ÇOK GÖRÜNÜMLÜ KÜMELEME İÇİN KARINCA KOLONİ OPTİMİZASYONU TEMELLİ BİR YAKLAŞIM

Melike ERTÜRK

T.C.
BURSA ULUDAĞ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ÇOK GÖRÜNÜMLÜ KÜMELEME İÇİN KARINCA KOLONİ OPTİMİZASYONU TEMELLİ BİR YAKLAŞIM

Melike ERTÜRK
0000-0003-3160-0968

Prof. Dr. Tülin İNKAYA
(Danışman)

YÜKSEK LİSANS TEZİ
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI

BURSA – 2024
Her Hakkı Saklıdır

ÖZET

Yüksek Lisans Tezi

ÇOK GÖRÜNÜMLÜ KÜMELEME İÇİN KARINCA KOLONİ OPTİMİZASYONU TEMELLİ BİR YAKLAŞIM

Melike ERTÜRK

Bursa Uludağ Üniversitesi
Fen Bilimleri Enstitüsü
Endüstri Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Tülin İNKAYA

Günümüzde sensör teknolojilerinin gelişmesi ve nesnelerin internetinin yaygın olarak kullanılması ile veriler farklı kaynaklardan farklı türlerde gelebilmektedir. Bu nedenle, farklı bakış açılarını ve/veya farklı veri kaynaklarını bütünleştiren çok görünümlü veri ile bu veri için geliştirilmiş makine öğrenmesi yöntemleri önem kazanmaktadır. Bu çalışmada, gözetimsiz öğrenme yaklaşımlarından biri olan çok görünümlü kümeleme problemi ele alınmıştır. Bu doğrultuda, çalışmada verideki farklı görünümleri bütünleşik ele alarak aynı kümedeki nesnelerin birbirlerine benzer, farklı kümelerdeki nesnelerin birbirlerinden farklı olacak şekilde gruplandırılması amaçlanmıştır. Bu doğrultuda, problemin çözümü için sürü zekasına dayalı meta-sezgisel yöntemlerden biri olan karınca koloni optimizasyonu (KKO) kullanılmıştır ve KKO temelli çok görünümlü kümeleme yaklaşımı önerilmiştir. Önerilen yaklaşım ile farklı görünümler için görünüm ağırlıkları optimize edilerek her kümedeki en iyi temsilci nesne belirlenmiştir. Önerilen yaklaşımın performansının değerlendirilmesi için karşılaştırmalı veri kümeleri ile deneysel çalışmalar gerçekleştirilmiştir. Yapılan deneysel çalışmalarda önerilen yaklaşımın performansı, klasik tek görünümlü kümeleme algoritmalarının performansı ile karşılaştırılmıştır ve çok görünümlü kümelemenin tamamlayıcı etkisi ile faydası değerlendirilmiştir. Ayrıca, önerilen algoritma sağlık sektöründe anomali tespitinde kullanılmıştır.

Anahtar Kelimeler: Kümeleme, çok görünümlü veri, karınca koloni optimizasyonu, anomali tespiti
2024, vii + 33 sayfa.

ABSTRACT

MSc Thesis

AN ANT COLONY OPTIMIZATION BASED APPROACH FOR MULTI-VIEW CLUSTERING

Melike ERTÜRK

Bursa Uludağ University
Graduate School of Natural and Applied Sciences
Department of Industrial Engineering

Supervisor: Prof. Dr. Tülin İNKAYA

Today, with the development of sensor technologies and the widespread use of the Internet of Things, data can come in different types from different sources. Therefore, multi-view data that integrate different perspectives and/or different data sources and machine learning methods developed for this data gain importance. In this study, the problem of multi-view clustering, which is one of the unsupervised learning approaches, is addressed. Accordingly, the aim of this study is to group objects in the same cluster similar to each other and objects in different clusters different from each other by integrating different views in the data. Accordingly, ant colony optimization (ACO), one of the meta-heuristics based on swarm intelligence, is used to solve the problem and a multi-view clustering approach based on ACO is proposed. With the proposed approach, the best representative object in each cluster is determined by optimizing the appearance weights for different views. To evaluate the performance of the proposed approach, experimental studies were carried out with comparative datasets. In the experimental studies, the performance of the proposed approach is compared with the performance of classical single-view clustering algorithms and the complementary effect and benefit of multi-view clustering is evaluated. In addition, the proposed algorithm has been used in anomaly detection in the health sector.

Key words: Clustering, multi-view data, ant colony optimization, anomaly detection
2024, vii +33 pages.

TEŞEKKÜR

Danışmanlığımı üstlenerek konu seçiminden tezin tamamlanmasına kadar olan tüm süreçlerde beni cesaretlendiren ve tecrübeleriyle bana yol gösteren sayın Prof. Dr. Tülin İnkaya hocama danışmanlığı için teşekkürlerimi sunarım.

Yüksek lisans eğitimim boyunca desteklerini esirgemeyen eşim Semih Ertürk'e teşekkürlerimi sunarım.

Melike ERTÜRK
…/…/…….

İÇİNDEKİLER

Sayfa
ÖZET iv
ABSTRACT v
TEŞEKKÜR vi
SİMGELER ve KISALTMALAR DİZİNİ viii
ŞEKİLLER DİZİNİ ix
ÇİZELGELER DİZİNİ x
1. GİRİŞ…….. 1
2. KAYNAK ARAŞTIRMASI 3
2.1. Çok Görünümlü Kümeleme 3
2.2. Karınca Koloni Optimizasyonu Temelli Kümeleme Çalışmaları 6
3. MATERYAL ve YÖNTEM 8
3.1. Çok Görünümlü Kümeleme 8
3.2. Karınca Kolonisi Optimizasyonu 9
3.3. Çok Görünümlü Kümeleme için Karınca Kolonisi Optimizasyonu 10
4. BULGULAR 15
4.1. Çok Görünümlü Veri Setleri ve Deneysel Koşullar 15
4.2. Parametre Optimizasyonu 17
4.3. Çok Görünümlü KKO Sonuçları 19
4.4. Sonuçların Diğer Yöntemlerle Karşılaştırılması 22
4.5. Sağlık Sektöründe Uygulama 24
5. TARTIŞMA ve SONUÇ 26
KAYNAKLAR 27
EKLER 29
EK 1 Literatür Tablosu 30
EK 2 Algoritma Çalışma Süresi Karşılaştırma Tablosu 32
ÖZGEÇMİŞ 33

SİMGELER ve KISALTMALAR DİZİNİ

Simgeler	Açıklama
A	Karınca sayısı
a_{best}	Elitist karınca sayısı
a	Feromon katsayısı
β	Sezgisel katsayı
C_k	k. kümedeki nesneler
C_l	Diğer kümelerdeki elemanlar
D	Benzerlik matrisi
d_j	Görünüm j için nesneler arası benzerlik matrisi
e	Nesne
J	Amaç fonksiyonu değeri
j	Görünüm indeksi
k^*	En iyi karınca çözümü
K	Küme sayısı
L_k	Karıncanın k'nın oluşturduğu amaç fonksiyonu değeri
m	Medoid
p	Görünüm sayısı
S_1	Feromon matrisine dayalı arama stratejisi
S_2	Seçilen rassal sayıya göre arama stratejisi
SS	Ağırlıklı silhouette kaysayısı
ρ	Feromon buharlaşma oranı
λ_j	Görünüm j'nin ağırlığı
τ_{ij}	i-j arası feromon miktarı

Kısaltmalar	Açıklama
KKO	Karınca Koloni Optimizasyonu

ŞEKİLLER DİZİNİ

Sayfa

Şekil 3.1. Çok Görünümlü Kümeleme için Karınca Kolonisi Optimizasyon……. 14
Şekil 4.1. İterasyona göre karınca keşifleri ve en iyi karınca çözümü…..... 18
Şekil 4.2. İterasyona göre görünüm ağırlıkları ve global en iyi çözüm…….. 18
Şekil 4.3. İterasyona göre global en iyi çözüm……... 19
Şekil 4.4. Görünüm ağırlığı ve J amaç fonksiyon değeri grafiği (MFeature)……. 20
Şekil 4.5. Görünüm ağırlığı ve J amaç fonksiyon değeri grafiği (Water)…….. 20
Şekil 4.6. Görünüm ağırlığı ve J amaç fonksiyon değeri grafiği (3Sources)……. 21
Şekil 4.7. Görünüm ağırlığı ve J amaç fonksiyon değeri grafiği (Image)…….. 21
Şekil 4.8. 207 Numaralı Hastanın EKG Raporu……. 25

ÇİZELGELER DİZİNİ

Sayfa

Çizelge 4.1. Veri setlerinin özellikleri……. 15
Çizelge 4.2. Görünümlerin tanımları……... 16
Çizelge 4.3. Deney tasarımında kullanılan başlangıç parametreleri……... 17
Çizelge 4.4. J amaç fonksiyonuna göre sonuçlar…….…… 19
Çizelge 4.5. SS amaç fonksiyonuna göre sonuçlar ……. 22
Çizelge 4.6. J amaç fonksiyonuyla yöntemlerin karşılaştırılması…… 22

1. GİRİŞ

Günümüzde gelişen teknoloji ile birçok alanda verilerin analizi ve yorumlanmasında veri madenciliği yöntemlerinden yararlanılmaktadır. Bu veriler tek bir kaynaktan geldiği gibi birden fazla kaynaktan farklı türlerde de gelebilir. Örneğin internette paylaşılan görseller aynı zamanda metinsel bir açıklamaya sahiptir. Görsel ve metin bu veri seti için farklı iki görünümü temsil etmektedir. Bir haberin birden fazla haber kuruluşu tarafından yayınlanması, birden fazla sensör verisinin alınması; bir tweet içinde metin, hashtag (#) ve görsellerin beraber kullanılması çok görünümlü veri için örnekler teşkil etmektedir.

Bu tezde, çok görünümlü kümeleme problemi ele alınmıştır. Kümeleme, denetimsiz öğrenme yaklaşımlarından biri olup etiketsiz verilerin belirli bir uzaklık ölçütüne göre gruplara ayrılmasıdır. Kümelenen verilerin birden fazla kaynaktan veya farklı özelliklerde geliyor olması çok görünümlü kümeleme olarak adlandırılmaktadır.

Literatürdeki çok görünümlü kümeleme yöntemleri beş grupta incelenmektedir: birlikte eğitim algoritmaları, çok çekirdekli öğrenme algoritmaları, grafik kümeleme algoritmaları, alt uzay kümeleme algoritmaları, çok görevli çok görünümlü kümeleme algoritmalarıdır (Yang & Wang, 2018). Bunlara ilave olarak geliştirilen algoritmaların performansının iyileştirilmesi amacıyla son yıllarda meta-sezgisel yöntemler öne çıkmaktadır. Meta-sezgisel yöntemlerin güçlü yönleri arasında yerel optimumlardan kaçabilmesi, daha geniş bir çözüm alanını taraması ve arama stratejileri ile yerel iyileştirme prosedürleri arasındaki etkileşimi etkili şekilde düzenleyebilmesi yer almaktadır (Zäpfel, Braune, & Bögl, 2010).

Bu tezde, sürü zekasının dağıtık optimizasyondaki avantajı nedeniyle çok görünümlü veri setleri için karınca koloni optimizasyonu temelli bir kümeleme algoritması geliştirilmiştir. Literatürde çok görünümlü kümeleme probleminin çözümünde meta-sezgisel yöntemler kullanan çalışmalar yer almaktadır (Gusmao & Carvalho, 2019). Bu çalışmalardan farklı olarak, bu tezde çok görünümlü kümeleme probleminin çözümü için karınca koloni optimizasyonu temelli bir yaklaşım geliştirilmiştir.

Bu tezin organizasyonu şöyledir: Tezin ikinci bölümünde çok görünümlü kümeleme probleminin çözümünde sezgisel ve meta-sezgisel yöntemler geliştiren çalışmalar incelenmiştir. Üçüncü bölümünde, karınca koloni optimizasyonu ve önerilen karınca koloni optimizasyonu temelli çok görünümlü kümeleme algoritması açıklanmıştır. Dördüncü bölümünde önerilen algoritmanın deneysel veri setlerindeki performansı analiz edilmiştir ve önerilen algoritma literatürdeki hem tek görünümlü hem de çok görünümlü kümeleme yöntemleriyle karşılaştırılmıştır. Ayrıca önerilen algoritma, sağlık sektöründeki gerçek bir veri setinde anomali tespiti amacıyla kullanılmıştır. Çalışmanın beşinci bölümünde sonuçlar özetlenmiş ve gelecek çalışmalar için önerilerde bulunulmuştur.

2. KAYNAK ARAŞTIRMASI

Bu bölümde kaynak araştırması iki başlık altında incelenmiştir. İlk olarak çok görünümlü kümeleme alanındaki çalışmalar ele alınmıştır. Sonrasında, kümeleme alanında karınca koloni optimizasyonu temelli yaklaşımlar kullanan çalışmalar incelenmiştir. Literatür çalışmalarının özet tablosu Ek 1'de verilmiştir.

2.1. Çok Görünümlü Kümeleme

Chao, Sun ve Bi (2004) çok görünümlü kümeleme alanındaki çalışmaları sınıflandırmıştır ve model tabanlı ile benzerlik tabanlı olmak üzere iki kategoride incelemiştir. Çok görünümlü kümelemenin bilgisayarlı görü, doğal dil işleme, sosyal medya, biyo-informatik ve sağlık bilişimi alanlarındaki çalışmalar hakkında literatür araştırmasına yer verilmiştir. Çok görünümlü ve çok terimli karışımı EM (Multiview mixture-of-multinomials EM - MVMMEM), birlikte regülarizasyonlu ve çok görünümlü spektral kümeleme (Co-regularization multiview spectral clustering - Co-Reg), çok görünümlü düşük dereceli seyrek alt uzay kümeleme (Multiview low-rank sparse subspace clustering - MVLRSSC), ortak negatif olmayan matris çarpanlarına ayırma yoluyla çoklu görünüm kümeleme (Multiview clustering via joint nonnegative matrix factorization - NMF), çekirdek tabanlı ağırlıklı çok görünümlü kümeleme (Kernel based weighted multiview clustering - MVKKM), kanonik korelasyon analizi aracılığıyla çok görünümlü kümeleme (Multiview clustering via canonical correlation analysis - MVCCA), derin matris çarpanlara ayırma yoluyla çok görünümlü kümeleme (Deep matrix factorization- Deep NMF) yöntemleri altı veri setinde uygulanmıştır. Yöntemlerin performansları doğruluk (accuracy), F-score, kesinlik (precision), duyarlılık (recall), ağırlıklandırılmış karşılıklı bilgi (normalized mutual information – NMI), düzeltilmiş Rand indeksi (Adjusted Rand index – ARI) kriterlerine göre karşılaştırılmıştır. Sonuç olarak, çok görünümlü alt uzay kümeleme, çok görünümlü çok çekirdekli kümeleme ve derin matris çarpanlara ayırma yoluyla çok görünümlü kümeleme algoritmalarının iyi performans gösterdiği sonucuna varılmıştır.

Bickel ve Scheffer (2004) tarafından metin verileri için bölümleme ve kümeleme, hiyerarşik çok görünümlü kümeleme algoritmaları geliştirilmiştir. K-ortalamalar ve EM yöntemleri ile kurulmuş modeller sekiz veri setinde tek görünümlü ve çok görünümlü olarak küme entropilerine göre karşılaştırılmıştır. Çalışmanın sonucunda k-ortalamalar ve EM (Expectation maximization) yöntemlerinin çok görünümlü versiyonlarında tek görünümlü versiyonlarına göre iyileşme gözlemlenmiştir. Buna rağmen, birleştirici hiyerarşik çok görünümlü kümeleme için negatif sonuçlar elde edilmiştir.

De Carvalho, Lechevallier ve Melo (2012) tarafından yapılan çalışmada, benzemezlik fonksiyonları ile oluşturulan matrisler kullanılarak katı kümeleme algoritması önerilmiştir. Bu algoritmada kümeler ve küme temsilcilerinin uyumunu ölçen yeterlilik kriterini optimize edecek şekilde görünümler arası ağırlıklandırma kullanılmıştır. Geliştirilen algoritmanın performansı kıyaslama veri setleri ile test edilmiştir.

Sun (2013) tarafından birden çok veri görünümünün avantajları ele alınmıştır. Ayrıca bu alanda yapılan çalışmalar çok görünümün kullanım şekline göre sekiz kategoriye ayrılmıştır, bu kategoriler çok görünümlü boyut azaltma, çok görünümlü yarı denetimli öğrenme, çok görünümlü denetimli öğrenme, çok görünümlü aktif öğrenme, çok görünümlü topluluk öğrenme, çok görünümlü aktarım öğrenme, çok görünümlü kümeleme ve üst düzey taksonomidir.

Xu, Tao ve Xu (2013) tarafından çok görünümlü birlikte eğitim, çoklu çekirdek (kernel) ve alt uzay öğrenimi algoritmaları kullanılarak modeller oluşturulmuştur. Oluşturulan modellerin performansları tek görünümlü Naive Bayes ve destek vektör makineleri ile karşılaştırılmıştır. Performansların değerlendirilmesinde literatürde yer alan temel kıyaslama veri setleri kullanılmıştır.

Wahid, Gao ve Andreae (2014) farklı kümeleme modelleri oluşturmak için çok görünümlü verilerden yararlanmıştır. Sonrasında nihai kümeleme çözümünü oluşturmak amacıyla baskın olmayan sıralama genetik algoritmasına dayanarak kümelerin kombinasyonunu seçen yeni bir kümeleme yöntemi sunmuştur. Önerilen yöntemin performansı, beş farklı algoritma ile çeşitli veri setlerinde karşılaştırılmıştır.

Yang ve Wang (2018) çok görünümlü kümeleme yöntemlerini beş grup altında incelemiştir. Bu gruplar aşağıda verilmiştir.

- Birlikte eğitim algoritmaları: Görünümlerin birbirinden önceki veya birbirinden öğrenilen bilgileri kullanarak farklı görünümlerin kümelenmesini sağlar. Bu strateji yinelemeli olarak gerçekleştirilerek tüm görünümlerin kümeleme sonuçlarındaki tutarlılık en üst seviyeye çıkartılır.
- Çok çekirdekli öğrenme: Bu yöntemde farklı görünümlere karşılık gelen önceden tanımlanmış çekirdekler kullanılır. Daha sonra birleşik bir çekirdek için bireysel çekirdekler doğrusal veya doğrusal olmayan bir şekilde birleştirilir.
- Çok görünümlü grafik kümeleme: Bu yöntemde füzyon grafiği ile ağ bulunmaya çalışılır.
- Çok görünümlü alt uzay kümeleme: Tüm görünümlerin alt uzaylarından birleşik bir temsil öğrenir.
- Çok görevli çok görünümlü kümeleme: Bu kategori, bir görev veya birden çok ilgili görevle her görünümü izler, görevler arasında bilgi aktarır ve kümeleme performansını iyileştirmek için çoklu görev ve çok görünümlü ilişkilerinden yararlanır.

Gusmão ve De Carvalho (2019) tarafından çok görünümlü kümeleme performansının iyileştirilmesi amacıyla hibrit yöntemler ele alınmıştır. Önerilen hibrit modelde çoklu benzemezlik matrisleri girdi olarak alınmıştır ve parçacık sürü optimizasyonunun küresel yakınmasa yeteneği kullanılmıştır. Böylece, sürü zekasından gelen ortak akıl ile kümeleme doğruluğunun arttırılması sağlanmıştır.

Fu, Lin, Vasilakos ve Wang (2020) tarafından yapılan çalışmada çok görünümlü kümeleme algoritmaları grafik tabanlı modeller, alt uzay öğrenme tabanlı modeller ve ikili kod öğrenme tabanlı modeller olarak üç bölümde incelenmiştir. Yedi adet çok görünümlü veri setine bu yöntemler uygulanmıştır. Algoritmaların sonuçları doğruluk, saflık ve ağırlıklandırılmış karşılıklı bilgi kriterlerine göre k-ortalamalar yöntemi ile kıyaslanmıştır.

Zhang, Li, Xu ve Zhang (2021) tarafından derin öğrenmede kullanılan softmin fonksiyonundan ilham alınarak çok görünümlü bulanık k-ortalamalar kümeleme algoritması geliştirilmiştir.

Zheng, Tang, Liu ve Zhu (2023) tarafından matris çarpanlarına ayırma ve k-ortalamalar tabanlı çok görünümlü kümeleme algoritması geliştirilmiştir. Geliştirilen algoritma her görünüm için katsayı matrisini hesaplarken aynı zamanda fikir birliği matrisini oluşturur. Önerilen algoritmanın deneysel veri setleri üzerindeki performansları karşılaştırılmıştır.

2.2. Karınca Koloni Optimizasyonu Temelli Kümeleme Çalışmaları

Shelokar, Jayaraman ve Kulkarni (2004) tarafından yapılan çalışmada *n* tane nesne karınca koloni optimizasyonu kullanılarak *k* adet kümeye ayrılmıştır. Yapılan çalışma diğer metasezgisel yöntemler olan genetik algoritma, tavlama benzetimi ve tabu arama algoritmalarıyla karşılaştırılmıştır.

Handl, Dorigo, ve Knowles (2006) tarafından yapılan çalışmada kümeleme uygulamaları için karınca koloni optimizasyonu temelli bir algoritma geliştirilmiştir. ATTA olarak adlandırılan bu algoritma küme sayısını otomatik olarak belirleyerek kümeleme kalitesini iyileştirmektedir. Algoritma hem sentetik hem de gerçek veriler kullanılarak kümeleme ve topografik haritalama konularında standart tekniklerle karşılaştırılmıştır. Geliştirilen algoritma ile kümeleme konusunda başarılı sonuçlar alınmıştır, ancak topografik haritalama konusunda hedeflenen başarı sağlanamamıştır.

Kao ve Cheng (2006) tarafından yapılan çalışmada kümeleme için karınca koloni optimizasyonu (ACOC) adı verilen yeni bir kümeleme algoritması önerilmiştir. Önerilen algoritma k-ortalamalar ve Shelokar, Jayaraman ve Kulkarni (2004) tarafından önerilen algoritmalar ile karşılaştırılmıştır.

Jafar ve Sivakumar (2010) tarafından karınca koloni sistemi tabanlı kümeleme çalışmalarını inceleyen bir derleme sunulmuştur.

Hung ve Sun (2010) tarafından görüntü segmentasyonu için k-ortalamalar ve karınca koloni optimizasyonu hibrit olarak kullanılmıştır. Başlangıçtaki kümeler, karınca koloni optimizasyonu ile oluşturulduktan sonra kümeleme hatasını en küçüklemek için k-ortalamalar algoritması kullanılmıştır.

İnkaya, Kayalıgil ve Özdemirel (2015) tarafından küme sayısının, şeklinin ve yoğunluğunun bilinmediği kümeleme problemleri için karınca koloni optimizasyonuna dayalı bir kümeleme metodolojisi sunulmuştur. Bu çalışmada çözümün değerlendirilmesi, komşuluk oluşturma ve veri setinin azaltılması gibi çeşitli zorluklar da ele alınmıştır.

Menéndez, Otero ve Camacho (2016) tarafından yapılan çalışmada veriler arasındaki mesafeyi kullanan iki farklı medoid tabanlı karınca koloni optimizasyonu algoritması önerilmiştir. Önerilen algoritmalardan birincisi optimal medoid kümesini belirlemek için karınca koloni optimizasyonunu kullanan bir algoritmadır. Önerilen ikinci algoritma ise küme sayısını otomatik belirleyen bir algoritmadır. Bu algoritmalar sentetik ve gerçek veri setleriyle hem medoid tabanlı hem de klasik kümeleme algoritmalarıyla karşılaştırılmıştır.

Almazini, Ku-Mahamud ve Almazini (2023) tarafından yapılan çalışmada yüksek korelasyonlu özellik kümeleri oluşturmak için karınca koloni optimizasyonunun yerel ve küresel arama yeteneğinden faydalanılmıştır. Önerilen algoritma; destek vektör makineleri, k-en yakın komşu, karar ağacı ve rastgele orman yöntemleriyle karşılaştırılmıştır ve deneysel veri testleri üzerindeki sonuçları verilmiştir.

3. MATERYAL ve YÖNTEM

Bu bölümde önerilen algoritmanın temelini oluşturan çok görünümlü kümeleme ve karınca koloni optimizasyonu yöntemi açıklanmıştır. Ayrıca, önerilen karınca koloni optimizasyonu temelli yaklaşımın adımları verilmiştir.

3.1. Çok Görünümlü Kümeleme

Bu tezde, çok görünümlü kümeleme için görünüm ağırlıklarına sahip çoklu benzemezlik matrislerine dayalı dinamik katı kümeleme algoritmasından ilham alınmıştır (Gusmão ve De Carvalho, 2019). Buna göre, E = $\{e_1, \ldots . e_n\}$ n nesneden oluşan bir veri kümesi olsun ve bu veri kümesinde p adet görünüm olsun. Burada, D_j = $[d_j(e_i, e_0)]$ j = 1,..,p, j görünümündeki benzerlik (uzaklık) matrisi olarak tanımlanır. $d_j(e_i, e_0)$, j görünümündeki uzaklık matrisinde e_i ve e_0 nesneleri arasındaki benzerliği ve λ_j, j görünümünün ağırlığını gösterir. K küme sayısı, C_k k. kümeye atanan nesneler kümesi, m_k ise k. kümenin temsilcisi olsun. Bu koşullar altında çok görünümlü kümeleme problemi için amaç fonksiyonu Denklem 3.1 ile verilmiştir.

$$J = \sum_{k=1}^{K} \sum_{e_i \in C_k} D_{\lambda_k}(e_i, m_k) = \sum_{k=1}^{K} \sum_{e_i \in C_k} \sum_{j=1}^{p} \lambda_j d_j(e_i, m_k) \tag{3.1}$$

$D_\lambda(e_i, m_k)$ e_i nesnesiyle küme temsilcisi olarak atanan m_k arasındaki ağırlıklı uzaklığı temsil etmektedir ve Denklem 3.2 ile hesaplanmaktadır.

$$D_\lambda(e_i, m_k) = \sum_{j=1}^{p} \lambda_j \, d_j(e_i, m_k) \tag{3.2}$$

Her küme için görünüm ağırlığı, $\prod_{j=1}^{p} \lambda_j = 1$ kısıtlaması altında Lagrange çarpanı yöntemi ile Denklem 3.3 ile hesaplanmaktadır.

$$\lambda_j = \frac{\{\prod_{h=1}^{p}[\sum_{e_i \in C_k} d_h(e_i, m_k)]\}^{\frac{1}{p}}}{\sum_{e_i \in C_k} d_j(e_i, m_k)} \quad (3.3)$$

3.2. Karınca Kolonisi Optimizasyonu

Karınca kolonisi optimizasyonu (KKO) ilk olarak 1991 yılında Dorigo, Maniezzo ve Colorni (1991) tarafından önerilmiştir. KKO, sürü zekâsı temelli meta-sezgisel bir algoritma olup karıncaların bıraktığı feromonlar aracılığıyla gıdaya ulaşmak için en kısa yolu bulmalarını taklit eder.

Kombinatoryal optimizasyon problemleri için karınca koloni optimizasyonu yaklaşımı aşağıdaki gibidir (Dorigo, Birattari ve Stützle, 2006).

Adım 1. Başlangıç parametreleri (iterasyon sayısı, karınca sayısı, başlangıç feromon seviyesi, buharlaşma miktarı) belirlenir.
Adım 2. Karıncalar feromon ve sezgisel bilgileri kullanarak rassal karar kuralına göre çözümler oluşturur.
Adım 3. Karıncalar tarafından oluşturulan çözümleri yerel arama ile iyileştirilir. (Bu adım opsiyoneldir.)
Adım 4. Çözümlerin kalitesine göre feromon miktarları güncellenir.
Adım 5. Sona erme koşulları sağlanmıyorsa Adım 2'ye geri dönülür.
Adım 6. En yüksek feromona sahip çözümü gösterilir.

Gezgin satıcı problemi için KKO algoritmasının adımlarında kullanılan formüller şöyledir: *A* adet karıncanın bıraktığı, feromon miktarları her iterasyonda ρ buharlaşma oranında buharlaşarak azalmaktadır. Buna göre, *i* ve *j* noktaları arasında bulunan feromon miktarı τ_{ij} Denklem 3.4 ile hesaplanır.

$$\tau_{ij} = (1-\rho).\tau_{ij} + \rho.\left(\sum_{k=1}^{A} \Delta\tau_{ij}^{k}\right) \quad (3.4)$$

Burada, $\Delta\tau_{ij}^{k}$, k karıncasının i ve j yolu arasına bıraktığı feromon miktarını; L_k ise k karıncasının oluşturduğu amaç fonksiyonu değerini göstermektedir. i ve j yolu arasına bırakılan feromon miktarı Denklem 3.5 ile gösterilmiştir.

$$\Delta\tau_{ij}^{k} = \begin{cases} \frac{1}{L_k}, \textit{k.} \text{ karınca için } i \text{ ve } j \text{ yolu yerel en iyi çözümdür.} \\ 0, \text{diğer} \end{cases} \tag{3.5}$$

Burada, i ve j noktaları arasındaki sezgisel bilgi n_{ij} ve uzaklık d_{ij} ile ifade edildiğinde, n_{ij} değerinin hesaplanması Denklem 3.6 ile verilmiştir.

$$n_{ij} = \frac{1}{d_{ij}} \tag{3.6}$$

Ayrıca α feromon katsayısı, β sezgisel katsayı, η_{ij} i ve j noktaları arasındaki sezgisel değer olarak ifade edildiğinde; P_{ij}^{k}, k karıncasının i-j yolunu kullanma olasılığı Denklem 3.7 ile verilmiştir.

$$P_{ij}^{k} = \frac{\tau_{ij}^{\alpha}.n_{ij}^{\beta}}{\sum((\tau_{il})^{\alpha}.(n_{il})^{\beta})} \tag{3.7}$$

3.3. Çok Görünümlü Kümeleme için Karınca Kolonisi Optimizasyonu

Bu bölümde, çok görünümlü kümeleme için önerilen karınca kolonisi optimizasyonu temelli algoritmanın adımları verilmiştir.

Algoritmada, karınca sayısı a=1...A ve küme sayısı k=1…K olduğunda a karıncası seçilen arama stratejisine göre k kümesi için E kümesindeki nesnelerden m_k^a ile ifade edilen temsilciler (medoidler) seçer ve seçilen medoidlere göre amaç fonksiyonu değeri hesaplanarak feromon matrisi ve görünüm ağırlıkları güncellenir. Amaç fonksiyonu, Denklem 3.8 ile verilmiştir.

$$J_a = \sum_{j=1}^{p} \lambda_j \sum_{k=1}^{K} \sum_{i \in C_k} d_j(e_i, m_k^a) \quad (3.8)$$

Önerilen algoritmanın adımları aşağıdaki verilmiştir:

Adım 1: Başlangıç parametreleri belirlenir. (Karınca sayısı (A), elitist karınca sayısı (a_{best}), buharlaşma oranı (ρ), arama için eşik değeri (q_0), iterasyon sayısı)

Adım 2: Her bir görünümün içinden rassal bir veri örneği (i) seçilir.

Adım 3: q_0 başlangıç parametresi olarak belirlenen arama olasılığıdır, q ise strateji seçimi için [0,1] aralığında düzgün dağılıma uygun rassal bir sayıdır. $P(i,u)$, i nesnesinin medoid olarak seçilip seçilmeme olasılığıdır ve "evet" veya "hayır" olarak tanımlanır: $u \in \{e, h\}$ (Menéndez, Otero ve Camacho, 2016). Buna göre,

a. Rassal bir sayı (q) çekilir ve bu rassal sayı kullanılarak Denklem 3.9'a göre S_1 ve S_2 olmak üzere iki arama stratejisinden biri seçilir.

$$S = \begin{cases} S_1\ ,\ q < q_0 \\ S_2\ ,\ q \geq q_0 \end{cases} \quad (3.9)$$

b. Eğer $q < q_0$ olursa S_1 stratejisine göre feromon matrisinde maksimum değer olan seçim dikkate alınır. Bu durum, Denklem 3.10'da tanımlanmıştır.

$$S_1 = \max_{u \in \{e,h\}} \{\tau(i,u)\} \quad (3.10)$$

c. Eğer $q \geq q_0$ olursa S_2 stratejisine göre feromon matrisinde "evet" olma olasılığı hesaplanarak karar verilir. Bu durum ise Denklem 3.11'de tanımlanmıştır.

$$S_2 = P(i,u) = \frac{\tau(i,u)}{\sum_{l \in \{e,h\}} \tau(i,l)} \quad (3.11)$$

Adım 4: Her karınca için tüm medoidlerin seçilip seçilmediği kontrol edilir. Eğer seçilmediyse seçilene kadar Adım 2 ve Adım 3 tekrar edilir.

Adım 5: Her karınca için veri setindeki nesneler, j görünümünün ağırlığının λ_j ve karıncanın seçtiği medoidin m_k^a olarak ifade edildiği durumda Denklem 3.12'ye göre ağırlıklı uzaklıklara göre en yakın medoide atanır.

$$k^* = argmin_{1\leq k\leq K} \sum_{j=1}^{p} \lambda_j d_j(e, m_k^a) \tag{3.12}$$

Adım 6: Her bir görünüm ağırlığı $\prod_{j=1}^{p} \lambda_j = 1$ kısıtlaması altında ve C_k küme elemanlarını, m_k^a seçilen küme temsilcisini ifade etmesi durumunda Lagrange çarpanı yöntemi ile Denklem 3.13'e göre güncellenir (Gusmão ve De Carvalho, 2019).

$$\lambda_j = \frac{\{\prod_{h=1}^{p}[\sum_{k=1}^{K}\sum_{e_i\in C_k} d_h(e_i, m_k^a)]\}^{\frac{1}{p}}}{\sum_{k=1}^{K}\sum_{e_i\in C_k} d_j(e_i, m_k^a)} \tag{3.13}$$

Adım 7: Her karınca için veri setindeki nesnelerin yeni görünüm ağırlıklarına göre en yakın medoide atanma işlemi yenilenir.

Adım 8: Her karıncanın oluşturduğu çözüm için amaç fonksiyonu değeri hesaplanır. Bu çalışmada J_a ve SS olarak iki farklı amaç fonksiyonu için sonuçlar değerlendirilmiştir. Toplam ağırlıklandırılmış uzaklığı en küçükleyen amaç fonksiyonu J_a Denklem 3.14 ile verilmiştir.

$$J_a = \sum_{j=1}^{p} \lambda_j \sum_{k=1}^{K} \sum_{e_i\in C_k} d_j(e_i, m_k^a) \tag{3.14}$$

Ağırlıklı Silhouette katsayısını en büyükleyen amaç fonksiyonu (SS) Denklem 3.15 ile hesaplanır (Gusmão ve De Carvalho, 2019). $D_\lambda(e_i, C_k)$ e_i nesnesinin atandığı C_k kümesindeki nesnelere olan ağırlıklı uzaklığını, $D_\lambda(e_i, C_l)$ e_i nesnesinin C_l ile ifade edilen diğer küme elemanlarına olan ağırlıklı uzaklığını, $a(e_i)$ e_i nesnesinin C_k kümesindeki nesnelere olan ortalama ağırlıklı uzaklığını ve $b(e_i)$ e_i nesnesinin C_l kümesindeki nesnelere olan ortalama ağırlıklı uzaklığını temsil etmektedir.

$$SS = \frac{1}{n}\sum_{e_i \in E} ss(e_i) \tag{3.15}$$

$$ss(e_i) = \begin{cases} \dfrac{b(e_i) - a(e_i)}{\max\left(b(e_i), a(e_i)\right)} & \text{eğer } b(e_i) \neq a(e_i) \\ 0 & \text{eğer } b(e_i) = a(e_i) \end{cases} \tag{3.16}$$

$$a(e_i) = \frac{D_\lambda(e_i, C_k)}{|C_k|} = \frac{\sum_{j=1}^{p} \lambda_j \sum_{e \in C_k} d_j(e_i, e)}{|C_k|} \tag{3.17}$$

$$b(e_i) = \frac{D_\lambda(e_i, C_l)}{|C_l|} = \frac{\sum_{j=1}^{p} \lambda_j \sum_{e \in C_l} d_j(e_i, e)}{|C_l|} \tag{3.18}$$

Adım 9: a_{best} adet en iyi çözüme ulaşmış elitist karınca seçilir.

Adım 10: Feromon matrisi a_{best} adet elitist karınca çözümüne göre güncellenir. Herhangi bir *t* anında *i* nesnesi için feromon matrisindeki feromon miktarı $u \in \{y, n\}$ koşulu altında $\tau_t(i,u)$ ile; feromon buharlaşma oranı ρ ile ifade edilmektedir.

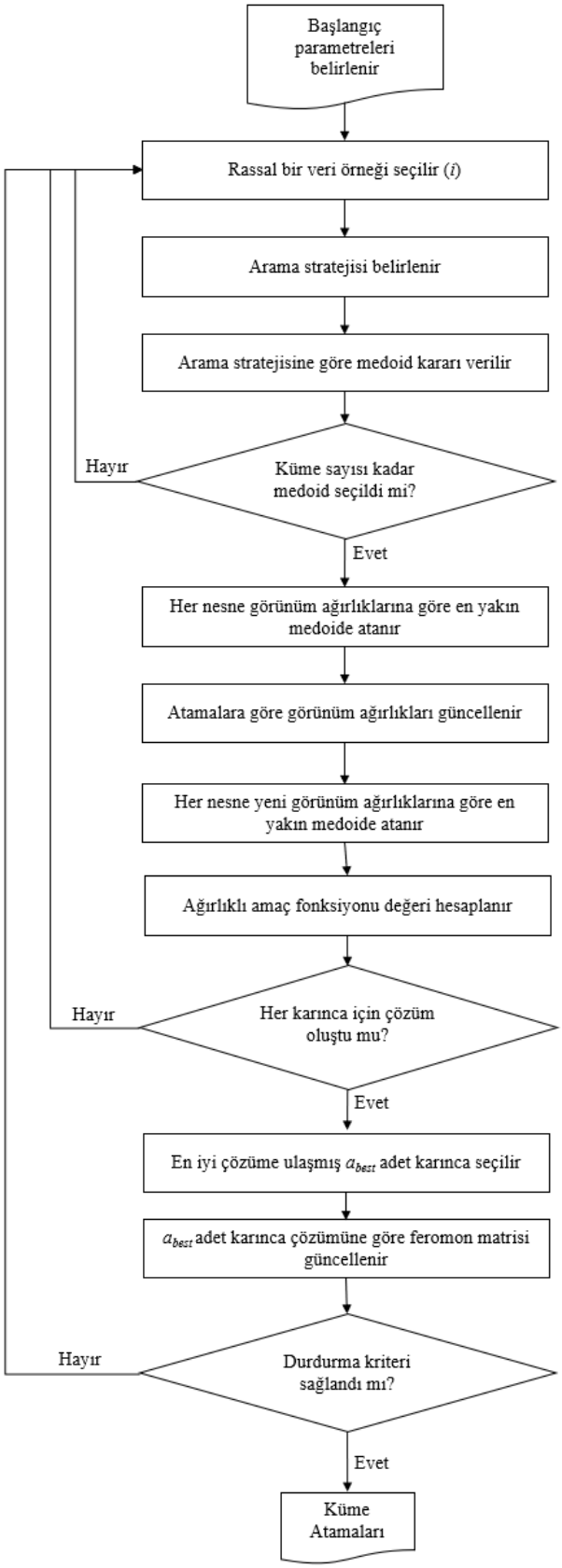
$$\tau_{t+1}(i,u) = (1-\rho).\tau_t(i,u) + \sum_{k=1}^{a_{best}} \Delta\tau_t\,(i,u) \tag{3.19}$$

$$\Delta\tau_t(i,u) = \frac{1}{J_a} \tag{3.20}$$

Adım 11: Her görünümdeki işlemler Adım 2'den itibaren tekrar edilir.

Adım 12: Durdurma kriteri olarak seçilen iterasyon sayısı tamamlanana kadar Adım 2 ile Adım 12 arasındaki adımlar tekrar edilir.

Önerilen algoritma Şekil 3.1'de gösterilmiştir.



Şekil 3.1. Çok Görünümlü Kümeleme için Karınca Kolonisi Optimizasyonu

4. BULGULAR

Bu bölümde, deneysel veri setleri kullanılarak çok görünümlü kümeleme için önerilen karınca koloni optimizasyonuna ilişkin performans değerlendirmeleri paylaşılmıştır ve önerilen yaklaşımın literatürdeki diğer yöntemlerle karşılaştırılması yapılmıştır. Bölüm 4.1'de deneysel koşullar ve veri setleri açıklanmıştır. Bölüm 4.2'de önerilen Çok görünümlü KKO algoritmasının başlangıç parametreleri seçiminden bahsedilmiştir. Bölüm 4.3'te ele alınan amaç fonksiyonları için sonuçlar paylaşılmıştır. Bölüm 4.4'te önerilen yöntem, tek görünümlü KKO, tek görünümlü k-medoid ve çok görünümü k-medoid yöntemleriyle karşılaştırılmıştır. Son olarak, Bölüm 4.5'te önerilen yaklaşımın sağlık sektöründe anomali tespiti amacıyla kullanımı açıklanmıştır.

4.1. Çok Görünümlü Veri Setleri ve Deneysel Koşullar

Bu tezde, performansın değerlendirilmesinde literatürde yer alan kıyaslama veri setleri kullanılmıştır. Kullanılan dört adet veri setinin özellikleri Çizelge 4.1'de verilmiştir ve bu veri setlerinin özellikleri aşağıda açıklanmıştır.

Çizelge 4.1. Veri setlerinin özellikleri

Veri Seti	Görünüm Sayısı	Küme Sayısı	Nesne Sayısı	Öznitelik Sayısı
Image Segmentation (Brodley, 1990)	2	7	2 310	19
Multiple Features (Duin, 1998)	6	10	2 000	649
Water Treatment Plant (Poch, 1993)	4	13	527	38
3-Sources (Greene, 2009)	3	6	169	10 259

Image veri seti UCI Machine Learning Repository'den alınmıştır ve yedi dış mekân görüntüsü içeren bir veri kümesinden rassal olarak seçilen 2 310 nesneden oluşmaktadır. Veride iki görünüm ve toplam 19 adet öznitelik vardır. Bu veri seti için görünümler Çizelge 4.2'de verilmiştir (Brodley, 1990).

Multiple Features veri seti UCI Machine Learning Repository'den alınmıştır ve 10 kümeye ayrılmış el yazısı verileri içeren bir veri kümesinden rassal olarak seçilen 2 000 nesneden oluşmaktadır. Veride altı görünüm ve toplam 649 adet öznitelik vardır. Bu veri seti için görünümler Çizelge 4.2'de verilmiştir (Duin, 1998).

Water Treatment Plant veri seti UCI Machine Learning Repository'den alınmıştır ve 13 kümeye ayrılmış kentsel bir atık su arıtma tesisindeki sensörlerin günlük ölçümlerinden rassal olarak seçilen 527 nesneden oluşmaktadır. Veride dört görünüm ve toplam 38 adet öznitelik vardır. Bu veri seti için görünümler Çizelge 4.2'de verilmiştir (Poch, 1993).

3-Sources veri seti ML Resources'den alınmıştır ve altı farklı haber başlığında incelenen üç çevrimiçi haber kaynağından (BBC, Reuters ve The Guardian) rassal olarak seçilen 169 adet haberden oluşmaktadır. Veride üç görünüm ve toplam 10 259 adet öznitelik vardır. Bu veri seti için görünümler Çizelge 4.2'de verilmiştir (Greene, 2009).

Çizelge 4.2. Görünümlerin tanımları

Veri Setleri	Görünüm	Görünüm Tanımları	Öznitelik Sayısı
Image Segmentation	Shape	Şekil bilgileri	9
	RGB	RGB değerleri	10
Multiple Features	mfeat-fac	Profil korelasyonu	216
	mfeat-fou	Fourier katsayısı	76
	mfeat-kar	Karhunen-Love katsayısı	64
	mfeat-mor	Morfolojik özellikleri	6
	mfeat-pix	2x3 pixel sayısı	240
	mfeat-zer	Zernike momenti	47
Water Treatment Plant	Input	Girdi koşulları	22
	Output	Çıktı talepleri	7
	Performance	Performans girdisi talepleri	5
	Global	Global girdi performans talepleri	4
3-Sources	BBC	BBC haberleri	3 561
	Reuters	Reuters haberleri	3 069
	The Guardian	The Guardian haberleri	3 632

Önerilen algoritma Python programlama dili kullanılarak kodlanmıştır ve tüm deneyler 3.00 GHz işlemcili, 32 GB RAM'li bir bilgisayarda gerçekleştirilmiştir.

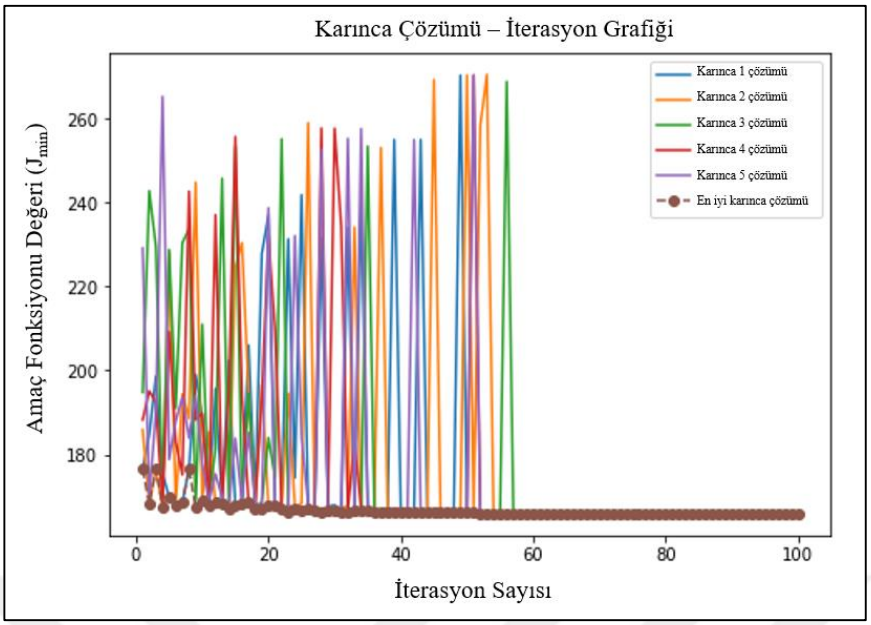
4.2. Parametre Optimizasyonu

Çok görünümlü KKO için başlangıç parametreleri olan karınca sayısı (a), feromon matrisini güncelleyecek olan en iyi çözüme ulaşmış karınca sayısı (a_{best}), feromon buharlaşma oranı (ρ), arama olasılığı için eşik değeri (q_0) ve iterasyon sayısı deneysel tasarım ile belirlenmiştir. Deneysel tasarımda 3Sources veri seti kullanılmıştır. Başlangıç feromon matrisindeki "evet" ve "hayır" olan her iki durum için 0,5 olarak kabul edilmiştir. Deneysel tasarımda kullanılan parametre değerleri ile seçilen parametre değerleri Çizelge 4.3 ile verilmiştir.

Çizelge 4.3. Deney tasarımında kullanılan başlangıç parametreleri

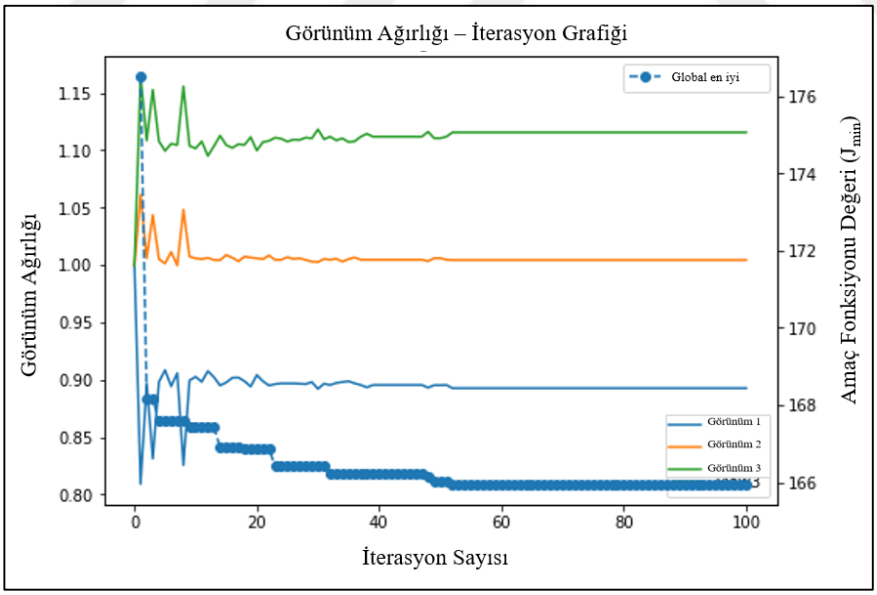
Parametre	Değerler	Seçilen Değer
Karınca Sayısı (a)	5, 10, 15	5
Elitist Karınca Sayısı (a_{best})	1, 4, 5	1
Buharlaşma Oranı (ρ)	0,5; 0,2; 0,05; 0,02; 0,004	0,02
Arama için Eşik Değeri (q_0)	0,0001; 0,001; 0,005	0,001
İterasyon Sayısı	30, 50, 100	100

Parametre değerlerinin amaç fonksiyonu değerini iyileştirmesi ve model çalışma süresi dikkate alınarak; karınca sayısı beş, elitist karınca sayısı bir, buharlaşma oranı 0,02; arama için eşik değeri 0,001 ve iterasyon sayısı 100 olarak seçilmiştir. Şekil 4.1 ile karıncaların keşif alanları ve iterasyon bazında en iyi çözümleri verilmiştir.



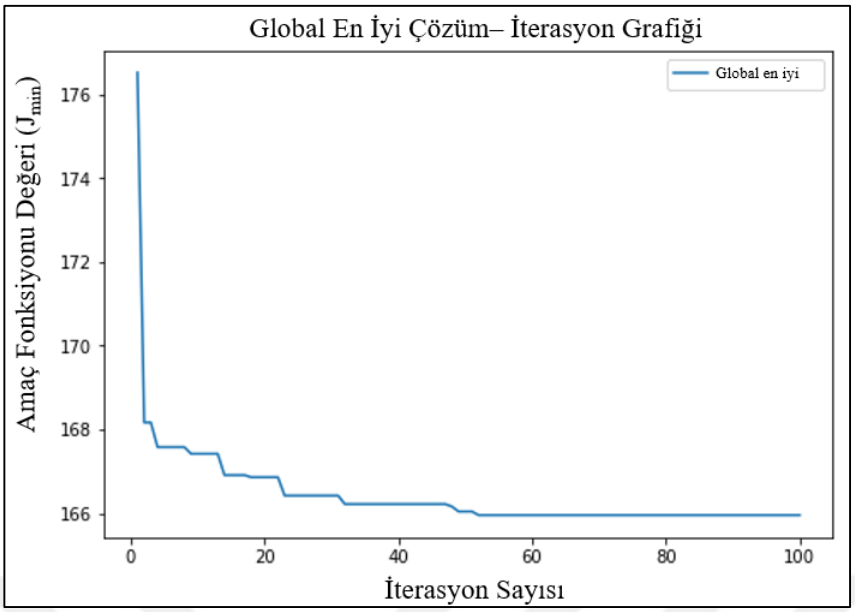
Şekil 4.1. İterasyona göre karınca keşifleri ve en iyi karınca çözümü

Şekil 4.2 ile görünüm ağırlıklarının ve global en iyi çözümün iterasyona göre değişim grafiği verilmiştir.



Şekil 4.2. İterasyona göre görünüm ağırlıkları ve global en iyi çözüm

Şekil 4.3 ile global en iyi çözümün iterasyona göre değişim grafiği verilmiştir.



Şekil 4.3. İterasyona göre global en iyi çözüm

Yapılan analizlerde modelin yakınsaması ve en iyi çözümün iterasyona göre değişimi parametrelerin doğru seçildiğini göstermektedir. Bu nedenle, bu tezdeki tüm deneylerde Çizelge 4.3'teki seçilen parametre değerleri kullanılmıştır.

4.3. Çok Görünümlü KKO Sonuçları

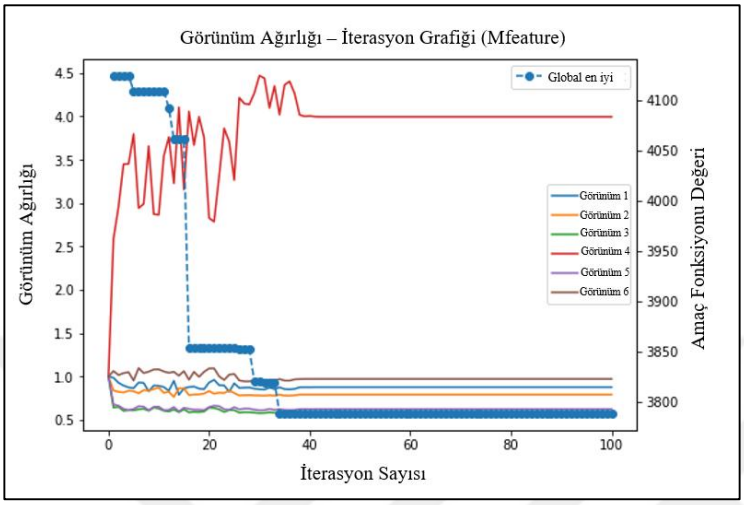
Bu bölümde önerilen algoritmanın veri setleri için, küme içi uzaklığı minimize eden J ve en iyi ağırlıklı Silhouette katsayısını veren *SS* amaç fonksiyonlarına göre sonuçları verilmiştir.

Deneysel veri setlerinin J amaç fonksiyonu kullanılarak hesaplanmış toplam küme için uzaklık değerleri Çizelge 4.4'te verilmiştir.

Çizelge 4.4. J amaç fonksiyonuna göre sonuçlar

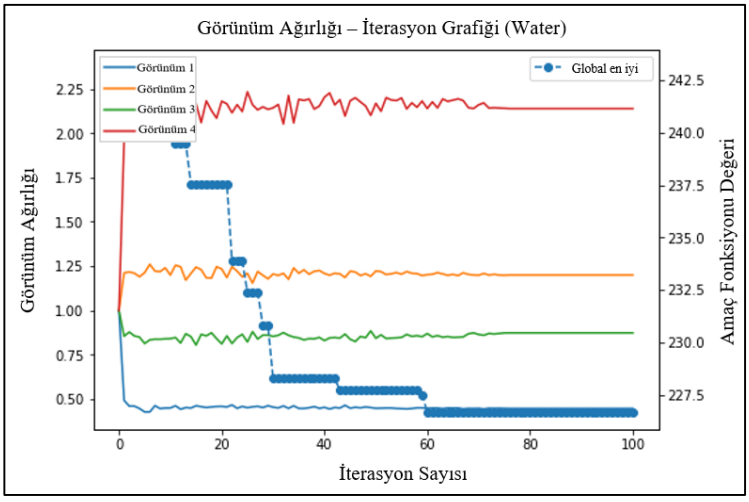
Veri Seti	Toplam Küme içi Uzaklık
Mfeature	3 731
Water	227
3Sources	165
Image	563

Multi-Feature veri seti için iterasyona göre görünüm ağırlıklarındaki ve amaç fonksiyonundaki değişimi gösteren grafik Şekil 4.4 ile verilmiştir.



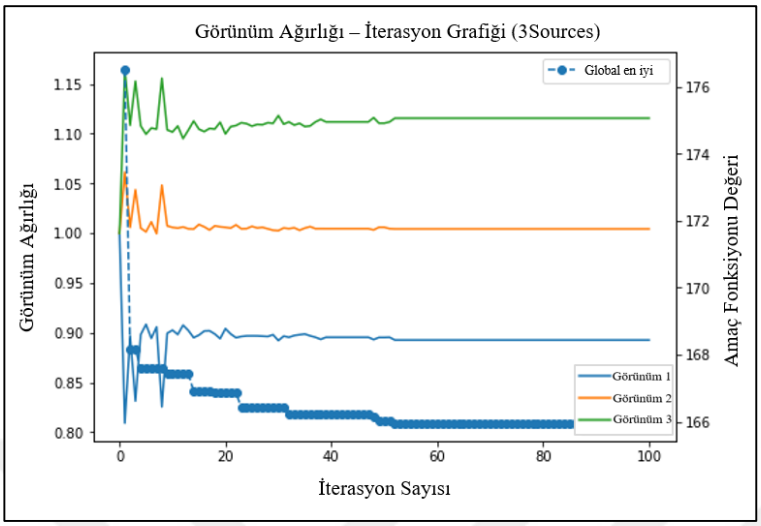
Şekil 4.4. Görünüm ağırlığı ve J amaç fonksiyon değeri grafiği (Multi-Feature)

Water veri seti için iterasyona göre görünüm ağırlıklarındaki ve amaç fonksiyonundaki değişimi gösteren grafik Şekil 4.5 ile verilmiştir.



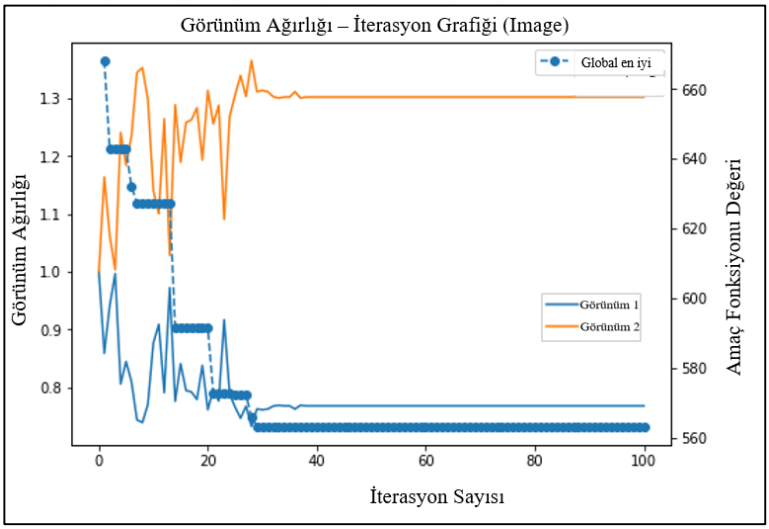
Şekil 4.5. Görünüm ağırlığı ve J amaç fonksiyon değeri grafiği (Water)

3Sources veri seti için iterasyona göre görünüm ağırlıklarındaki ve amaç fonksiyonundaki değişimi gösteren grafik Şekil 4.6 ile verilmiştir.



Şekil 4.6. Görünüm ağırlığı ve J amaç fonksiyon değeri grafiği (3Sources)

Image veri seti için iterasyona göre görünüm ağırlıklarındaki ve amaç fonksiyonundaki değişimi gösteren grafik Şekil 4.7 ile verilmiştir.



Şekil 4.7. Görünüm ağırlığı ve J amaç fonksiyon değeri grafiği (Image)

Deneysel veri setlerinin SS fonksiyonu kullanılarak hesaplanmış ağırlıklı Silhouette katsayısı değerleri Çizelge 4.5'te verilmiştir.

Çizelge 4.5. SS amaç fonksiyonuna göre sonuçlar

Veri Seti	Ağırlıklı Silhouette Katsayısı
Mfeature	0,22
Water	0,07
3Sources	0,08
Image	0,36

4.4. Sonuçların Diğer Yöntemlerle Karşılaştırılması

Bu bölümde önerilen çok görünümlü KKO modelinin performansı; tek görünümlü KKO, tek görünümlü k-medoid ve çok görünümü k-medoid yöntemleriyle karşılaştırmalı olarak verilmiştir. Yöntemlerin her iki amaç fonksiyonu için karşılaştırmalı tablosu Çizelge 4.6'da verilmiştir. Her veri seti içim en iyi performansa sahip olan yöntem koyu renk ile gösterilmiştir.

Çizelge 4.6. J amaç fonksiyonuyla yöntemlerin karşılaştırılması

Veri Seti	Yöntem	Toplam Küme içi Uzaklık
Mfeature	Çok Görünümlü KKO	**3 731**
	Tek Görünümlü KKO	4 481
	Tek Görünümlü K-medoid	4 439
	Çok Görünümlü K-medoid	5 681
Water	Çok Görünümlü KKO	**227**
	Tek Görünümlü KKO	263
	Tek Görünümlü K-medoid	308
	Çok Görünümlü K-medoid	352
3Sources	Çok Görünümlü KKO	**165**
	Tek Görünümlü KKO	167
	Tek Görünümlü K-medoid	172
	Çok Görünümlü K-medoid	180
Image	Çok Görünümlü KKO	**563**
	Tek Görünümlü KKO	618
	Tek Görünümlü K-medoid	743
	Çok Görünümlü K-medoid	638

Tüm veri setlerinde amaç fonksiyonu olarak küme içi ağırlıklı uzaklığı minimize eden J fonksiyonu kullanıldığında önerilen çok görünümlü kümeleme algoritmasının en iyi sonucu verdiği gözlemlenmiştir.

Çizelge 4.7. SS amaç fonksiyonuyla yöntemlerin karşılaştırılması

Veri Seti	Yöntem	Ağırlıklı Silhouette Katsayısı
Mfeature	Çok Görünümlü KKO (SS)	**0,22**
	Tek Görünümlü KKO	0,14
	Tek Görünümlü K-medoid	0,18
	Çok Görünümlü K-medoid	0,00
Water	Çok Görünümlü KKO (SS)	**0,07**
	Tek Görünümlü KKO	0,04
	Tek Görünümlü K-medoid	0,04
	Çok Görünümlü K-medoid	0,01
3Sources	Çok Görünümlü KKO (SS)	0,08
	Tek Görünümlü KKO	0,14
	Tek Görünümlü K-medoid	**0,23**
	Çok Görünümlü K-medoid	0,16
Image	Çok Görünümlü KKO (SS)	**0,36**
	Tek Görünümlü KKO	0,25
	Tek Görünümlü K-medoid	0,33
	Çok Görünümlü K-medoid	0,27

Mfeature, Water, Image veri setlerinde amaç fonksiyonu olarak ağırlıklı Silhouette katsayısını minimize eden SS fonksiyonu kullanıldığında önerilen çok görünümlü kümeleme algoritmasının en iyi sonucu verdiği gözlemlenmiştir.

3Sources veri setinde amaç fonksiyonu olarak SS kullanıldığında ise ağırlıklı Silhouette katsayısı için önerilen algoritma beklenen performansı sağlayamamıştır. Önerilen yaklaşım, SS amaç fonksiyonunda küme içi uzaklığı minimize ederken kümeler arası ayrıklığı maksimize etmeyi amaçlamaktadır. Bu nedenle, bu veri setindeki çok etiketli nesneler kümeler arasındaki ayrıklığı sağlama konusunda modeli zayıflatmıştır.

Karşılaştırılan algoritmaların çalışma süreleri Ek-2'de sunulmuştur. Buna göre, KKO algoritmaları sürü zekasına dayalı oldukları için k-medoid algoritmalarından daha uzun çalışma süresine sahiptir.

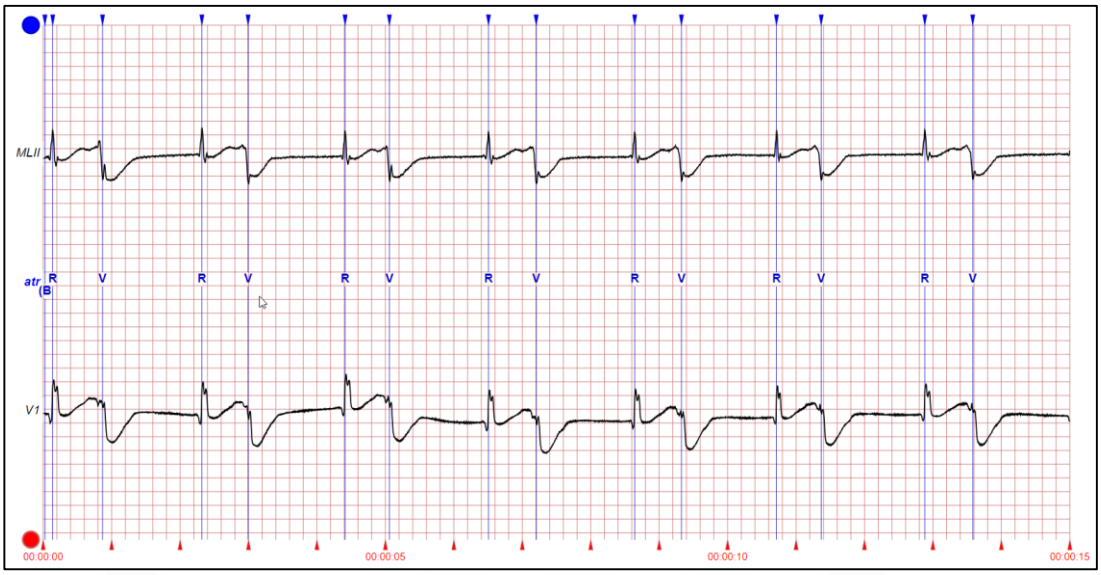
4.5. Sağlık Sektöründe Uygulama

Bu bölümde elektrokardiyografi (EKG) kayıtlarındaki anomali tespiti, önerilen kümeleme yaklaşımı kullanılarak yapılmıştır. Bu kapsamda, gerçek bir veri seti olan MIT-BIH aritmi veri seti kullanılmıştır (Moody & Mark, 2001). Bu veri seti, laboratuvar ortamında 48 hastanın yarım saat boyunca kaydedilmiş, frekansı 360Hz olan EKG sinyallerinden ve bu sinyallere eklenen uzman notlarından oluşmaktadır.

Seçilen veri örnekleri pencere boyutunun 240, kayma oranının %5 olduğu kayan pencereler metoduyla ele alınmıştır. Kayan pencerelerde otokorelasyon katsayıları hesaplanarak önerilen modelde veri girdisi olarak kullanılmıştır (Izakian & Pedrycz, 2013). Hasta verisinde mevcut olan üst sinyal (MLII) ve alt sinyal (V1) değerleri farklı iki görünüm olarak ele alınmıştır. Veriler normal ve anormal olmak üzere iki kümeye ayrılmıştır.

Uzman notlarına göre anomalinin fazla olması sebebiyle 207 numaralı hastanın kayıtlarından sıralı 5 400 veri örneği seçilmiştir. Hastanın uzman notlarına göre seçilen 5 400 veri örneğinde yedi adet Prematüre Ventriküler Kasılma (V) ve yedi adet Sağ Dal Bloğu Atımı (R) anomalisi mevcuttur. Bu hastanın EKG raporu Şekil 4.8 ile verilmiştir.

Önerilen model bu veri setinde uygulandığında örneklem içinde bulunan 14 adet anomali bir kümeye atanmıştır, diğer normal veriler ise başka bir kümeye atanmıştır. Böylece, kümeleme tabanlı anomali tespiti başarılı bir şekilde uygulanmıştır.



Şekil 4.8. 207 Numaralı Hastanın EKG Raporu

5. TARTIŞMA ve SONUÇ

Bu çalışmada küme sayısının bilinmesi ve küme merkezlerinin küme içindeki bir temsilci olması durumunda çok görünümlü veriler için kümeleme problemi ele alınmıştır. Problemin çözümü için farklı görünümlerin görünüm ağırlıklarını sürü zekasına dayalı karınca koloni optimizasyonu kullanarak optimize eden ve buna bağlı olarak her kümedeki en iyi temsilciyi küme merkezi kabul eden bir kümeleme algoritması önerilmiştir.

Önerilen algoritmanın performansı çeşitli veri setleri kullanılarak literatürdeki kümeleme algoritmalarıyla karşılaştırmalı olarak test edilmiştir. Deneysel sonuçlarda önerilen çok görünümlü KKO algoritmasının diğer algoritmalara göre daha iyi performans gösterdiği gözlemlenmiştir. Ayrıca, çok görünümlü algoritmaların tek görünümlü algoritmalara göre kümeleme performansını iyileştirdiği görülmüştür. Diğer yandan, önerilen yaklaşım sürü zekası temelli olduğu için çalışma süresi k-medoid temelli algoritmaların çalışma sürelerinden uzundur. Önerilen algoritma sağlık sektöründe EKG verilerindeki anomalilerin tespitinde uygulanmıştır ve kümeleme tabanlı yaklaşım ile başarılı sonuçlar elde edilmiştir.

Gelecek çalışmalarda, çok görünümlü kümeleme için küme sayısının bilinmediği durum ele alınabilir. Bir diğer gelecek çalışma ise problemin çözümünde genetik algoritmalar, tavlama benzetimi gibi diğer meta-sezgisel yöntemlerin kullanılmasıdır.

KAYNAKLAR

Almazini, H., Almazini, H.F.,Ku-Mahamud K.R, (2023). Enhanced feature clustering method based on ant colony optimization for Feature Selection. *International Journal of Advanced in Intelligent Informatics 9*(1).

Bickel, S., & Scheffer, T. (2004). Multi-View Clustering. *IEEE International Conference on Data Mining.*

Brodley, C. (1990). Image. *Erişim adresi: https://archive.ics.uci.edu/ml/datasets/image+segmentation*.

C. Fraley : A. E. Raftery, A. E. (2007). Bayesian regularization for normal mixture estimation and model-based clustering. *Journal of Classification, 24*(2), 155-181.

Chao, G., Sun, S., & Bi, J. (2021). A Survey on Multiview Clustering. *IEEE Transactions on Artificial Intelligence, 2*(2).

De Carvalho, F. d., Lechevallier, Y., & Melo, F. M. (2019). Clustering of multi-view relational data based on particle swarm optimization. *Expert Systems with Applications, 34*(53)

Dorigo, M., Birattari, M., & Stützle, T. (2006). Ant colony optimization. *IEEE Computational Intelligence Magazine, 1*(4), 28-39. doi:10.1109/MCI.2006.329691

Dorigo, M., Maniezzo, V., & Colorni, A. (1991). *Positive feedback as a search strategy.* Teknik Rapor, Dipartimento di Elettronica, Politecnico di Milano, İtalya.

Duin, R. P. (1998). Multiple Features. *Erişim adresi: https://archive.ics.uci.edu/ml/datasets/Multiple+Features*.

Fu, L., Lin, P., Vasilakos, A. V., & Wang, S. (2020). An overview of recent multi-view clustering. *Neurocomputing, 402*, 148-161.

Greene, D. (2009). 3-Sources. *Erişim adresi: http://mlg.ucd.ie/datasets/3sources.html*.

Gusmão, R. d., & De Carvalho, F. d. (2019). Clustering of multi-view relational data based on particle swarm optimization. *Expert Systems With Applications*(123), 34-53.

Handl, J., Dorigo, M., & Knowles, J. D. (2006). Ant-based clustering and topographic mapping. *Artificial Life, 12*(1), 35-62.

Hastie, T., Tibshirani, R., & Friedman, J. (2017). Phoneme. *Erişim adresi: https://hastie.su.domains/ElemStatLearn/*.

Herrmann, L., & Ultsch, A. (2008). The architecture of ant-based clustering to improve topographic mapping. *Ant Colony Cptimization and Swarm Intelligence International Conference, 5217*, s. 379-386.

Hung, C. C., & Sun, M. (2010). Ant colony optimization for the K-ortalamalar algorithm in image segmentation. *Proceedings of the 48th Annual Southeast Regional Conference*, (s. 15-17). Oxford.

Izakian, H., & Pedrycz, W. (2013). Anomaly Detection in Time Series Data using a Fuzzy C-Means Clustering *Joint IFSA World Congress and NAFIPS Annual Meeting.* Canada.

İnkaya, T., Kayalıgil, S., & Özdemirel, N. E. (2015). Ant colony optimization based clustering methodology. *Applied Soft Computing*(28), 301-311.

Jafar, O. M., & Sivakumar, R. (2010). Ant-based Clustering Algorithms: A Brief. *International Journal of Computer Theory and Engineering, 2*(5), 1793-8201.

Kao, Y., & Cheng, K. (2006). An ACO-based clustering algorithm. *Ant colony optimization and Swarm Intelligence International Conference, 4150*, s. 340-347.

Kaufman, L., & Rousseeuw, P. (2009). *Finding groups in data: An introduction to cluster analysis.* New Jersey.

Kushmerick, N. (1998). Internet Advertisements. *Erişim adresi: https://archive.ics.uci.edu/ml/datasets/internet+advertisements*.

Menéndez, H. D., Otero, F. E., & Camacho, D. (2016). Medoid-based clustering using ant colony optimization. *Swarm Intell*(10), 123-145.

Moody, G.B., Mark, R.G. (2001) The impact of the MIT-BIH Arrhythmia Database. *IEEE Eng in Med and Biol 20*(3):45-50

Nilsback, M. E., & Zisserman, A. (2006). Flowers. *Erişim adresi: https://www.robots.ox.ac.uk/~vgg/data/flowers/17/index.html*.

Poch, M. (1993). Water. *Erişim adresi: https://archive.ics.uci.edu/ml/datasets/water+treatment+plant*.

Shelokar, P. S., Jayaraman, V. K., & Kulkarni, B. D. (2004). An ant colony approach for clustering. *Analytica Chimica Acta*, 187-195.

Shelokar, P., Jayaraman, V., & Kulkarni, B. (2004). An ant colony approach for clustering. *Analytica Chimica Acta*(509), 187-195.

Sun, S. (2013). A survey of multi-view machine learning. *Neural Computing and Applications*(23), 7-8.

Wahid, A., Gao, X., & Andreae, P. (2014). Multi-view clustering of web documents using multi-objective genetic algorithm. *IEEE Congress on Evolutionary Computation.* Beijing.

Xu, C., Tao, D., & Xu, C. (2013, Nisan 20). A survey on multi-view learning. https://arxiv.org/abs/1304.5634 adresinden alındı

Yan, Y., & Wang, H. (2018). Multi-view clustering: A survey. *Big Data Mining and Analytics, 1*(6), 83-107.

Zäpfel, G., Braune, R., & Bögl, M. (2010). *Metaheuristic Search Concepts.* Berlin: Springer. doi:0.1007/978-3-642-11343

Zheng, X., Tang, C., Liu, X., Zhu, E., (2023). Multi-view Clustering via Matrix Factorization Assisted K-means. *Neurocomputing (534)*,45-54.

Zhang, H., Li, X., Xu, Y.,Zhang, R. (2021). Robust Multi-view Fuzzy Clustering via Softmin. *Neurocomputing (458)*, 47-55

Zhao, J., Xie, X., Xu, X., & Sun, S. (2017). Multi-view learning overview: Recent progress and new. *Information Fusion*(38), 43-54.

EKLER

EK 1 Literatür Tablosu Özeti
EK 2 Algoritma Çalışma Süresi Karşılaştırma Tablosu

EK 1 Literatür Tablosu

Yazar	Veri Setleri	Yöntem
Bickel ve Scheffer (2004)	-WebKB -University Web page -Reuters-21578 -TREC-5 -Los Angeles Times -San Jose Mercury -TREC -WebACE project	-Tek görünümlü k-ortalamalar -Çok görünümlü k-ortalamalar -Tek görünümlü EM -Çok görünümlü EM
Shelokar,Jayaraman ve Kulkarni (2004)	-Iris -Wine -Human Thyroid Disease -Ortalama ve varyans verilerek oluşturulmuş 2 veri seti	Karınca Koloni Optimizasyonuna dayalı kümeleme algoritması
Handl,Knowles ve Dorigo (2006)	-Bir adet sentetik veri seti -UCI'den seçilmiş 7 veri seti	Kümeleme için geliştirilmiş karınca koloni optimizasyonu (ATTA)
Kao ve Cheng (2006)	-İki adet sentetik veri seti -Irıs -Vine	Kümeleme için geliştirilmiş karınca koloni optimizasyonu (ACOC)
Hung ve Sun (2010)	Görsel veri seti	K-ortalamalar Kümeleme Karınca Koloni Optimizasyonu
De Carvalhoa, Lechevallier ve Melo (2012)	-Abalone -Image -Iris -Thyroid Gland -Vine	-Benzemezlik matrisine göre yerel ağırlıklandırılmış çok görünümlü kümeleme (MRDCA- RWL) -Benzemezlik matrisine göre global ağırlıklandırılmış çok görünümlü kümeleme (MRDCA- RWG)
Wahid,Gao ve Andreae (2014)	-Vikipedia'dan oluşturulmuş iki tane dökuman veri seti -Açık kaynaklardan oluşturulmuş doküman veri seti	Çok görünümlü ve çok amaçlı evrimsel algoritma ile kümeleme (Multi-view multi-objective evolutionary algorithm - MMOEA)
Inkaya, Kayalıgil ve Özdemirel (2015)	-32 adet veri seti	Karınca Koloni Optimizasyonuna dayalı kümeleme algoritması (ACO-C)
Menéndez, Otero ve Camacho (2016)	-Üç adet sentetik veri seti -UCI'den seçilmiş 20 veri seti -UCR'den seçilmiş 10 veri seti	Medoid bazlı karınca koloni optimizasyon algoritması

EK 1 Literatür Tablosu (Devam)

Yazar	Veri Setleri	Yöntem
Gusmão ve De Carvalho (2019)	-Animals -Corel -Flowers -Image -Internet -Multiple Features -Phoneme -Water -3-Sources	Parçacık sürü optimizasyonu ve benzemezlik matrisine dayalı kümeleme algoritması
Fu,Lin,Vasilakos ve Wangl (2020)	-ALOI -MNIST -NUS-WIDE -MSRC-v1 -Extended Yale-B -Caltech101 -3-Sources Tex	Grafik tabanlı modeller Uzay öğrenme tabanlı modeller İkili kod öğrenme tabanlı modeller
Chao,Sun ve Bi (2021)	-3 Sources -Reuters -Handwritten Digits -COIL20 -YALE -Movies	- Çok görünümlü ve çok terimli karışımı EM -Birlikte regülarizasyonlu ve çok görünümlü spektral kümeleme -Çok görünümlü düşük dereceli seyrek altuzay kümeleme -Negatif olmayan matris çarpanlarına ayırma aracılığıyla çoklu görüntü kümeleme -Çekirdek tabanlı ağırlıklı çoklu görüntülü kümeleme –Kanonik korelasyon analizi aracılığıyla çok görünümlü kümeleme -Derin matris çarpanlara ayırma yoluyla çok görünümlü kümeleme
Zheng, Tang, Liu, Zhu (2023)	-BBC -Reuters -BBCSport -ORL -CiteSeer -HW	- Matris çarpanlarına ayırma ve k-ortalamalar tabanlı çok görünümlü kümeleme algoritması
Almazini, Ku-Mahamud, Almazini (2023	-Wine -Hepatits -Lonosphere -Arrhythmia -Madelon -Colon -Leukemia	-Karınca koloni optimizasyonu temelli özellik seçimi yapan kümeleme algoritması

EK 2 Algoritma Çalışma Süresi Karşılaştırma Tablosu

Veri Seti	Yöntem	Çalışma Süresi (saat)
Mfeature	Çok Görünümlü KKO	2,8
	Tek Görünümlü KKO	2,1
	Tek Görünümlü K-medoid	< 0,01
	Çok Görünümlü K-medoid	< 0,01
Water	Çok Görünümlü KKO	0,6
	Tek Görünümlü KKO	0,3
	Tek Görünümlü K-medoid	< 0,01
	Çok Görünümlü K-medoid	< 0,01
3Sources	Çok Görünümlü KKO	0,13
	Tek Görünümlü KKO	0,11
	Tek Görünümlü K-medoid	< 0,01
	Çok Görünümlü K-medoid	< 0,01
Image	Çok Görünümlü KKO	2
	Tek Görünümlü KKO	1.78
	Tek Görünümlü K-medoid	< 0,01
	Çok Görünümlü K-medoid	< 0,01