Mühendislik Bilimleri ve Tasarım Dergisi
8(5), 173-184, 2020
e-ISSN: 1308-6693
Araştırma Makalesi

Journal of Engineering Sciences and Design
DOI: 10.21923/jesd.828575
Research Article

Özel Sayı: Uluslararası Mühendislikte Yapay Zekâ ve Uygulamalı Matematik Konferansı (UMYMK 2020)
Special Issue: International Conference on Artificial Intelligence and Applied Mathematics in Engineering (ICAIAME 2020)

K-ORTALAMALAR TABANLI EN ETKİLİ META-SEZGİSEL KÜMELEME ALGORİTMASININ ARAŞTIRILMASI

Ömer Köroğlu [1], Hamdi Tolga Kahraman[2*]

[1] Enerji SA, Veri Mühendisliği Müdürlüğü, Veri Mühendisliği Uzmanı, İstanbul, Türkiye
[2] Karadeniz Teknik Üniversitesi, Teknoloji Fakültesi, Yazılım Mühendisliği Bölümü, Trabzon, Türkiye

Anahtar Kelimeler

Kümeleme, K-Ortalamalar Yöntemi, Meta-Sezgisel Arama Algoritması, Meta-Sezgisel Kümeleme Algoritması.

Öz

Kümeleme uygulamalarında en sık kullanılan algoritmalardan biri olan k-ortalamalar yönteminin tatbik edilmesinde karşılaşılan başlıca zorluk, gözlem sayısına bağlı olarak hesaplama karmaşıklığının artması ve problem için küresel en iyi çözüme yakınsayamamadır. Üstelik problem boyutunun ve karmaşıklığının artması halinde k-ortalamalar yönteminin performansı daha da kötüleşmektedir. Tüm bu nedenlerden ötürü klasik k-ortalamalar prosedürü yerine daha hızlı ve başarılı bir kümeleme algoritması geliştirme çalışmaları önem kazanmaktadır. Meta-sezgisel kümeleme (MSK) algoritmaları bu amaçla geliştirilmişlerdir. MSK algoritmaları sahip oldukları arama yetenekleri sayesinde karmaşık kümeleme problemlerinde yerel çözüm tuzaklarından kurtulabilmekte ve küresel çözüme başarılı bir şekilde yakınsayabilmektedirler. Bu makale çalışmasında literatürde yer alan güncel ve güçlü meta-sezgisel arama (MSA) teknikleri kullanılarak MSK algoritmaları geliştirilmekte ve performansları karşılaştırılarak en etkili yöntem araştırılmaktadır. Bu amaçla güncel ve güçlü MSA teknikleri ile k-ortalamalar yöntemi melezlenerek 10 farklı MSK algoritması geliştirilmiştir. Geliştirilen algoritmaların performanslarını ölçmek için 5 farklı kümeleme veri seti kullanılmıştır. Deneysel çalışmalardan elde edilen veriler istatistiksel test yöntemleri kullanılarak analiz edilmiştir. Analiz sonuçları, makalede geliştirilen MSK algoritmaları arasında AGDE tabanlı yöntemin hem yakınsama hızı hem de küresel optimum çözüme yakınsama miktarı açısından kümeleme problemlerinde rakiplerine kıyasla üstün bir performansa sahip olduğunu göstermektedir.

RESEARCH OF MOST EFFECTIVE K-MEANS BASED META HEURISTIC SEARCH ALGORITHM

Keywords

Clustering, K-Means, Meta-Heuristic Search Algorithm, Meta-Heuristic Clustering Algorithm.

Abstract

One of the most frequently used algorithms in clustering analysis, the main difficulty encountered in applying the k-means method is that the calculation complexity increases due to the number of observations and it cannot converge to the global best solution for the problem. Moreover, if the problem size and complexity increases, the performance of the k-means method gets worse. For all these reasons, it is important to develop a faster and successful clustering algorithm instead of the classical k-means procedure. Meta-heuristic clustering (MSK) algorithms have been developed for this purpose. Thanks to their search capabilities, MSK algorithms can get rid of local solution traps in complex clustering problems and successfully converge to the global solution. Therefore, the cluster success of MSK methods is directly affected by the search success of MSA techniques. In this article, MSK methods are developed by using current and powerful MSA techniques in the literature and the most effective method is investigated by comparing the performance of these algorithms. For this purpose, ten different MSK algorithms have been developed by hybridizing the k-means method with current and powerful MSA techniques. Five different clustering data sets were used to measure the performance of the developed algorithms. Data obtained from experimental studies

[*] İlgili yazar / Corresponding author: htolgakahraman@ktu.edu.tr, +90-505-319-2015

were analyzed using statistical test methods. The results of the analysis show that among the MSK algorithms developed in the article, the AGDE-based method has a superior performance compared to its competitors in cluster problems in terms of both the convergence rate and the amount of convergence to the global optimum solution.

Alıntı / Cite

Kahraman, H, Koroglu, O (2020). K-Ortalamalar Tabanlı En Etkili Meta-Sezgisel Kümeleme Algoritmasının Araştırılması, Mühendislik Bilimleri ve Tasarım Dergisi, 8(5), 173-184.

Yazar Kimliği / Author ID (ORCID Number)	Makale Süreci / Article Process	
Hamdi Tolga Kahraman, 0000-0001-9985-6324	**Başvuru Tarihi / Submission Date**	20.11.2020
Ömer Köroğlu, 0000-0003-4456-1320	**Revizyon Tarihi / Revision Date**	07.12.2020
	Kabul Tarihi / Accepted Date	09.12.2020
	Yayım Tarihi / Published Date	29.12.2020

1. Giriş (Introduction)

Kümeleme, nesnelerin benzerliklerine göre gruplandırılması ve benzer niteliklere sahip olanların bir araya getirilmesi işlemidir. Bu amaç doğrultusunda bulanık-c-ortalamalar (Yu vd., 2019; Miao vd., 2020), yoğunluk-tabanlı kümeleme (DBSCAN) (Pandey vd., 2020; Ghazizadeh vd., 2020; Galán, 2019; Chen vd., 2019), destek vektör makineleri (Pouladzadeh vd., 2015; Borkar vd., 2019), öz yinelemeli yapay sinir ağları (Nan vd., 2019; Xu vd., 2020; Jin vd., 2015), k-ortalamalar (Jothi vd., 2019; Wu vd., 2019; Yu vd., 2020), yapay sinir ağları ile melezlenmiş k-ortalamalar (Nithya vd., 2020; Amiri vd., 2016; Arunkumar vd., 2019), bulanık yapay sinir ağı (Pandeeswari ve Kumar, 2016), fuzzy c-ortalamalar (Alam vd., 2019) ve meta-sezgisel kümeleme algoritmaları (Deng vd., 2019; Zhao vd., 2019; Pal vd., 2020; Singh vd., 2019; Kushwaha ve Pant, 2020) geliştirilmiştir. Kümeleme algoritmaları veri madenciliği, makine öğrenmesi ve matematiksel programlama gibi birçok alanda kullanılmaktadır. Kümeleme algoritmaları iki kategoriye ayrılabilir. Bunlar hiyerarşik ve hiyerarşik olmayan kümeleme algoritmalarıdır. K-ortalamalar algoritması hiyerarşik olmayan bir kümeleme algoritmasıdır. K-ortalamalar algoritmasının kümeleme problemlerindeki başarısı arama uzayının karmaşıklığına bağlı olarak değişmektedir. Yerel çözüm tuzaklarının olmadığı arama uzaylarında başarılı olurken konveks olmayan ve çok sayıda yerel çözümlerin bulunduğu arama uzaylarında küresel en iyi çözüme yakınsayamamaktadır. Bu problemlerin üstesinden gelebilmek için modern meta-sezgisel arama teknikleri kullanılarak etkili çözümler geliştirilmeye çalışılmıştır (Huang vd., 2019; Kurada ve Kanadam, 2019; Bonab vd., 2019; Mohamed vd., 2020; Zhou vd., 2019).

Bu çalışmada, güncel MSA teknikleri kullanılarak güçlü meta-sezgisel kümeleme (MSK) algoritmaları geliştirme üzerine çalışmalar yürütülmüştür. Çalışmada önerilen MSK algoritmaları, k-ortalamalar yöntemi ile MSA tekniklerinin melezlenmesi ile geliştirilmiştir. K-ortalamalar yönteminin amacı olan en küçük kare hataya sahip küme merkezlerini keşfetme süreci, meta-sezgisel kümeleme algoritması için çözüm adaylarının uygunluk değerlerini hesaplamada kullanılmıştır. Böylelikle karmaşık arama uzaylarında çok sayıda çözüm adayı ile etkili arama yetenekleri sergileyen MSK yöntemleri geliştirilmiştir. Bu amaçla 10 farklı MSK algoritması geliştirilmiş ve kapsamlı bir deneysel çalışma ile bu algoritmaların kümeleme performansları araştırılmıştır. Deneysel çalışmalardan elde edilen veriler ise parametrik olmayan istatistiksel test yöntemleri (Carrasco vd., 2020; Eftimov vd., 2017) kullanılarak analiz edilmiştir. Analiz sonuçlarına göre bu makalede geliştirilen MSK algoritmalarının büyük bir çoğunluğu k-ortalamalar yöntemine kıyasla üstün bir performans sergilemişlerdir.

Çalışmanın literatüre katkıları aşağıdaki gibi özetlenebilir:

- Güncel meta-sezgisel arama yöntemlerinin tatbik edildiği yeni meta-sezgisel kümeleme algoritmaları geliştirilmiştir. Çalışmada geliştirilen MSK algoritmaları kümeleme konusunda çalışan araştırmacılar tarafından kullanılabilirler.
- Geliştirilen MSK algoritmaları hakkında kapsamlı bir deneysel çalışma yürütülmüş ve 10 farklı MSK yönteminin performansları karşılaştırmalı bir şekilde ve sunulmuştur.
- Deneysel çalışmalardan elde edilen veriler istatistiksel test yöntemleri ile analiz edilerek en etkili MSK yöntemi önerilmiştir.

Makalenin takip eden bölümünde k-ortalamalar yönteminden bahsedilmekte ve MSK algoritmalarının geliştirilme süreci tüm adımlarıyla tanıtılmaktadır.

2. Materyal ve Yöntem (Material and Method)

Bu bölümde k-ortalamalar ve meta-sezgisel arama teknikleri esaslı meta-sezgisel kümeleme algoritmaları hakkında bilgiler verilmektedir. Önce k-ortalamalar yöntemi tanıtılmakta sonrasında melez MSK algoritmalarının geliştirilme süreci açıklanmaktadır.

2.1. K-Ortalamalar Algoritması (K-Means Algorithm)

Kümeleme problemlerinde en çok kullanılan algoritma k-ortalamalar algoritmasıdır. K-ortalamalar gözetimsiz öğrenme yöntemlerinden olup her bir gözlemin sadece bir kümeye ait olabilmesine olanak tanır. Bu yönüyle de bulanık kümelemeden ayrılır ve keskin kümeleme algoritması olarak nitelendirilir. k-ortalamalar algoritması birbirine benzer niteliklerde olan örnek gözlemleri aynı kümeye alır. Bir örnek gözlem en az ve en fazla bir kümede bulunmak zorundadır. Her kümenin merkezini temsil eden ve o kümedeki gözlem vektörlerinin ortalaması olarak hesaplanan "küme merkez vektörü" değeri vardır. Küme sayısı ise kullanıcı tarafından belirlenen ve "k" adı verilen bir parametre ile belirlenir.

K-ortalamalar algoritmasında, kümelerin oluşturduğu hata değerini uygun k değeri ile minimize edilmelidir. Hata değerini hesaplarken en yaygın kullanılan yöntem kare hata değeridir. Kare hata, kümedeki gözlem örneklerinin küme merkezine olan uzaklık değerleri ile ölçülür. Kısacası kümeleme probleminde esas amaç kümeler içindeki gözlemler arasındaki uzaklığın en az (minimum), küme merkezleri arasındaki uzaklığın ise azami (maksimum) olmasını sağlamaktır.

Bu isterler sonucunda algoritma adımları aşağıdaki şekilde izlenmelidir:

1. Küme sayısı olan k değeri seçilir. (Bu değer kullanıcıdan alınan keyfi bir değer veya optimum değer bulunup belirlenebilir.) P={C_1, C_2, C_3,….., C_k}
2. Gözlemler tesadüfi şekilde en az bir ve en fazla bir kümede olmak koşuluyla dağıtılır.
3. Küme merkezleri hesaplanır.
4. Kümedeki her gözlemin küme merkezlerine olan uzaklıkları (küme içi değişmeler) hesaplanır. Bu çalışmada gözlemler arası uzaklık hesaplamaları için Öklid metriği kullanılmıştır:

$$d(Xi, Cj) = \sqrt{\sum_{n=1}^{p} \left(x_{in} - c_{jn}\right)^2} \quad (1)$$

5. Küme içi değişmeler toplanarak kare-hata değeri Eşitlik 2'de verildiği gibi elde edilir. Kare-hatanın amacı, kare-hatayı minimize eden k değerini bulmaktır. Kare-hatanın hesaplanması:

$$E = \sum_{i=1}^{n} \|x_i - c_k\|^2 \mid k = 1, 2, \ldots, K \quad (2)$$

6. Küme merkez değerleri ile gözlemler arası uzaklıklar hesaplanır. Gözlem hangi kümeye yakınsa o kümeye dahil edilir ve kümeler güncellenir.
7. Kümelerde herhangi bir değişiklik olmayana kadar 4., 5. ve 6. Adımlar tekrar edilir.

2.2. Önerilen Yöntem - Meta Sezgisel Kümeleme Algoritması (Proposed Method - Meta Heuristic Clustering Algorithm)

MSK algoritması, meta sezgisel arama algoritması ve kümeleme algoritmasının melezlenmesi ile geliştirilir. Meta sezgisel arama (MSA) algoritmalarında arama süreci, popülasyonun yaratılış evresi ve arama süreci yaşam döngüsü olarak iki genel evreden oluşmaktadır. MSK algoritmalarında küme merkezlerini temsil eden vektörler sürekli değerli oldukları ve k-ortalamalar yönteminde amaç en küçük kare hata değerine sahip küme merkezleri kombinasyonunu bulmak olduğu için popülasyon oluşturulurken sürekli değerli ve kısıtsız optimizasyon problemlerini tanımlayan parametreler kullanılır. Bu parametreler: problem boyutu 'm', çözüm adayları X= [x_1, x_2, x_3,….., x_m] ve çözüm adayları için kısıtlar $a, b \in R$ ve $-\infty < (a, b) < +\infty$ şeklinde (a,b)'den oluşmaktadır. Oluşturulan çözüm adaylarının uygunluk değerlerinin hesaplanması ve birbirleriyle kıyaslanması için bir amaç fonksiyon kullanılır. Buna göre, MSK yöntemi Eşitlikler 3-9 kullanılarak aşağıda tanımlanmaktadır.

$$D = \begin{bmatrix} d_{11} & \cdots & d_{1m} \\ \vdots & \ddots & \vdots \\ d_{r1} & \cdots & d_{rm} \end{bmatrix} \quad (3)$$

$$X = \begin{bmatrix} x_{11} & \cdots & x_{1m} \\ \vdots & \ddots & \vdots \\ x_{k1} & \cdots & x_{km} \end{bmatrix} \quad (4)$$

$$P = \begin{bmatrix} X_1 \\ \vdots \\ X_n \end{bmatrix} \quad (5)$$

$$e = \sum_{j=1}^{k} \sqrt{\sum_{z=1}^{m} \left(d_{iz} - x_{jz}\right)^2} \quad (6)$$

(d_{iz}= *z. kümeye ait i. gözlem*)

$$E = \sqrt{\sum_{i=1}^{k} e_i} \quad (7)$$

$$a = \min(d_1, d_2, \ldots, d_m)$$
$$b = \max(d_1, d_2, \ldots, d_m) \quad (8)$$

$$F = \begin{bmatrix} E_1 \\ \vdots \\ E_n \end{bmatrix} \quad (9)$$

Meta sezgisel kümeleme (MSK) algoritması, MSA algoritması ve k-ortalamalar algoritmasının melezleştirilmesi ile oluşmaktadır. İki algoritma melezleştirilirken kümeleme problemi, MSA algoritmalarının parametreleri kullanılarak bir optimizasyon problemine dönüştürülmelidir. Bu amaçla herhangi bir nesne grubunu temsil eden *D* setinin (Eşitlik 3) *r*-adet gözlemden ve gözlemlerin de *m*-adet nitelikten oluştuğunu varsayarsak; kümeleme problemi için k-ortalamalar tabanlı bir optimizasyon modeli oluşturmak için öncelikle küme merkezleri belirlenir. *k*-adet küme merkezini temsil eden *X*-vektörü Eşitlik 4'de verilmektedir. Buna göre *X*-vektörü kümeleme problemi için bir çözüm adayını temsil etmektedir. Meta-sezgisel bir algoritmada ise popülasyonlar *n*-adet çözüm adayı ile oluşturulurlar. Buna göre Eşitlik-4'de verilen çözüm adayından *n*-adet oluşturularak Eşitlik 5'deki çözüm adayları topluluğu yani popülasyon yaratılır. k-ortalamalar yönteminde amaç, kare hatayı en aza düşürmektir. Dolayısıyla MSK yönteminde de amaç, kümeleme problemi için en küçük kare hata değerine sahip küme merkezleri vektörünü (çözüm adayını) bulmaktır. Kare hata değeri ise küme içi değişimlerin toplamıdır. Buna göre *k*-adet kümenin küme içi değişimlerinin toplamı Eşitlik 6'de verildiği gibi hesaplanır. Küme içi değişim değeri, kümeye ait olan gözlemlerin küme merkezine olan uzaklıklarının toplamıdır. Küme içi değişimler Eşitlik 6'de verildiği gibi toplanarak Eşitlik 7'deki kare hata değeri elde edilir. Dolayısıyla Eşitlikler 6 ve 7, MSK yönteminde çözüm adaylarının uygunluk değerlerinin hesaplanması için kullanılmaktadırlar. Bu denklemlere bağlı kalarak çözüm adaylarının kısıtları Eşitlik 8'de verildiği gibi uygulanmaktadır. Popülasyondaki çözüm adaylarının uygunluk değerini temsil eden *F* vektörü (Eşitlik 9) ise amaç fonksiyon kullanılarak elde edilen kare hata değerleri ile oluşturulmaktadır.

MSA algoritması ve k-ortalamalar algoritmasının melezlenmesi ile oluşturulan MSK algoritmasının sözde kodu aşağıda verilmiştir.

Algoritma1. Meta Sezgisel Kümeleme (MSK) Algoritmasının Sözde Kodu (Pseudo Code of Meta Heuristic Clustering Algorithm)

1. **Başla**
2. *D:* Kümeleme probleminin tanımlanması (Eşitlik 3)
3. *k:* Problemin küme sayısını belirle.
4. *P:* Kısıtlara uygun çözüm adayları topluluğunu rastgele oluştur. (Eşitlik 5
5. **for** *i*=1:n

```
6.          P-topluluğu içerisindeki adaylar için r-adet gözlemi kümelere dağıt.
7.          Çözüm adaylarının küme içi değişmesini hesapla. (Eşitlik 6)
8.          F: Bütün çözüm adayları için kare hata değeri hesapla. (Eşitlik 7)
9.      end
10.     while (amaç fonksiyonu azami çağırma sayısına ulaşılıncaya kadar)
11.         En az kare hata değerine sahip çözüm adayını belirle.
12.         Seçim süreci:
                P-topluluğu içerisinden arama sürecinin yöneleceği k-adet çözüm adayı
                belirle.
13.         Arama süreci:
                Komşuluk araması yap.
                Çeşitlilik araması yap.
14.         Oluşturulan yeni çözüm adayları ile verileri kümelere dağıt.
15.         Yeni çözüm adayının kare hata değerini hesapla.
16.         if (yeni çözüm adayının kare hata değeri eskisinden küçük ise)
17.             Popülasyondaki çözüm adayını güncelle.
18.         end
19.     end
20. end
```

Algoritma 1'de MSK algoritması kullanılarak kümeleme probleminin çözüm aşamaları gösterilmiştir. Çözümün ilk kısmında, kümeleme problemi ve küme sayısı tanımlanarak oluşturulan çözüm adaylarına sahip popülasyon *(P)* oluşturulmuştur. Popülasyon oluşturulduktan sonra her çözüm adayına ait kare hata denklemi (Eşitlik 7) kullanılarak uygunluk değerleri *(F)* bulunmuştur. Böylelikle kümeleme problemi için çözüm adaylarını oluşturma evresi tamamlanır (Adımlar 1-9). Arama süreci yaşam döngüsü evresinde (Adımlar 10-19) ise her MSA algoritmasının kendine özgü yöntemleri ile probleme ait çözüm adaylarının optimum değerlerini keşfetmeye çalışılmaktadır. Algoritmaların birbirine karşı üstünlükleri ve başarıları bu evrede ortaya çıkmaktadır.

Özetle, MSA teknikleri ile k-ortalamalar yönteminin melezlenmesi yoluyla MSK algoritmalarının geliştirilmesi süreci Algoritma 1'de verildiği gibi MSA tekniklerinden ve k-ortalamalar yönteminden bağımsız bir şekilde modüler bir tasarıma uygun olarak gerçekleştirilebilmektedir.

3. Deneysel Ayarlar (Experimental Settings)

Bu çalışmada k-ortalamalar algoritmasının optimum performansı ile 10 adet meta sezgisel algoritmalarla melezlenmiş MSK algoritmaları karşılaştırılmıştır. Bu meta sezgisel algoritmalar: ABC (Karaboga ve Basturk, 2007), AEO (Zhao vd., 2019), AGDE (Mohamed ve Mohamed, 2019), COA (Pierezan ve Coelho, 2018), CS (Yang ve Deb, 2009), GWO (Mirjalili vd., 2014), PSO (Eberhart ve Kennedy, 1995), SOS (Cheng ve Doddy, 2014), FDB SOS(Kahraman vd., 2020) ve SFS (Salimi, 2015) olarak belirlenmiştir.

Belirlenen algoritmalar için maksimum uygunluk değişim sayısı kullanılmıştır. Bu uygunluk değişim sayısı algoritmanın amaç fonksiyonu kaç kez çağırıldığını belirtmektedir. Algoritmaların parametreleri Bölüm 3.3'te ayrıntılı bir şekilde belirtilmiştir. Her problem için maksimum uygunluk değişim sayısı $(Nitelik\ Sayısı + Küme\ Sayısı) \times 100$ olarak belirlenmiştir. Kullanılan her problemin detayları Bölüm 3.2'de belirtilmiştir.

3.1. Veri Setleri (Datasets)

Bu çalışmada kullanılan veri setleri gerçek hayattan elde edilmiş (Iris (Dasarathy, 1980), CMC (Lim vd., 1999), Glass (Jiang ve Zhou, 2004), User Knowledge Modeling (Kahraman vd., 2013)) ve kümeleme algoritmalarının kıyaslanmasında kullanılan (Compound (Zhan, 1971)) verilerden oluşturulmuştur.

Tablo 1. Kullanılan Veri Setlerinin Özellikleri (Properties of Datasets)

Veri Setleri	Nitelik Sayısı	Küme Sayısı	Gözlem Sayısı
Iris Dataset	4	3	150
CMC Dataset	9	3	1473

Glass Dataset	9	6	214
User Knowledge Modeling	5	4	403
Compound Dataset	2	6	399

3.2. Algoritma Parametreleri (Parameters of Algorithm)

Meta-sezgisel kümeleme algoritmalarının geliştirilmesi için literatürde iyi bilinen ve en sık kullanılan üç meta-sezgisel arama yönteminin PSO, ABC ve CS yanı sıra güncel ve güçlü GWO, SFS, AEO, AGDE, COA, SOS, FDB-SOS yöntemleri kullanılmıştır. Bu on MSA yöntemi kullanılarak yöntem bölümünde açıklandığı gibi MSK algoritmaları geliştirilmiştir. Algoritmaların parametre ayarları için tanıtıldıkları çalışmalardaki ayarlar referans alınmıştır.

Tablo 2. MSK Algoritmalarının Parametre Ayarları (Parameter Settings of MHC Algorithms)

ABC		AEO	
Parametreler	Değer	Parametreler	Değer
NP	100	nPop	50
FoodNumber	NP/2	u	randn(dim)
limit	FoodNumber * dim	v	randn(dim)
		C	1/2 * u./abs(v)
AGDE		**COA**	
Parametreler	Değer	Parametreler	Değer
NP	50	n_coy	5
CR_1	[0.05,0.15]	n_packs	20
CR_2	[0.9,1]	p_leave	0.005*n_coy^2
F	[0.1,1]	Pop_total	n_packs*n_coy
		Ps	1/dim
CS		**GWO**	
Parametreler	Değer	Parametreler	Değer
pa	0.25	SearchAgents_no	30
n	25		
nd	dim	a	2 - 2/MaxFE
u	randn(size(X_i))*sigma		
v	randn(size(X_i))	A	2 * a * rand()-a
step	u./abs(v).^(1/beta)		
stepsize	0.01*step.*(X_i-X_{best})	C	2 * rand()
PSO		**SOS**	
Parametreler	Değer	Parametreler	Değer
nPop	50		
c1	2	ecosize	50
c2	2		
w	0.9	BF1	round(1+rand)
VelMax	0.1*(UpperBound-LowerBound)	BF2	round(1+rand)
VelMin	-UpperBound		
FDB-SOS		**SFS**	
Parametreler	Değer	Parametreler	Değer
ecosize	50	nPop	50
BF1	round(1+rand)	u	randn(dim)
BF2	round(1+rand)	v	randn(dim)
		C	1/2 *u./abs(v)

4. Deneysel Sonuçlar (Experimental Results)

Bu bölümde on adet MSK algoritmasının ve klasik k-ortalamalar yönteminin beş farklı veri setindeki kümeleme performansları araştırılmaktadır. Algoritmaların performanslarının istatistiksel olarak analiz edilmesi için her bir deneysel çalışma 21 defa tekrarlanmıştır. Her bir çalıştırmada algoritmaların her bir problem için elde ettikleri

toplam kare hata değerleri kayıt altına alınmıştır. Buna göre algoritmaların her bir veri seti için 21 kez çalışma neticesinde elde ettikleri en iyi, en kötü, ortalama ve standart sapma sonuçları Tablo 3'de verilmektedir.

Tablo 3. 21 Kez Çalıştırılarak Algoritmalardan Elde Edilen Sonuçlar (Results of Algorithms with 21 Runs)

Algoritmalar		Iris	Compound	Glass	UKM	CMC
ABC	**En İyi**	78,94084	3880,6758	339,3702	68,8451	23690,2522
	En Kötü	78,94184	4040,8806	343,4296	68,8958	23690,2727
	Ortalama	78,94105	3969,2884	340,8874	68,8699	23690,2630
	Std. Sapma	0,000324	42,419486	1,288396	0,01728	0,00632250
AEO	**En İyi**	78,94084	3865,9421	336,2712	68,7919	23690,2438
	En Kötü	78,94507	4704,8339	422,7769	72,2899	23691,7605
	Ortalama	78,94326	3919,0307	379,4771	69,6325	23690,3578
	Std. Sapma	0,002090	184,75590	24,98172	0,91773	0,33251366
AGDE	**En İyi**	78,94084	3865,9421	336,0605	68,7919	23690,2378
	En Kötü	78,94084	3865,9421	398,8778	70,3384	23720,5298
	Ortalama	78,94084	3865,9421	363,4494	69,0305	23694,9433
	Std. Sapma	4,26E-14	3,898E-08	18,10030	0,52014	9,68056151
COA	**En İyi**	78,94192	3923,5158	343,7911	68,8364	23690,8149
	En Kötü	79,76430	5049,2355	491,9245	69,3800	23691,1916
	Ortalama	79,01006	4256,3205	395,3357	69,0593	23690,9549
	Std. Sapma	0,173984	284,25971	39,67533	0,14058	0,10723223
CS	**En İyi**	78,94177	3867,2723	373,6944	68,8408	23691,2069
	En Kötü	78,95002	3879,2851	438,8140	68,9842	23707,8165
	Ortalama	78,94406	3869,8151	407,1534	68,8822	23696,6622
	Std. Sapma	0,002084	3,3387587	19,09442	0,03307	5,63505207
GWO	**En İyi**	78,94747	3866,0625	491,2983	68,7923	24749,9917
	En Kötü	144,6420	3867,4981	688,8156	74,9673	25077,5489
	Ortalama	90,17829	3866,3161	583,4453	70,0641	24787,3990
	Std. Sapma	23,42483	0,2860455	49,96469	1,95196	76,1779375
PSO	**En İyi**	78,94084	3865,9421	336,0608	68,7919	23690,2378
	En Kötü	78,94507	4695,2921	443,2596	71,0607	23690,9874
	Ortalama	78,94305	4197,8702	397,3143	69,4947	23690,3654
	Std. Sapma	0,002110	386,39195	35,67926	0,77205	0,24400290
SFS	**En İyi**	79,29446	3909,4574	480,8739	69,1788	24695,6944
	En Kötü	89,86196	4928,0533	678,5885	74,7395	24706,3850
	Ortalama	82,78958	4306,2215	591,5701	71,7666	24700,8560
	Std. Sapma	2,619000	350,60032	53,40081	1,38231	4,12650593
SOS	**En İyi**	78,94830	3982,9071	356,8353	69,8078	23721,0241
	En Kötü	81,44516	4554,5247	589,3014	71,8312	23756,2731
	Ortalama	79,28766	4227,7026	470,0607	70,5061	23730,7609
	Std. Sapma	0,534414	154,79833	56,34905	0,62074	11,5003692
FDB SOS	**En İyi**	78,94300	4005,6696	355,7311	69,4633	23722,6265
	En Kötü	82,51778	4552,9935	585,6527	71,3147	23745,4077
	Ortalama	79,41507	4236,0567	491,4391	70,4368	23732,2518
	Std. Sapma	0,800651	170,99762	47,43432	0,51140	7,19172386
k-ortalamalar	**En İyi**	78,94084	4485,9375	336,2687	68,8902	23705,4414
	En Kötü	78,94507	5453,3334	412,3417	72,8140	23705,4414
	Ortalama	78,94245	5284,6973	368,0197	70,1804	23705,4414
	Std. Sapma	0,002050	340,72445	21,60930	1,01941	0,0000

Tablo 4. Algoritmaların Kullanılan Problemlerde Wilcoxon Sıralı Testler ile Elde Edilen Sonuçlar (Results Obtained by the Wilcoxon Rank Test)

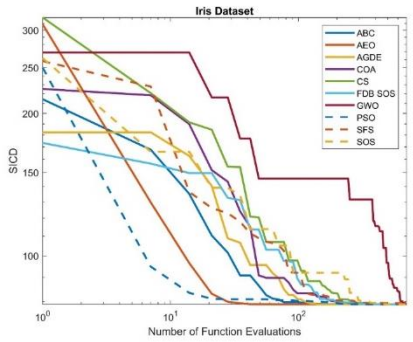
	k-ortalamalar vd.									
	AGDE	ABC	AEO	PSO	CS	COA	GWO	SOS	FDB SOS	SFS
Galibiyet	4	4	3	3	3	3	2	1	1	1
Beraberlik	1	1	1	0	0	1	0	1	1	0
Mağlubiyet	0	0	1	2	2	1	3	3	3	4

K-ortalamalar yönteminin ile MSK algoritmalarının kümeleme performanslarını ikili olarak karşılaştırmak ve deneysel çalışma verilerini analiz etmek için Wilcoxon testi uygulanmıştır. Buna göre 5 veri seti için k-ortalamalar yönteminin rakip MSK algoritmaları ile arasındaki skorlar Tablo 4'de verildiği gibi elde edilmiştir.

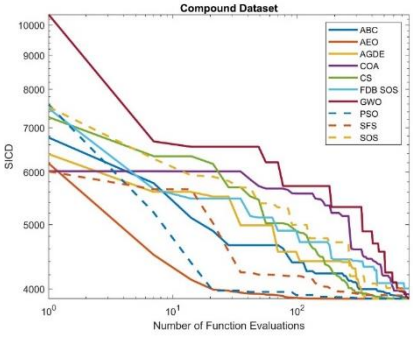
Tablo 5. Friedman Testi ile Elde Edilen Sıralamalar (Ranks Obtained by the Friedman Test)

Algoritma	**Ortalama Sıra**
AGDE	2,262
ABC	3,620
AEO	4,028
PSO	4,143
CS	5,152
COA	6,076
k-ortalamalar (k-means)	6,357
GWO	7,657
SOS	8,4
FDBSOS	8,505
SFS	9,8

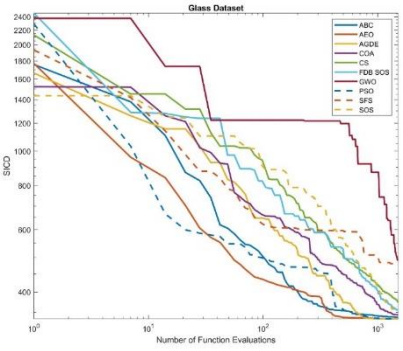
Tablo 5'te verilen sıralamaya göre AGDE esaslı MSK algoritması rakiplerine karşı bariz bir üstünlüğe sahiptir.



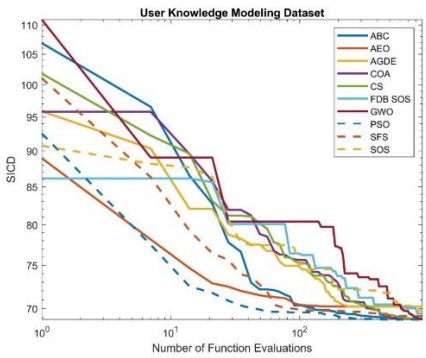
(a) Iris Veri Seti (Iris Dataset)



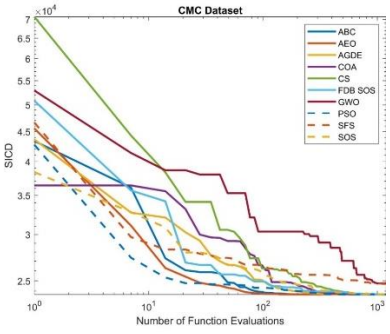
(b) Compound Veri Seti (Compound Dataset)



(c) Glass Veri Seti (Glass Dataset)



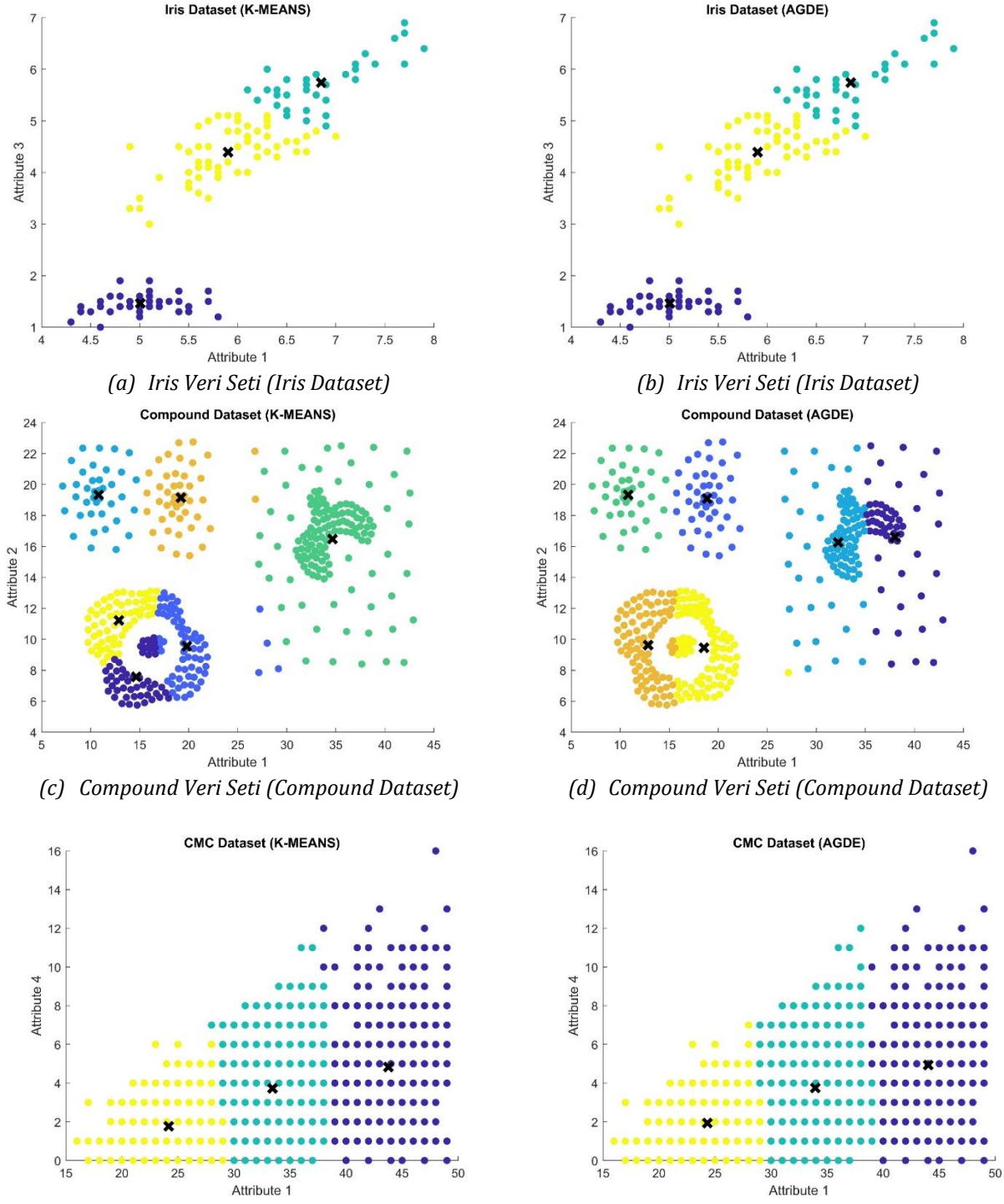
(d) User Knowledge Modeling Veri Seti (User Knowledge Modeling Dataset



(e) CMC Veri Seti (CMC Dataset)

Şekil 1. Algoritmaların İterasyona Göre Kare Hata Değeri Değişim Grafikleri (Square Error Value Change Charts of Algorithms According to Iteration)

MSK algoritmalarının yakınsama performanslarını gözlemlemek için 5 veri seti için algoritmaların arama süreci yaşam döngüsü boyunca elde ettikleri en iyi kare hata değerleri iterasyon sayısına bağlı olarak Şekil 1'de görülen grafikler çizdirilmiştir. Grafikler incelendiğinde beş veri setinde de yakınsama hızı ve miktarı açısından GWO algoritmasının rakip MSK yöntemlerine yenildiği görülmektedir. ABC, AGDE, PSO ve AEO'nun yakınsama performansları birbirine yakındır. Bunun yanında grafiklerden çıkarılması gereken bir sonuçta algoritmalara arama için daha fazla fırsat verilmesi gerektiğidir. Çünkü algoritmaların yakınsama eğrileri henüz sabit bir hata değerine yakınsamamışlar ve oturma eğilimi göstermeye başlamamışlardır. Ancak bu çalışmanın amaçlarından biri de MSK yöntemlerinin yakınsama miktarlarını ve hızlarını gözlemlemektir.



(a) Iris Veri Seti (Iris Dataset) *(b) Iris Veri Seti (Iris Dataset)*

(c) Compound Veri Seti (Compound Dataset) *(d) Compound Veri Seti (Compound Dataset)*

(e) CMC Veri Seti (CMC Dataset) *(f) CMC Veri Seti (CMC Dataset)*

Şekil 2. AGDE ve K-Ortalamalar algoritmalarının nokta dağılımlı grafikleri (Scatter Charts of AGDE and K-Means Algorithms)

Şekil2'de problemlere dair çözümler nokta dağılımlı grafik ile görselleştirilmiştir. Nokta dağılımlı grafikte her nokta problemdeki gözlemi, X işareti ise küme merkezlerini belirtmektedir. Grafikteki her renk bir kümeyi temsil etmektedir. Aynı renkli veriler aynı kümede bulunmaktadır. Şekil 2(a) ve Şekil 2(b) incelendiğinde k-ortalamalar ve AGDE algoritmalarının Iris problemini başarılı bir şekilde çözdüğü görülüyor. Fakat problem boyutu arttıkça kümeleme işlemi karmaşıklaşmaktadır. Şekil 2(c) incelendiğinde turuncu renkli ve mavi renkli verilerin yeşil renkli veriler ile karıştığını ve veri setindeki gözlemlerin 6 kümede düzensiz bir şekilde gruplandığı görülmektedir. Buna rağmen veri setinin 6 kümeye keskin bir şekilde ayrıldığı görülmektedir. Şekil 2 (d) incelendiğinde ise AGDE algoritmasının veri setini 6 kümeye homojen bir şekilde ayırdığı görülmektedir. Bunun yanı sıra mavi ve sarı renkli verilerin mümkün olduğunca diğer kümelerle karışması engellenmiştir. Aynı şekilde CMC probleminde görünürde az da olsa AGDE algoritmasının küme merkezlerinin optimum noktasını bularak kare hata değerini başarılı bir şekilde azalttığı görülmektedir.

5. Sonuç ve Tartışma (Result and Discussion)

Bu çalışmada, daha önce MSK algoritması geliştirmek için kullanılmamış AGDE, AEO, FDB-SOS gibi güncel meta-sezgisel arama yöntemleri kullanılarak melez yöntemler geliştirilmiştir. Geliştirilen on farklı MSK algoritması ve klasik k-ortalamalar yöntemi, 5 farklı kümeleme problemini çözmek için başarılı bir şekilde tatbik edilmişlerdir. Geliştirilen MSK algoritmalarının genel anlamda k-ortalamalar algoritmasına üstünlük sağladığı görülmektedir. İstatistiksel analiz sonuçlarına göre k-ortalamalar algoritmasının yerel minimum noktasına takıldığı problemlerde MSK algoritmalarının başarılı kümeleme sonuçları elde ettiği görülmektedir. MSK algoritmaları arasında ise en başarılı olanın AGDE algoritması olduğu, sonuca en hızlı yakınsama sağlayan algoritmaların ise ABC ve PSO algoritmaları oldukları anlaşılmaktadır. Deneysel çalışmalarda meta-sezgisel kümeleme algoritmaları için arama süreci sonlandırma kriteri olarak tanımlanan amaç fonksiyonu çağırma sayısı ((nitelik sayısı+küme sayısı)*100) oldukça az tutularak MSK yöntemlerinin hızlı yakınsama performansları gözlemlenmiştir. Amaç fonksiyonu azami çağırma sayısının artırılması halinde MSK yöntemlerinin tümü k-ortalamalar yönteminden çok daha üstün bir yakınsama performansı sergileyebilirler. Hatta MSK algoritmaları arasındaki sıralama dahi değişebilir. Gelecekteki çalışmalarda daha fazla veri seti, daha fazla amaç fonksiyonu çağırma sayısı ve daha fazla rakip algoritmalar ile kümeleme problemlerindeki en iyi MSK yöntemleri geliştirilmeye ve araştırılmaya çalışılacaktır. MSK algoritmalarının karmaşık kümeleme problemlerini çözme yeteneğini arttırmak için yeni geliştirilmiş yöntemlerle melezlenmeleri üzerine araştırmalar yürütülecektir.

Teşekkür (Acknowledgement)

Bu çalışmada yürütülen faaliyetler, 2020 yılında TÜBİTAK 2209-A Üniversite Öğrencileri Yurt İçi Araştırma Projeleri Destek Programı kapsamında 1919B011904077 numaralı proje olarak TUBİTAK tarafından desteklenmiştir.

Çıkar Çatışması (Conflict of Interest) Yazarlar tarafından herhangi bir çıkar çatışması beyan edilmemiştir. No conflict of interest was declared by the authors.

Kaynaklar (References)

Alam, M. S., Rahman, M. M., Hossain, M. A., Islam, M. K., Ahmed, K. M., Ahmed, K. T., ... & Miah, M. S. (2019). Automatic Human Brain Tumor Detection in MRI Image Using Template-Based K Means and Improved Fuzzy C Means Clustering Algorithm. Big Data and Cognitive Computing, 3(2), 27.

Amiri, M., Amnieh, H. B., Hasanipanah, M., & Khanli, L. M. (2016). A new combination of artificial neural network and K-nearest neighbors models to predict blast-induced ground vibration and air-overpressure. Engineering with Computers, 32(4), 631-644.

Arunkumar, N., Mohammed, M. A., Ghani, M. K. A., Ibrahim, D. A., Abdulhay, E., Ramirez-Gonzalez, G., & de Albuquerque, V. H. C. (2019). K-means clustering and neural network for object detecting and identifying abnormality of brain tumor. Soft Computing, 23(19), 9083-9096.

Bonab, M. B., Hashim, S. Z. M., Haur, T. Y., & Kheng, G. Y. (2019). A New Swarm-Based Simulated Annealing Hyper-Heuristic Algorithm for Clustering Problem. Procedia Computer Science, 163, 228-236.

Borkar, G. M., Patil, L. H., Dalgade, D., & Hutke, A. (2019). A novel clustering approach and adaptive SVM classifier for intrusion detection in WSN: a data mining concept. Sustainable Computing: Informatics and Systems, 23, 120-135.

Carrasco, J., García, S., Rueda, M. M., Das, S., & Herrera, F. (2020). Recent trends in the use of statistical tests for comparing swarm and evolutionary computing algorithms: Practical guidelines and a critical review. Swarm and Evolutionary Computation, 54, 100665.

Chen, S., Liu, X., Ma, J., Zhao, S., & Hou, X. (2019). Parameter selection algorithm of DBSCAN based on K-means two

classification algorithm. The Journal of Engineering, 2019(23), 8676-8679.
Cheng, Min-Yuan, and Doddy Prayogo. Symbiotic organisms search: a new metaheuristic optimization algorithm, Computers & Structures 139 (2014): 98-112.
Dasarathy, B.V. (1980) "Nosing Around the Neighborhood: A New System Structure and Classification Rule for Recognition in Partially Exposed Environments". IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-2, No. 1, 67-71.
Deng, W., Yao, R., Zhao, H., Yang, X., & Li, G. (2019). A novel intelligent diagnosis method using optimal LS-SVM with improved PSO algorithm. Soft Computing, 23(7), 2445-2462.
Eberhart, R., & Kennedy, J. (1995, October). A new optimizer using particle swarm theory. In Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on (pp. 39-43). IEEE.
Eftimov, T., Korošec, P., & Seljak, B. K. (2017). A novel approach to statistical comparison of meta-heuristic stochastic optimization algorithms using deep statistics. Information Sciences, 417, 186-215.
Galán, S. F. (2019). Comparative evaluation of region query strategies for DBSCAN clustering. Information Sciences, 502, 76-90.
Ghazizadeh, G., Gheibi, M., & Matwin, S. (2020, May). CB-DBSCAN: A Novel Clustering Algorithm for Adjacent Clusters with Different Densities. In Canadian Conference on Artificial Intelligence (pp. 232-237). Springer, Cham.
Huang, K. W., Wu, Z. X., Peng, H. W., Tsai, M. C., Hung, Y. C., & Lu, Y. C. (2019). Memetic Particle Gravitation Optimization Algorithm for Solving Clustering Problems. IEEE Access, 7, 80950-80968.
Jiang, Y., & Zhou, Z. H. (2004, August). Editing training data for kNN classifiers with neural network ensemble. In International symposium on neural networks (pp. 356-361). Springer, Berlin, Heidelberg.
Jin, C. H., Pok, G., Lee, Y., Park, H. W., Kim, K. D., Yun, U., & Ryu, K. H. (2015). A SOM clustering pattern sequence-based next symbol prediction method for day-ahead direct electricity load and price forecasting. Energy conversion and management, 90, 84-92.
Jothi, R., Mohanty, S. K., & Ojha, A. (2019). DK-means: a deterministic k-means clustering algorithm for gene expression analysis. Pattern Analysis and Applications, 22(2), 649-667.
Kahraman, H. T., Aras, S., & Gedikli, E. (2020). Fitness-distance balance (FDB): A new selection method for meta-heuristic search algorithms. Knowledge-Based Systems, 190, 105169.
Kahraman, H. T., Sagiroglu, S., Colak, I., Developing intuitive knowledge classifier and modeling of users' domain dependent data in web, Knowledge Based Systems, vol. 37, pp. 283-295, 2013.
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.
Kurada, R. R., & Kanadam, K. P. (2019). A Novel Evolutionary Automatic Clustering Technique by Unifying Initial Seed Selection Algorithms into Teaching–Learning-Based Optimization. In Soft Computing and Medical Bioinformatics (pp. 1-9). Springer, Singapore.
Kushwaha, N., & Pant, M. (2020). Fuzzy Particle Swarm Page Rank Clustering Algorithm. In Soft Computing: Theories and Applications (pp. 895-904). Springer, Singapore.
Lim, T.-S., Loh, W.-Y. & Shih, Y.-S. (1999). A Comparison of Prediction Accuracy, Complexity, and Training Time of Thirty-three Old and New Classification Algorithms. Machine Learning.
Miao, J., Zhou, X., & Huang, T. Z. (2020). Local segmentation of images using an improved fuzzy C-means clustering algorithm based on self-adaptive dictionary learning. Applied Soft Computing, 106200.
Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. Advances in engineering software, 69, 46-61.
Mohamed, A. W., & Mohamed, A. K. (2019). Adaptive guided differential evolution algorithm with novel mutation for numerical optimization. International Journal of Machine Learning and Cybernetics, 10(2), 253-277.
Mohamed, A., Saber, W., Elnahry, I., & Hassanien, A. E. (2020, April). Clustering Analysis Based on Coyote Search Technique. In Joint European-US Workshop on Applications of Invariance in Computer Vision (pp. 182-192). Springer, Cham.
Nan, F., Li, Y., Jia, X., Dong, L., & Chen, Y. (2019). Application of improved som network in gene data cluster analysis. Measurement, 145, 370-378.
Nithya, A., Appathurai, A., Venkatadri, N., Ramji, D. R., & Palagan, C. A. (2020). Kidney disease detection and segmentation using artificial neural network and multi-kernel k-means clustering for ultrasound images. Measurement, 149, 106952.
Pal, S. S., Hira, R., & Pal, S. (2020). Comparison of Four Nature Inspired Clustering Algorithms: PSO, GSA, BH and IWD. In Computational Intelligence in Pattern Recognition (pp. 669-674). Springer, Singapore.
Pandeeswari, N., & Kumar, G. (2016). Anomaly detection system in cloud environment using fuzzy clustering based ANN. Mobile Networks and Applications, 21(3), 494-505.
Pandey, S., Samal, M., & Mohanty, S. K. (2020). An SNN-DBSCAN Based Clustering Algorithm for Big Data. In Advanced Computing and Intelligent Engineering (pp. 127-137). Springer, Singapore.
Pierezan, J., & Coelho, L. D. S. (2018, July). Coyote optimization algorithm: a new metaheuristic for global optimization problems. In 2018 IEEE Congress on Evolutionary Computation (CEC) (pp. 1-8). IEEE.

Pouladzadeh, P., Shirmohammadi, S., Bakirov, A., Bulut, A., & Yassine, A. (2015). Cloud-based SVM for food categorization. Multimedia Tools and Applications, 74(14), 5243-5260.

Salimi, H. (2015). Stochastic fractal search: a powerful metaheuristic algorithm. Knowledge-Based Systems, 75, 1-18.

Singh, H., Kumar, Y., & Kumar, S. (2019). A new meta-heuristic algorithm based on chemical reactions for partitional clustering problems. Evolutionary Intelligence, 12(2), 241-252.

Wu, M., Li, X., Liu, C., Liu, M., Zhao, N., Wang, J., ... & Zhu, L. (2019). Robust global motion estimation for video security based on improved k-means clustering. Journal of Ambient Intelligence and Humanized Computing, 10(2), 439-448.

Xu, G., Zhang, L., Ma, C., & Liu, Y. (2020). A mixed attributes oriented dynamic SOM fuzzy cluster algorithm for mobile user classification. Information Sciences, 515, 280-293.

Yang, X. S., & Deb, S. (2009, December). Cuckoo search via Lévy flights. In 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC) (pp. 210-214). IEEE.

Yu, H., Fan, J., & Lan, R. (2019). Suppressed possibilistic c-means clustering algorithm. Applied Soft Computing, 80, 845-872.

Yu, H., Wen, G., Gan, J., Zheng, W., & Lei, C. (2020). Self-paced learning for k-means clustering algorithm. Pattern Recognition Letters, 132, 69-75.

Zhan, Charles T. Graph-theoretical methods for detecting and describing gestalt clusters. IEEE Transactions on computers, 1971, 100.1: 68-86.

Zhao, F., Chen, Y., Liu, H., & Fan, J. (2019). Alternate PSO-based adaptive interval type-2 intuitionistic fuzzy C-means clustering algorithm for color image segmentation. IEEE Access, 7, 64028-64039.

Zhao, W., Wang, L. & Zhang, Z. Artificial ecosystem-based optimization: a novel nature-inspired meta-heuristic algorithm. Neural Comput & Applic (2019). https://doi.org/10.1007/s00521-019-04452-x

Zhou, Y., Wu, H., Luo, Q., & Abdel-Baset, M. (2019). Automatic data clustering using nature-inspired symbiotic organism search algorithm. Knowledge-Based Systems, 163, 546-557.