimal küme sayısını elde etmiştir. Sonuç olarak önerilen GBABC ve IDisABC yöntemleri, görüntü kümelemede hem kümeleme kalitesi sağlamada hem de optimal küme sayısı elde etmede iyi performans göstermiştir. Ancak iki yöntem arasında GBABC otomatik kümeleme yöntemi IDisABC'ye göre optimal küme sayısı elde etmede daha iyi konumdadır. GBABC ile kümelemeye ait görsel sonuçlar Şekiller 5.10-5.16 arasında sunulmuştur.

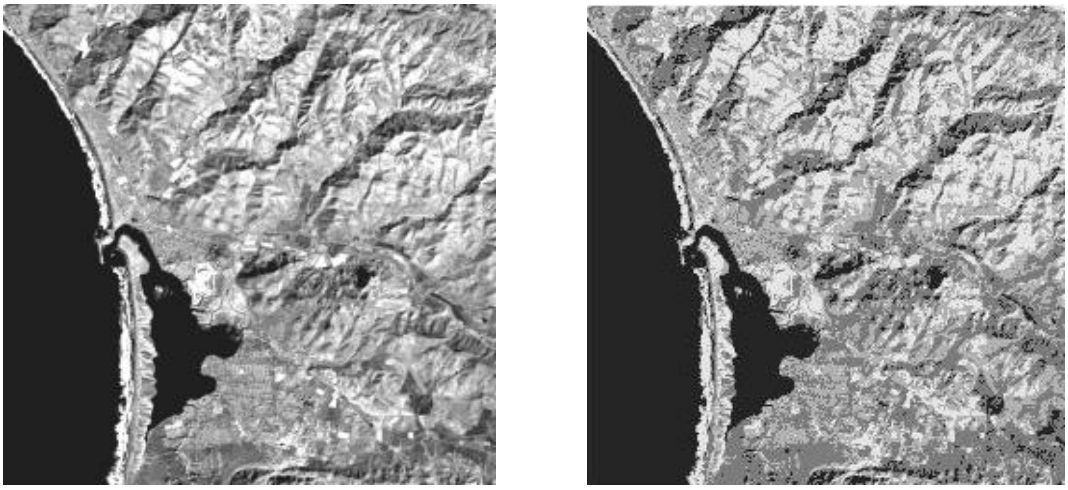


Şekil 5.9. İkili modellerde görüntü kümeleme için 30 koşma üzerinden toplamda kaç defa optimal küme sayısının elde edildiğini gösteren grafik.



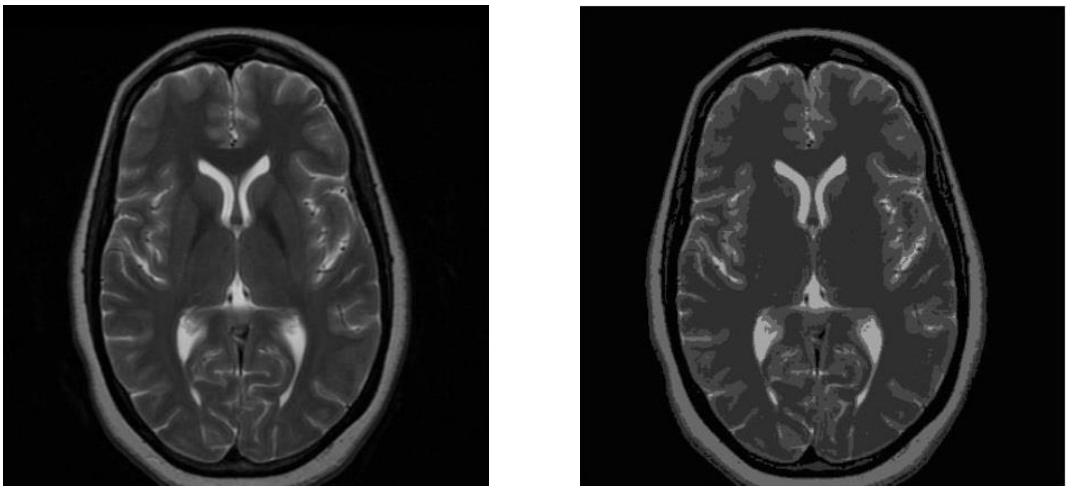
a) b)

Şekil 5.10. a) Orijinal ve b) GBABC ile kümelenmiş Tahoe görüntüsü.



a) b)

Şekil 5.11. a) Orijinal ve b) GBABC ile kümelenmiş Morro Bay görüntüsü.



a) b)

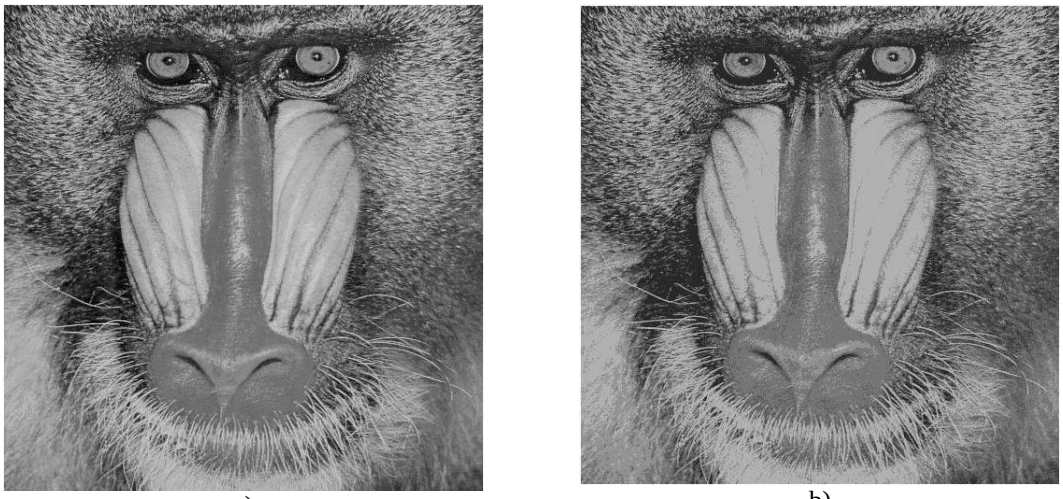
Şekil 5.12. a) Orijinal ve b) GBABC ile kümelenmiş MRI görüntüsü.

a) b)

Şekil 5.13. a) Orijinal ve b) GBABC ile kümelenmiş Airplane görüntüsü.

a) b)

Şekil 5.14. a) Orijinal ve b) GBABC ile kümelenmiş Lena görüntüsü.



a) b)

Şekil 5.15. a) Orijinal ve b) GBABC ile kümelenmiş Mandril görüntüsü.

a) b)

Şekil 5.16. a) Orijinal ve b) GBABC ile kümelenmiş Pepper görüntüsü.

5.3.2. Veri Kümeleme

5.3.2.1. Veri Setleri için VI İndeks Sonuçları

Veri kümeleme için elde edilen VI indeks değerleri Tablo 5.4'te ve istatistiksel test sonuçları Tablo 5.5'te sunulmuştur. Tablo 5.4'e göre önerilen GBABC ve IDisABC otomatik kümeleme yöntemleri, Diabet ve Knowledge veri setlerindeki üç durum dışında diğer yöntemlerden daha iyi VI ortalama değerleri elde etmiştir. Diabet veri setinde DisABC yöntemi önerilen yöntemlerden daha iyi performans göstermiştir. Knowledge veri setinde ise, DisABC yöntemi IDisABC'den daha iyi performans göstermiştir. Performanslarına bakıldığında genel olarak GBABC ve IDisABC yöntemlerinden sonra DisABC yöntemi gelmektedir. Diğer yöntemler değerlendirildiğinde, GA yönteminin diğer yöntemlere nispeten daha kötü değerler ürettiği görülmektedir. AMABC yöntemi de Ecoli ve Dermatology veri setlerinde diğer yöntemlere göre VI indeksini minimize etmede yetersiz kalmıştır.

Önerilen yöntemler ile diğer yöntemler arasındaki fark Wilcoxon Rank Sum Testine göre değerlendirildiğinde, Wisconsin ve Diabet gibi iki kümeli veri setlerinde yöntemlerin birbirine benzer sonuçlar ürettikleri Tablo 5.5'te görülmektedir. Buna ek olarak E-coli ve Dermatology veri setleri dışında önerilen yöntemler ile DisABC arasında anlamlı bir farklılık söz konusu değildir. Bunun dışındaki yöntemler ile

önerilen GBABC ve IDisABC yöntemleri arasında genel olarak anlamlı bir farklılıktan söz etmek mümkündür ve bu farklılık önerilen yöntemler lehinedir.

Önerilen yöntemler kendi aralarında karşılaştırıldığında Knowledge ve Dermatology dışında birbirlerine benzer sonuçlar üretmiştir. Wilcoxon Rank Sum Test istatistiğine göre değerlendirildiğinde aralarında net bir farkın olduğunu söylemek mümkün değildir. Ancak, elde edilen ortalama değerlerine bakılacak olursa, GBABC yöntemi IDisABC'ye göre bir adım öne çıkmıştır. Sonuç olarak VI indeks minimizasyonunda önerilen yöntemler diğer yöntemlere oranla daha iyi performans ortaya koymuştur.

Tablo 5.4. Veri kümeleme için elde edilen VI indeks sonuçları.

Veri Setleri	GBABC	IDisABC	DisABC	AMABC	BPSO	NBPSO	QBPSO	GA
Wisconsin	**0.1348** (0.022)	**0.1348** (0.022)	**0.1348** (0.022)	0.1396 (0.020)	0.1368 (0.018)	0.1350 (0.019)	0.1373 (0.022)	0.1424 (0.039)
Diabet	0.2505 (0.090)	0.2559 (0.083)	**0.2195** (0.092)	0.2957 (0.104)	0.2741 (0.089)	0.2878 (0.105)	0.2757 (0.085)	0.2907 (0.092)
İris	**0.0965** (0.009)	0.0974 (0.010)	0.0982 (0.010)	0.1069 (0.011)	0.1042 (0.013)	0.1104 (0.017)	0.1026 (0.012)	0.1181 (0.020)
Wine	**0.3241** (0.035)	0.3251 (0.035)	0.3365 (0.029)	0.3622 (0.038)	0.3518 (0.055)	0.3666 (0.045)	0.3583 (0.053)	0.4426 (0.145)
Knowledge	**0.4358** (0.051)	0.4551 (0.046)	0.4416 (0.052)	0.5039 (0.054)	0.4557 (0.064)	0.5148 (0.070)	0.5065 (0.078)	0.5233 (0.056
Ecoli	0.3160 (0.063)	**0.3073** (0.057)	0.3841 (0.069)	0.4376 (0.088)	0.3691 (0.086)	0.3468 (0.084)	0.3434 (0.092)	0.5351 (0.176)
Dermatology	0.4146 (0.052)	**0.3968** (0.043)	0.4328 (0.048)	0.5034 (0.110)	0.4600 (0.045)	0.4892 (0.078)	0.4544 (0.108)	0.5717 (0.138)

Tablo 5.5. VI indeks değerlerinin Wilcoxon Rank Sum Test istatistiğine göre değerlendirme sonuçları.

	Wisconsin		**Diabet**		**Iris**		**Wine**		**Knowledge**		**E-coli**		**Dermatology**	
	GB	**IDis**	**GB**	**IDis**	**GB**	**IDis**	**GB**	**IDis**	**GB**	**IDis**	**GB**	**IDis**	**GB**	**IDis**
DisABC	=	=	=	=	=	=	=	=	=	=	+	+	=	+
AMABC	=	=	=	=	+	+	+	+	+	+	+	+	+	+
BPSO	=	=	=	=	+	+	+	=	+	=	+	+	+	+
NBPSO	=	=	+	+	+	+	+	+	+	+	=	=	+	+
QBPSO	=	=	=	=	+	=	+	+	+	+	=	=	=	+
GA	+	+	=	=	+	+	+	+	+	+	+	+	+	+

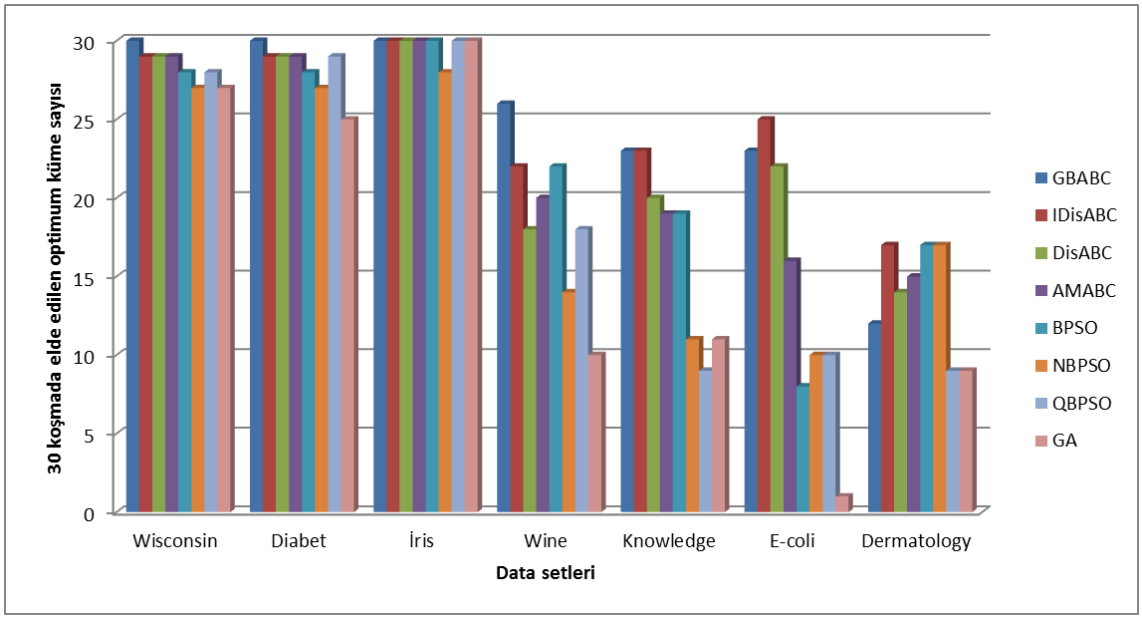
Tablo 5.6. Veri kümeleme için küme sayısı sonuçları.

Veri Setleri	No	GBABC	IDisABC	DisABC	AMABC	BPSO	NBPSO	QBPSO	GA
Wisconsin	2	**2** (0)	2.033 (0.182)	2.033 (0.182)	2.033 (0.182)	2.066 (0.253)	2.1 (0.305)	2.066 (0.253)	2.133 (0.681)
Diabet	2	**2** (0)	2.033 (0.182)	2.033 (0.182)	2.033 (0.182)	2.066 (0.253)	2.133 (0.434)	2.033 (0.182)	2.266 (0.691)
İris	3	**3** (0)	**3** (0)	**3** (0)	**3** (0)	**3** (0)	2.933 (0.253)	**3** (0)	**3** (0)
Wine	3	**3.133** (0.345)	3.3 (0.534)	3.4 (0.498)	3.366 (0.556)	3.9 (0.712)	3.7 (0.866)	3.533 (0.776)	4.3 (1.557)
Knowledge	4	**4.233** (0.430)	4.266 (0.520)	4.333 (0.660)	4.4 (0.855)	4.443 0.817	5 (1.231)	5.266 (1.412)	5.333 (1.470)
E-coli	5	5.230 (0.430)	**5.166** (0.379)	5.333 (0.606)	5.6 (0.813)	5.866 0.776	6.166 (1.205)	6.033 (0.999)	7.7 (1.600)
Dermatology	6	5.433 (0.568)	5.56 **(0.504)**	5.533 (0.571)	5.633 (0.614)	**6.033** (0.999)	6.166 (1.116)	5.733 (0.980)	6.966 (1.790)

5.3.2.2. Veri Setleri için Küme Sayıları

Veri kümeleme için elde edilen ortalama küme sayısı değerleri Tablo 5.6'da sunulmuştur. İlk iki veri seti Wisconsin ve Diabet'te genel olarak bütün yöntemler ortalama olarak optimum küme sayısına yaklaşmıştır. Önerilen GBABC yöntemi bu veri setlerinde optimum küme sayısına en çok yaklaşan yöntem olmuştur. İris veri setinde de bütün yöntemler optimum küme sayısına yaklaşarak başarılarını devam ettirmiştir. Wine veri setinde birçok yöntemin performansı düşmüştür. Ancak, önerilen GBABC ve IDisABC otomatik kümeleme yöntemleri elde ettikleri küme sayısı değerleri ile optimum küme sayısına en çok yaklaşan yöntemler olmuştur. Knowlege ve E-coli veri setlerinde de optimal küme sayısına en çok yaklaşan yöntemler GBABC ve IDisABC otomatik kümeleme yöntemleri olmuştur. Onları DisABC takip etmiştir. Kalan diğer yöntemlerin genel olarak performansı bu veri setlerinde oldukça düşük seviyede kalmıştır. Örneğin, QBPSO yöntemi Knowledge ve E-coli veri setlerinde sırasıyla 5.266 ve 6.033 gibi optimum küme sayısına oldukça uzak değerler elde etmiştir. Bu örneklerin sayısı epey fazladır. Ayrıca, önerilen yöntemlerin elde ettikleri standart sapma değerlerinin de diğer yöntemlere göre genellikle küçük olduğu görülmektedir. Buradan önerilen yöntemlerin diğerlerine nazaran daha tutarlı veya optimum küme sayısına daha uygun değerler buldukları çıkarılabilir. Her ne kadar

Dermatology veri setinde BPSO yöntemi optimum küme sayısına daha yakın ortalama değer elde etmiş olsa da standart sapmasının yüksek oluşu, elde edilen küme sayıları arasında genel bir tutarlılığın olmadığını göstermektedir.



Şekil 5.17. İkili modellerde veri kümeleme için 30 koşma üzerinden toplamda kaç defa optimum küme sayısının elde edildiğini gösteren grafik.

Yöntemlerin veri kümeleme için 30 koşma üzerinden toplamda kaç defa optimum küme sayısını elde ettiğini gösteren sütun grafiği Şekil 5.17'de sunulmuştur. İki kümeli Wisconsin ve Diabet veri setlerinde, bütün otomatik kümeleme yöntemleri %90 ve üzeri bir oranla optimum küme sayısını bulmuştur. GBABC yöntemi bu veri setlerinde optimum küme sayısını her koşmada bularak %100'lük bir oranla diğer yöntemlerden bir adım öne çıkmıştır. Iris veri setinde yine bütün yöntemler optimum küme sayısını elde etmede başarılı olmuştur. NBPSO yöntemi %93'lük bir oranla optimum küme sayısını elde ederken, diğer bütün yöntemler optimum küme sayısı elde etmede %100'lük bir başarı sağlamıştır. Diğer dört veri setinde, ilk üç veri setindeki yüksek performans söz konusu değildir. Wine veri setinde, önerilen GBABC yöntemi %87'lik bir oranla optimum küme sayısını elde ederek diğer yöntemlerin üstünde bir performans göstermiştir. Diğer önerilen IDisABC ile birlikte BPSO yöntemi %74'lük bir oranla optimum küme sayısını bulmuştur. Onları QBPSO ve AMABC %66'lık oranla takip etmektedir. Diğer kalan NBPSO ve GA yöntemlerinde bu oran %50'lerin altına düşmüştür. 4 kümeli Knowledge veri setinde, önerilen GBABC ve IDisABC yöntemleri yaklaşık %77'lik bir optimum küme sayısı bulma oranıyla en iyi performansı

göstermiştir. Onları yaklaşık %66'lik oranla DisABC ve %63'lük oranla AMABC ve BPSO yöntemleri takip etmektedir. Diğer yöntemlerde optimum küme sayısı bulma oranı %33'lere kadar düşmektedir. 5 kümeli E-coli veri setinde, önerilen IDisABC ve GBABC otomatik kümeleme yöntemleri sırasıyla %83 ve %77'lik optimum küme sayısı bulma oranlarıyla ilk iki sırayı almıştır. Onların ardından %73'lük optimum küme sayısı bulma oranıyla DisABC gelmektedir. Diğer yöntemlerde bu oran %53'lerden %33'lere kadar düşmektedir ve GA yönteminde bu oran sadece %3.3 gibi düşük bir seviyededir. 6 kümeli Dermatology veri setinde optimum küme sayısı bulma oranları genel olarak düşük seviyededir. IDisABC, BPSO ve NBPSO yöntemleri %53'lük bir optimum küme sayısı bulma oranıyla ilk sırayı almıştır. Diğer yöntemlerde bu oran %50'lerin altına düşmektedir. Dolayısıyla önerilen GBABC yöntemi sadece Dermatology veri setinde performans olarak düşük seviyede kalmıştır. Sonuç olarak önerilen GBABC ve IDisABC otomatik kümeleme yöntemleri diğer yöntemlerden hem VI indeksini optimize etme hem de optimum küme sayısını bulma kabiliyetleri açısından diğerlerinden genel olarak daha iyidir. Önerilen yöntemler kendi aralarında değerlendirildiğinde, GBABC yöntemi IDisABC yönteminden özellikle optimum küme sayısını bulmada bir adım daha öndedir.

5.3.2.3. Veri Setleri için Rand İndeks Sonuçları

Elde edilen kümeleme kalitesini daha sağlıklı ölçmek amacıyla Rand indeks değerleri Tablo 5.7'de sunulmuştur. Tablo 5.7'ye göre Wisconsin veri setinde GA dışındaki tüm yöntemlerin elde ettikleri Rand indeks değerleri birbirine oldukça yakındır. Diabet veri setinde, önerilen GBABC yöntemi diğerlerinden daha iyi bir kümeleme kalitesi elde etmiştir. Diğer yöntemlerin Rand indeks performansları GA dışında birbirinin aynı veya birbirine çok yakındır. Iris veri setinde tüm yöntemlerin performansı hemen hemen birbirinin aynısıdır ve %96 gibi yüksek bir performans ortaya koymuştur. Geriye kalan Wine, Knowledge ve E-coli veri setlerinde, önerilen GBABC ve IDisABC yöntemleri diğer yöntemlerden daha iyi sonuçlar elde ederek farklarını ortaya koymuştur. Geriye kalan DisABC dışındaki tüm yöntemlerin performansları, bu veri setlerindeki önerilen yöntemlere kıyasla genel olarak oldukça düşüş göstermiştir. Dermatology veri setinde, ilk iki sırayı IDisABC ve BPSO yöntemleri almıştır. Onları GBABC yöntemi takip etmiştir. Diğer yöntemlerin elde ettikleri kümeleme indeks sonuçları bu yöntemlere kıyasla oldukça düşük seviyede seyretmiştir. Sonuç olarak önerilen GBABC ve

IDisABC otomatik kümeleme yöntemleri dışsal indeks doğrulama sonuçlarında da diğerlerinden daha iyi performans sağlamıştır.

Tablo 5.7. Veri Kümeleme Rand indeks Sonuçları.

Veri Setleri	GBABC	IDisABC	DisABC	AMABC	BPSO	NBPSO	QBPSO	GA
Wisconsin	**0.9548** (0.0027)	0.9541 (0.0036)	0.9541 (0.0036)	0.9532 (0.0030)	0.9531 (0.0039)	0.9491 (0.0118)	0.9435 (0.0491)	0.9296 (0.0731)
Diabet	**0.6119** (2.8E-16)	0.6093 (0.0142)	0.6093 (0.0142)	0.6093 (0.0142)	0.6062 (0.0217)	0.5990 (0.0443)	0.6093 (0.0142)	0.5640 (0.1176)
İris	**0.9600** (0)	**0.9600** (0)	**0.9600** (0)	**0.9600** (0)	**0.9600** (0)	0.9404 (0.0744)	**0.9600** (0)	0.9404 (0.0744)
Wine	**0.9239** (0.0077)	0.9192 (0.0118)	0.9183 (0.0119)	0.9177 (0.0159)	0.8918 (0.0404)	0.8215 (0.1240)	0.8593 (0.1143)	0.7164 (0.2064)
Knowledge	**0.5133** (0.0543)	0.5043 (0.0480)	0.5024 (0.0597)	0.4913 (0.0923)	0.4943 (0.0777)	0.4441 (0.0751)	0.4503 (0.0964)	0.4559 (0.0925)
E-coli	**0.7353** (0.1005)	0.7246 (0.1135)	0.7105 (0.1244)	0.6739 (0.1518)	0.6376 (0.1363)	0.5786 (0.1719)	0.5780 (0.1515)	0.4992 (0.1517)
Dermatology	0.7454 (0.1162)	**0.7622** (0.0965)	0.7218 (0.1286)	0.7138 (0.1291)	0.7578 (0.1108)	0.6124 (0.1148)	0.6968 (0.1428)	0.6828 (0.1443)

5.4. Sonuçlar

Bu bölümün temel amacı doğru küme sayısının saptanmasına yönelik ikili ABC yöntemlerinin geliştirilmesiydi. Bu amacı gerçekleştirmek için iki tane ayrık ABC modeli önerilmiştir. İlk model olan GBABC, genetik operatörlerden oluşan komşuluk arama mekanizmasının tanımlanması sonucu ortaya çıkmıştır. İkinci model olarak IDisABC, DisABC modelindeki benzerlik kombinasyonlarının tümünü içeren bir arama mekanizmasının tasarlanması üzerine kurulmuştur. Geliştirilen modellerin performans analizi, bilinen ikili ABC, PSO ve GA modelleri ile karşılaştırılarak bilinen görüntü ve veri setleri üzerinde yapılmıştır. Elde edilen sonuçlar neticesinde geliştirilen ikili ABC otomatik kümeleme yöntemlerinin diğer yöntemlere karşı hem kümeleme kalitesi hem de optimal veya optimum küme sayısı elde etme açısından daha iyi performans sağladığı anlaşılmıştır. Bunun yanında, ikili ABC modelleri ile geliştirilen otomatik kümeleme yöntemlerinin sürekli ABC modellerinden veri kümeleme için daha iyi Rand indeks değerleri elde ettiği açıkça görülmektedir. Bu çalışma ile geliştirilen ikili ABC otomatik kümeleme yöntemleri literatürdeki ilk çalışmalar olma niteliği taşımaktadır. Bununla beraber görüntü setleri için tasarlanmış olan VI indeksi veri setlerine ilk olarak bu çalışma ile uygulanmıştır.

6. BÖLÜM

TARTIŞMA, SONUÇ VE ÖNERİLER

6.1. Tartışma ve Sonuçlar

Bu tez çalışması, görüntü kümeleme problemlerinin çözümü için önemli bir evrimsel hesaplama tekniği olan yapay arı koloni algoritmasına (ABC) dayalı yeni yöntemlerin geliştirilmesini ve uygulamalarını ele almıştır.

Bölüm 2'de ele alınan problemlerin genel bir tanıtımı, değerlendirmesi ve analizi yapılmıştır. Bu değerlendirme ve analiz kapsamında verideki küme sayısını saptayan yöntemlerin ortak özellikleri temel alınarak genel bir taksonomi ortaya koyulmuştur.

Bölüm 3'te küme sayısını dışardan bir kontrol parametresi olarak alan ABC tabanlı görüntü kümeleme yöntem ve uygulamaları sunulmuştur. İlk olarak ABC ile kümeleme tabanlı beyin tümörü segmentasyon metodolojisi gerçekleştirilmiştir. Bu metodolojide ABC kümeleme yönteminin temel görevi, kümeler arası uzaklığın maksimize ve aynı zamanda kuantalama hatası ve küme içi uzaklığın minimize edilerek beyin MRI görüntülerinin bölgelere ayrılmasıdır. Önerilen ABC tümör segmentasyon metodolojisi bilinen klasik kümeleme metodolojileri ile karşılaştırılmıştır. Genel olarak ABC segmentasyon metodolojisi klasik segmentasyon metodolojilerinden küme içi uzaklık, kümeler arası uzaklık ve kümeleme doğrulama indeksleri olarak çok üstün bir performans göstermiştir. Sayısal değerlerdeki üstünlük görsel sonuçlara da yansımıştır. Klasik segmentasyon metodolojileri bazı MRI görüntülerinde tümörlü bölgeyi doğru belirleyemezken, ABC segmentasyon metodolojisi genel olarak tümörlü bölgeleri doğru olarak belirlemiştir. Ancak bu metodolojideki ABC kümeleme yönteminin amaç fonksiyonu olarak kullandığı kümeleme kriteri kullanıcı tarafından belirlenmesi gereken ağırlıklandırma parametreleri içermekte ve Je hata kriteri minimizasyonunda problemler yaşayabilmektedir. Bu kümeleme kriterinde görülen eksiklikleri gidermek amacıyla bu

bölümde ikinci olarak yeni bir kümeleme kriteri geliştirilmiştir. Bu kümeleme kriteri, ortalama hata kareler toplamı ve Je hata kriterinin birlikte entegrasyonu sağlanarak oluşturulmuş ve hiçbir parametre değeri içermemektedir. Geliştirilen kümeleme kriterinin performans analizi, ABC, PSO ve GA algoritmaları kullanılarak ve bilinen üç kümeleme kriteri ile karşılaştırılarak gerçekleştirilmiştir. Elde edilen sonuçlar neticesinde, geliştirilen kümeleme kriterinin mevcut kümeleme kriterlerinden daha iyi kümeleme kalitesi elde ettiği ve birlikte kullanıldığı evrimsel hesaplama tekniklerinin arasında da en iyi performansı ABC ile sağladığı görülmüştür. Bu bölümde son olarak K-means algoritması ABC'ye entegre edilerek yeni bir kümeleme tabanlı renk kuantalama yöntemi geliştirilmiştir. Geliştirilen renk kuantalama yöntemi RGB ve L*a*b* uzaylarında test edilmiş ve elde edilen sonuçlar değerlendirildiğinde önerilen yöntemin GCMA, CQ-PSO, K-means, FCM ve minimum varyans yöntemlerinden daha iyi performans sağladığı görülmüştür. Geliştirilen yöntem ABC'nin renk kuantalama problemindeki ilk uygulaması olma niteliği taşımaktadır.

Bölüm 4'te, verilerin küme sayısı bilinmeden kümelenmesini gerçekleştirecek yeni bir ABC tabanlı otomatik kümeleme yöntemi geliştirilmiştir. Geliştirilen yöntem sadece görüntü setlerinde değil, aynı zamanda veri setlerinin kümelenmesinde de kullanılmıştır. Yöntemin geliştirilmesinde PSO'daki vektörel arama ve küresel en iyi kaynak mekanizmaları ABC algoritmasına entegre edilerek ABC'nin daha güçlü bir modeli elde edilmiştir. Geliştirilen yeni ABC modeli, bilinen ABC modellerinden ve PSO algoritmasından hem kümeleme kalitesi hem de optimal küme sayısını bulma açısından daha iyi sonuçlar üretmiştir. CPU hesaplama zamanı açısından değerlendirildiğinde ise, standart ABC ve PSO algoritmalarının hesaplama karmaşıklığının birbirine yakın olduğu bilinen bir gerçektir. Tez çalışmasında kullanılan sürekli ABC modellerinin yapısında köklü değişiklik yapılmadan, sadece farka dayalı çözüm üretme mekanizmasında küçük değişiklikler yapılarak oluşturulması sebebiyle bu modellerin de hesaplama zamanı açısından birbirine yakın maliyet değerleri elde ettiği görülmüştür.

Bölüm 5'te, ikili ABC ve K-means'e dayalı yeni hibrit otomatik kümeleme yöntemleri geliştirilmiştir. Geliştirilen yöntemlerden ilki genetik operatör ve evrimsel algoritmalardan esinlenerek oluşturulmuştur. Diğer yöntem ise bilinen bir ikili ABC modelinin geliştirilmiş bir versiyonudur. Önerilen ikili ABC otomatik kümeleme yöntemleri literatürdeki bilinen ABC, PSO ve GA tabanlı yöntemler ile karşılaştırılarak

geniş bir performans analizi ortaya konulmuştur. Elde edilen sonuçlar geliştirilen yöntemlerin diğer yöntemlerden hem görüntü hem de veri kümeleme problemlerinde daha iyi performans sergilediğini göstermiştir. Ayrıca, önerilen ikili otomatik kümeleme yöntemleri sürekli otomatik kümeleme yöntemlerine göre daha uygun Rand indeks değerleri elde etmiştir. CPU hesaplama zamanı açısından değerlendirildiğinde ise, yöntemlerin hemen hemen birbirine yakın maliyet değerleri elde ettiği görülmüştür. Ancak, GBABC'nin genetik operatörleri biraz daha sık kullanması sebebiyle hesaplama maliyetinin diğerlerinden daha fazla olması beklenen bir gerçektir.

Tez kapsamında yapılan çalışmalar genel olarak değerlendirildiğinde, ABC algoritmasına dayalı olarak geliştirilen kümeleme yöntemlerinin görüntü kümeleme problemlerine başarılı bir şekilde uygulanabileceği görülmüştür. Elde edilen sonuçlar analiz edildiğinde, önerilen yöntemlerin görüntü kümeleme problemlerinin çözümünde literatürdeki mevcut evrimsel algoritmalara dayalı yöntemlere göre daha iyi veya eşdeğer performans sergilediği sonucuna varılmıştır.

Geliştirilen otomatik kümeleme yöntemleri kendi aralarında değerlendirildiğinde ise, sürekli ABC'ye dayalı yöntemlerin ikili ABC ve K-means'e dayalı hibrit yöntemlerden daha iyi VI indeks değerleri elde ettiği görülmüştür. Bunun en temel sebebi sürekli modeller sınırlamasız bir optimizasyon işlemi yürütürken, ikili modeller sınırlamalı bir optimizasyon işlemi yürütmektedir. Ancak, ikili ABC ve K-means algoritmalarının senkronize edilmesiyle geliştirilen hibrit otomatik kümeleme yöntemleri sürekli ABC'ye dayalı yöntemlerden daha iyi Rand indeks sonuçları elde etmiştir. Ürettiği VI indeks değerlerinin daha iyi olmasına rağmen, sürekli ABC modellerinin hibrit modellerden daha kötü Rand indeks değerleri elde etmesinin temel sebepleri şunlardır: 1) Sınırlamasız optimizasyon işlemi nedeniyle sürekli ABC'nin küme içi uzaklığı aşırı derecede azaltması bazı elde edilen kümelerdeki eleman sayısını aşırı düşürmektedir ve 2) Hibrit modellerin sadece küme seti seçmeye odaklanması ve merkez güncelleme işlemini K-means algoritmasına bırakmasıdır. Bundan dolayı tez çalışmasında dikkate alınan problemler için ikili ABC ve K-means'e dayalı hibrit yöntemlerin küme sayısı saptamada kullanımının sürekli ABC'ye dayalı yöntemlere göre daha uygun olduğunu söylemek mümkündür.

6.2. Öneriler

Bu tez çalışması ile bağlantılı olarak gelecekte aşağıda bulunan konular araştırılabilir:

Amaç fonksiyonu olarak kullanılan kümeleme kriteri: Mekânsal bilgi henüz ABC tabanlı kümeleme yöntemlerinde kullanılmış değildir. Örneğin, sekizli komşuluk bilgisi kullanarak yeni bir amaç fonksiyonu tasarlanabilir.

Kuantalama yöntemi: CQ-ABC kuantalama yöntemi, K-means algoritması ile senkronize bir şekilde çalışarak başlangıç koşullarının etkisini azaltmaya çalışmıştır. Ancak K-means yerine FCM veya K-medoids gibi farklı kümeleme yöntemlerinin CQ-ABC'ye entegre edilmesi ile oluşturulan yeni alternatiflerin daha iyi performans verip vermeyeceği araştırılabilir.

Kümeleme performans analizi: Her ne kadar elde edilen kümeleme sonuçlarını değerlendirecek birçok kriter ve doğrulama indeksi mevcut olsa da, bu kriter ve indekslerin hangi durumlarda ve verilerde daha uygun olacağına dair detaylı bir analiz çalışması mevcut değildir. Bu konu araştırılabilir.

Otomatik kümeleme yöntemleri: ABC'nin verideki küme sayısını otomatik ve doğru olarak belirlemesine yönelik kullanılması ile ilgili çalışmalar henüz başlangıç aşamasındadır. Dolayısıyla bu problemin çözümü için mevcut ABC modellerinden daha güçlü ABC modellerinin geliştirilmesi mümkün olabilir. Ayrıca, VI indeksi yerine farklı indeksler de amaç fonksiyonu olarak optimizasyon işleminde kullanılabilir.

KAYNAKÇA

1. Bocij, P., Chaffey, D., Greasley, A., Hickie, S., 2009. Business Information Systems: Technology, Development & Management for the E-business. FT Press.
2. Zaki, M. J., Wagner Meira, J., 2014. Data Mining and Analysis: Fundamental Concepts and Algorithms. Cambridge University Press.
3. Koylu, F., 2015. Yapay Arı Koloni Algoritması Tabanlı Veri Madenciliği Algoritmalarının Geliştirilmesi. Erciyes Universitesi, Fen Bilimleri Enstitüsü, Doktora Tezi.
4. Agrawal, R., Srikant, R., 1994. Fast Algorithms for Mining Association Rules in Large Databases, pp. 487-499. *20th International Conference on Very Large Databases*.
5. Anderberg, M. R., 1973. Cluster analysis for applications. Academic Press, New York.
6. Omran, M., Al-Sharhan, S., 2007. Barebones particle swarm methods for unsupervised image classification, pp. 3247-3252. *IEEE Congress on Evolutionary Computation (CEC'2007),* 25-28 Sept. 2007.
7. Abul Hasan, M., Ramakrishnan, S., 2011. A survey: hybrid evolutionary algorithms for cluster analysis. **Artificial Intelligence Review, 36** (3): 179-204.
8. Hruschka, E. R., Campello, R. J. G. B., Freitas, A. A., De Carvalho, A. C. P. L. F., 2009. A Survey of Evolutionary Algorithms for Clustering. **IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 39** (2): 133-155.
9. Sheikh, R. H., Raghuwanshi, M. M., Jaiswal, A. N., 2008. Genetic Algorithm Based Clustering: A Survey, pp. 314-319. *International Conference on Emerging Trends in Engineering and Technology (ICETET '08),* 16-18 July 2008.
10. Alam, S., Dobbie, G., Koh, Y. S., Riddle, P., Ur Rehman, S., 2014. Research on particle swarm optimization based clustering: A systematic review of literature and techniques. **Swarm and Evolutionary Computation, 17**: 1-13.

11. Karaboga, D., 2005. An idea based on honey bee swarm for numerical optimization. Erciyes University, Engineering Faculty, Computer Engineering Department Technical Report-TR06.
12. Bandyopadhyay, S., Maulik, U., 2002. Genetic clustering for automatic evolution of clusters and application to image classification. **Pattern Recognition, 35** (6): 1197-1208.
13. Das, S., Abraham, A., Konar, A., 2008. Automatic Clustering Using an Improved Differential Evolution Algorithm. **IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, 38** (1): 218-237.
14. Su, Z.-g., Wang, P.-h., Shen, J., Li, Y.-g., Zhang, Y.-f., Hu, E.-j., 2012. Automatic fuzzy partitioning approach using Variable string length Artificial Bee Colony (VABC) algorithm. **Applied Soft Computing, 12** (11): 3421-3441.
15. Karaboga, D., Gorkemli, B., Ozturk, C., Karaboga, N., 2014. A comprehensive survey: artificial bee colony (ABC) algorithm and applications. **Artificial Intelligence Review, 42** (1): 21-57.
16. Hansen, P., Jaumard, B., 1997. Cluster analysis and mathematical programming. **Mathematical Programming, 79** (1-3): 191-215.
17. Everitt, B. S., Landau, S., Leese, M., 2009. Cluster Analysis. Wiley Publishing.
18. Jain, A. K., Murty, M. N., Flynn, P. J., 1999. Data clustering: a review. **ACM Computing Surveys, 31** (3): 264-323.
19. Sarstedt, M., Mooi, E., 2011. A Concise Guide to Market research: The process, data, and methods using IBM SPSS statistics. Springer Verlag.
20. Frigui, H., Krishnapuram, R., 1997. Clustering by competitive agglomeration. **Pattern Recognition, 30** (7): 1109-1119.
21. Han, J., Kamber, M., Tung, A. K. H., 2001. Spatial Clustering Methods in Data Mining: A Survey. *In* Geographic Data Mining and Knowledge Discovery (Eds: Miller, H. J.Han, J.). Taylor and Francis, Bristol, PA, USA.
22. Demiralay, M., 2005. Hiyerarşik kümeleme metotları ile veri madenciliği uygulamaları. Marmara Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi.
23. Sneath, P. H. A., Sokal, R. R., 1973. Numerical Taxonomy: The Principles and Practice of Numerical Classification. W H Freeman Limited.

24. Zhang, T., Ramakrishnan, R., Livny, M., 1997. BIRCH: A New Data Clustering Algorithm and Its Applications. **Data Mining and Knowledge Discovery, 1** (2): 141-182.
25. Guha, S., Rastogi, R., Kyuseok, S., 1999. ROCK: a robust clustering algorithm for categorical attributes, pp. 512-521. *15th International Conference on Data Engineering,* Washington, DC, USA.
26. Andritsos, P., Tsaparas, P., Miller, R., Sevcik, K., 2004. LIMBO: Scalable Clustering of Categorical Data. *In* Advances in Database Technology - EDBT 2004 (Eds: Bertino, E., Christodoulakis, S., Plexousakis, D., Christophides, V., Koubarakis, M., Böhm, K.Ferrari, E.). Springer Berlin Heidelberg.
27. Murtagh, F., 1983. A Survey of Recent Advances in Hierarchical Clustering Algorithms. **The Computer Journal, 26** (4): 354-359.
28. Murtagh, F., Contreras, P., 2012. Algorithms for hierarchical clustering: an overview. **WIREs Data Mining Knowl Discov, 2**: 86–97.
29. Omran, M., 2004. Particle Swarm Optimization Methods for Pattern Recognition and Image Processing. University of Pretoria, PHD Thesis, Environment and Information Technology.
30. MacQueen, J., 1967. Some methods for classification and analysis of multivariate observations, pp. 281–297. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*.
31. Bezdek, J. C., 1981. Pattern Recognition with Fuzzy Objective Function Algoritms. Plenum Press, New York.
32. Kaufman, L., Rousseeuw, P., 1987. Clustering by means of medoids. **Statistical Data Analysis Based on the L1-Norm and Related Methods, North-Holland**: 405–416.
33. Kaufman, L., Rosseeuw, P. J., 1990. Finding Groups in Data: An Introduction to Cluster Analysis. Wiley, New York.
34. Ng, R. T., Han, J., 1994. Efficient and Effective Clustering Methods for Spatial Data Mining, pp. 144-155. *20th International Conference on Very Large Data Bases*.
35. Rijsbergen, C. J. V., 1979. Information Retrieval. Butterworth-Heinemann.
36. Rand, W. M., 1971. Objective Criteria for the Evaluation of Clustering Methods. **Journal of the American Statistical Association, 66** (336): 846-850.

37. Banerjee, A., Dhillon, I. S., Ghosh, J., Sra, S., 2005. Clustering on the Unit Hypersphere using von Mises-Fisher Distributions. **The Journal of Machine Learning Research, 6**: 1345-1382.
38. Zhao, Y., Karypis, G., 2004. Criterion functions for document clustering: Experiments and analysis. **Machine Learning, 55**: 311-331.
39. Calinski, R. B., Harabasz, J., 1974. A dendrite method for cluster analysis. **Communications in Statistics, 3** (1): 1–27.
40. Dunn, J., 1974. Well separated clusters and optimal fuzzy partitions. **Journal of Cybernetics, 4** (95): 95-104.
41. Davies, D. L., Bouldin, D. W., 1979. A cluster separation measure. **IEEE Transactions on Pattern Analysis and Machine Intelligence, 1** (4): 224–227.
42. Chou, C. H., Su, M. C., Lai, E., 2004. A new cluster validity measure and its application to image compression. **Pattern Analysis and Applications, 7** (2): 205-220.
43. Zhao, Q., Xu, M., Fränti, P., 2009. Sum-of-Squares Based Cluster Validity Index and Significance Analysis Adaptive and Natural Computing Algorithms. (Eds: Kolehmainen, M., Toivanen, P.Beliczynski, B.). Springer Berlin / Heidelberg.
44. Saha, S., Bandyopadhyay, S., 2009. Performance Evaluation of Some Symmetry-Based Cluster Validity Indexes. **IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 39** (4): 420-425.
45. Xie, X., Beni, G., 1991. A validity measure for fuzzy clustering. **IEEE Transactions on Pattern Analysis and Machine Intelligence, 13** (4): 841–846.
46. Cheng, H. D., Jiang, X. H., Sun, Y., Wang, J., 2001. Color image segmentation: advances and prospects. **Pattern Recognition, 34** (12): 2259-2281.
47. Gonzelez, R. C., Woods, R. E., 2012. Digital Image Processing. Prentice Hall.
48. Xiaolin, W., Kaizhong, Z., 1991. A better tree-structured vector quantizer, pp. 392-401. *Data Compression Conference (DCC '91)*, 8-11 Apr 1991.
49. Heckbert, P., 1982. Color image quantization for frame buffer display. **Computer Graphics, 16** (3): 297-307.
50. Kruger, A., 1994. Median-cut color quantization. **Dr. Dobb's Journal**: 46-54, 91-92.

51. Gervautz, M., Purgathofer, W., 1990. A simple method for color quantization: octree quantization. *In* Graphics gems (Eds: Andrew, S. G.). Academic Press Professional, Inc.
52. Clark, D., 1995. The popularity algorithm. **Dr. Dobb's Journal**: 121-128.
53. Freisleben, B., Schrader, A., 1997. An evolutionary approach to color image quantization, pp. 459-464. *IEEE International Conference on Evolutionary Computation,* 13-16 Apr 1997.
54. Scheunders, P., 1997. A comparison of clustering algorithms applied to color image quantization. **Pattern Recogniton Letters, 18** (11-13): 1379-1384.
55. Omran, M., Engelbrecht, A., 2005. A Color Image Quantization Algorithm Based On Particle Swarm Optimization. **Informatica, 29**: 261-269.
56. Dekker, A. H., 1994. Kohonen neural networks for optimal colour quantization. **Network: Computation in Neural Systems, 5** (3): 351-367.
57. Papamarkos, N., Atsalakis, A. E., Strouthopoulos, C. P., 2002. Adaptive color reduction. **IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 32** (1): 44-56.
58. Bock, H. H., 1996. Probabilistic models in cluster analysis. **Computational Statistics & Data Analysis, 23** (1): 5-28.
59. Baraldi, A., Blonda, P., 1999. A survey of fuzzy clustering algorithms for pattern recognition. I. **IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 29** (6): 778-785.
60. Baraldi, A., Blonda, P., 1999. A survey of fuzzy clustering algorithms for pattern recognition II. **IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 29** (6): 786-801.
61. Wei, C.-P., Lee, Y.-H., Hsu, C.-M., 2000. Empirical Comparison of Fast Clustering Algorithms for Large Data Sets, pp. 1-10. *33rd Hawaii International Conference on System Sciences*.
62. Xu, R., Wunsch, D., II, 2005. Survey of clustering algorithms. **IEEE Transactions on Neural Networks, 16** (3): 645-678.
63. Mukhopadhyay, A., Maulik, U., Bandyopadhyay, S., 2015. A Survey of Multiobjective Evolutionary Clustering. **ACM Computing Surveys, 47** (4): 1-46.

64. Dharmarajan, A., Velmurugan, T., 2013. Applications of partition based clustering algorithms: A survey, pp. 1-5. *IEEE International Conference on Computational Intelligence and Computing Research (ICCIC'2013),* 26-28 Dec. 2013.
65. Fahad, A., Alshatri, N., Tari, Z., Alamri, A., Khalil, I., Zomaya, A. Y., Foufou, S., Bouras, A., 2014. A Survey of Clustering Algorithms for Big Data: Taxonomy and Empirical Analysis. **IEEE Transactions on Emerging Topics in Computing, 2** (3): 267-279.
66. Omran, M. G. H., Engelbrecht, A. P., Salman, A., 2007. An overview of clustering methods. **Intelligent Data Analysis, 11** (6): 583-605.
67. Grira, N., Crucianu, M., Boujemaa, N., 2004. Unsupervised and Semi-supervised Clustering: a Brief Survey, pp. *A Review of Machine Learning Techniques for Processing Multimedia Content, Report of the MUSCLE European Network of Excellence (FP6).*
68. Aggarwal, C., Zhai, C., 2012. A Survey of Text Clustering Algorithms. *In* Mining Text Data (Eds: Aggarwal, C. C.Zhai, C.). Springer US.
69. Abbasi, A. A., Younis, M., 2007. A survey on clustering algorithms for wireless sensor networks. **Computer Communications, 30** (14–15): 2826-2841.
70. Mirkin, B., 2011. Choosing the number of clusters. **Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 1** (3): 252-260.
71. Zhao, Q., 2012. Cluster validity in clustering methods. University of Eastern Finland, Faculty of Science and Forestry Dissertation, Phd Thesis, Finland.
72. Milligan, G., Cooper, M., 1985. An examination of procedures for determining the number of clusters in a data set. **Psychometrika, 50** (2): 159-179.
73. Dimitriadou, E., Dolničar, S., Weingessel, A., 2002. An examination of indexes for determining the number of clusters in binary data sets. **Psychometrika, 67** (1): 137-159.
74. Ratkowsky, D. A., Lance, G. N., 1978. A criterion for determining the number of groups in a classification. **Australian Computer Journal, 10**: 115–117.
75. Xu, L., 1997. Bayesian Ying-Yang machine, clustering and number of clusters. **Pattern Recognition Letters, 18**: 1167-1178.
76. Scott, A. J., Symons, M. J., 1971. Clustering Methods Based on Likelihood Ratio Criteria. **Biometrics, 27** (2): 387-397.

77. Hubert, L., Schultz, J., 1976. Quadric Assignment As a General Data Analysis Strategy. **British Journal of Mathematical and Statistical Psychology, 29** (2): 190-241.
78. Tibshirani, R., Walther, G., Hastie, T., 2001. Estimating the number of clusters in a data set via the gap statistic. **Journal of the Royal Statistical Society: Series B (Statistical Methodology), 63** (2): 411-423.
79. Yan, M., Ye, K., 2007. Determining the Number of Clusters Using the Weighted Gap Statistic. **Biometrics, 63** (4): 1031-1037.
80. Mohajer, M., Englmeier, K.-H., Schmid, V. J., 2011. A comparison of Gap statistic definitions with and without logarithm function. **CoRR,**
81. Salvador, S., Chan, P., 2005. Learning States and Rules for Detecting Anomalies in Time Series. **Applied Intelligence, 23** (3): 241-255.
82. Sugar, C. A., James, G. M., 2003. Finding the number of clusters in a data set—an information theoretic approach. **Journal of American Statistics Association, 98**: 750–763.
83. Zhao, Q., Hautamaki, V., Fränti, P., 2008. Knee Point Detection in BIC for Detecting the Number of Clusters. *In* Advanced Concepts for Intelligent Vision Systems (Eds: Blanc-Talon, J., Bourennane, S., Philips, W., Popescu, D.Scheunders, P.). Springer Berlin Heidelberg.
84. Schwarz, G., 1978. Estimating the Dimension of a Model. **Annals of Statistics, 6** (2): 461-464.
85. Qinpei, Z., Mantao, X., Franti, P., 2008. Knee Point Detection on Bayesian Information Criterion, pp. 431-438. *20th IEEE International Conference on Tools with Artificial Intelligence (ICTAI '08)* 3-5 Nov. 2008.
86. Fujita, A., Takahashi, D. Y., Patriota, A. G., 2014. A non-parametric method to estimate the number of clusters. **Computational Statistics & Data Analysis, 73**: 27-39.
87. Rousseeuw, P. J., 1987. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. **Journal of Computational and Applied Mathematics, 20**: 53-65.
88. Tibshirani, R., Walther, G., 2005. Cluster Validation by Prediction Strength. **Journal of Computational and Graphical Statistics, 14** (3): 511-528.

89. Peng, Y., Zhang, Y., Kou, G., Shi, Y., 2012. A Multicriteria Decision Making Approach for Estimating the Number of Clusters in a Data Set. **PLoS ONE, 7** (7): e41713.
90. Pedersen, T., Kulkarni, A., 2006. Automatic cluster stopping with criterion functions and the gap statistic, pp. 276-279. *NAACL-Demonstrations '06,* New York.
91. Hartigan, J. A., 1975. Clustering Algorithms. John Wiley & Sons, Inc.
92. Mojena, R., 1977. Hierarchical grouping methods and stopping rules: an evaluation. **The Computer Journal, 20** (4): 359-363.
93. Dice, L. R., 1945. Measures of the Amount of Ecologic Association Between Species. **Ecology, 26** (3): 297-302.
94. Charrad, M., Ghazzali, N., Boiteau, V., Niknafs, A., 2014. NbClust: An R Package for Determining the Relevant Number of Clusters in a Data Set. **Journal of Statistical Software, 61** (6): 1-36.
95. Zhang, D., Ji, M., Yang, J., Zhang, Y., Xie, F., 2014. A novel cluster validity index for fuzzy clustering based on bipartite modularity. **Fuzzy Sets and Systems, 253**: 122-137.
96. Chiang, M.-T., Mirkin, B., 2010. Intelligent Choice of the Number of Clusters in K-Means Clustering: An Experimental Study with Different Cluster Spreads. **Journal of Classification, 27** (1): 3-40.
97. Minaei-Bidgoli, B., Topchy, A. P., Punch, W. F., 2004. A Comparison of Resampling Methods for Clustering Ensembles, pp. 939-945. *International Conference on Machine Learning; Models, Technologies and Application (MLMTA04).*
98. Dudoit, S., Fridlyand, J., 2002. A prediction-based resampling method for estimating the number of clusters in a dataset. **Genome Biology, 3** (7): 1-21.
99. McLachlan, G. J., Khan, N., 2004. On a resampling approach for tests on the number of clusters with mixture model-based clustering of tissue samples. **Journal of Multivariate Analysis, 90** (1): 90-105.
100. Fang, Y., Wang, J., 2012. Selection of the number of clusters via the bootstrap method. **Computational Statistics & Data Analysis, 56** (3): 468-477.
101. Bel Mufti, G., Bertrand, P., El Moubarki, L., 2005. Determining the number of groups from measures of cluster stability, pp. 404-413. *10th International*

Symposium on Applied Stochastic Models and Data Analysis (ASMDA2005), Brest, France.

102. Ball, G., Hall, L., 1965. ISODATA, A novel method of data analysis and pattern classification

Standford Research Institute, Tech. Rep. NTIS No. AD 699616, Menlo Park, CA.

103. Tou, J., 1979. Dynoc—A dynamic optimal cluster-seeking technique. **International Journal of Computer & Information Sciences, 8** (6): 541-547.

104. Huang, K., 2002. A Synergistic Automatic Clustering Technique (SYNERACT) for Multispectral Image Analysis. **Journal of the American Society for Photogrammetry and Remote Sensing 68** (1): 33-40.

105. Wallace, C., Dowe, D., 1994. Intrinsic Classification by MML – the snob program, pp. 37-44. *Seventh Australian Joint Conference on Artificial Intelligence,* NSW, Australia.

106. Kanungo, T., Dom, B., Niblack, W., Steele, D., 1994. A fast algorithm for MDL-based multi-band image segmentation, pp. 609-616. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '94),* 21-23 Jun 1994.

107. Bischof, H., Leonardis, A., Selb, A., 1999. MDL Principle for Robust Vector Quantization. **Pattern Analysis and Applications, 2**: 59-72.

108. Veenman, C. J., Reinders, M. J. T., Backer, E., 2002. A Maximum Variance Cluster Algorithm. **IEEE Trans. Pattern Anal. Mach. Intell., 24** (9): 1273-1280.

109. Pelleg, D., Moore, A., 2000. X-means: Extending K-means with Efficient Estimation of the Number of Clusters, pp. 727-734. *Seventeenth International Conference on Machine Learning,* San Francisco.

110. Shahbaba, M., Beheshti, S., 2012. Improving X-means clustering with MNDL, pp. 1298-1302. *11th International Conference on Information Science, Signal Processing and their Applications (ISSPA'2012)* 2-5 July 2012.

111. Beheshti, S., Dahleh, M. A., 2010. Noisy Data and Impulse Response Estimation. **Signal Processing, IEEE Transactions on, 58** (2): 510-521.

112. Hamerly, G., Elkan, C., 2003. Learning the k in k-means, pp. 281-288. *Seventeenth Annual Conference on Neural Information Processing Systems (NIPS2003).*

113. Foina, A., Badia, R., Ramirez-Fernandez, J., 2010. G-Means Improved for Cell BE Environment. *In* Facing the Multicore-Challenge (Eds: Keller, R., Kramer, D.Weiss, J.-P.). Springer Berlin Heidelberg.

114. Feng, Y., Hamerly, G., 2006. PG-means: learning the number of clusters in data, pp. 393-400. *Advances in Neural Information Processing Systems (NISP2006).*

115. Vatsavai, R. R., Symons, C. T., Chandola, V., Jun, G., 2011. GX-Means: A model-based divide and merge algorithm for geospatial image clustering. **Procedia Computer Science, 4**: 186-195.

116. Lei, H., Tang, L. R., Iglesias, J. R., 2007. S-means: similarity driven clustering and its application in gravitational-wave astronomy data mining, pp. 25-36. *Int. Workshop on Knowledge Discovery from Ubiquitous Data Streams,* Warsaw

117. Tang, L. R., Lei, H., Mukherjee, S., Mohanty, S., 2008 Cluster analysis of simulated gravitational wave triggers using S-means and constrained validation clustering. **Classical and Quantum Gravity, 25** (18): 184023.

118. Kalogeratos, A., Likas, A., 2012. Dip-means: an incremental clustering method for estimating the number of clusters, pp. 2402-2410. *Advances in Neural Information Processing Systems*.

119. Welling, M., Kurihara, K., 2008. Bayesian k-Means as a "Maximization-Expectation" Algorithm. **Neural Computation, 21** (4): 1145-1172.

120. Zhang, Y., Xiaofei, X., Ye, Y., 2010. NSS-AKmeans: An Agglomerative Fuzzy K-means clustering method with automatic selection of cluster number, pp. 32-38. *2nd International Conference on Advanced Computer Control (ICACC),* 27-29 March 2010.

121. Koonsanit, K., Jaruskulchai, C., Eiumnoh, A., 2012. Parameter-Free K-Means Clustering Algorithm for Satellite Imagery Application, pp. 1-6. *International Conference on Information Science and Applications (ICISA),* 23-25 May 2012.

122. He, Z., Cichocki, A., Xie, S., Choi, K., 2010. Detecting the Number of Clusters in n-Way Probabilistic Clustering. **IEEE Transactions on Pattern Analysis and Machine Intelligence, 32** (11): 2006-2021.

123. Gupta, U. D., Menon, V., Babbar, U., 2010. Detecting the Number of Clusters during Expectation-Maximization Clustering Using Information Criterion, pp. 169-173. *Second International Conference on Machine Learning and Computing*.

124. Shahbaba, M., Beheshti, S., 2014. MACE-means clustering. **Signal Processing, 105**: 216-225.

125. Liang, J., Zhao, X., Li, D., Cao, F., Dang, C., 2012. Determining the number of clusters using information entropy for mixed data. **Pattern Recognition, 45** (6): 2251-2265.

126. Gluck, M. A., Corter, 1985. Information, uncertainty and the utility of categories, pp. 283-287. *7th Annual Conference of the Cognitive Science Society*.

127. Bandyopadhyay, S., Maulik, U., 2001. Nonparametric genetic clustering: comparison of validity indices. **IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 31** (1): 120-125.

128. D. E. Goldberg, Deb, K., Korb, B., 1989. Messy genetic algorithms: Motivation, analysis, and first results. **Complex Systems, 3**: 493-530.

129. Maulik, U., Bandyopadhyay, S., 2003. Fuzzy partitioning using a real-coded variable-length genetic algorithm for pixel classification. **IEEE Transactions on Geoscience and Remote Sensing, 41** (5): 1075-1081.

130. Hruschka, E. R., Ebecken, N. F. f., 2003. A genetic algorithm for cluster analysis. **Intelligent Data Analysis, 7** (1): 15-25.

131. Holland, J., 2012. Genetic algorithms. **Scholarpedia, 7**: 1482.

132. Hruschka, E. R., de Castro, L. N., Campello, R. J. G. B., 2004. Evolutionary algorithms for clustering gene-expression data, pp. 403-406. *Fourth IEEE International Conference on Data Mining (ICDM '04),* 1-4 Nov. 2004.

133. Hruschka, E. R., Campello, R. J. G. B., de Castro, L. N., 2006. Evolving clusters in gene-expression data. **Information Sciences, 176** (13): 1898-1927.

134. Alves, V. S., Campello, R. J. G. B., Hruschka, E. R., 2006. Towards a Fast Evolutionary Algorithm for Clustering, pp. 1776-1783. *IEEE Congress on Evolutionary Computation,* Santos, Brazil.

135. Dai, W., Jiao, C., He, T., 2007. Research of K-means Clustering Method Based on Parallel Genetic Algorithm, pp. 158-161. *Third International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP'07),* 26-28 Nov. 2007.

136. Lai, C.-C., 2005. A Novel Clustering Approach using Hierarchical Genetic Algorithms. **Intelligent Automation & Soft Computing, 11** (3): 143-153.

137. Man, K. F., Tang, K. S., Kwong, S., 1996. Genetic algorithms: concepts and applications [in engineering design]. **IEEE Transactions on Industrial Electronics, 43** (5): 519-534.

138. Lin, H.-J., Yang, F.-W., Kao, Y.-T., 2005. An Efficient GA-based Clustering Technique. **Tamkang Journal of Science and Engineering, 8** (2): 113-122.

139. Liu, Y., Wu, X., Shen, Y., 2011. Automatic clustering using genetic algorithms. **Applied Mathematics and Computation, 218** (4): 1267-1279.

140. Sheng, W., Swift, S., Zhang, L., Liu, X., 2005. A weighted sum validity function for clustering with a hybrid niching genetic algorithm. **IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 35** (6): 1156-1167.

141. Leon, E., Nasraoui, O., Gomez, J., 2006. ECSAGO: Evolutionary Clustering with Self Adaptive Genetic Operators, pp. 1768-1775. *IEEE Congress on Evolutionary Computation (CEC'06),* Bogota, Colombia.

142. Bandyopadhyay, S., Saha, S., 2008. A Point Symmetry-Based Clustering Technique for Automatic Evolution of Clusters. **IEEE Transactions on Knowledge and Data Engineering, 20** (11): 1441-1457.

143. Bandyopadhyay, S., Saha, S., 2007. GAPS: A clustering method using a new point symmetry-based distance measure. **Pattern Recognition, 40** (12): 3430-3451.

144. Saha, S., Bandyopadhyay, S., 2008. Application of a New Symmetry-Based Cluster Validity Index for Satellite Image Segmentation. **IEEE Geoscience and Remote Sensing Letters, 5** (2): 166-170.

145. Saha, S., Bandyopadhyay, S., 2009. A new point symmetry based fuzzy genetic clustering technique for automatic evolution of clusters. **Information Sciences, 179** (19): 3230-3246.

146. He, H., Tan, Y., 2012. A two-stage genetic algorithm for automatic clustering. **Neurocomputing, 81**: 49-59.

147. Agustı´n-Blas, L. E., Salcedo-Sanz, S., Jiménez-Fernández, S., Carro-Calvo, L., Del Ser, J., Portilla-Figueras, J. A., 2012. A new grouping genetic algorithm for clustering problems. **Expert Systems with Applications, 39** (10): 9695-9703.

148. Falkenauer, E., 1992. The grouping genetic algorithm–widening the scope of the GAs, pp. 79–102. *Proceedings of the Belgian Journal of operations research, statistics and computer science*.

149. Falkenauer, E., 1998. Genetic Algorithms and Grouping Problems. John Wiley.

150. Salcedo-Sanz, S., Carro-Calvo, L., Portilla-Figueras, A., Cuadra, L., Camacho, D., 2013. Fuzzy Clustering with Grouping Genetic Algorithms. *In* Intelligent Data Engineering and Automated Learning – IDEAL 2013 (Eds: Yin, H., Tang, K., Gao, Y., Klawonn, F., Lee, M., Weise, T., Li, B.Yao, X.). Springer Berlin Heidelberg.

151. Tseng, L. Y., Yang, S. B., 2001. A genetic approach to the automatic clustering problem. **Pattern Recognition, 34** (2): 415-424.

152. Cowgill, M. C., Harvey, R. J., Watson, L. T., 1999. A genetic algorithm approach to cluster analysis. **Computers & Mathematics with Applications, 37** (7): 99-108.

153. Chang, D.-X., Zhang, X.-D., Zheng, C.-W., Zhang, D.-M., 2010. A robust dynamic niching genetic algorithm with niche migration for automatic clustering problem. **Pattern Recognition, 43** (4): 1346-1360.

154. Raposo, C., Antunes, C., Barreto, J., 2014. Automatic Clustering Using a Genetic Algorithm with New Solution Encoding and Operators. *In* Computational Science and Its Applications – ICCSA 2014 (Eds: Murgante, B., Misra, S., Rocha, A. C., Torre, C., Rocha, J., Falcão, M., Taniar, D., Apduhan, B.Gervasi, O.). Springer International Publishing.

155. Rahman, M. A., Islam, M. Z., 2014. A hybrid clustering technique combining a novel genetic algorithm with K-Means. **Knowledge-Based Systems, 71**: 345-365.

156. Othman, R. M., Deris, S., Zakaria, Z., Illias, R. M., Mohamad, S. M., 2006. Automatic clustering of gene ontology by genetic algorithm. **International Journal of Information Technology, 3** (1): 37-46.

157. Omran, M. G. H., Salman, A., Engelbrecht, A. P., 2006. Dynamic clustering using particle swarm optimization with application in image segmentation. **Pattern Analysis and Applications, 8** (4): 332-344.

158. Kohonen, T., Honkela, T., 2007. Kohonen network. **Scholarpedia, 2**: 1568.

159. Das, S., Abraham, A., Konar, A., 2006. Spatial Information Based Image Segmentation Using a Modified Particle Swarm Optimization Algorithm, pp. 438-444. *Sixth International Conference on Intelligent Systems Design and Applications (ISDA '06)*, 16-18 Oct. 2006.

160. Ouadfel, S., Batouche, M., Taleb-Ahmed, A., 2010. A Modified Particle Swarm Optimization Algorithm for Automatic Image Clustering, pp. 49-57. *International Symposium on Modelling and Implementation of Complex Systems (MISC'2010)*.

161. Kuo, R. J., Syu, Y. J., Chen, Z.-Y., Tien, F. C., 2012. Integration of particle swarm optimization and genetic algorithm for dynamic clustering. **Information Sciences, 195**: 124-140.

162. Kao, Y., Lee, S.-Y., 2009. Combining K-means and particle swarm optimization for dynamic data clustering problems, pp. 757-761. *IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS'2009),* 20-22 Nov. 2009.

163. Jarboui, B., Cheikh, M., Siarry, P., Rebai, A., 2007. Combinatorial particle swarm optimization (CPSO) for partitional clustering problem. **Applied Mathematics and Computation, 192** (2): 337-345.

164. Kuo, R. J., Zulvia, F. E., 2013. Automatic Clustering Using an Improved Particle Swarm Optimization. **Journal of Industrial and Intelligent Information, 1** (1): 46-51.

165. Das, S., Konar, A., 2009. Automatic image pixel clustering with an improved differential evolution. **Applied Soft Computing, 9** (1): 226-236.

166. Das, S., Konar, A., Chakraborty, U. K., 2006. Automatic Fuzzy Segmentation of Images with Differential Evolution, pp. 2026-2033. *IEEE Congress on Evolutionary Computation (CEC'2006),* Vancouver, BC.

167. Das, S., Chowdhury, A., Abraham, A., 2009. A Bacterial Evolutionary Algorithm for Automatic Data Clustering, pp. 2403-2410. *IEEE Congress on Evolutionary Computation (CEC'2009)*.

168. Das, S., Sil, S., 2010. Kernel-induced fuzzy clustering of image pixels with an improved differential evolution algorithm. **Information Sciences, 180** (8): 1237-1256.

169. Das, S., Sil, S., Chakraborty, U. K., 2008. Kernel-based clustering of image pixels with modified Differential Evolution, pp. 3472-3479. *IEEE Congress on Evolutionary Computation (CEC'2008),* 1-6 June 2008.

170. Das, S., Abraham, A., Konar, A., 2008. Automatic kernel clustering with a Multi-Elitist Particle Swarm Optimization Algorithm. **Pattern Recognition Letters, 29** (5): 688-699.

171. Kuo, R. J., Huang, Y. D., Lin, C.-C., Wu, Y.-H., Zulvia, F. E., 2014. Automatic kernel clustering with bee colony optimization algorithm. **Information Sciences, 283**: 107-122.

172. Maulik, U., Saha, I., 2010. Automatic Fuzzy Clustering Using Modified Differential Evolution for Image Classification. **IEEE Transactions on Geoscience and Remote Sensing, 48** (9): 3503-3510.

173. Saha, I., Maulik, U., Bandyopadhyay, S., Plewczynski, D., 2011. Improvement of new automatic differential fuzzy clustering using SVM classifier for microarray analysis. **Expert Systems with Applications, 38** (12): 15122-15133.

174. Rui, X., Jie, X., Wunsch, D. C., 2012. A Comparison Study of Validity Indices on Swarm-Intelligence-Based Clustering. **IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 42** (4): 1243-1256.

175. Pan, S.-M., Cheng, K.-S., 2007. Evolution-Based Tabu Search Approach to Automatic Clustering. **IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 37** (5): 827-838.

176. Bandyopadhyay, S., Pakhira, M. K., Maulik, U., 2004. Validity index for crisp and fuzzy clusters. **Pattern Recognition, 37**: 487-501.

177. Kumar, V., Chhabra, J. K., Kumar, D., 2014. Automatic cluster evolution using gravitational search algorithm and its application on image segmentation. **Engineering Applications of Artificial Intelligence, 29**: 93-103.

178. Liu, C., Zhou, A., Zhang, G., 2013. Automatic clustering method based on evolutionary optimisation. **IET Computer Vision, 7** (4): 258-271.

179. Husseinzadeh Kashan, A., Rezaee, B., Karimiyan, S., 2013. An efficient approach for unsupervised fuzzy clustering based on grouping evolution strategies. **Pattern Recognition, 46** (5): 1240-1254.

180. Kashan, A. H., Jenabi, M., Kashan, M. H., 2009. A New Solution Approach for Grouping Problems Based on Evolution Strategies, pp. 88-93. *International Conference of Soft Computing and Pattern Recognition (SOCPAR '09),* 4-7 Dec. 2009.

181. Alia, O. M., Mandava, R., Ramachandram, D., Aziz, M. E., 2009. Dynamic Fuzzy Clustering using Harmony Search with Application to Image Segmentation, pp. 538-543. *IEEE International Symposium on Signal Processing and Information Technology*.

182. Liu, R., Zhu, B., Bian, R., Ma, Y., Jiao, L., 2015. Dynamic local search based immune automatic clustering algorithm and its applications. **Applied Soft Computing, 27**: 250-268.

183. Taherdangkoo, M., Hossein Shirzadi, M., Yazdi, M., Hadi Bagheri, M., 2013. A robust clustering method based on blind, naked mole-rats (BNMR) algorithm. **Swarm and Evolutionary Computation, 10**: 1-11.

184. Handl, J., Knowles, J., 2007. An Evolutionary Approach to Multiobjective Clustering. **IEEE Transactions on Evolutionary Computation, 11** (1): 56-76.

185. Corne, D. W., Jerram, N. R., Knowles, J. D., Oates, M. J., 2001. PESA-II: Region-based Selection in Evolutionary Multiobjective Optimization, pp. 283-290. *Genetic and Evolutionary Computation Conference (GECCO'2001),* San Francisco, USA.

186. Bandyopadhyay, S., Maulik, U., Mukhopadhyay, A., 2007. Multiobjective Genetic Clustering for Pixel Classification in Remote Sensing Imagery. **IEEE Transactions on Geoscience and Remote Sensing, 45** (5): 1506-1511.

187. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. **IEEE Transactions on Evolutionary Computation, 6** (2): 182-197.

188. Mukhopadhyay, A., Maulik, U., 2009. Unsupervised Pixel Classification in Satellite Imagery Using Multiobjective Fuzzy Clustering Combined With SVM Classifier. **IEEE Transactions on Geoscience and Remote Sensing, 47** (4): 1132-1138.

189. Kundu, D., Suresh, K., Ghosh, S., Das, S., Abraham, A., Badr, Y., 2009. Automatic Clustering Using a Synergy of Genetic Algorithm and Multi-objective Differential Evolution. *In* Hybrid Artificial Intelligence Systems (Eds: Corchado, E., Wu, X., Oja, E., Herrero, Á.Baruque, B.). Springer Berlin Heidelberg.

190. Dutta, D., Dutta, P., Sil, J., 2012. Clustering by multi objective genetic algorithm, pp. 548-553. *1st International Conference on Recent Advances in Information Technology (RAIT'12)*, 15-17 March 2012.

191. Suresh, K., Kundu, D., Ghosh, S., Das, S., Abraham, A., 2009. Automatic clustering with multi-objective Differential Evolution algorithms, pp. 2590-2597. *IEEE Congress on Evolutionary Computation (CEC'2009)*, 18-21 May 2009.

192. Suresh, K., Kundu, D., Ghosh, S., Das, S., Abraham, A., 2009. Multi-Objective Differential Evolution for Automatic Clustering with Application to Micro-Array Data Analysis. **Sensors, 9** (5): 3981-4004.

193. Robič, T., Filipič, B., 2005. DEMO: Differential Evolution for Multiobjective Optimization. *In* Evolutionary Multi-Criterion Optimization (Eds: Coello Coello, C., Hernández Aguirre, A.Zitzler, E.). Springer Berlin Heidelberg.

194. Ripon, K. S. N., Chi-Ho, T., Sam, K., Man-Ki, I., 2006. Multi-Objective Evolutionary Clustering using Variable-Length Real Jumping Genes Genetic Algorithm, pp. 1200-1203. *18th International Conference on Pattern Recognition (ICPR'2006),* Vancouver, BC.

195. Saha, S., Bandyopadhyay, S., 2009. A new multiobjective simulated annealing based clustering technique using symmetry. **Pattern Recognition Letters, 30** (15): 1392-1403.

196. Bandyopadhyay, S., Saha, S., Maulik, U., Deb, K., 2008. A Simulated Annealing-Based Multiobjective Optimization Algorithm: AMOSA. **IEEE Transactions on Evolutionary Computation, 12** (3): 269-283.

197. Maulik, U., Bandyopadhyay, S., 2000. Genetic algorithm based clustering technique. **Pattern Recognition, 33**: 1455–1465.

198. Saha, S., Bandyopadhyay, S., 2010. A new multiobjective clustering technique based on the concepts of stability and symmetry. **Knowledge and Information Systems, 23** (1): 1-27.

199. Saha, S., Bandyopadhyay, S., 2010. A symmetry based multiobjective clustering technique for automatic evolution of clusters. **Pattern Recognition, 43** (3): 738-751.

200. Saha, S., Bandyopadhyay, S., 2011. Automatic MR brain image segmentation using a multiseed based multiobjective clustering approach. **Applied Intelligence, 35** (3): 411-427.

201. Saha, S., Bandyopadhyay, S., 2013. A generalized automatic clustering algorithm in a multiobjective framework. **Applied Soft Computing, 13** (1): 89-108.

202. Saha, S., Bandyopadhyay, S., 2012. Some connectivity based cluster validity indices. **Applied Soft Computing, 12** (5): 1555-1565.

203. Wikaisuksakul, S., 2014. A multi-objective genetic algorithm with fuzzy c-means for automatic data clustering. **Applied Soft Computing, 24**: 679-691.

204. Mukhopadhyay, A., Maulik, U., Bandyopadhyay, S., 2013. An Interactive Approach to Multiobjective Clustering of Gene Expression Patterns. **IEEE Transactions on Biomedical Engineering, 60** (1): 35-41.

205. Yanfei, Z., Shuai, Z., Liangpei, Z., 2013. Automatic Fuzzy Clustering Based on Adaptive Multi-Objective Differential Evolution for Remote Sensing Imagery. **IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 6** (5): 2290-2301.

206. Maulik, U., Bandyopadhyay, S., 2003. Fuzzy partitioning using a real-coded variable-length genetic algorithm for pixel classification. **Geoscience and Remote Sensing, IEEE Transactions on, 41** (5): 1075-1081.

207. Michalewicz, Z., 1996. Genetic Algorithms + Data Structures = Evolution Programs. Springer-Verlag, Berlin.

208. Salman, A., 1999. Linkage Crossover Operator for Genetic Algorithms. School of Syracuse University, USA, PhD.

209. Goldberg, D., 1989. Genetic Algorithms in search, optimization and machine learning. Addison-Wesley.

210. Koza, J., 1992. Genetic Programming: On the Programming of Computers by means of Natural Selection. MIT Press, Cambridge, Massachusetts.

211. Auger, A., 2005. Convergence results for the -SA-ES using the theory of -irreducible Markov chains. **Theoretical Computer Science, 334** (1–3): 35-69.

212. Xin, Y., Yong, L., Guangming, L., 1999. Evolutionary programming made faster. **IEEE Transactions on Evolutionary Computation, 3** (2): 82-102.

213. 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.

214. Dorigo, M., 1992. Optimization Learning and Natural Algorithms. Politecnico Di Milano, Phd Thesis, Italy.

215. Eberhart, R., Kennedy, J., 1995. A new optimizer using particle swarm theory, pp. 39-43. *6th International Symposium on Micro Machine and Human Science,* Nagoya.

216. Karaboga, D., Basturk, B., 2007. A powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm. **Journal of Global Optimization, 39** (3): 459-471.

217. Yang, X.-S., 2008. Nature-Inspired Metaheuristic Algorithms. Luniver Press.

218. Dasgupta, S., Das, S., Abraham, A., Biswas, A., 2009. Adaptive Computational Chemotaxis in Bacterial Foraging Optimization: An Analysis. **IEEE Transactions on Evolutionary Computation, 13** (4): 919-941.

219. Angeline, P., 1998. Using Selection to Improve Particle Swarm Optimization, pp. 84-89. *International Conference on Evolutionary Computation,* Piscataway, New Jersey, USA.

220. Lovberg, M., 2002. Improving Particle Swarm Optimization by Hybridization of Stochastic Search Heuristics and Self Organized Critically. University of Aarhus, Department of Computer Science, Denmark.

221. Kennedy, J., Eberhart, R. C., 1997. A discrete binary version of the particle swarm algorithm, pp. 4104-4108. *IEEE International Conference on Systems, Man, and Cybernetics,* 12-15 Oct 1997.

222. Yuan, X., Nie, H., Su, A., Wang, L., Yuan, Y., 2009. An improved binary particle swarm optimization for unit commitment problem. **Expert Systems with Applications, 36** (4): 8049-8055.

223. Pampara, G., Franken, N., Engelbrecht, A. P., 2005. Combining particle swarm optimisation with angle modulation to solve binary problems, pp. 89-96. *IEEE Congress on Evolutionary Computation (CEC'2005).*

224. Nezamabadi-pour, H., Rostami-shahrbabaki, M., Farsangi, M. M., 2008. Binary Particle Swarm Optimization: challenges and New Solutions. **Journal of**

Computer Society of Iran (CSI) On Computer Science and Engineering (JCSE), 6 (1-A): 21-32.

225. Khanesar, M. A., Teshnehlab, M., Shoorehdeli, M. A., 2007. A novel binary particle swarm optimization, pp. 1-6. *Mediterranean Conference on Control & Automation (MED '07)*, 27-29 June 2007.

226. Yun-Won, J., Jong-Bae, P., Se-Hwan, J., Lee, K. Y., 2010. A New Quantum-Inspired Binary PSO: Application to Unit Commitment Problems for Power Systems. **IEEE Transactions on Power Systems, 25** (3): 1486-1495.

227. Pham, D., Ghanbarzadeh, A., Koc, E., Otri, S., Rahim, S., Zaidi, M., 2005. The bees algorithm. Cardiff University, UK, Manufacturing Engineering Centre, Technical report.

228. Yang, X.-S., 2005. Engineering optimizations via nature-inspired virtual bee algorithms. *In* Artificial intelligence and knowledge engineering applications: a bioinspired approach, Lecture notes in computer science (Eds: Mira, J.lvarez, J.). Springer.

229. Lucic, P., Teodorovic, D., 2001. Bee system: modeling combinatorial optimization transportation engineering problems by swarm intelligence, pp. 441–445. *Preprints of the TRISTAN IV Triennial Symposium on Transportation Analysis,* Sao Miguel, Azores Islands, Portugal.

230. Teodorovic, D., Dell'orco, M., 2005. Bee colony optimization - a cooperative learning approach to complex transportation problems, pp. 51–60. *16th mini-EURO Conference on Advanced OR and AI Methods in Transportation*.

231. Karaboga, D., Basturk, B., 2008. On the performance of artificial bee colony (ABC) algorithm. **Applied Soft Computing, 8** (1): 687-697.

232. Akay, B., Karaboga, D., 2012. A modified artificial bee colony algorithm for real-parameter optimization. **Information Sciences, 192**: 120-142.

233. Jadhav, H. T., Sharma, U., Patel, J., Roy, R., 2012. Gbest guided artificial bee colony algorithm for emission minimization incorporating wind power, pp. 1064-1069. *11th International Conference on Environment and Electrical Engineering (EEEIC'12),* 18-25 May 2012.

234. Pampara, G., Engelbrecht, A. P., 2011. Binary artificial bee colony optimization, pp. 1-8. *IEEE Symposium on Swarm Intelligence (SIS),* 11-15 April 2011.

235.Kashan, M. H., Nahavandi, N., Kashan, A. H., 2012. DisABC: A new artificial bee colony algorithm for binary optimization. **Appl. Soft Comput., 12** (1): 342-352.

236.Kiran, M. S., Gunduz, M., 2013. XOR-based artificial bee colony algorithm for binary optimization. **Turkish Journal of Electrical Engineering & Computer Sciences, 21**: 2307-2328.

237.Wei, L., Ben, N., Hanning, C., 2012. Binary artificial bee colony algorithm for solving 0-1 knapsack problem. **Advances in Information Sciences and Service Sciences, 4** (22): 464-470.

238.Pampara, G., 2012. Angle modulated population based algorithms to solve binary problems. University of Pretoria, Faculty of Engineering, Built Environment and Information Technology, MSc dissertation, Pretoria.

239.Choi, S. S., Cha, S. H., Tappert, C., 2010. A Survey of Binary Similarity and Distance Measures. **Journal on Systemics, Cybernetics and Informatics, 8** (1): 43-48.

240.Jaccard, P., 1912. The Distribution of the Flora in the Alpine Zone. **New Phytologist, 11** (2): 37-50.

241.Soille, P., 2002. On morphological operators based on rank filters. **Pattern Recognition, 35** (2): 527-535.

242.Hancer, E., Ozturk, C., Karaboga, D., 2013. Extraction of Brain Tumors from MRI Images with Artificial Bee Colony based Segmentation Methodology, pp. 516-520. *ELECO'2013,* Bursa, Turkiye.

243.Hancer, E., Ozturk, C., Karaboga, D., 2012. Artificial Bee Colony Based Image Clustering Method, pp. 1-5. *IEEE Congress on Evolutionary Computation (CEC'2012),* Brisbane, Australia.

244.Omran, M., Engelbrecht, A., Salman, A., 2005. Particle swarm optimization method for image clustering. **International Journal of Pattern Recognition and Artificial Intelligence, 19** (3): 297–322.

245.Omran, M., Engelbrecht, A., Salman, A., 2006. Particle Swarm Optimization for Pattern Recognition and Image Processing. *In* Swarm Intelligence and Data Mining (Eds: A. Abraham, C. G. a. V. R.). Springer-Verlag, SCI Series 'Studies in Computational Intelligence'.

246. Ozturk, C., Hancer, E., Karaboga, D., 2015. Improved clustering criterion for image clustering with artificial bee colony algorithm. **Pattern Analysis and Applications, 18** (3): 587-599.

247. Ozturk, C., Hancer, E., Karaboga, D., 2014. Color Quantization: A Short Review and An Application with Artificial Bee Colony Algorithm. **Informatica, 25** (3): 485-503.

248. M. Savic, Z. Peric, Dincic, M., 2012. An Algorithm for Grayscale Images Compression based on the forward Adaptive Quantizer Designed for Signals with Discrete Amplitudes. **Electronics and Electrical Engineering, 118** (2): 13-16.

249. Gao, W., Liu, S., Huang, L., 2012. A global best artificial bee colony algorithm for global optimization. **Journal of Computational and Applied Mathematics, 236** (11): 2741-2753.

250. Zhu, G., Kwong, S., 2010. Gbest-guided artificial bee colony algorithm for numerical function optimization. **Applied Mathematics and Computation, 217** (7): 3166-3173.

251. Tuba, M., Bacanin, N., Stanarevic, N., 2011. Guided artificial bee colony algorithm, pp. 398-403. *5th European conference on European computing conference,* Paris, France.

252. Abro, A. G., Mohamad-Saleh, J., 2014. Multiple-global-best guided artificial bee colony algorithm for induction motor parameter estimation. **Turkish Journal of Electrical Engineering & Computer Sciences, 22**: 620-636.

253. Zhang, Y., Zeng, P., Wang, Y., Zhu, B., Kuang, F., 2012. Linear Weighted Gbest-Guided Artificial Bee Colony Algorithm, pp. 155-159. *Fifth International Symposium on Computational Intelligence and Design (ISCID'2012)*, 28-29 Oct. 2012.

254. Huo, Y., Zhuang, Y., Gu, J., Ni, S., Xue, Y., 2015. Discrete gbest-guided artificial bee colony algorithm for cloud service composition. **Applied Intelligence, 42** (4): 661-678.

255. Shah, H., Herawan, T., Ghazali, R., Naseem, R., Aziz, M., Abawajy, J., 2014. An Improved Gbest Guided Artificial Bee Colony (IGGABC) Algorithm for Classification and Prediction Tasks. *In* Neural Information Processing (Eds:

Loo, C., Yap, K., Wong, K., Teoh, A.Huang, K.). Springer International Publishing.

256. Ozturk, C., Hancer, E., Karaboga, D., 2014. Küresel En İyi Yapay Arı Koloni Algoritması ile Otomatik Kümeleme. **Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi, 29** (4): 677-687.

257. Turi, R. H., 2001. Clustering-Based Colour Image Segmentation. Monash University, PhD Thesis, Australia.

258. Bache, K., Lichman, M., 2013. UCI Machine Learning Repository. (Web Page: http://archive.ics.uci.edu/ml).

259. Yew-Soon, O., Meng-Hiot, L., Xianshun, C., 2010. Memetic Computation-Past, Present & Future [Research Frontier]. **IEEE Computational Intelligence Magazine, 5** (2): 24-31.

260. Neri, F., Cotta, C., 2012. Memetic algorithms and memetic computing optimization: A literature review. **Swarm and Evolutionary Computation, 2**: 1-14.

261. Ozturk, C., Hancer, E., Karaboga, D., 2015. A novel binary artificial bee colony algorithm based on genetic operators. **Information Sciences, 297**: 154-170.

262. Ozturk, C., Hancer, E., Karaboga, D., 2015. Dynamic clustering with improved binary artificial bee colony algorithm. **Applied Soft Computing, 28**: 69-80.

263. Kuncheva, L. I., Bezdek, J. C., 1998. Nearest prototype classification: clustering, genetic algorithms, or random search? **IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 28** (1): 160-164.

264. Mirjalili, S., Lewis, A., 2013. S-shaped versus V-shaped transfer functions for binary Particle Swarm Optimization. **Swarm and Evolutionary Computation, 9**: 1-14.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı, Soyadı: Emrah HANÇER

Uyruğu: Türkiye (T.C.)

Doğum Tarihi ve Yeri: 1986, Ankara

Tel: +90 352 437 49 01/32583

Email: emrahhancer@erciyes.edu.tr

Adresi: Erciyes Üniversitesi (ERÜ), Mühendislik Fakültesi,
Bilgisayar Mühendisliği Bölümü, 38039, Melikgazi/KAYSERİ.

EĞİTİM

Derece	Kurum	Mez. Yılı
Yüksek Lisans	ERÜ Fen Bil. Ens. Bilgisayar Müh. Ab.D. , Kayseri	2012
Yüksek Lisans	Ankara Üniversitesi Fen. Bil. Ens. Bilgisayar Müh. ABD, Ankara	2011
Lisans	Çankaya Üniversitesi Fen-Edeb. Fak. Matematik-Bilgisayar Bilimleri, Ankara	2009

İŞ DENEYİMLERİ

Yıl	Kurum	Görev
2011- Halen	ERÜ Mühendislik Fak. Bilgisayar Müh. Bölümü, Bilgisayar Yazılım ABD, KAYSERİ	Arş. Gör.
2010-2011	MAKÜ ZTYO Bilgisayar Tek. ve Bilişim Sistemleri	Arş.Gör.

TEZ İLE İLGİLİ YAYINLARI

1. Ozturk, C., Hancer, E., Karaboga, D., 2015. Improved clustering criterion for image clustering with artificial bee colony algorithm. **Pattern Analysis and Applications, 18** (3): 587-599.
2. Ozturk, C., Hancer, E., Karaboga, D., 2015. A novel binary artificial bee colony algorithm based on genetic operators. **Information Sciences, 297**: 154-170.

3. Ozturk, C., Hancer, E., Karaboga, D., 2015. Dynamic clustering with improved binary artificial bee colony algorithm. **Applied Soft Computing, 28**: 69-80.
4. Ozturk, C., Hancer, E., Karaboga, D., 2014. Küresel En İyi Yapay Arı Koloni Algoritması ile Otomatik Kümeleme. **Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi, 29** (4): 677-687.
5. Ozturk, C., Hancer, E., Karaboga, D., 2014. Color Quantization: A Short Review and An Application with Artificial Bee Colony Algorithm. **Informatica, 25** (3): 485-503.

TEZ İLE İLGİLİ BİLDİRİLERİ

1. Hancer, E., Ozturk, C., Karaboga, D., 2013. Extraction of Brain Tumors from MRI Images with Artificial Bee Colony based Segmentation Methodology, pp. 516-520. *ELECO'2013,* Bursa, Turkiye.
2. Hancer, E., Ozturk, C., Karaboga, D., 2012. Artificial Bee Colony Based Image Clustering Method, pp. 1-5. *IEEE Congress on Evolutionary Computation (CEC'2012),* Brisbane, Australia.