

Veri Kümelemede Yapay Atom Algoritması ve Cırcır Böceği Algoritmasının Karşılaştırılmalı Analizi

*¹Murat CANAYAZ ve ²Murat Demir

*¹ Mühendislik Fakültesi, Bilgisayar Mühendisliği, Yüzüncü Yıl Üniversitesi, Türkiye

² Mühendislik Fakültesi, Bilgisayar Mühendisliği, Muş Alparslan Üniversitesi, Türkiye

Özet

Günümüzde veri kümeleme analizi verilerin hızla artış göstermesi nedeniyle önemli bir araç haline gelmiştir. Tıp, makine öğrenmesi, imge işleme, istatistiksel analiz gibi birçok alanda kullanılan bu işlemde benzer verilerin aynı grup içerisinde yer alması istenmektedir. Ayrıca verilerin hızlı ve güvenilir bir şekilde gruplanması için sezgisel yöntemler gibi bazı yardımcı yöntemlerde kullanılmaktadır. Sezgisel yöntemler kümelemedeki zaman ve işlem karmaşıklığını azaltarak verimli bir şekilde kümeleme yapılmasını sağlamaktadır. Yapay Atom Algoritması kimyasal bileşim oluşumuna dayalı bir algoritmadır. Cırcır Böceği Algoritması ise cırcır böceğinin hareketlerinden esinlenilerek oluşturulmuş sürü temelli bir algoritmadır. Öncelikle bu çalışmada algoritmaların özellikleri tanıtılmaya çalışılacaktır. Veri kümeleme alanındaki performansları göğüs kanseri veri seti üzerinde karşılaştırılmalı olarak incelenecektir.

Anahtar Kelimeler: Veri kümeleme, sezgisel yöntemler, yapay atom algoritması, cırcır böceği algoritması, evrimsel hesaplama

Abstract

Data clustering analysis has become important tool today, due to the rapid growth of data. It is expected that similar data take place in the same group used in many areas such as medicine, machine learning, image processing and statistical analysis. It is also used some help methods such as heuristics for grouping of data quickly and reliably. Heuristic methods provide efficient clustering by reducing the time and complexity of clustering. Artificial Atom Algorithm is heuristic algorithm that based on chemical process of compound formation. Cricket algorithm is swarm based heuristic algorithm that arose from the inspiration of the behavior of crickets. Primarily, it will attempt to introduce characteristics of algorithms in this study. Then performance of algorithms will examined in data clustering by comparing on breast cancer data set.

Key words: Data clustering, heuristic methods, artificial atom algorithm, cricket algorithm, evolutionary computation

1. Giriş

Kümeleme işlemi verilerin her geçen gün arttığı günümüz şartlarında önemli avantajlar sunmaktadır. Benzer verileri bir araya getirerek gruplama işlemi olarak bilinen veri kümeleme kaynaklardan elde edilen veri yığınlarının anlamlandırılması açısından önemli bir araçtır. Eğitimsiz bir yöntem olan veri kümeleme işleminden elde edilen, aynı grup içerisindeki verilerin birbirine benzemesi beklenir. Tıp, imge işleme, istatistiksel analiz gibi alanlardaki veri kümeleme işleminde sezgisel yöntemlere dayalı birçok algoritma, farklı özelliklere sahip bu verilerin doğru grup içerisinde yer alması için kullanılmaktadır [1-3]. Bilindiği üzere sezgisel yöntemler optimizasyon problemlerinde en iyiyi bulmaya çalışan, klasik yöntemlere nazaran daha verimli sonuçlar elde

*Corresponding author: Address: Faculty of Engineering, Department of Computer Engineering Yuzuncu Yil University, 65100, Van TURKEY. E-mail address: mcanayaz@yyu.edu.tr, Phone: +904322251027

eden yöntemler olarak değerlendirilmektedir. Bu yöntemlerden sürü temelli olanları Parçacık Sürü Optimizasyonu [4], Yapay Arı Algoritması [5], Karınca Kolonisi Algoritması [6], Cırcır Böceği Algoritması [7], fizik temelli Elektromanyetizma algoritması [8], Yer Çekimsel Arama algoritması [9], kimya temelli, Yapay Atom Algoritması [10], Yapay Kimyasal Tepkime Optimizasyon algoritması [11] gibi algoritmalar literatürde karşımıza çıkmaktadır. Bu algoritmaların veri kümeleme veya sınıflandırma alanında kullanılması ile bu alanda yapılacak işlemlerin zaman ve hız açısından performanslarının arttığı görülmektedir. Bu çalışmada da kümeleme ve sınıflandırma işlemlerinde kullanılabilen sezgisel yöntemlerden olan Yapay Atom Algoritması ve Cırcır Böceği algoritmasının karşılaştırılmalı olarak performansları incelenmiştir. Bu nedenle çalışmada ilk olarak kullanılan Cırcır Böceği ve Yapay Atom algoritmaları hakkında bilgi verilecek, daha sonra kullanılan yöntem ve sonuçlar değerlendirilecektir.

2. Sezgisel Yöntemler

Doğrusal olmayan optimizasyon problemlerinin çözümünde sezgisel yöntemler sıklıkla kullanılmaktadır. Problemlerin çözümünde klasik yöntemlere göre üstünlükleri mevcuttur. Sezgisel yöntemler sosyal tabanlı, fizik tabanlı, biyoloji tabanlı, kimya tabanlı, müzik tabanlı, sürü tabanlı, spor tabanlı, matematik tabanlı ve melez olmak üzere 9 farklı kategoride değerlendirilmektedir [12]. Bu çalışmada kullandığımız Cırcır Böceği Algoritması (CBA) sürü tabanlı, Yapay Atom Algoritması (YAA) ise kimya tabanlı algoritmalarındandır.

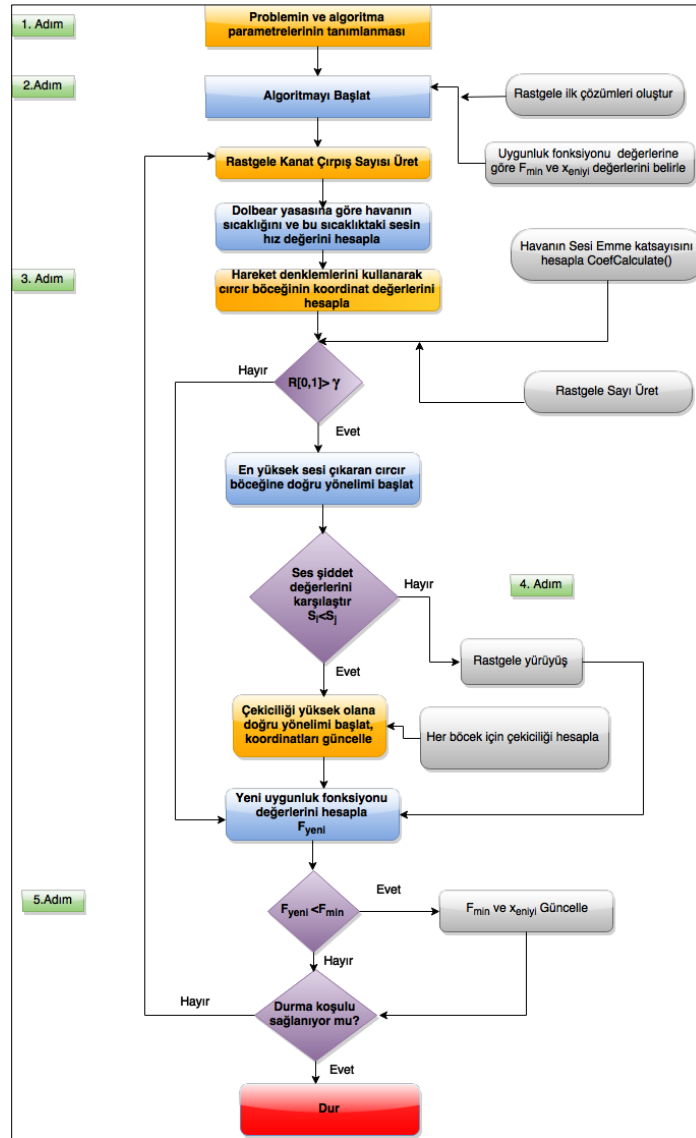
2.1. Cırcır Böceği Algoritması

Cırcır böcekleri ses ile iletişim kabiliyeti olan yazın sıcak günlerinde yol kenarlarında veya ormanlık alanlarda gördüğümüz bir canlı türüdür. Erkek cırcır böcekleri bu kabiliyetlerini dişi cırcır böceklerini etkilemek için kullanırlar. Yüksek sese sahip olan erkek cırcır böceğine doğru dişi cırcır böcekleri tarafından bir yönelim gerçekleştirilir. Cırcır böceğinin bu davranışlarının modellenmesi ile elde edilen bir meta sezgisel algoritma olan Cırcır Böceği Algoritması (CBA) sürü temelli sezgisel yöntemlerdendir. Yapmış oldukları yönelim işlemi birçok metasezgisel algoritmada esinlenen canlılar ile benzer özellikler barındırır. Örneğin parçacık sürü optimizasyonundaki kuşların en önde giden kuşu takibi, ateş böceği algoritmasındaki ateş böceklerinin en yüksek ışığa sahip ateş böceğine yönelmeleri, yarasalar algoritmasındaki yarasaların ses ile yön bulmada yapmış oldukları hareketler Cırcır böceği algoritmasının esinlendiği ortak özellikler olarak sayılabilir. Bunun yanında Cırcır böceğine has olan bazı özellikler kullanılarak CBA geliştirilmiştir. Amos Dolbear isimli bir bilim adamının ortaya attığı hipoteze göre Cırcır böceklerinin kanat çırpış sayısı ile o anki hava sıcaklığı arasında bir korelasyon bulunmaktadır [13]. Dolbear bu hipotezini Denklem 1 ve 2 ile ifade etmiştir. Dolbeardan sonra bu hipotezi doğrulayan yayınlar ortaya çıkmıştır [14]. Algoritmada bu denklemlerin kullanılmasının yanında bu canlılar ses ile iletişime geçtiklerinden dolayı algoritmanın geliştirilme safhasında gerçek hayata uygunluğunun artırılması için sesin doğadaki yayılımı ile alakalı fiziki kanunlarda göz önünde bulundurulmaktadır. Sesin yayılımı ile alakalı kullanılan denklemler Tablo'1 de verilmektedir.

Tablo 1. CBA'da kullanılan denklemler

Açıklama	Formül	Açıklama	Formül
{Dolbear Kanunu}	$T_F = 50 + \left(\frac{N - 40}{4} \right)$ T _F :Havanın sıcaklığı (Fahrenayt) N: kanat çırpış sayısı	(1) {Frekans Değerleri}	$f_i = f_{min} + (f_{max} - f_{min})\beta$ f:frekans (9)
“	$T_c = 10 + \left(\frac{N - 40}{7} \right)$ T _C :Havanın sıcaklığı (Santigrat)	(2) {Hız değerleri}	$v_i^t = v_i^{t-1} + (x_i - x_*)f_i + V_i$ V:sesin hızı x*:o anki en iyi çözüm (10)
{ Sıcaklık-Hız İlişkisi}	$V = 20.1 * \sqrt{273 + C}$ V:Sesin Hızı C: Havanın sıcaklığı	(3) {Koordinat Değerleri}	$x_i^t = x_i^{t-1} + v_i^t$ (11)
{Frekans}	$f = \frac{V}{\lambda}$ f:Frekans, V:Hız, λ:Dalga boyu	(4) {Rastgele yürüyüş}	$x_i = x_{best} + 0.01 * rand(0,1)$ x _{best} : mevcut en iyi çözüm (12)
{Sesin Gücü}	$P = I * 4\pi r^2$ P:Sesin gücü, I:Sesin şiddeti, r:Mesafe	(5) {Öklid Mesafesi}	$r_{ij} = \ x_i - x_j\ = \sqrt{\sum_{k=1}^d (x_{i,k} - x_{j,k})^2}$ (13)
{Ses Basıncı Seviyesi}	$L_p = L_w + 10 * \log[Q / (4\pi r^2)]$ L _p :Ses basınç seviyesi, L _w : Kaynağın ses gücü düzeyi (dB), Q: Yönelme katsayısı yön faktörü,r: Kaynaktan olan uzaklıktır(m.)	(6) {Çekicilik}	$K = K_0 e^{-\gamma r^2}$ (14)
{ Atmosferin sesi tutumu}	$A_{att} = 7.4 (f^2 r / \emptyset) 10^{-8}$ f = İletilen sesin frekansı (Hz), r = Kaynaktan olan uzaklık (m), \emptyset = Havanın bağıl nemi (%)	(7) {Koordinatların Güncellenmesi}	$x_i = x_i + K_0 e^{-\gamma r_i^2} + \alpha \epsilon_i$ γ : havanın sesi absorbe etme katsayısı, αε _i : verilen probleme göre üst ve alt limitlerde ölçekleme için kullanılan katsayılar, xi: cırcır böceğinin koordinatı (aday çözüm) (15)
{ Serbest Alanda Gerçek Ses Basınç Düzeyi }	$L_p' = L_p - A_{att}$ L _p ': Gerçek ses basınç düzeyi	(8)	

Algoritmaya ait akış diyagramı Şekil 1’de verilmektedir.

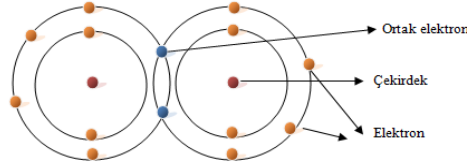


Şekil 1. CBA algoritması akış diyagramı

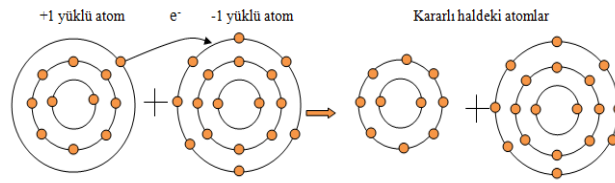
Algoritmada aşağıdaki parametre değerleri kullanılmıştır. α : 0.5 (Her seferinde 0.97 ile çarpılarak azaltılır), θ : %50, β : Random[0,1], ϵ : Random[0,1], n : 25.

2.2. Yapay Atom Algoritması

YAA, atomlar içerisindeki ve atomlar arasındaki çekim kuvvetlerini esas alan bir algoritmadır. Atomların içyapılarında çekirdek ve çekirdeğin etrafında elektronlar vardır. Son yörüngede bulunan elektronların bir kısmı atomlar arası bağlarda ortaklaşa kullanılır. Kovalent bağ elektronların ortaklaşa kullanılması, iyonik bağ ise zıt yüklü iyonların çekim kuvvetlerinden ortaya çıkar. Atomlarda her iki bağ çeşidinin de etkileri mevcuttur [15].



Şekil 2. Kovalent bağ şekilsel temsili



Şekil 3. İyonik bağ şekilsel temsili

YAA'nın çözüm temsili, kovalent ve iyonik bağlar çözüm matrislerinde temsili tutulur. Bu matrisler çözüm değerlerini içerisinde barındırır. Tablo 2 yapay atom algoritmasının genel çözüm yapısının temsili göstermektedir.

Tablo 2. Yapay atom algoritması genel çözüm matrisi temsili

K_1	K_2	K_3	K_4	K_5	...	K_n	\dot{I}_1	\dot{I}_2	\dot{I}_3	...	\dot{I}_n
-------	-------	-------	-------	-------	-----	-------	-------------	-------------	-------------	-----	-------------

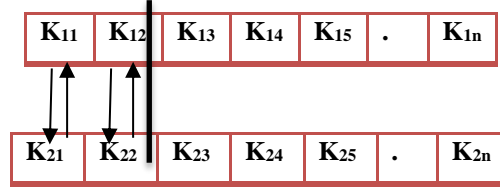
K_1, K_2, \dots, K_n 'ler kovalent bağ aday çözüm değerlerini, $\dot{I}_1, \dot{I}_2, \dots, \dot{I}_n$ 'ler ise güncellenecek iyonik bağ aday çözümlerini temsil etmektedir. Bu şekilde bir popülasyon oluşturulur [16-17].

Tablo 3. Yapay atom algoritması popülasyon temsili

K_{11}	K_{12}	K_{13}	...	K_{1n}	\dot{I}_{11}	\dot{I}_{12}	\dot{I}_{13}	...	\dot{I}_{1n}
K_{21}	K_{22}	K_{23}	...	K_{2n}	\dot{I}_{21}	\dot{I}_{22}	\dot{I}_{23}	...	\dot{I}_{2n}
K_{31}	K_{32}	K_{33}	...	K_{3n}	\dot{I}_{31}	\dot{I}_{32}	\dot{I}_{33}	...	\dot{I}_{3n}
K_{41}	K_{42}	K_{43}	...	K_{4n}	\dot{I}_{41}	\dot{I}_{42}	\dot{I}_{43}	...	\dot{I}_{4n}
.
.
K_{n1}	K_{n2}	K_{n3}	...	K_{nn}	\dot{I}_{n1}	\dot{I}_{n2}	\dot{I}_{n3}	...	\dot{I}_{nn}

Kovalent ve iyonik bağ çözümlerinin temsil edildiği matrislerdeki özellik sayıları aynı (n) olmak zorunda değildir. Fakat bu çalışmada geliştirilen yöntemde farklı matriste temsil edildiklerinden özellik sayıları eşit seçilmiştir.

Popülasyonun kovalent tarafındaki çözüm değerlerinin uygunluk (fitness) değerleri hesaplanıp sıralanır. Kovalent bölgede uygunluk değeri en düşük elemanlar içerisinden; seçilen özellik (attribute) sayısı kadar özellik, uygunluğu yüksek olan bireyler ile değer değiştirilir. Böylelikle farklı çözümler arası karşılıklı eleman değişikliği ile çeşitlilik sağlanır.



Şekil 4. Yapay atom algoritması elektron ortaklığının temsili

Bu yolla çözüm çeşitliliği sağlanır. Daha sonrasında iyonik bağları temsilen oluşturulan çözümlerde popülasyona dahil edilerek, tüm popülasyon tekrar bir uygunluk hesaplamasından geçirilir. Daha sonra sıralama işlemi tekrar gerçekleştirilir. İyonik bağlar için oluşturulan rastgele çözümler, çözüm uzayına eklenerek, çözüm uzayındaki çeşitlilik artırılmış olmaktadır. Algoritma işlemler belirli bir durdurma kriteri sağlanana kadar devam ettirilir.

3. Önerilen Yöntem

Bu çalışmada geliştirilen yöntemde BEST matris adlı bir çözüm kümesi oluşturulmaktadır. BEST matrisin boyutu, kullanılan veri setinin özellik sayısına göre uygulama yazılımında belirlenmiştir. Öncelikle her algoritma için bir başlangıç popülasyonu oluşturulur. Popülasyon BEST matris adayları olabilecek çözümler topluluğudur. Algoritmaya göre popülasyonda oluşturulacak çözüm satır sayısı tasarımcı tarafından belirlenir. Sütun sayısı çalışılacak olan veri setinin özellik sayısına eşit olmak zorundadır. Tablo 4 genel popülasyon temsili ifade etmektedir. Tablo 4'deki m tasarımcıya bağlıdır. n ise veri setindeki özellik sayısına bağlıdır.

Tablo 4. Başlangıç popülasyonlarının genel temsili

özellik ₁	özellik ₂	.	.	.	özellik _n
pop(1,1)	pop(1,2)	.	.	.	pop(1,n)
pop(2,1)	pop(2,2)	.	.	.	pop(2,n)
.
.
pop(m,1)	pop(m,2)	.	.	.	pop(m,n)

BEST matris her uygulama yazılımında farklı yöntemlerle elde edilir. BEST matris elde edilirken uygulama yazılımında oluşturulan popülasyon ile eğitim kümesinin özelliklerinin farkları alınır. Algoritmanın yapısına göre sonuca gidilir ve BEST matris elde edilir.

$$\text{for } i=1,2,3,\dots,m \quad p_i = \sum_{k=1}^n |(\text{popülasyon}(i, k) - \text{eğitim_kümesi_özellik}(i, k))|$$

p_i , popülasyondaki her bir çözümün manhattan uzaklık değeri, $\text{popülasyon}(i,k)$, uygulama yazılımına özel oluşturulmuş çözüm değeri, $\text{eğitim_kümesi_özellik}(i,k)$, eğitim veri seti değerlerinin her biri

Amaç p_i değerinin minimum olduğu değeri tespit etmektir. Çünkü p_i 'nin en küçük olduğu çözüm eğitim setinin tümüne en yakın olan optimum çözümdür. Algoritma sonlanma adımına kadar, toplam uzaklık değeri (hata değeri) en küçük olan çözüm, popülasyondan çıkan en iyi, yani BEST çözümdür.

Tablo 5. p_i değer matrisinin temsili

$p_i(1,1)$	$p_i(1,2)$	$p_i(1,3)$.	.	.	$p_i(1,n)$
$p_i(2,1)$	$p_i(2,2)$	$p_i(2,3)$.	.	.	$p_i(2,n)$
.
.
$p_i(m,1)$	$p_i(m,2)$	$p_i(m,3)$.	.	.	$p_i(m,n)$
$\sum (\text{fark}(n,:))$						

Tablo 6. p_i değerlerinin toplamının matris temsili

$p_{i_top}(1,1)$	$p_{i_top}(1,2)$.	.	.	$p_{i_top}(1,m)$
-------------------	-------------------	---	---	---	-------------------

Her çözüm için bir uzaklıklar toplamı hesaplanacak olursa toplam m tane farklar toplamı ortaya çıkar. Tablo 6'nın değerleri içerisinde en küçük olan değer, indeks değerine karşılık gelen popülasyondaki çözüm matrisi o anki adımın BEST matrisidir. Örneğin Tablo 6'daki en küçük değer indeks değeri 7. olan çözüm olsun; bu durumda çözüm popülasyonunun 7. satırı o anki adımın BEST matrisi olur. Algoritma sonlanana kadar bu adımlar tekrar edilir. Her seferinde BEST matris güncellenir. En son elde edilen BEST nihai sonuçtur ve BEST matris olarak çıktı verilir.

BEST matris elde edildikten sonra, BEST matrisin test için ayrılan veriler ile manhattan uzaklık farklarının toplamı alınır. Aslında bu uzaklıkların toplamı bizim için, BEST matrise göre, test verisinin hata değerlerinin toplamı olmaktadır. Bu fark değerlerinin sonuç aralığına göre sınıflar belirlenir. Bu durum Şekil 5'de şekilsel olarak temsil edilmiştir.

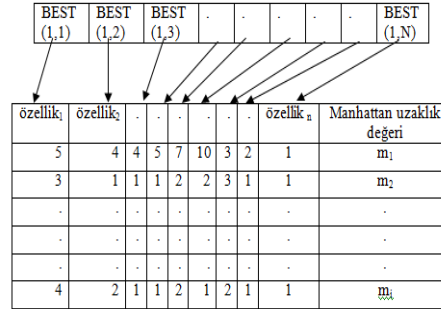
for $i=1,2,3,\dots,t$

$$m_i = \sum_{k=1}^n | \text{BEST}(i,k) - \text{test_kümesi_özellik}(i,k) |$$

m_i , test değerinin manhattan uzaklık değeri,

BEST(i,k), yazılımın üretmiş olduğu BEST matrisin her bir satır elemanı,

test kümesi özellik(i,k) , test için ayrılan veri setinin her bir satır elemanıdır.



Şekil 5. Manhattan uzaklıkları hesabının şekilsel temsili

Algoritmaların karşılaştırılmasında Breast Cancer Wisconsin [18] adlı veri seti kullanılmaktadır. Bu veri seti 9 özneliğe sahip 687 veri içermektedir. Bu verilerin bir kısmı eğitim, bir kısmı ise test verisi olarak seçilmiştir. Veri setine ait verilerin bir kısmı Şekil 6'da gösterilmektedir. Algoritmalar her biri 100, 500 ve 1000 iterasyon olmak üzere bağımsız olarak çalıştırılmaktadır.

Veri setinde bulunan özellikler; Kitle Kalınlığı:1-10, Hücre Boyutu birbirine benzerliği:1-10, Hücre Şeklinin birbirine benzerliği:1-10, Marjinal yapışkanlık:1-10, Tek Epitel Hücre Boyutu:1-10, Saf Çekirdekler:1-10, Donuk Kromatin:1-10, Normal çekirdek:1-10, Mitoz bölünmeler: 1-10.

5	1	1	1	2	1	3	1	1
5	4	4	5	7	10	3	2	1
3	1	1	1	2	2	3	1	1
6	8	8	1	3	4	3	7	1
4	1	1	3	2	1	3	1	1

Şekil 6. Veri seti örnek

3. Deneysel Bulgular

Tablo 7 ve 8'de sırasıyla YAA ve CBA algoritmalarından elde edilen sonuçlar gösterilmektedir. Sonuçlar incelendiğinde 100 iterasyonda CBA'dan elde edilen doğruluk oranı ortalama % 86 iken bu oran YAA'da % 92,6 civarındadır. 500 iterasyonda doğruluk oranları birbirine yakın değerlerdir. İterasyon arttıkça CBA'daki doğruluk oranının arttığı görülmektedir. 1000 iterasyon da ise YAA'daki doğruluk oranı ortalama %90,6 iken, CBA algoritmasında % 93 olduğu görülmektedir. Aynı veri seti üzerinde yapılan karşılaştırmalar sonucunda düşük iterasyonda YAA algoritmasının daha iyi sonuçlar verdiğini söylemek mümkündür.

Tablo 7. YAA Sonuçları

Yapay Atom Algoritmasından Elde Edilen Sonuçlar										
100 iterasyon										
Adım	att.1	att.2	att.3	att.4	att.5	att.6	att.7	att.8	att.9	Doğruluk
1	4,6017	1,5434	1,2517	1,2325	2,0052	2,2804	1,7219	1,544	1,2406	94%
2	3,9281	1,1804	1,2678	1,2157	2,3369	1,1769	4,2009	1,6727	1,5808	93%
3	4,1947	2,1218	1,676	1,0327	2,0267	1,5091	2,5721	1,9016	1,0347	91%
500 iterasyon										
Adım	att.1	att.2	att.3	att.4	att.5	att.6	att.7	att.8	att.9	Doğruluk
1	3,4412	1,1496	1,3661	2,1392	1,9654	1,7654	2,4269	1,1757	1,6698	93%
2	3,8866	1,7122	1,4765	1,8183	2,2208	1,2228	1,9893	1,0464	1,2469	94%
3	4,3989	1,0968	1,3013	1,4938	2,0501	1,0002	2,8477	1,0511	1,431	93%
1000 iterasyon										
Adım	att.1	att.2	att.3	att.4	att.5	att.6	att.7	att.8	att.9	Doğruluk
1	4,0787	3,0024	1,803	1,6369	2,246	1,006	3,0327	1,1299	1,0985	89%
2	4,2699	1,1319	2,2	1,2706	2,4901	1,6017	2,7095	1,3119	1,5693	91%
3	4,3917	1,5213	1,1719	1,6102	2,0494	1,8646	2,8599	1,2628	1,1057	92%

Tablo 8. CBA Sonuçları

Cırcır Böceği Algoritmasından Elde Edilen Sonuçlar										
100 iterasyon										
Adım	att.1	att.2	att.3	att.4	att.5	att.6	att.7	att.8	att.9	Doğruluk
1	3,3203	3,5315	1,9830	6,7667	2,5905	1,1869	5,1111	4,0183	5,3363	82%
2	1,6749	1,9939	6,4073	4,9009	4,8020	1,4018	3,8305	2,3927	1,8047	89%
3	2,4407	2,0101	4,2048	7,0053	6,9764	1,7631	4,5962	1,2526	5,0455	87%
500 iterasyon										
Adım	att.1	att.2	att.3	att.4	att.5	att.6	att.7	att.8	att.9	Doğruluk
1	2,0078	2,0221	1,9368	1,9920	2,0276	2,0105	1,9966	1,8270	1,8129	93%
2	2,7077	1,6068	1,5298	1,6351	8,5996	1,8678	1,2483	2,0085	1,2354	93%
3	4,5467	2,7279	1,5659	2,6096	4,7587	1,0497	1,9595	4,2073	2,1633	92%
1000 iterasyon										
Adım	att.1	att.2	att.3	att.4	att.5	att.6	att.7	att.8	att.9	Doğruluk
1	1,7315	1,6897	1,6197	1,6654	1,8160	1,7054	1,6623	1,6505	1,7001	93%
2	1,8249	1,5324	1,7791	1,5967	1,8266	5,7817	1,7077	1,5518	1,4821	92%
3	2,8791	1,2373	1,1590	1,3201	2,1363	1,1712	1,2958	2,0737	1,2564	94%

Sonuçlar

Bu çalışmada sezgisel yöntemlerden olan Yapay Atom Algoritması ve Cırcır Böceği Algoritması ile göğüs kanseri verileri üzerinde kümeleme analizinin karşılaştırması yapılmıştır. Veri kümelemede kullanılan birçok yöntem olmakla beraber; sonuçlara bakıldığında bu algoritmaların % 94'e varan bir düzeyde doğruluk oranına sahip olarak, kümeleme işlemini gerçekleştirmesi, kümeleme işlemi için başarılı yöntemler olarak uygulanabileceğini göstermiştir. Sadece düşük

iterasyon sayısında Cırcır Böceği Algoritmasının doğruluk yüzdesi %90'nın altında kalmaktadır. Buda Yapay Atom Algoritmasının çözüme yakınsamasının daha çabuk olabileceğini göstermektedir. İleriki çalışmalarda düşük iterasyon sayılarında Cırcır Böceği Algoritmasının performansının artırılması için neler yapılabileceği üzerinde çalışma yapılması düşünülmektedir.

Kaynaklar

- [1] Demir M, Karcı A. Data clustering on breast cancer data using firefly algorithm with golden ratio method. *AECE* 2015;15:75-84.
- [2] Öztürk C, Hancer E, Karaboga D. Improved clustering criterion for image clustering with artificial bee colony algorithm. *Pattern Anal Applic* 2015;18:587–599.
- [3] Jothi G, Hannah Inbarani H. Hybrid tolerance rough set–firefly based supervised feature selection for MRI brain tumor image classification. *Appl Soft Comput* 2016;46:639–651.
- [4] Kennedy J, Eberhart RC. Particle swarm optimization. *IEEE International Conference on Neural Network*, 1995, pp. 1942-1948, Perth, WA, Australia.
- [5] Akay B, Karaboga D. Artificial bee colony algorithm for large-scale problems and engineering design optimization. *J. Intell. Manuf.* 2012; 23: 1001–1014.
- [6] Dorigo M, Stützle T. *Ant colony optimization*. MIT Press: Cambridge; 2004.
- [7] Canayaz M, Karcı A. Cricket behavior-based evolutionary computation technique in solving engineering optimization problems. *Appl Intell* 2016;44:362–376.
- [8] Özdağ R, Karcı A. Probabilistic dynamic distribution of wireless sensor networks with improved distribution method based on electromagnetism-like algorithm. *Measurement* 2016;79:66-76.
- [9] Rashedi, E, Nezamabadi-pour H, Saryazdi, S. GSA: A gravitational search algorithm. *Inform. Sciences* 2009;179:2232–2248.
- [10] Karcı A. A new metaheuristic algorithm based chemical process: atom algorithm. *1st International Eurasain Conference on Mathematical Sciences and Applications*, 2012, pp. 83-84, Prishtine, Kosovo.
- [11] Alataş B. ACROA: Artificial chemical reaction optimization algorithm for global optimization. *Expert Syst. Appl.* 2011; 38: 13170–13180.
- [12] Canayaz M. Cırcır böceği algoritması: yeni bir meta-sezgisel yaklaşım ve uygulamaları. *Doktora Tezi, İnönü Üniversitesi, Fen Bilimleri Enstitüsü, Malatya*, 2015.
- [13] Dolbear EA. The cricket as a thermometer. *Amer Nat* 1897; 31: 970-971.
- [14] Larsen JL, LeMone P. The sound of crickets. *Science Teacher* 2009; 76: 37-41.
- [15] Erdoğan Yıldırım A, Karcı A. Solutions of travelling salesman problem using genetic algorithm and atom algorithm. *2nd International Eurasain Conference on Mathematical Sciences and Applications*, 2013, pp. 134, Sarajevo, Bosnia and Hercegovina.
- [16] Karadoğan A, Karcı A. Artificial atom algorithm for reinforcement learning. *2nd International Eurasain Conference on Mathematical Sciences and Applications*, 2013, pp. 379, Sarajevo, Bosnia and Hercegovina.
- [17] Demir M, Karcı A. Yapay atom algoritması yöntemi ile veri kümeleme. *Muş Alparslan Üniversitesi Fen Bil. Enst. Dergi.* 2015; 3: 340-353.
- [18] Mangasarian OL, Wolberg WH. Cancer diagnosis via linear programming. *SIAM News* 1990;23:1-18