

YAPAY ATOM ALGORİTMASI YÖNTEMİ İLE VERİ KÜMELEME
DATA CLUSTERING WITH ARTIFICIAL ATOM ALGORITHM METHOD

Murat DEMİR¹, Ali KARCI²

Murat DEMİR¹, Muş Alparslan Üniversitesi Mühendislik-Mimarlık Fakültesi Bilgisayar Mühendisliği, 49250, Muş, Ali KARCI², İnönü Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği, Malatya

ÖZET

Bu çalışmada, Yapay Atom Algoritması (A³: Artificial Atom Algorithm) ile veri kümeleme gerçekleştirilmiştir. Yapay Atom Algoritması sezgisel bir algoritmadır. Çözümde, atomların elektron ortaklaşması sonucu oluşan kovalent bağları ve zıt iyonların çekim kuvveti sonucu oluşan iyonik bağları temsil eden çözüm popülasyonları kullanarak gider. Veri kümeleme işlemi için Iris-Plants veri seti kullanılmıştır. Veri seti 3 sınıf bitkiden oluşmaktadır. Geliştirilen uygulama yazılımı ile veri seti tekil deneme sonuçlarında %91,67, ortalama deneme sonuçlarında ise % 90,5 oranında başarı ile veri setini kümelemiştir.

Anahtar Kelimeler: *Yapay Atom Algoritması, Veri Kümeleme*

ABSTRACT

In this study, with the A³: Artificial Atom Algorithm the data clustering has been made. Artificial Atom Algorithm is a heuristic algorithm. It comes to a conclusion by using solution populations that represents the covalent bonds, which are formed as a result of the collective of electrons, and ionic bonds, which are formed as a result of the gravitational force of the counter ions. Iris-Plants data sets have been used for data clustering. The data set consists of three plant classes. With the developed software application, with a 91.67% success in the individual database test results, and with the 90,5 % success in the average test results, the data set has been clustered.

Key Words: *Artificial Atom Algorithm, Data Clustering*

1.GİRİŞ

Çok hızlı bir şekilde değişim gösteren teknolojilerin, giderek daha da karmaşıklaşan sistemlerin beraberinde getirmiş olduğu problemlerin, matematiksel ya da analitik bilinen klasik yöntemlerle çözülmesi güçleşmiştir. Bu sebeplerden dolayı optimizasyon kavramı son zamanlarda en çok bahsi geçen alanlardan olmuştur. Tüm bilim dallarının içerisinde optimizasyon kavramı yerini almıştır.

Optimum kelimesi Latince kökenli bir kelimedir. Olabilmesi mümkün olan en ideal manasındadır. Optimizasyon ise, bir problemin en iyi çözümünü veya tasarımını bulma işlemi olarak tanımlanabilir. Bir başka deyişle mümkün olan alternatifler arasında en iyisini seçebilme işlemidir. Mühendislik alanında, tasarım, imalat ve bakım aşamalarında kararlar alınır. Bu kararların amacı, gerekli çaba, sermaye, malzeme veya teknolojinin minimum tutulması ve faydanın maksimum yapılabilmesidir.

Bu bakış açısıyla optimizasyon, hedeflenen amacı maksimum veya minimum yapacak şartları bulmak olarak ifade edilebilir [1]. Belirli sınırlamaları sağlayacak şekilde, bilinmeyen parametrelerin değerlerinin hesaplanmasını içeren herhangi bir problem, optimizasyon problemi olarak adlandırılabilir [2].

2.SEZGİSEL YÖNTEMLER

Gerçek dünyada optimizasyon problemlerinin çoğu doğrusal olmayan yapıdadır. Bu problemlerin çözümü için birçok teknik ve yaklaşım mevcuttur. Özellikle çok parametre içeren sistemlerin optimizasyon aşamalarında, parametre sayısına ve veri tipine bağlı olarak problemlerin zorluk derecesi değişiklik gösterebilmektedir. Bu tip problemlerin klasik yöntemler ile çözümü, hem modellemede, hem de çözüm aşamasında güçlükler getirmektedir. Bu tür zorlukları aşabilmek için doğada var olan sistemleri ve olayları taklit eden sezgisel yöntemler geliştirilmiştir.

Sezgisel yöntemler, büyük boyutlu optimizasyon problemleri için, makul bir sürede

optimuma yakın çözümler sunabilen yöntemlerdir. Genel amaçlı sezgisel optimizasyon algoritmaları, biyoloji tabanlı, fizik tabanlı, sürü tabanlı, sosyal tabanlı, müzik tabanlı ve kimya tabanlı olmak üzere altı farklı grupta değerlendirilmektedir [3].

Fizik tabanlı: Çok-noktalı benzetimli tavlama algoritması [4], Yerçekimi arama algoritması [5], Elektromanyetizma algoritması [6, 7], Büyük patlama-Büyük çöküş algoritması [8].

Biyoloji tabanlı: Yarasa algoritması [9-11], Genetik algoritma [12, 13], Karınca koloni algoritması [14], Arı koloni algoritması [15, 16], Yapay bağışıklık algoritması [17], Fidan gelişim algoritması [18-23], İstilacı ot (Invasive weed) optimizasyon algoritması [24], Maymun arama algoritması [25], Bakteri yiyecek arama (Bacterial foraging) algoritması [26], Cırcır böceği algoritması [27], Evrimsel algoritma [28].

Sosyal tabanlı: Çok-noktalı tabu arama algoritması [29], Emperyalist rekabetçi algoritma [30].

Kimya tabanlı: Yapay kimyasal reaksiyon optimizasyon algoritması [31], Yapay Atom Algoritması [32-34].

Sürü tabanlı: Ateşböceği algoritması [35-38], Parçacık sürü optimizasyon algoritması [39-43], Kedi sürü optimizasyon algoritması [44].

Müzik tabanlı: Harmony search algoritması [38, 45].

Sezgisel (heuristic) yöntemler bilgisayar bilimlerinde, bir problem çözme tekniğidir. Genelde problemlerin çözümünde iyiye yakın olacak şekilde çözümler elde ederler. Elde edilen sonuçların doğruluğunun kanıtlanabilir olup olmadığını önemsememektedir. Sezgisel yöntemler, en iyi sonucu bulabileceklerini garanti etmezler, fakat makul bir sürede bir çözüm elde edeceklerini garanti ederler. Genellikle en iyiye yakın olan çözüme bilinen matematiksel ya da analitik çözümlerden daha hızlı ve kolay bir şekilde ulaşırlar. Bu nedenle çözümü güç ve karmaşık olan sistemlerin çözümünde tercih edilirler.

Sezgisel algoritmalar; anlaşılabilirlik yönünden

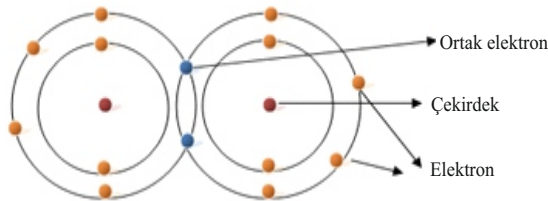
çok daha basit olmasından, optimizasyon problemlerinin kesin çözüme giden yollarının bazen çok karmaşık olabilmesinden ve bu öğrenme amaçlı çözüm yöntemlerinde kullanılabilmesinden dolayı çeşitli problemlerin çözümünde ihtiyaç duyulmaktadır [46].

Sezgisel yöntemlere örnek olarak :

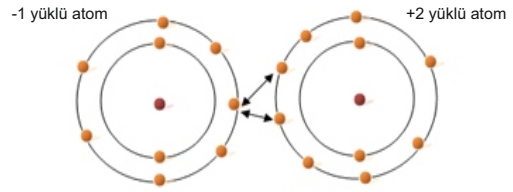
- 1- Genetik Algoritma
- 2- Yapay Sinir Ağları
- 3- Karınca Koloni Algoritması
- 4- Ateş Böceği Algoritması
- 5- Yarasa Algoritması
- 6- Yapay Atom Algoritması
- 7- Fidan Gelişim Algoritması
- 8- Parçacık Sürü Optimizasyon Algoritması
- 9- Destek Vektör Makineleri
- 10- Benzetimli Tavlama
- 11- Arı Koloni Algoritması
- 12- Kurt Koloni Algoritması
- 13- Yapay Balık Sürüsü Algoritması v.b. verilebilir.

3.YAPAY ATOM ALGORİTMASI

Bu metot, atomlar arası çekimlerde etkili olan kovalent ve iyonik bağları esas almaktadır. Atomların yapısında çekirdek ve etrafında elektronlar vardır. Son yörüngedeki elektronların bir kısmı bağ esnasında ortaklaşa kullanılır. Kovalent bağ elektron ortaklaşmasından, iyonik bağ ise zıt yüklü çekim kuvvetlerinden oluşur. Atomlarda her iki bağ çeşidinin de etkileri mevcuttur [32-34]. Bu bağlar Şekil 3.1.ve Şekil 3.2.'de temsilen gösterilmiştir.



Şekil 3.1. Kovalent bağ temsili



Şekil 3.2. İyonik bağ temsili

Yapay Atom Algoritmasında kovalent ve iyonik bağların çözümlerini temsil eden çözüm matrisleri mevcuttur. Bu durum Tablo 3.1.'de gösterilmiştir. Bu matrisler çözüm değerlerini içerisinde barındırır.

Tablo 3.1. Yapay Atom Algoritması 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 çözümlerini, $\dot{I}_1, \dot{I}_2, \dots, \dot{I}_n$ ler ise iyonik bağ çözümlerini ifade etmektedir. Bu temsil yardımı ile Tablo 3.2.'de gösterildiği gibi bir popülasyon oluşturulur [32-34].

Tablo 3.2. 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}

Bu çalışmada Yapay Atom Algoritmasında aynı matriste temsil edilen kovalent ve iyonik bağ çözümleri Tablo 3.3. ve 3.4.'te gösterildiği gibi farklı iki matriste temsil edilmiştir.

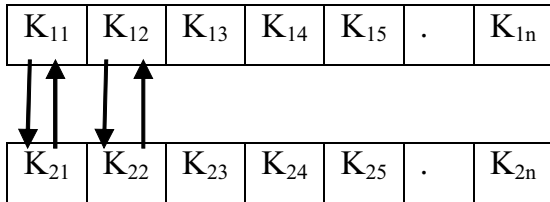
Tablo 3.3. Yapay Atom Algoritması kovalent bağ temsili

K_{11}	K_{12}	K_{13}	K_{14}	K_{15}	.	K_{1n}
K_{21}	K_{22}	K_{23}	K_{24}	K_{25}	.	K_{2n}
K_{31}	K_{32}	K_{33}	K_{34}	K_{35}	.	K_{3n}
K_{41}	K_{42}	K_{43}	K_{44}	K_{45}	.	K_{4n}
.
.
K_{m1}	K_{m2}	K_{m3}	K_{m4}	K_{m5}	.	K_{mn}

Tablo 3.4. Yapay Atom Algoritması iyonik bağ temsili

\dot{I}_{11}	\dot{I}_{12}	\dot{I}_{13}	\dot{I}_{14}	\dot{I}_{15}	.	\dot{I}_{1n}
\dot{I}_{21}	\dot{I}_{22}	\dot{I}_{23}	\dot{I}_{24}	\dot{I}_{25}	.	\dot{I}_{2n}
\dot{I}_{31}	\dot{I}_{32}	\dot{I}_{33}	\dot{I}_{34}	\dot{I}_{35}	.	\dot{I}_{3n}
.
.
\dot{I}_{m1}	\dot{I}_{m2}	\dot{I}_{m3}	\dot{I}_{m4}	\dot{I}_{m5}	.	\dot{I}_{mn}

Popülasyonun kovalent bölgedeki tüm çözüm matrislerinin uygunluk (fitness) değerleri hesaplanıp sıralanır. Kovalent bölgede uygunluğu en zayıf elemanlar içerisinden seçilen özellik (attribute) sayısı kadar özellik uygunluğu yüksek olan bireyler ile değer değiştirilir. Böylelikle karşılıklı eleman değişikliği çeşitliliği sağlanır. Şekil 3.3. bunu göstermektedir.

**Şekil 3.3.** Elektron ortaklığının Yapay Atom Algoritması temsili

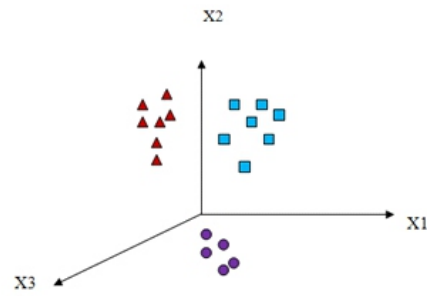
Böylelikle çözüm çeşitliliği de sağlanabilmektedir. Daha sonrasında iyonik bağları temsilen oluşturulan çözümlerde işin içine katılarak tüm popülasyon tekrar bir uygunluk hesaplamasından geçirilir. Sıralama işlemi tekrar gerçekleştirilir. İyonik bağlar için oluşturulan rastgele çözümler, çözüm uzayına eklenerek, çözüm uzayındaki çeşitlilik artırılır. Yeni bireylerle beraber oluşan popülasyon sıralandıktan sonra yani işlemler belirli bir durdurma kriteri sağlanan kadar uygulanır.

4.VERİ KÜMELEME

Çoğu zaman kümeleme ve sınıflandırma birbirlerinin yerine kullanılan kavramlar olmalarına rağmen, ikisi farklı şeylerdir. Sınıflandırma, eğitici bir öğrenme metodudur. Fakat kümeleme eğitici bir öğrenme metodudur. Kümeleme, nesnelerin öğreticisiz olarak farklı gruplara (kümelere) ayrıldığı bir sınıflandırma yöntemidir. Aynı küme içerisindeki nesnelere diğer kümelerdeki nesnelere göre birbirlerine daha benzerler [47, 48].

Veri kümeleme yöntemleri, hiyerarşik ve hiyerarşik olmayan yöntemler olarak ikiye ayrılır. Hiyerarşik kümeleme yöntemleri: Tek bağlantı yöntemi, tam bağlantı yöntemi, ortalama bağlantı yöntemi, merkezi bağlantı yöntemi ve varyans yöntemi gibi yöntemlerdir. Hiyerarşik olmayan kümeleme yöntemleri ise: K-ortalama yöntem ve en çok olabilirlik yöntemi gibi yöntemlerdir.

Veri kümelemede kullanılan birçok algoritma vardır [48-51]. Bu algoritmalarda amaç, birbirine çok benzediği düşünülen, fakat özellikleri bakımından birbirinden farklı olan verilerin birbirinden ayrılması ve farklı kümelere bölünmesidir. Şekil 4.1. veri kümelemenin şekilsel temsili ifade etmektedir.

**Şekil 4.1.** Verileri kümelemenin grafiksel temsili

Verilerin birbirlerine olan benzerlikleri veriler arası uzaklık matrisleri ile belirlenir. Uzaklık matrislerinin hesaplanmasında farklı yöntemler mevcuttur. Her uzaklık ölçütü aşağıdaki şartları sağlar. [52]

- 1- $d(i,j) \geq 0$, uzaklık fonksiyonunun değeri negatif olamaz.
- 2- $d(i,i) = 0$, her nokta çözümün kendisine olan uzaklığı sıfırdır.
- 3- $d(i,j) = d(j,i)$, uzaklık fonksiyonunun simetri özelliği vardır.
- 4- $d(i,j) \leq d(i,h) + d(h,j)$, Üçgen Eşitsizliği: İki nokta çözümün arasındaki uzaklık bu iki noktanın üçüncü bir noktaya olan uzaklıkları toplamından küçük olamaz.

Veri kümelemede uzaklık ölçüleri hesaplanırken öklit, manhattan, minkowski ve pearson uzaklık ölçüleri en bilinen yöntemlerdir.

5.ÖNERİLEN YÖNTEM VE SONUÇLAR

Yapay atom algoritmasının çözümlerini temsil eden başlangıç popülasyonu, kovalent ve iyonik çözümler olmak üzere rastgele oluşturulur. Oluşturulan kovalent ve iyonik çözümler toplam popülasyonu temsil eder. Tablo 5.1. popülasyonun genel temsili göstermektedir.

Tablo 5.1. 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)

Tablo 5.1.'deki m tasarımcıya bağlıdır. n ise veri setindeki özellik sayısına bağlıdır. Bu çalışmada m=100 seçilmiştir. n ise 4'tür. Çünkü Iris-Plants veri setinin 4 tane özelliği mevcuttur. Geliştirilen uygulama yazılımında sonuç olarak BEST matris isimli bir çözüm matrisi elde edilmektedir. Popülasyon BEST matris aday olabilecek çözümler topluluğudur. BEST matris elde edilirken uygulama yazılımında oluşturulan popülasyon ile eğitim kümesinin özelliklerinin farkları alınır.

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

$$p_i = \sum_{k=1}^n |(\text{popülasyon}(1, k) - \text{eğitim_kümesi_özellik}(i, k))|$$

p_i , popülasyondaki her bir çözümün manhattan uzaklık değeri, popülasyon(1,k), uygulama yazılımına özel oluşturulmuş çözüm değeri, eğitim kümesi özellik(i,k), eğitim veri seti değerlerinin her biridir.

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 5.2. p_i değerlerinin tutulduğu matrisi temsil etmektedir.

Tablo 5.2. p_i değer matrisi temsili

$p_i(1,1)$	$p_i(1,2)$	$p_i(1,3)$.	.	.	$p_i(1,n)$
$p_i(2,1)$	$p_i(2,2)$	$p_i(2,3)$.	.	.	$p_i(2,n)$
.
.
$p_i(m,1)$	$p_i(m,2)$	$p_i(m,3)$.	.	.	$p_i(m,n)$

$\underbrace{\hspace{15em}}_{\Sigma(\text{fark}(n,:))}$

Tablo 5.3. p_i değerlerinin toplamının matris temsili

$p_top(1,1)$	$p_top(1,2)$.	.	.	$p_top(1,m)$
---------------	---------------	---	---	---	---------------

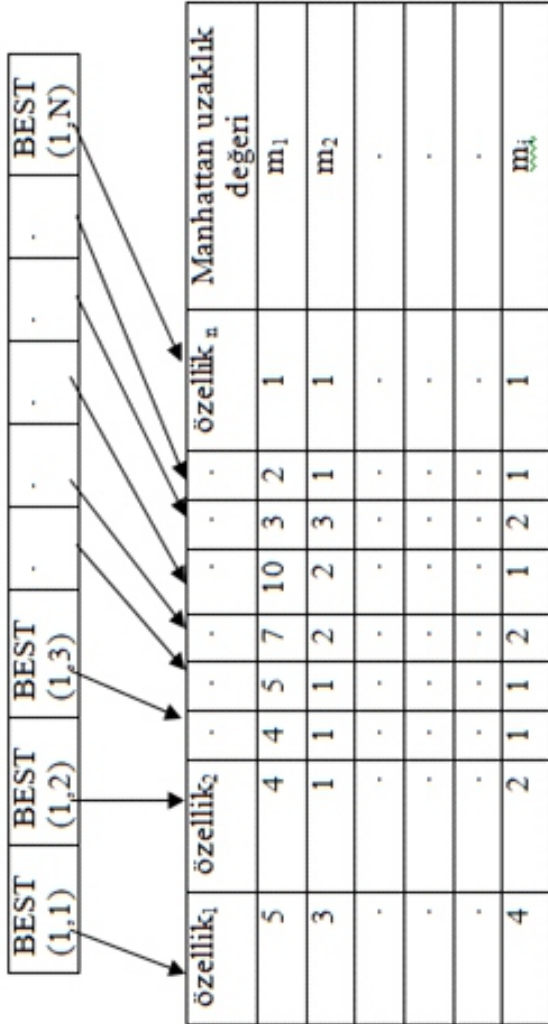
Her çözüm için bir uzaklıklar toplamı hesaplanacak olursa toplam m tane farklar toplamı ortaya çıkar. Tablo 5.3.'ün değerleri içerisinde en küçük olan değer, indeks değerine karşılık gelen popülasyondaki çözüm matrisi o anki adımın BEST matrisidir. Örneğin Tablo 5.3.'teki en küçük değer indeks değeri 7. olan çözüm olsun; bu durumda çözüm popülasyonunun 7. satırı o anki adımın BEST matrisi olur. Algoritma sonlanan 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 matrisine göre, test verisinin hata değerlerinin toplamı olmaktadır. Bu fark değerlerinin sonuç aralığına göre sınıflar belirlenir. Şekil 5.1. bu yöntemin şekilsel temsilini ifade etmektedir.

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

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

m_i , test değerinin manhattan uzaklık değeri, $\text{BEST}(1,k)$, yazılımın üretmiş olduğu BEST matrisin her bir satır elemanı, $\text{test_kümesi_özellik}(i,k)$, test için ayrılan veri setinin her bir satır elemanıdır.



Şekil 5.1. Manhattan uzaklıkları hesabını şekilsel temsili

Geliştirilen yöntemin sonuçları göstermiştir ki, BEST matris test gurubunu manhattan uzaklıklarına göre kümelemeyi başarılı bir şekilde yapmıştır. Tablo 5.5.'te uygulama yazılımının adımlarından birinin BEST matrisi ile, test gurubunun arasındaki manhattan uzaklıklarının hesaplamasından elde edilen sonuçlar sunulmuştur. Bu deneme için BEST matris değerleri Tablo 5.4.'teki gibidir.

Tablo 5.4. Örnek uygulama BEST matris değerleri

BEST(1,1)= 4,3321	BEST(1,2)= 2,1088	BEST(1,3)= 1,1588	BEST(1,4)= 0,2431
----------------------	----------------------	----------------------	----------------------

Tablo 5.5. Uygulama yazılımı manhattan uzaklık değerleri örnek temsili

1.test sınıfı hata değerleri				2.test sınıfı hata değerleri				3.test sınıfı hata değerleri						
5,4	3,4	1,5	0,4	4,4341	5,7	2,9	4,2	1,3	0,3395	6	3	4,8	1,8	1,2659
5,7	3,8	1,7	0,4	4,4341	5,6	3	4,1	1,3	0,4395	5,8	2,7	5,1	1,9	1,7913
5,4	3,9	1,7	0,4	4,7341	5,7	2,8	4,1	1,3	1,5395	6,3	2,7	4,9	1,8	1,9659
5	3,4	1,6	0,4	4,7341	5,7	2,8	4,5	1,3	0,6549	6,5	3,2	5,1	2	2,3605
4,8	3,4	1,9	0,2	4,8341	5,6	3	4,5	1,5	0,6913	6,4	3,1	5,5	1,8	2,3605
4,9	3,1	1,5	0,1	4,9341	6	2,9	4,5	1,5	0,7659	6,5	3	5,2	2	2,3659
4,9	3,1	1,5	0,1	4,9341	6,1	2,8	4	1,3	0,8141	6,5	3	5,5	1,8	2,4659
5,4	3,7	1,5	0,2	4,9341	6,4	2,9	4,3	1,3	0,8295	6,4	2,7	5,3	1,9	2,5659
5,2	3,5	1,5	0,2	4,9341	6,1	2,9	4,7	1,4	0,9659	6,7	3	5,2	2,3	2,8659
5,1	3,4	1,5	0,2	4,9341	5,5	2,5	4	1,3	1,1395	6,9	3,1	5,1	2,3	2,9605
4,8	3	1,4	0,3	5,9395	6,7	3,1	4,4	1,4	1,1605	6,4	2,8	5,6	2,1	2,9659
5	3,4	1,5	0,2	5,0341	6,6	2,9	4,6	1,3	1,3295	6,3	3,4	5,6	2,4	3,2605
5	3,3	1,4	0,2	5,0341	6,7	3,1	4,7	1,5	1,5605	6,7	3,3	5,7	2,1	3,3605
5,1	3,5	1,4	0,3	5,0341	5,5	2,4	3,8	1,1	1,6395	6,9	3,2	5,7	2,3	3,6605
5,1	3,7	1,5	0,4	5,0341	6	2,7	5,1	1,6	1,6659	7,2	3,2	6	1,8	3,7605
5,1	3,8	1,6	0,2	5,2341	5,5	2,4	3,7	1	1,8395	7,3	2,9	6,3	1,8	4,1659
5	3,5	1,3	0,3	5,2341	6,8	2,8	4,8	1,4	1,8659	7,6	3	6,6	2,1	4,9659
4,6	3,2	1,4	0,2	5,3341	6,3	2,5	4,9	1,5	1,8659	7,7	2,8	6,7	2	5,2659
4,6	3,4	1,4	0,3	5,4341	6,9	3,1	4,9	1,5	1,9605	7,9	3,8	6,4	2	5,6605
4,4	3	1,3	0,2	5,5395	5	2,3	3,3	1	2,8395	7,7	3,8	6,7	2,2	5,9605

Tablo 5.5. dikkatlice incelenecek olursa, her test sınıfının örnekleri için manhattan uzaklık değerleri belirli bir değer aralığına çekilmektedir. Tablo 5.5.'te ilk sınıfın manhattan uzaklık değeri (4,4341-5,5395) aralığında, ikinci sınıfın manhattan uzaklık değeri (0,3395-2,8395) aralığında, üçüncü sınıfın manhattan uzaklık değeri ise (1,2659-5,9605) aralığında çıkmaktadır. Bu durumda ikinci gurubun 1, üçüncü gurubun 4 değeri hatalı kümelenebilmektedir. Doğruluk oranı aşağıdaki gibi hesaplanmaktadır.

$$\text{Doğruluk oranı} = (100/\text{Örnek sayısı}) * \text{Doğru hesaplanan örnek sayısı} \quad (5.1)$$

Tablo 5.5.'teki testin doğruluk oranı %91,67 çıkmaktadır.

Bu çalışmada Iris Plants veri seti ile çalışılmıştır. Üzerinde çalışılan veriler 2013 yılında aşağıdaki kaynaktan elde edilmiştir.

[59] : <http://archive.ics.uci.edu/ml/machine-learning-databases/iris/>

Geliştirilen yazılım sonuç olarak BEST matris isimli 1X4 boyutlu bir matris vermektedir. Çünkü veri seti 4 farklı özellik içermektedir. Bu matrisin eğitim kümesi ile farklarının toplamı hata değerleri olacaktır. Bu sonuç Tablo 5.6.'da temsil edilmiştir.

Tablo 5.6. Yapay Atom Algoritması uygulama yazılımı BEST matrisi temsili

BEST(1,1)	BEST(1,2)	BEST(1,3)	BEST(1,4)
-----------	-----------	-----------	-----------

Veri setinin özellikleri Tablo 5.7.'de gösterilmiştir.

Tablo 5.7. Iris Dataset veri örnekleri temsili

sepal length	sepal width	petal length	petal width	class
5,4	3,7	1,5	0,2	1
5,4	3,4	1,5	0,4	1
5,7	3,8	1,7	0,3	1
5	2,3	3,3	1	2
5,5	2,4	3,8	1,1	2
5,5	2,4	3,7	1	2
5,8	2,7	5,1	1,9	3
6	3	4,8	1,8	3
6,3	2,7	4,9	1,8	3
6,3	3,4	5,6	2,4	3

Sepal length: Çanak yaprak uzunluğu,

Sepal width: Çanak yaprak genişliği,

Petal length: Taç yaprak uzunluğu,

Petal width: Taç yaprak genişliğini ifade etmektedir.

Bu örnek değerlere karşılık class değeri ise bitkinin sınıfını göstermektedir. Uygulama yazılımında bu özellik kullanılmayacaktır. Çünkü kümelemede sınıf özellik değeri kullanılmadan kümeler belirlenmeye çalışılır.

Veri setinde 150 farklı örnek mevcuttur. Her bir sınıftan 50 adet örnek vardır. Geliştirilen uygulama yazılımında, her sınıf için 30 örnek algoritmanın eğitim verileri için kullanılmıştır. Her sınıftan 20 örnekte test için bırakılmıştır.

Uygulama yazılımı için kovalent ve iyonik bağları temsilen oluşturulan toplam 100 elemanlı bir çözüm kümesi oluşturulmuştur. Popülasyon oluşturulurken her kümenin en küçük ve en büyük değerlerinden faydalanılır. Amaç sınır değerleri taşmayan bir popülasyon çözümü oluşturabilmektir.

$$\begin{aligned} \text{kovalent}(:,1) &= 4.3 + (7.7 - 4.3) * \text{rand}(n,1) \\ \text{kovalent}(:,2) &= 2 + (4.4 - 2) * \text{rand}(n,1) \\ \text{kovalent}(:,3) &= 1 + (6.9 - 1) * \text{rand}(n,1) \\ \text{kovalent}(:,4) &= 0.1 + (2.5 - 0.1) * \text{rand}(n,1) \end{aligned}$$

$$\begin{aligned} \text{iyonik}(:,1) &= 4.3 + (7.7 - 4.3) * \text{rand}(m,1) \\ \text{iyonik}(:,2) &= 2 + (4.4 - 2) * \text{rand}(m,1) \\ \text{iyonik}(:,3) &= 1 + (6.9 - 1) * \text{rand}(m,1) \\ \text{iyonik}(:,4) &= 0.1 + (2.5 - 0.1) * \text{rand}(m,1) \end{aligned}$$

Burada n ve m değerleri oluşturulacak 100 elemanlı popülasyonun indeks değerleri olup; uygulamada değişik kombinasyonlarda denenmişlerdir. Tablo 5.8. ve 5.9.'da temsilen gösterilmişlerdir.

Tablo 5.8. Kovalent bağ çözüm temsili

K ₁₁	K ₁₂	K ₁₃	K ₁₄
K ₂₁	K ₂₂	K ₂₃	K ₂₄
K ₃₁	K ₃₂	K ₃₃	K ₃₄
K ₄₁	K ₄₂	K ₄₃	K ₄₄
.	.	.	.
.	.	.	.
K _{n1}	K _{n2}	K _{n3}	K _{n4}

Tablo 5.9. İyonik bağ çözüm temsili

\dot{I}_{11}	\dot{I}_{12}	\dot{I}_{13}	\dot{I}_{14}
\dot{I}_{21}	\dot{I}_{22}	\dot{I}_{23}	\dot{I}_{24}
\dot{I}_{31}	\dot{I}_{32}	\dot{I}_{33}	\dot{I}_{34}
\dot{I}_{41}	\dot{I}_{42}	\dot{I}_{43}	\dot{I}_{44}
.	.	.	.
.	.	.	.
\dot{I}_{m1}	\dot{I}_{m2}	\dot{I}_{m3}	\dot{I}_{m4}

Denklem (5.2)'de verildiği üzere rastgele oluşturulan popülasyonun eğitim kümesi ile manhattan uzaklıklarının toplamları hesaplanır. Bu toplam değerleri Tablo 5.10.'da olduğu gibi fark matrisinde temsil edilmiştir.

$$\|x_i - x_j\| = \sum_{k=1}^d |(x_{i,k} - x_{j,k})|$$

Tablo 5.10. Yapay atom algoritması uygulama yazılımı farklar matrisinin temsili

fark(1,1)	fark(1,2)	fark(1,3)	fark(1,4)
fark(2,1)	fark(2,2)	fark(2,3)	fark(2,4)
.	.	.	.
.	.	.	.
fark(100,1)	fark(100,2)	fark(100,3)	fark(100,4)

$$\sum (\text{fark}(k,:))$$

Her çözüm için bir farklar toplamı hesaplanacak olursa toplam 100 tane farklar toplamı ortaya çıkar. Tablo 5.11. bu çözümleri temsil etmektedir

Tablo 5.11. Yapay Atom Algoritması uygulama yazılımı farklar matrisinin toplamının temsili.

farktop(1,1)	farktop(1,2)	.	.	.	farktop(1,100)
--------------	--------------	---	---	---	----------------

Farklar matrisinin değerleri küçükten büyüğe doğru sıralanır ve buna uygun olarak popülasyon çözümleri yeni baştan düzenlenirse elimizde toplam fark değerlerine göre

sıralanmış bir popülasyon oluşur. Bu sıralanmış popülasyonunda 1. çözümü en iyi çözüm olur. Bu işleme algoritma sonlanana kadar devam edilir. Sonuçta en son adımın elde ettiği popülasyonun 1. çözümü BEST matrisi olarak seçilmiş olur.

Yapay Atom Algoritması yapısında parametre barındırmayan bir algoritmadır. Yeni değerleri oluşturmada hesapsal bir yöntem mevcut değildir. En iyi çözümleri bir sonraki jenerasyona taşırken, bir çözüm kendisinden daha iyi olan bir çözüm değerini üzerine kopyalar. Yani bir nevi elektron ortaklaşması gibi iyi çözümler ortaklaşılır. İyonik çözümler ise her jenerasyonda belirli bir oranda yeniden üretilir.

Şimdi uygulama yazılımındaki 5 adımdaki tüm adımlar ve doğruluk oranları sırasıyla verilmiştir.

Adım 1: Bu adımda popülasyondaki kovalent çözüm oranı %30, iyonik çözüm oranı ise %70'dir. 4 farklı özelliğin sadece ilki değiştiğinde doğruluk oranları Tablo 5.12.'de, iki özellik değiştiğinde doğruluk oranları Tablo 5.13.'te, üç özellik değiştiğinde doğruluk oranları Tablo 5.14.'te ve 4 özellik de değiştiğinde doğruluk oranları Tablo 5.15.'te verildiği gibidir.

Tablo 5.12. Adım 1 için 1 özellik değiştiğinde sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,8127	3,0527	4,2423	1,3318	91,67
5,8352	2,9950	4,2515	1,3678	90
5,7522	2,9615	4,2277	1,3749	90
5,6593	2,9996	4,2249	1,3205	91,67
5,7577	3,0303	4,2955	1,3846	90
5,6871	2,9989	4,2131	1,2308	90
5,7702	3,0094	4,2932	1,3013	90
5,7968	3,0349	4,2135	1,2300	91,67
5,6658	3,0198	4,2139	1,3235	90
5,8024	3,0312	4,2004	1,2658	90
				Ortalama = 90,50

Tablo 5.13. Adım 1 için 2 özellik değiştiğinde sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,7630	3,0089	4,2808	1,3006	90
5,7784	3,0163	4,2112	1,3005	90
5,7248	2,9863	4,2497	1,3299	90
5,7350	3,0016	4,2108	1,2954	91,67
5,7863	3,0035	4,2576	1,3102	90
5,7989	3,0036	4,2727	1,3057	90
5,7803	2,9974	4,2373	1,2947	90
5,7808	3,0003	4,2996	1,2799	90
5,7514	2,9902	4,2127	1,3007	90
5,7748	3,0009	4,2314	1,3000	90
				Ortalama = 90,17

Tablo 5.14. Adım 1 için 3 özellik değiştiğinde sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,7532	3,0064	4,2072	1,3021	90
5,7909	2,9986	4,2979	1,3063	90
5,7851	3,0042	4,2741	1,2963	90
5,7817	2,9987	4,2871	1,2884	90
5,8000	2,9985	4,2574	1,3070	90
5,7980	2,9994	4,2563	1,3000	90
5,7636	2,9969	4,2236	1,2998	90
5,7991	3,0023	4,2689	1,3066	90
5,7757	3,0047	4,2827	1,3026	90
5,7874	2,9999	4,2931	1,3093	90
				Ortalama = 90

Tablo 5.15. Adım 1 için 4 özellik değiştiğinde sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,7830	2,9989	4,3074	1,2446	90
5,7328	3,0079	4,2811	1,2943	90
5,7013	3,0004	4,2125	1,3239	91,67
5,7901	3,0085	4,2898	1,2713	90
5,7920	2,9876	4,2864	1,2879	90
5,7518	2,9969	4,2722	1,3064	90
5,7489	3,0201	4,2881	1,2972	90
5,7868	3,0073	4,2442	1,2698	90
5,7615	3,0039	4,2465	1,3032	90
5,7728	3,0081	4,2791	1,3061	90
				Ortalama = 90,17

Adım 2: Bu adımda popülasyondaki kovalent çözüm oranı %40, iyonik çözüm oranı ise %60'dir. 4 farklı özelliğin sadece ilki değiştiğinde doğruluk oranları Tablo 5.16.'da, iki özellik değiştiğinde doğruluk oranları Tablo 5.17.'de, üç özellik değiştiğinde doğruluk oranları Tablo 5.18.'de ve 4 özellik de değiştiğinde doğruluk oranları Tablo 5.19.'da verildiği gibidir.

Tablo 5.16. Adım 2 için 1 özellik değiştiğinde sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,7723	2,9846	4,2414	1,3304	90
5,7276	2,9817	4,2308	1,3456	90
5,7559	2,9582	4,2653	1,2932	90
5,8321	2,9596	4,2163	1,3018	90
5,7638	2,9658	4,2805	1,3055	90
5,7043	3,0287	4,2093	1,3857	90
5,7234	3,0006	4,2771	1,4187	90
5,8538	2,9984	4,2138	1,3574	90
5,7454	3,0156	4,2582	1,3748	90
5,8072	3,0455	4,2079	1,2507	90
				Ortalama = 90

Tablo 5.17. Adım 2 için 2 özellik değiştiğinde sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,7743	3,0071	4,2548	1,2991	90
5,7840	2,9843	4,2604	1,2936	90
5,7475	2,9901	4,2925	1,2972	90
5,7301	2,9968	4,2059	1,2995	91,67
5,7726	2,9853	4,2491	1,2999	90
5,8127	2,9947	4,2826	1,3471	90
5,8076	3,0126	4,2657	1,29992	90
5,7896	2,9975	4,2831	1,2760	90
5,7508	2,9989	4,2878	1,3214	90
5,7671	2,9970	4,2738	1,2911	90
				Ortalama = 90,17

Tablo 5.18. Adım 2 için 3 özellik değiştiğinde sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,7971	2,9989	4,2718	1,3070	90
5,7909	2,9999	4,2679	1,2938	90
5,7968	3,0048	4,2954	1,3022	90
5,8015	3,0018	4,2491	1,3016	90
5,7908	2,9985	4,2078	1,2942	90
5,7957	3,0004	4,2614	1,3039	90
5,7659	2,9954	4,2610	1,2945	90
5,8005	3,0141	4,2506	1,3033	90
5,7790	2,9956	4,2375	1,3286	90
5,7892	3,0027	4,2289	1,2984	90
				Ortalama = 90

Tablo 5.19. Adım 2 için 4 özellik değiştiğinde sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,7457	2,9974	4,2810	1,3290	90
5,7697	3,0080	4,2531	1,3410	90
5,8037	2,9978	4,2515	1,2883	90
5,8051	3,0001	4,2585	1,3389	90
5,7871	3,0029	4,2865	1,3594	90
5,7762	2,9886	4,2086	1,3041	90
5,7975	3,0004	4,2639	1,2961	90
5,6886	3,0213	4,2961	1,3218	90
5,7879	3,0182	4,3090	1,3004	90
5,8076	2,9926	4,2046	1,2900	90
				Ortalama = 90

Adım 3: Bu adımda popülasyondaki kovalent çözüm oranı %50, iyonik çözüm oranı ise %50'dir. 4 farklı özelliğin sadece ilki değiştiğinde doğruluk oranları Tablo 5.20.'de, iki özellik değiştiğinde doğruluk oranları Tablo 5.21.'de, üç özellik değiştiğinde doğruluk oranları Tablo 5.22.'de ve 4 özellik de değiştiğinde doğruluk oranları Tablo 5.23.'te verildiği gibidir.

Tablo 5.20. Adım 3 için 1 özellik değiştiğinde sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,7826	3,0179	4,2815	1,2799	90
5,7748	2,9933	4,2861	1,2933	90
5,6820	3,0049	4,2505	1,2920	91,67
5,7777	2,9760	4,2248	1,3160	90
5,6515	3,0267	4,2150	1,4084	90
5,7490	3,0116	4,2383	1,3320	90
5,7584	3,0059	4,2003	1,1634	91,67
5,7901	2,9890	4,2031	1,2459	90
5,7842	2,9971	4,2218	1,2425	90
5,7671	3,0092	4,2967	1,3145	90
				Ortalama = 90,33

Tablo 5.21. Adım 3 için 2 özellik değiştiğinde sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,8027	2,9981	4,2140	1,2987	90
5,6723	3,0005	4,2542	1,3117	91,67
5,8082	3,0058	4,2944	1,3014	90
5,7662	3,0071	4,2249	1,2900	90
5,7312	2,9939	4,2688	1,2977	90
5,8066	2,9903	4,2980	1,2877	90
5,7925	3,0020	4,2694	1,2951	90
5,7486	2,9983	4,2927	1,3078	90
5,7662	3,0169	4,2568	1,2809	90
5,7723	2,9858	4,2399	1,3026	90
				Ortalama = 90,17

Tablo 5.22. Adım 3 için 3 özellik değiştiğinde sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,8027	2,9987	4,2856	1,2993	90
5,7985	3,0001	4,2293	1,3065	90
5,7737	2,9997	4,2252	1,3098	90
5,7853	2,9939	4,2929	1,3173	90
5,7970	3,0005	4,2857	1,3090	90
5,7927	2,9933	4,2094	1,2955	90
5,7853	3,0001	4,2372	1,3202	90
5,8010	3,0012	4,2376	1,2970	90
5,7743	3,0027	4,2642	1,3198	90
5,7657	2,9963	4,2326	1,3022	90
				Ortalama = 90

Tablo 5.23. Adım 3 için 4 özellik değiştiğinde sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,7670	3,0001	4,3395	1,3279	90
5,7681	3,0162	4,2111	1,3050	90
5,8089	3,0013	4,3410	1,3154	90
5,7649	2,9909	4,2560	1,3251	90
5,7488	2,9951	4,2059	1,3027	90
5,7395	2,9753	4,3007	1,2911	90
5,7710	3,0235	4,2342	1,2989	90
5,7627	2,9862	4,2125	1,3038	90
5,7727	3,0032	4,2629	1,3239	90
5,7615	3,0081	4,1850	1,2664	91,67
				Ortalama = 90,17

Adım 4: Bu adımda popülasyondaki kovalent çözüm oranı %62, iyonik çözüm oranı ise %38'dir. 4 farklı özelliğin sadece ilki değiştiğinde doğruluk oranları Tablo 5.24.'te, iki özellik değiştiğinde doğruluk oranları Tablo 5.25.'te, üç özellik değiştiğinde doğruluk oranları Tablo 5.26.'da ve 4 özellik de değiştiğinde doğruluk oranları Tablo 5.27.'de verildiği gibidir.

Tablo 5.24. Adım 4 için 1 özellik değiştiğinde sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,8542	3,0121	4,2836	1,4034	88,33
5,7048	2,9736	4,2710	1,2730	90
5,8754	2,9961	4,2070	1,2907	90
5,8672	3,0105	4,2215	1,2998	90
5,7111	3,0646	4,2127	1,3008	91,67
5,7864	3,0198	4,2241	1,2779	90
5,6896	2,9737	4,2576	1,3904	90
5,6501	3,0007	4,2150	1,3832	91,67
5,7361	3,0198	4,2859	1,2817	90
5,8031	2,9363	4,2152	1,2398	90
				Ortalama = 90,17

Tablo 5.25. Adım 4 için 2 özellik değiştiğinde sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,8093	2,9998	4,2208	1,2916	90
5,7713	3,0008	4,2804	1,3013	90
5,7611	3,0052	4,2862	1,3083	90
5,7169	2,9976	4,2651	1,3083	90
5,7903	3,0026	4,2393	1,3213	90
5,7357	2,9957	4,2516	1,2979	90
5,7982	2,9941	4,2310	1,3232	90
5,7605	3,0097	4,2661	1,2995	90
5,7784	3,0010	4,2639	1,2973	90
5,7901	2,9875	4,2491	1,2992	90
				Ortalama = 90

Tablo 5.26. Adım 4 için 3 özellik değiştiğinde sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,7917	3,0017	4,2689	1,2972	90
5,7998	2,9990	4,2460	1,3004	90
5,8008	3,0006	4,2153	1,2909	90
5,7928	2,9998	4,2778	1,3062	90
5,7464	3,0012	4,2156	1,3014	90
5,8015	2,9995	4,2220	1,3078	90
5,7905	2,9981	4,2911	1,2962	90
5,7728	2,9990	4,2729	1,3021	90
5,8004	3,0040	4,2513	1,3148	90
5,7999	3,0171	4,2840	1,2890	90
				Ortalama = 90

Tablo 5.27. Adım 4 için 4 özellik değiştiğinde sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,7375	3,0080	4,2806	1,3313	90
5,6879	2,9970	4,2694	1,3200	90
5,7461	2,9919	4,2608	1,3029	90
5,7793	3,0033	4,4425	1,2932	88,33
5,7480	3,0031	4,2744	1,3426	90
5,7912	2,9841	4,2202	1,2649	90
5,7203	3,0112	4,2183	1,2943	91,67
5,8001	3,0061	4,3453	1,3012	90
5,8055	2,9993	4,3582	1,2992	90
5,7852	3,0094	4,3451	1,3175	90
				Ortalama = 90

Adım 5: Bu adımda popülasyondaki kovalent çözüm oranı %80, iyonik çözüm oranı ise %20'dir. 4 farklı özelliğin sadece ilki değiştiğinde doğruluk oranları Tablo 5.28.'de, iki özellik değiştiğinde doğruluk oranları Tablo 5.29.'da, üç özellik değiştiğinde doğruluk oranları Tablo 5.30.'da ve 4 özellik de değiştiğinde doğruluk oranları Tablo 5.31.'de verildiği gibidir.

Tablo 5.28. Adım 5 için 1 özellik değiştiğinde sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,7493	30462	4,2334	1,2543	90
5,7272	2,9985	4,2120	1,3224	90
5,7738	2,9713	4,2663	1,4242	90
5,8793	3,0090	4,2709	1,2933	90
5,7330	2,9458	4,2516	1,3994	90
5,7505	2,9924	4,2549	1,2987	90
5,7167	2,9373	4,2651	1,3575	90
5,8580	3,0333	4,2872	1,3098	90
5,7738	3,0255	4,3048	1,2734	90
5,6819	2,9948	4,2548	1,2420	90
				Ortalama = 90

Tablo 5.29. Adım 5 için 2 özellik değiştiğinde sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,7607	3,0098	4,2801	1,2974	90
5,7035	2,9983	4,2574	1,2908	90
5,7859	3,0016	4,2247	1,2993	90
5,7407	3,0022	4,2339	1,2969	90
5,7824	2,9963	4,2213	1,2921	90
5,7572	3,0094	4,2921	1,3150	90
5,7677	3,0063	4,2822	1,3061	90
5,7635	3,0065	4,2357	1,2979	90
5,8318	2,9962	4,2388	1,2974	90
5,7853	2,9995	4,2134	1,2981	90
				Ortalama = 90

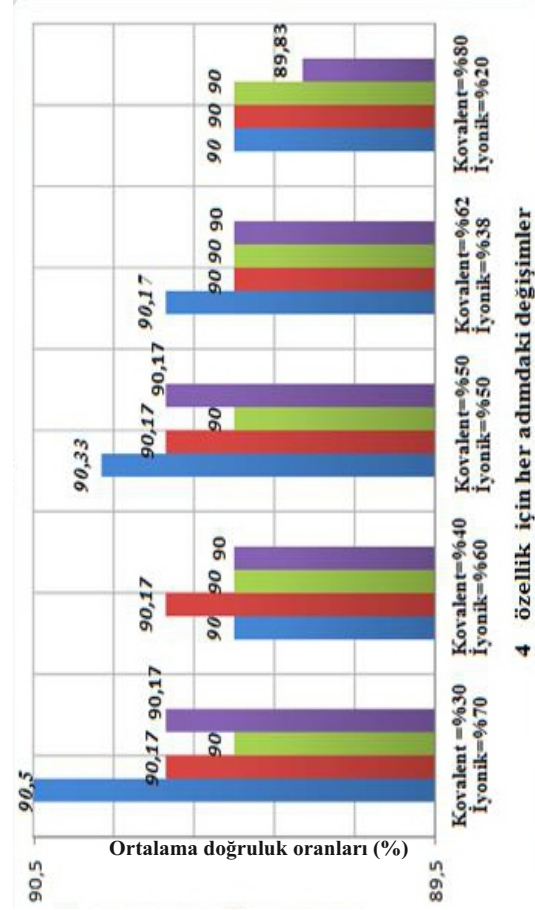
Tablo 5.30. Adım 5 için 3 özellik değiştiğinde sonuçlar

BEST (1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,7969	2,9931	4,2451	1,3002	90
5,7988	3,0008	4,2717	1,2934	90
5,7788	3,0000	4,2547	1,3122	90
5,7930	3,0044	4,2926	1,3013	90
5,7959	2,9976	4,2025	1,3005	90
5,7692	3,0018	4,2417	1,3018	90
5,7992	2,9974	4,2008	1,3543	90
5,7999	2,9998	4,2465	1,2998	90
5,7853	2,9985	4,2937	1,3042	90
5,8016	2,9996	4,2923	1,2975	90
				Ortalama = 90

Tablo 5.31. Adım 5 için 4 özellik değiştiğinde sonuçlar

BEST(1,1)	BEST (1,2)	BEST (1,3)	BEST (1,4)	Doğruluk oranı (%)
5,7345	3,0005	4,2952	1,2986	90
5,7782	3,0048	4,2857	1,2796	90
5,7716	3,0044	4,4009	1,3153	90
5,7718	2,9960	4,2410	1,3222	90
5,7891	2,9985	4,2900	1,3110	90
5,7268	3,0143	4,2065	1,3612	90
5,7966	2,9520	4,2023	1,2607	90
5,7959	3,0212	4,4019	1,3280	88,33
5,7500	3,0131	4,3019	1,3548	90
5,7921	2,9968	4,2959	1,2941	90
				Ortalama = 89,83

Şekil 5.2. tüm adımların ortalama başarı oranlarını göstermektedir. Sonuç olarak veri seti %90,5 oranında ortalama başarı oranı ile kümelenebilmiştir.

**Şekil 5.2.** Yapay Atom Algoritması tüm çözüm durumları grafik temsili

6.TARTIŞMA VE SONUÇ

Yapay atom algoritması, hesapsal yapısı gayet sade olan bir çözüm yöntemi olmasına rağmen, bu çalışma göstermiştir ki; yapısındaki bu sadeliğe rağmen, ayrı zamanlı veriler üzerinde gerçekleştirilen veri kümeleme işlemi için başarılı olmuştur.

Sezgisel yöntemlerde, her jenerasyonda oluşan iyi çözümler etrafında oluşturulacak yeni çözümler, belirli bir formüle bağlı kalınarak oluşturulur. Bu yeni çözümleri oluştururken, en iyi çözüm etrafında bazı parametreler yardımıyla yeni çözümler hesaplanır.

Yapay atom algoritmasında parametre yoktur. Sadece popülasyonun kovalent ve iyonik çözüm oranı ve ortaklaşılan özellik sayısında değişiklik yapılarak sonuca gidilmeye çalışılmıştır.

Uygulamadaki 5 adımın tüm durumları göz önüne alındığına görülmüştür ki; Yapay atom algoritması yapısında parametre bulundurmaması ve yeni çözümleri sadece çözüm ortaklaşma ile oluşturmasına rağmen, %90,5 oranında bir başarı ile veri setini doğru kümeleyebilmiştir.

KAYNAKÇA

- [1] Kaymaz, İ., Optimizasyon Teknikleri, <http://194.27.49.11/makine/ikaymaz/optimizasyon/dosyalar/DERS1OptimizasyonGiris.pdf>, 2014.
- [2] Murty, K. G., Optimization Models For Decision Making, 1, Internet Edition, Chapter 1: Models for Decision Making, 1-18, 2003.
- [3] Akyol, S., Alataş, B. Güncel Sürü Zekası Optimizasyon Algoritmaları, *Nevşehir Üniversitesi Fen Bilimleri Enstitü Dergisi*, 1, 1, 36-50, 2012.
- [4] Lamberti, L. Pappalettere, C. Weight optimization of skeletal structures with multi-point simulated annealing, *Computer Modelling in Engineering and Sciences*, 18, 3, 183-221, 2007.
- [5] Precup, R. E., David, R. C., Petriu, E. M., Preitl, S., Paul, A. S. Gravitational search algorithm-based tuning of fuzzy control systems with a reduced parametric sensitivity. *Soft Computing in Industrial Applications*, 141-150, 2011.
- [6] Birbil, S. I. Fang, S. C. An electromagnetism-like mechanism for global optimization, *Journal of Global Optimization*, 25, 263-282, 2003.
- [7] Özdağ, R. Karcı, A. The Application of Electromagnetism-like Algorithm for the Dynamic Deployment Problem in Wireless Sensor Networks, in Proc. 2nd International Eurasian Conference on Mathematical Sciences and Applications, Sarajevo, Bosnia and Hersegovina, 199, 2013.
- [8] Erol, O. K. Eksin, İ. A new optimization method: Big bang-big crunch, *Advances in Engineering Software*, 37, 2, 106-111, 2006.
- [9] Yang, X. A New Metaheuristic Bat-Inspired Algorithm, *NICSO* 284, 65-74, 2010.
- [10] Yang, X., Gandomi, A.H. Bat Algorithm: A Novel Approach for Global Engineering Optimization, *Engineering Computations*, 29, 5, 464-483, 2012.
- [11] Yang, X. Bat Algorithm for Multi-Objective Optimisation, *Int. J. Bio-Inspired Computation*, 3, 5, 267-274, 2011.
- [12] Tsai, J.-T. Solving Japanese nograms by Taguch-based genetic algorithm, *Applied Intelligence*, 37, 3, 405-419, 2012.
- [13] Xing, H. Qu, R. A compact genetic algorithm for the network coding based resource minimization problem, *Applied Intelligence*, 36, 4, 809-823, 2011.
- [14] Rivero, J., Coadra, D., Calle, J., Isasi, P. Using the ACO algorithm for path searches in social networks, *Applied Intelligence*, 36, 4, 899-917, 2011.
- [15] Karaboğa, D., Baştürk, B. A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm, *Journal of Global Optimization*, 39, 3, 459-471, 2007.
- [16] Akay, B., Karaboğa, D. A Modified Artificial Bee Colony Algorithm for Real-Parameter Optimization, *Information Sciences*, 192, 1, 120-142, 2012.
- [17] De Castro, L. N. Von Zuben, F. J. Learning and optimization using the clonal selection principle, *IEEE Transactions on Evolutionary Computation*, 6, 3, 239-251, 2002.
- [18] A. Karcı, Theory of saplings growing-up algorithm, in Proc. ICANNGA-2007: Adaptive and Natural Computing Algorithms, Editors: Bartłomiej Beliczynski, Andrej Dzielinski, Marcin Iwanowski, Bernardete Ribeiro, LNCS, vol. 4431, 450-460, 2007.
- [19] Karcı, A. Alataş, B. Thinking Capability of Saplings Growing Up Algorithm, in Proc. IDEAL-2006: 7th International Conference on Intelligent Data Engineering and Automated Learning,

LNCS, 4224, 386-393, 2006.

- [20] Karcı, A. Saplings Sowing and Growing up Algorithm Convergence Properties, in Proc. INISTA-2007: International Symposium on Innovations in Intelligent Systems and Applications, Yıldız Technical University, İstanbul, 322-326, 2007.
- [21] Karcı, A. Yiğiter, M. Demir, M. Natural Inspired Computational Intelligence Method: Saplings Growing Up Algorithm, in Proc. Ikecco'2007 International Kyrgyz-Kazak Electronics And Computer Conference, Bishkek-Almaty, 1-8, 2007.
- [22] Demir, M., Yiğiter, M., Karcı, A. Application of Saplings Growing Up Algorithm to Clustering Medical Data, in Proc. Ikecco'2007 International Kyrgyz-Kazak Electronics And Computer Conference, Bishkek-Almaty, 9-15, 2007.
- [23] Demir, M., Karcı, A., Özdemir, M. Fidan Gelişim Algoritması Yardımı ile DNA Motiflerinin Keşfi, *Çankaya University Journal of Science and Engineering*, 8, 1, 51-62, 2011.
- [24] Mehrabian, A. R., Lucas, C. A novel numerical optimization algorithm inspired from weed colonization, *Ecological Informatics*, 1, 4, 355-366, 2006.
- [25] Mucherino, A. Seref, O. Monkey search: A novel metaheuristics search for global optimization I. Continuous parameter optimization, *Evolutionary Computation*, 953, 1, 25-49, 2007.
- [26] Passino, K. M. Biomimicry of bacterial foraging for distributed optimization and control, *IEEE Control Systems Magazine*, 22, 3, 52-67, 2002.
- [27] Canayaz, M., Karcı, A. A New Metaheuristic Cricket-Inspired Algorithm, in Proc. 2nd International Eurasain Conference on Mathematical Sciences and Applications, Prishtine, Kosovo, 83-84, 2012.
- [28] Ülker, E. D., Haydar, A. Comparing the Robustness of Evolutionary Algorithms on the Basis of Benchmark Functions, *Advances in Electrical and Computer Engineering*, 13, 59-64, 2013.
- [29] Niizuma D., Yasuda, K. A. Ishigame, Multi-point tabu search for traveling salesman problems, *IEEE Transactions on Electrical and Electronic Engineering*, 1, 1, 126-129, 2006.
- [30] Gargari, E. A. Lucas, C. Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition, in Proc. IEEE congress on evolutionary computation, Singapore, 4661-4667, 2007.
- [31] Alataş, B. ACROA: Artificial Chemical Reaction Optimization Algorithm for Global Optimization, *Expert Systems with Applications*, 38, 13170-13180, 2011.
- [32] Karcı, A. A new Metaheuristic Algorithm Based Chemical Process: Atom Algorithm, in Proc. 1st International Eurasain Conference on Mathematical Sciences and Applications, Prishtine, Kosovo, 83-84, 2012.
- [33] Yıldırım, A. E., Karcı, A. Solutions of Travelling Salesman Problem Using Genetic Algorithm and Atom Algorithm, in Proc. 2nd International Eurasain Conference on Mathematical Sciences and Applications, Sarajevo, Bosnia and Hercegovina, 134, 2013.
- [34] Karadoğan, A., Karcı, A. Artificial Atom Algorithm for Reinforcement Learning, in Proc. 2nd International Eurasain Conference on Mathematical Sciences and Applications, Sarajevo, Bosnia and Hercegovina, 379, 2013.
- [35] Yang, X.-S. Firefly algorithm, Levy flights and global optimization, in Proc. Research and Development in Intelligent Systems XXVI (Eds M. Bramer, R. Ellis, M. Petridis), Springer London, 209-218, 2010.

- [36] Yang, X.-S. Firefly algorithms for multimodal optimization, *Stochastic Algorithms: Foundations and Applications, Lecture Notes in Computer Science*, Springer-Verlag, Berlin, 5792, 169-178, 2009.
- [37] Yang, X.-S. Firefly algorithm, stochastic test functions and design optimisation, *International Journal of Bio-Inspired Computation*, 2, 2, 78–84, 2010.
- [38] Yang, X.-S. Harmony Search as a Metaheuristic Algorithm, *Music-Inspired Harmony Search Algorithm: Theory and Applications, Studies in Computational Intelligence*, Springer Berlin, 191, 1-14, 2009.
- [39] Kennedy, J. Eberhart, R. C. Particle swarm optimization, in Proc. of IEEE international conference on neural Networks, Australia, 4, 1942-1948, 1995.
- [40] El-Hefnawy, N. A. Solving Bi-level Problems Using Modified Particle Swarm Optimization Algorithm, *International Journal of Artificial Intelligence*, 12, 2, 88-101, 2014.
- [41] Ortakçı, Y., Göloğlu, C., Parçacık Sürü Optimizasyonu İle Küme Sayısının Belirlenmesi, Akademik Bilişim'12 - XIV. Akademik Bilişim Konferansı, Uşak, 335-341, 2012.
- [42] Eberhart, R., Kennedy, J. A New Optimizer Using Particle Swarm Theory, Sixth International Symposium on Micro Machines and Human Science, Nagoya, 39-43, 1995.
- [43] Özsağlam, M. Y., Cunkaş, M. Optimizasyon Problemlerinin Çözümü için Parçacık Sürü Optimizasyonu Algoritması , *Politeknik Dergisi*, 11, 4, 299-305, 2008.
- [44] Chu, S. C., Tsai, P.W., Pan, J. S. Cat swarm optimization, *PRICAI 2006: Trends in Artificial Intelligence Lecture Notes in Computer Science*, 4099, 854-858, 2006.
- [45] Lee, K. S., Geem, Z. W. A new metaheuristics algorithm for continues engineering optimization: Harmony search theory and practice, *Computer Methods in Applied Mechanics and Engineering*, 194, 3902-3933, 2005.
- [46] Karaboğa, D., Yapay Zekâ Optimizasyon Algoritmaları, Nobel Yayın Dağıtım, Ankara, 2011.
- [47] Durmuş, M. S., İplikçi, S. Veri Kümeleme Algoritmalarının Performansları Üzerine Karşılaştırmalı Bir Çalışma, Akademik Bilişim'07-IX. Akademik Bilişim Konferansı Bildirileri, Kütahya, 393-400, 2007.
- [48] Jain, A. K., Murty, M. N., Flynn, P. J. Data Clustering: A Review, *ACM Computing Surveys*, 31, 3, 265-323, 1999.
- [49] Hartigan J. A., *Clustering Algorithms*, John Wiley & Sons Inc., New York, 1975