T.C.
ERCİYES ÜNİVERSİTESİ
BİLİMSEL ARAŞTIRMA PROJELERİ
KOORDİNASYON BİRİMİ

VERİ MADENCİĞİ PROBLEMLERİ İÇİN ÖLÇEKLENEBİLİR ARI KOLONİ ALGORİTMALARININ GELİŞTİRİLMESİ

Proje No: EÜBAP - FBA-09-889

Normal Araştırma Projesi

SONUÇ RAPORU

Proje Yürütücüsü:
Yrd. Doç. Dr. Mete ÇELİK
Mühendislik Fakültesi / Bilgisayar Mühendisliği

Aralık 2016

KAYSERİ

TEŞEKKÜR

Veri Madenciliği Problemleri için Ölçeklenebilir Arı Koloni Algoritmalarının Geliştirilmesi başlıklı FBA-09-889 nolu araştırma projesinin yapılabilmesi için ihtiyaç duyulan maddi desteği sağlayan Erciyes Üniversitesi Bilimsel Araştırma Projeleri birimine teşekkür ederim.

Desteklerinden ötürü sayın Prof. Dr. Derviş Karaboğa'ya ve sayın Arş. Gör. Fehim Köylü'ye teşekkürü borç bilirim. Ayrıca çalışmaların tamamlanması için projenin çeşitli iş paketlerinde katkıda bulunan lisans ve yüksek lisans öğrencilerine teşekkür ederim.

Yrd. Doç. Dr. Mete ÇELİK

Kayseri, Aralık 2016

İÇİNDEKİLER

ÖZET 5
ABSTRACT 6
1. GİRİŞ / AMAÇ VE KAPSAM 7
2. GENEL BİLGİLER 9
2.1 Literatür Özeti 9
3. YÖNTEM 11
3.1. Yapay Arı Koloni Algoritmasının Analizi 11
3.2. Paralel ABC Algoritması 13
3.3. Yapay Arı Koloni Algoritması Tabanlı (ABC-tabanlı) Sınıflandırma Algoritması 16
3.4. İşbirlikçi Yapay Arı Koloni Algoritması Tabanlı (ABC-tabanlı)Sınıflandırma Algoritması 19
4. BULGULAR 22
4.1. Paralel ABC Algoritması Performans Deneyleri 22
4.2. ABC-tabanlı Sınıflandırma Algoritmasının Performans Deneyleri 25
4.3. İşbirlikçi ABC-tabanlı Sınıflandırma Algoritmasının Performans Deneyleri 27
5. TARTIŞMA VE SONUÇ 29
KAYNAKLAR 30

ÖZET

Son yıllarda yaşanan teknolojik gelişim çok sayıda verinin veritabanlarında tutulmasına imkân sağlamıştır. Veri tabanlarında tutulan bu verilerin etkin ve doğru bir biçimde analizi pek çok uygulama alanında karar verme süreçlerine büyük katkılar sağlayacaktır. Ancak üretilen verilerin analizi için yapılan manuel çabalar çoğu zaman yetersiz kalmaktadır. Veri analizindeki başarıyı artırmak ve doğru sonuçlara en kısa sürede ulaşabilmesi için büyük veri kümelerinden anlamlı ve ilginç örüntülerin çıkarılması (madenciliği) için otomatik ve yarı-otomatik tekniklerin geliştirilmesi gerekmektedir.

Bu projenin amacı veri madenciliği problemlerini çözmek için (sınıflandırma problemi gibi) yeni ve özgün yapay arı algoritması (ABC-Artificial Bee Colony) tabanlı veri madenciliği algoritmalarının geliştirilmesidir.

Proje kapsamında yapılan çalışmalarda ABC algoritmasının analizi yapılmıştır. Daha sonra paralel ABC algoritması geliştirilerek ABC algoritmanın hızlandırılması amaçlanmış ve ABC-tabanlı sınıflandırma algoritması ve işbirlikçi ABC-tabanlı sınıflandırma algoritması geliştirilecek veri madenciliği sınıflandırma problemlerinin çözümü için uygulanmıştır. Geliştirilen algoritmaların performansları bilinen fonksiyonların ve sentetik ve gerçek veri kümelerinin analizinde incelenmiştir. Bu algoritmalar literatürdeki benzerleri ile karşılaştırılmıştır. Deney sonuçları geliştirilen algoritmaların literatürdeki diğer algoritmalara göre başarılı sonuçlar verdiğini göstermiştir.

Anahtar Sözcükler: Yapay Arı Kolonisi, Veri Madenciliği, Sınıflandırma

ABSTRACT

Technological developments in recent have lead the storage of big amount of data in the databased. Effective analysis of this data will contribute to the decision-making process in several application domains. However, manuel analysis of such data is insufficient and takes lots of time. Automatic and semi automatic data analysis techniques should be developed to analyze these datasets to achieve correct and meaningful patterns.

The aim of this project is to develop new artificial bee colony (ABC) based data mining algorithms.

In this project the analysis of ABC algorithm is conducted. Then a parallel ABC algorithm is developed to speedup the classical ABC algorithm and ABC-based classification algorithm and cooperative ABC-based classification algorithm are developed to solve classification problems in data mining. The performances of the algorithms are compared the performances of the algorithms proposed in the literature. In the analysis known function and datasets were used. Exerimental results show that the algorithm showed high performance with respect to the ones proposed in the literature.

Keywords: Artificial Bee Colony, Data Mining, Classification

1. GİRİŞ / AMAÇ VE KAPSAM

Son yıllarda yaşanan teknolojik gelişim çok sayıda verinin veritabanlarında tutulmasına imkân sağlamıştır. Veri tabanlarında tutulan bu verilerin etkin ve doğru bir biçimde analizi pek çok uygulama alanında karar verme süreçlerine büyük katkılar sağlayacaktır. Ancak üretilen verilerin analizi için yapılan manuel çabalar çoğu zaman yetersiz kalmaktadır. Veri analizindeki başarıyı artırmak ve doğru sonuçlara en kısa sürede ulaşabilmesi için ve büyük veri kümelerinden anlamlı ve ilginç örüntülerin çıkarılması (madenciliği) için otomatik ve yarı-otomatik tekniklerin geliştirilmesi gerekmektedir.

Doğadaki arı sürüsünün davranışı kendi başına organize olabilmesi ve iş bölümü yapabilmesi ile karakterize edilebilir. Son yıllarda, araştırmacılar çeşitli yapay sistemleri geliştirebilmek için sosyal böceklerin (arı, karınca gibi) davranışlarını çalışarak sürü zekâsı kavramını kullanmaya başlamışlardır. Arı koloni optimizasyonu en son geliştirilen sürü zekâsı algoritmalarındandır. Yapılan çalışmalarda arı koloni optimizasyonunun diğer sürü zekası algoritmalarına göre daha doğru sonuçlar verdiği görülmüştür.

Bu projenin amacı veri madenciliği problemlerini çözmek için (sınıflandırma problemi gibi) yeni yapay arı algoritması (ABC-Artificial Bee Colony) tabanlı veri madenciliği algoritmalarının geliştirilmesidir.

Proje kapsamında yapılan çalışmalarda ABC algoritmasının analizi yapılarak algoritmanın darboğazları ve en çok zaman tükettiği yerler belirlenmiştir. Daha sonra paralel ABC algoritması geliştirilerek ABC algoritmanın hızlandırılması amaçlanmış ve ABC-tabanlı sınıflandırma algoritması ve işbirlikçi ABC-tabanlı sınıflandırma algoritması geliştirilecek veri madenciliği sınıflandırma problemlerinin çözümü için uygulanmıştır. Algoritmaların performansları bilinen fonksiyonların ve sentetik ve gerçek veri kümelerinin analizinde incelenmiştir.

ABC-tabanlı sınıflandırma algoritması sonuçları derlenerek " International Symposium On Innovations in Intelligent Systems and Applications" konferansında bildiri olarak sunulmuştur [22].

İşbirlikçi ABC-tabanlı sınıflandırma algoritması sonuçları derlenerek " International Journal On Artificial Intelligence Tools" dergisinde makale olarak yayınlanmıştır [24].

2. GENEL BİLGİLER

Bu projenin amacı veri madenciliği problemleri için (sınıflandırma gibi) etkin yapay arı kolonisi algoritmalarının geliştirilmesidir.

2.1 Literatür Özeti

Literatürde şu ana kadar yapılan çalışmalar arı algoritmasının teorisini ve çeşitli alanlarına uygulamasını kapsamaktadır [2-6]. Yang iki boyutlu nümerik problemler için Virtual Bee algoritmasını geliştirmiştir [6]. Fakat geliştirilen bu algoritma iki boyuttan fazla boyutlu problemleri çözmekte yetersiz kalmıştır. Buna karşılık Karaboğa ikiden fazla boyutlu problemlerin çözümü için Yapay Arı Kolonisi algoritmasını önermiştir [7-10]. Pham ve arkadaşları da benzer amaçla arı algoritması geliştirmişlerdir [14]. Geliştirilen bu algoritmalar çok-boyutlu problemlere uygulandığı zaman çözüm vermesine rağmen özellikle veritabanının çok-büyük olduğu durumlarda düşük performans sergilemişlerdir yani ölçeklenebilir olamamışlardır.

Yukarıda listelenen arı algoritmaları çeşitli uygulama alanlarındaki problemleri çözmek için kullanılmışlardır. Karaboğa ve arkadaşları tarafından yapay sinir ağlarının eğitilmesi için yapay arı koloni algoritması kullanılmıştır [15, 16]. Geliştirilen ABC algoritması literatürdeki populasyon-tabanlı yaklaşımlarla karşılaştırılmıştır (Differantial Evolution, DE) [11] ve Parçacık Sürüsü Optimizasyon (Particle Swarm Optimizasyon, PSO) [12]) ve arı algoritmasının doğruluğu daha yüksek çözümler sunduğu görülmüştür. Pham ve arkadaşları tarafından arı algoritması bir robot jimnastikçi için kullanılan bulanık lojik kontrolörünü ayarlamak için kullanılmıştır [14]. Bu çalışmada bulanık lojik kontrolörünün üyelik fonksiyonunun eniyileştirilmesi amaçlanmıştır. Bahamish ve arkadaşları tarafından bal arısı koloni optimizasyonu kullanılarak protein oluşumlarının aranması ve tahmin edilmesi sağlanmıştır [1]. Wong ve arkadaşları gezgin satıcı problemini çözmek için arı algoritmasını kullanmışlardır [13].

Literatürde çeşitli sezgisel algoritmalar kullanılarak veri madenciliği sınıflandırma algoritmaları geliştirilmiştir. Genetik algoritma kullanılarak sınıflandırma kurallarının

çıkarımı yapılmıştır [17, 18, 19]. Karınca kolonisi algoritması kullanılarak Ant-Miner algoritması önerilmiştir [23]. Parçacık sürü optimizasyonu kullanılrak PSO DataMiner önerilmiştir [20].

Klasik sezgisel veri madenciliği algoritmalarına ek olarak literatürde işbilikçi yaklaşımlar da çalışılmıştır. Liretaürdeki bu algoritmalara örnek olarak HİDER [26] ve CORE [27], verilebilir.

Literatürde geliştirilen arıların davranışını modelleyen algoritmalar, diğer popülasyon-tabanlı algoritmalara göre daha iyi ve doğru çözümler sunmasına rağmen, miktarı ve boyutu büyük yani çok değişkenli veri kümelerini analiz etmekte yetersiz kalmışlardır ve gerektiği zamanlarda gereken çözümleri hızlı bir şekilde üretmekte istenen performansı gösterememişlerdir. Bu nedenle de geliştirilen arı algoritmalarını çoğu veri madenciliği problemlerine uygulanması sınırlı kalmıştır.

Bu kapsamda, bu çalışmada paralel ABC algoritması ve iki adet ABC sınıflandırmaları önerilmiştir.

3. YÖNTEM

Proje kapsamında öncelikle yapay arı algoritmasının analizi yapılarak hangi işlem adımlarının ne kadar zaman tükettikleri belirlenmeye çalışılmıştır ve daha sonra 3 adet yapay arı algoritması tabanlı algoritma önerilmiştir. Birinci algoritma paralel yapay arı algoritmasıdır ve diğer iki algoritma ise yapay arı algoritmasına dayalı veri madenciliği algoritmasıdır. Önerilen veri madenciliği algoritmaları sınıflandırma problemine odaklanmıştır [22, 24].

Yapay arı algoritmasının analizindeki amaç hangi işlem adımının veya fonksiyonun ne kadar zaman tükettiğinin tespit edilmesidir. Elde edilen sonuçlar özellikle paralel yapay arı algoritmasının geliştirilmesinde ve diğer algoritmaların geliştirilmesinde faydalı olmuştur. Önerilen algoritmaların veri madenciliği problemlerine uygulanması ve sonuçların değerlendirilmesi ve yorumlanması işlemleri ise bu raporun sırasıyla Bulgular ve Tartışma ve Sonuç bölümlerinde verilmiştir.

3.1. Yapay Arı Koloni Algoritmasının Analizi

Bu aşamada öncelikle ABC algoritmasının analizi yapılarak algoritmanın darboğazları ortaya konulmaya çalışılmıştır. Bu amaçla algoritmanın değişkenlerinin ve fonksiyonlarının arasındaki bağımlılıklar incelenmiştir. İdeal olarak bir algoritmada fonksiyonlar arasında fonksiyonlar çalışırken birbirlerine bağımlılıklarının olmaması tercih edilir. Böyle bir algoritmayı ölçeklenebilir hale getirmek kolaydır, çünkü fonksiyonlar bağımsız olarak eş-zamanlı bir şekilde çalışabilirler.

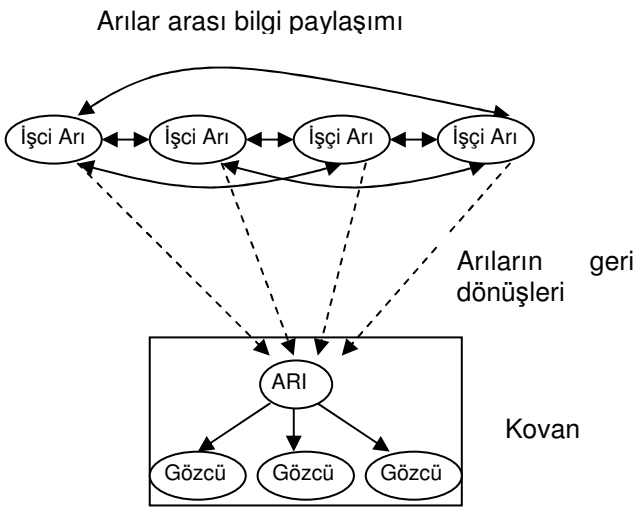
Yapay arı koloni algoritması kabaca üç ana aşamada çalışmaktadır (Algoritma 1). Birinci aşamada işçi arılarının (EmployedBees) kalite değerleri hesaplanmakta ve daha sonra işçi arılar kovandaki gözcü arılar (OnlookerBees) ile bilgi paylaşımı yapmaktadır (Adım 3 ve 4). İkinci aşamada daha önce işçi arılık vazifesi yapmamış olan gözcü arılar, işçi arılardan bilgileri almaktadırlar ve bu aşamada gözcü arılarında kalite fonksiyonları hesaplanmaktadır (Adım 5 ve 6). Son aşamada ise algoritmanın arama uzayında lokal minimuma takılmaması için kaşif arılar rastgele çözümler üretmekte ve bu çözümlerin kalite değerleri hesaplanmaktadır (Adım 7 ve 8). Yapay arı koloni algoritmasının işlem adımları Şekil 2'de

şematize edilmiştir. Yapay arı koloni algoritması hakkındaki detaylı bilgiye [7] nolu referanstan ulaşılabilir.

Algoritma 1: Yapay arı koloni algoritması

Girişler
MDöngü: Toplanm iterasyon sayısı
CS: Toplam arı sayısı
Min_yiyecek: Bir yiyecek kaynağının terk edilemesi için gerekli alt yiyecek limiti
Çıktılar: Optimizasyon sonuçları
Algoritma
1- Algoritmayı ilklendir : Kullanıcı-tanımlı parametrelere göre değişkenleri ata.
2- do
3- İşçi arıları yiyecek kaynaklarına gönder ve nektar miktarlarını hesapla
4- Komşu arılar ile bilgilerini paylaş ve kısmı çözümü üret
5- Gözcü arıları yiyecek kaynaklarına gönder ve nektar miktarlarını hesapla
6- Komşu arılar ile bilgilerini paylaş ve kısmı çözümü üret
7- Kâşif arıları yeni kaynaklar bulmak için gönder
8- Komşu arılar ile bilgilerini paylaş ve kısmı çözümü üret
9- Arıları kovana gonder ve bilgi paylaşımını sağla.
10- En iyi çözümü hafızada tut
11- while (toplam iterasyon sayısı kadar veya kriterler sağlanana kadar)

Yapılan çalışmalarda en çok zaman tüketen işlemin en iyi çözümlerin bulunmasında yapılan tekrarlama işleminin olduğu tespit edilmiştir ve bu işlemler Algoritma 1'de verilen Adım 2 ve 11 arasındaki işlemlere karşılık gelir. Buna ek olarak işçi, gözcü ve kaşif arıları arasındaki veri alış-veriş işlemleri sırasında bilgi paylaşımından ve kovana geldiklerinde arıların birbirleri arasındaki bilgi paylaşımlarından dolayı da birbirlerine karşı bağımlılıkları bulunmaktadır (Adım 3, 4, 6 ve 8) (Şekil 1).



Şekil 1. Arıların birbirlerine bağımlılıkları. Kesikli çizgiler arıların kovana dönüp bilgi paylaşımlarını ifade eder. Diğerleri arıların kendi aralarında olan bilgi paylaşımlarını ifade eder.

Yapay arı koloni algoritması üzerinde yapılan analizlerde, algoritmada en çok işlem gücünün harcandığı yerler olarak %42'lik oran ile işçi arıları işlemleri safhasında (Adım 3 ve 4) ve %57 oran ile gözcü arıları işlemleri safhasında (Adım 5 ve 6) ve geri kalan işlemlerinde %1'lik oran ile gerçekleştiği tespit edilmiştir. Proje kapsamında işçi arıları işlemleri safhasının paralelleştirilmesi işlemleri üzerine yoğunlaşılmıştır. Çalışma detayları bu raporun Bulgular ve Tartışma ve Sonuç bölümlerinde verilmiştir.

Proje kapsamında, yapay arı koloni algoritması tabanlı 3 yeni algoritmanın geliştirilmesi çalışmaları yapılmıştır. Bunlar Paralel ABC algoritması, ABC-tabanlı sınıflandırma algoritması ve İşbirlikçi ABC-tabanlı sınıflandırma algoritmasıdır. Herbir algoritmanın detayları alt bölümlerde tartışılmıştır.

3.2. Paralel ABC Algoritması

Bir önceki bölümde yapılan çalışmalar neticesinde yapay arı koloni algoritmasının en çok zaman tüketen adımlardan birinin de işçi ayrıları safhasının olduğu ortaya konulmuştur. Bu kısmın işlem zamanının azaltılması için proje kapsamında paralel ABC algoritması geliştirilmiştir. Paralel ABC algoritması Algoritma 2'de de görüldüğü üzere işçi arıları işlem

safhası olan Adım 4 ve 5 'in paralelleştirilmesi için önerilmiştir. Bu safhanın ilk safha olması ve diğer gözcü ve kaşif arıları safhalarını etkilemesi nedeniyle işçi arıları safhasının paralelleştirilmesine odaklanılmıştır.

Algoritma 2: Paralel ABC Algoritması

```
Girişler
MDöngü: Toplanm iterasyon sayısı
CS: Toplam arı sayısı
Min_yiyecek: Bir yiyecek kaynağının terk edilemesi için gerekli alt yiyecek limiti
Çıktılar: Optimizasyon sonuçları
Algoritma
    1- Algoritmayı ilklendir : Kullanıcı-tanımlı parametrelere göre değişkenleri ata.
    2- do
    3-         Paralel Başla
    4-                 İşçi arıları yiyecek kaynaklarına gönder ve nektar miktarlarını
       hesapla
    5-                         Komşu arılar ile bilgilerini paylaş ve kısmı çözümü üret
    6-         Paralel Son
    7-         Gözcü arıları işçi arılardan alınan bilgilere göre yiyecek kaynaklarına
       gönder ve nektar miktarlarını hesapla
    8-                 Komşu arılar ile bilgilerini paylaş ve kısmı çözümü üret
    9-         Kâşif arıları yeni kaynaklar bulmak için gönder
    10-                Komşu arılar ile bilgilerini paylaş ve kısmı çözümü üret
    11-        Arıları kovana gonder ve bilgi paylaşımını sağla.
    12-        En iyi çözümü hafızada tut
    13- while (toplam iterasyon sayısı kadar veya kriterler sağlanana kadar)
```

Paralelleştirme işlemi C/C++ ortamında yapılmıştır ve OpenMP paralelleştirme komutları kullanılarak gerçekleştirilmiştir. OpenMP yaklaşımda herbir paralel işlem ortak bir hafızayı kullaranak veri paylaşımı yaparlar. Yapay arı koloni algoritmasındaki işçi arıların davranışları düşünüldüğünde bu yaklaşımın mantıklı olacağı görülecektir. Çünkü işçi arılar öncelikle birbirlerinden bağımsız olarak bir çözüm bulurlar daha sonra kovana gelerek (paylaşımlı hafıza) bilgilerini diğer arılara aktarırlar ve diğer arılardan yeni bilgiler alırlar ve daha sonra

tekrar bir çözüm bulma işlemine giderler. Benzer şekilde OpenMP yaklaşımında da işlemler işlemcilerde ayrı ayrı işlenir veri gerektiği zaman ise ortak hafızadan alırlar. Ortak hafızada yapılan herhangi bir güncelleme ise güncellemeden sonraki işlemleri etkiler. Bu yaklaşımda arıların birbirleri arasındaki bilgi paylaşımı olarak düşünülebilir.

Çalışmada her bir işlemciye ait ön-bellek (cache) yapısının işlemciler arasındaki en güncel halinin tutulabilmesi için, öncelikli olarak önceden tanımlanmış 1. dizi üzerinden okunduktan sonra, okunan bu veriler işlenip 2. başka bir diziye yazılmaktadır. İşçi arılar safhası bitirildikten sonra ise 2. dizideki veriler 1. dizinin üstüne yazılmaktadır. Paralel yapay arı koloni algoritmasının çalıştırılması sonucunda elde edilen sonuçlar bu raporun Bulgular ve Tartışma ve Sonuç kısımlarında sunulmuştur. Paralel ABC algoritmasının çalışma adımları aşağıda açıklanmıştır.

Adım 1'de yapay arı koloni algoritmasına öncelikle kullanıcı tarafından belirli parametreler girilir. Bu parametreler literatürde hafif farklılık göstermesine rağmen, toplam iterasyon sayısı, yiyecek miktarları, arı sayısı (işçi, gözcü ve kaşif arı sayılar teker teker belirlenebilir veya belli bir orana göre toplam arı sayısından pay alırlar) ve bir yiyecek kaynağının kayda değer olup olmadığını tespit etmek için eşik değeri gibi parametreler olabilir.

Adım 2 ve 13 arasında tekrarlamalı olarak toplam iterasyon sayısı veya belirli kriterler sağlanana kadar işlemler tekrarlanır. Bu tekraklama işlemi sırasında işçi, gözcü ve kaşif arıları öncelikle kovan dışında komşu sırasıyla işçi, gözcü ve kaşif arılar ile haberleşerek kısmı çözümleri yani en iyi olabilecek yiyecek kaynaklarını (nektar miktarı) belirlemeye çalışırlar. Daha sonra bu arılar kovana donerek bilgilerini kovandaki arılar ile (özellikle gözcü arılar ile) paylaşırlar ve tespit ettikleri nektar miktarına göre bir arı ekibi toplamaya gayret ederler. Bu bilgi paylaşım sırasında en iyi olan çözüm yada çözümler hafızada tutulur ve eskisinden iyi ise eskisi yerine yeni çözüm tercih edilir. Ayarlanan arı ekipleri tercih edilen yiyecek kaynaklarına giderler ve bu şekilde algoritmanın 2. ve 13. adımları tekrarlanır.

Bu proje kapsamında adım 3 ve 6 arası paralel işleminin uygulandığı kısımdır. Her bir algoritma iterasyonunda, adım 3 ve 6 arası işlemciler arasında paylaştırılarak paralel olarak hesaplanması sağlanır. Daha sonra her işlemci sonucu alınarak birleştirilir adım 7'ye geçilir.

Paralel hesaplama işleminde paylaşımlı hafıza yaklaşımı olan OpenMP yaklaşımı kullanılmıştır.

Adım 13'de toplam iterasyon sayısı veya kullanıcı tarafından istenen kriterler sağlanınca algoritma sonlandırılır. Bu kriterler çeşitli olabilir: yiyecek kaynaklarında yeterli yiyecek kalmaması, art arta üretilen en iyi sonuçların değişmemesi gibi.

3.3. Yapay Arı Koloni Algoritması Tabanlı (ABC-tabanlı) Sınıflandırma Algoritması

Bu bölümde proje kapsamında geliştirilen yapay arı koloni algoritması tabanlı veri madenciliği algoritması (ABC-tabanlı sınıflandırma algoritması) önerilmiştir. ABC-tabanlı sınıflandırma algoritması veri madenciliği problemlerinden birisi olan sınıflandırma için geliştirilmiştir (Algoritma 3). Algoritma verilen veri için eğer-öyleyse (if-then) sınıflandırma kurallarının çıkarılmasını sağlamaktadır. Çıkarılan bu kurallar daha sonra sınıfı bilinmeyen verilerin sınıflandırılması işleminde kullanılmaktadır. ABC-tabanlı sınıflandırma algoritması sonuçları derlenerek " International Symposium On Innovations in Intelligent Systems and Applications" konferansında bildiri olarak sunulmuştur [22].

ABC-tabanlı sınıflandırma algoritması öncelikle çok farklı sayıda kural üretimini sağlar daha sonra uygunluk kriterlerini sağlayan ve güçlü sınıflandırma yapan kuralları belirler. Daha sonra bulunan en iyi kural kümesi bir eleme işlemine daha tabi tutularak aynı işlemi yapan mükerrer kurallar elenir. Sonuç olarak algoritma tespit etmiş olduğu kurallar çıktı olarak verir.

Çıktı olarak verilen kurallar bir veri kümesinin sınıflandırılması için verilen sıra ile uygulanır. Sınıflandırm kuralları iki kısımdan oluşur. Kuralın ilk kısmında (eğer kısmı) şartlar listelenir, ikinci kısmında ise (öyleyse kısmı) ilk kısımda şartları sağlayan verilerin atanacağı sınıf bulunur. Bu nedenle herbir kuralın (veya sınıfın) sınıflandırma şartı farklıdır. Bu nedenle verilen veriyi sınıflandırmak için 1. kuraldan itibaren uygulama yapılır. Eğer 1. kural verilen veriyi sınıflandırmış ise diğer kuralların uygulanmasına gerek yoktur. Eğer sınıflandırmamış ise algoritmanın çıktı olarak verdiği 2. kural uygulanır ve sınıflandırma işlemi bu şekilde devam eder. Eğer verilen veri hiçbir kural tarafından sınıflandırılamamış ise verilen veri

varsayılan sınıfına dahil edilir. Bu dahil edilme işlemi algoritmanın son kuralı tarafından yapılır.

ABC-tabanlı sınıflandırma algoritması 4 ana adımdan oluşur (Algoritma 3). Bunlar ilklendirme (Adım 1), kural keşfi (Adım 2-8), varsayılan kuralların belirlemesi (Adım 9) ve gereksiz kuralların elenmesi (Adım 10) aşamalarıdır. Bu aşamaların açıklamaları aşağıda verilmiştir.

Algoritma 3: ABC-tabanlı Sınıflandırma Algoritması

Girişler

Veritabanı

Max_unc: Öğrenme işlemini durdurmak için kullanılan keşfedilmemiş kural sayısı

MDöngü: Toplanm iterasyon sayısı

CS: Toplam arı sayısı

Min_yiyecek: Bir yiyecek kaynağının terk edilemesi için gerekli alt yiyecek limiti

Çıktılar: Sınıflandırma Kuralları

Algoritma

1. İlklendir: Değişkenlerin ilk değerlerini ata ve veriyi hazırla
2. Bütün kurallar keşfedilene kadar (Adım 3 ve 8 arası tekrarlanır)
3. Baskın sınıfı bul
4. ABC_algoritmasını uygula
5. Kuralı oluştur
6. Kuraldan gereksiz şartları temizle
7. Sınıflandırılmış örnekleri veritabanından kaldır
8. Kural veritabanına yeni kuralı ekle
9. Varsayılan kuralları belirle
10. Gereksiz kuralları temizle

ABC-tabanlı sınıfladırma algoritmasında öncelikle veriler hazırlanır. Veri hazırlama adımında gürültü giderimi, normalizasyon gibi işlemler yapılır. Daha sonra kural keşfi aşamasına geçilir. Bu aşamada öncelikle en büyük sınıftan başlamak üzere kurallar birer birer belirlenir. Belirlenen kuralı karşılayan veriler veritabanından çıkarılarak sınıflandırılmayan veri için yeni kuralların keşfi işlemi yapılır. Keşfedilen kurallardan birden fazlasının aynı

sınıflandırmayı yapar şekilde olması muhtemeldir. Bu nedenle aynı sınıflandırmayı yapan kuralların bir kısmı elenerek optimum kural listesi bulunur. Bu kural listesi daha sonra algoritma tarafından çıktı olarak verilir (Algritma 3). Algoritmanın işlem adımları aşağıda açıklanmıştır.

İlklendirme: Bu aşamada değişkenlerin ilk değerleri atanır ve veritabanındaki veriler [0, MaksDeğer] aralığında normalize edilir. Buradaki MaksDeğer değişkeni kullanıcı tarafından algoritmaya giriş olarak verilir. Sürekli verilerin normalizasyonu için eşitlik 1'de verilen formül kullanılmıştır. Ayrık verilerin normalizasyonu için ise eşitlik 2'de verilen formül kullanılmıştır.

$$Değer_{normalize} = \frac{MaksDeğer(Değer - Değer_{\min})}{Değer_{maks} - Değer_{\min}} \quad (1)$$

$$Değer_{normalize} = \frac{MaksDeğer * Değer}{Toplam_Değer_Sayısı} \quad (2)$$

Kural Keşfi Aşaması: Yapay arı koloni algoritması tabanlı sınıflandırma algoritması öncelikle eğitim verilerini kullanarak kuralları belirler. Kurallar eğer-öyleyse yapısında olup hangi şartlar sağlandığında veriyi hangi sınıfa ait olduğunu belirlemek için kullanılır.

Bu aşamada (adım 2 ve 8 arası) ilk adım olarak en büyük sınıfın kuralı belirlenir. Kural belirlendikten sonra kuralın doğru olarak sınıflandırdığı verileri eğitim kümesinden çıkarılır. Algoritma eğitim kümesinde bulunan henüz sınıfı belirlenmemiş veriler için gerekli kuralları bulana kadar veya eğitim kümesindeki verilerin sayısı kullanıcı tarafından belirlenen bir eşik değerin altına düşene kadar devam eder.

Varsayılan Kural Aşaması: Bu aşamada (adım 9), algoritmada şimdiye kadar sınıflandırılmamış geri kalan veriler için varsayılan kural belirlenir.

Gereksiz Kuralları Temizeleme Aşaması: Bu aşamada kurallar birbirleri içerisinde kontrol edilerek tekrar eden kurallar kural kümesinden çıkarılır.

Sonuç olarak elde edilen kurallar algoritma tarafından çıktı olarak verilir.

3.4. İşbirlikçi Yapay Arı Koloni Algoritması Tabanlı (ABC-tabanlı)Sınıflandırma Algoritması

Sınıflandırma işleminde bütün kuralların aynı anda bulunması için İşbirlikçi ABC-tabanlı sınıflandırma algoritması önerilmiştir. Önceki bölümde sunulan ABC-tabanlı sınıflandırma algoritması kuralları teker teker bulmakta idi. İşbirlikçi yaklaşımda her iterasyonda veriyi sınıflandırılan bütün kurallar bulunmaktadır. Bu nedenle literatürdeki algoritmalara iteratif olarak kuralları bulurken İşbirlikçi ABC-tabanlı sınıflandırma algoritması bütün kuralları aynı anda bulur. Bu stratejiye çoklu-kural çıkarımı denir. Bu yaklaşımda herbir döngüde algoritma veri kümesinin sınıflandırması için gerekli bütün kuralları üretir. Temel olarak ABC algoritmasının herbir bireyi bir sınıflandırma kuralı üretir. Bu nedenle işbirlikçi ABC-tabanlı sınıflandırma algoritmasının performansı bütün bireylerin işbirliğine bağlıdır (Algoritma 4).

İşbirlikçi ABC-tabanlı sınıflandırma algoritmasnın 4 giriş parametresi vardır. Üç tanesi temel yapay arı koloni algoritması parametresidir. Yeni parametre olan ilklendirme oranı ilk yiyecek kaynağında bir özelliğin bulunma ihtimalidir. Bu parametre yüksek ise uzun kuralların üretilmesi mümkündür ve bu durum ise algoritmanın çalışma süresini arttırır.

İlklendirme Aşaması: Bu aşamada (adım 1) farklı parametreler için ilk değerler atanır, herbir birey için rastgele kurallar üretilir. Algoritmaya giriş olarak küme sayısı K verilir. Daha sonra bireyler K adet gruba ayrılır. Herbir grup kendi sınıfı için kurallar üretmeye çalışır. Bu aşamada ayrıca işçi, onlooker ve kaşif arıların sayıları belirlenir.

Rekabet Aşaması: Bu fonksiyonda (adım 2 ve 7) bir grup kural kümesi (yani yiyecek kaynağı) seçilerek çoklu-kural çıkarım işlemi yapılır. Öncelikle kural kümesi tarafından veri kümesinin sınıflandırılması sağlanır. Eğer sınıflandırma işlemi bir kural tarafından doğru bir şekilde yapıldıysa bu kurala bir jeton (token) verilir. Jeton verme işleminde kuralların uygunluk değerlendirmesi de yapılır. En uygun kurala ödül verilir. Son olarak bir sınıfa ait olan jetonlara sahip kurallar güçlü çözüm olarak kaydedilir.

Bütün Kuralların Üretilmesi Aşaması: Bir önceki aşamada belirlenen yeni kurallar bu aşamda (adım 3 ve 8) uygunluk değerlerine göre azalan olarak sıralanır. Sınıflandırma

işleminde öncelikle uygunluk değeri en yüksek olan kural uygulandığı için kuralların hangi sırada kullanıldığı önemlidir.

Algoritma 4: İşbirlikçi ABC-tabanlı Sınıflandırma Algoritması

Girişler:
Veritabanı
Max_unc: Öğrenme işlemini durdurmak için kullanılan keşfedilmemiş kural sayısı
MDöngü: Toplanm iterasyon sayısı
CS: Toplam arı sayısı
Min_yiyecek: Bir yiyecek kaynağının terk edilemesi için gerekli alt yiyecek limiti
initratio: ilk yiyecek kaynağında bir özelliğin bulunma ihtimali
K: Küme sayısı
Çıktılar: Sınıflandırma Kuralları
Algoritma

1. İlklendir, Değikenleri ata ve veriyi hazırla, sınıf bireylerini belirle.
2. Rekabet
3. Bütün kuralları üret
4. Kural doğruluklarını hesapla
5. Verilen döngü sayısı kadar
6. işci, onlooker ve kaşif arıları belirle
7. Rekabet
8. Bütün kuralları üret
9. Kural doğruluklarını hesapla
10. En iyi kuralları kaydet
11. Varsayılan kuralları belirle
12. Gereksiz kuralları temizle

Kural Doğruluklarının Hesaplanması Aşaması: Bu aşamada (adım 4 ve 9) her bir kuralın sınıflandırma doğruluğu hesaplarnır. Sınıflandırma doğruluğu veri kümesinde doğru olarak sınıflandırılan kayıtların toplam kayıt sayısına oranı olarak ifade edilmiştir.

En İyi Kuralların Kaydedilmesi Aşaması: Bu aşamada (adım 10) daha önce kaydedilen kurallar ile yeni üretilen kuralların performansları karşılaştırılır. Eğer yeni üretilen kurallar

önceki üretilen kurallardan daha iyi performans gösterdilerse, yeni üretilen kurallar hafızaya alınarak kaydedilir, aksi durumda en iyi kural listesinde değişiklik yapılmaz.

Varsayılan Kural Aşaması: Bu aşamada (adım 11), algoritmada şimdiye kadar sınıflandırılmamış geri kalan veriler için varsayılan kural belirlenir.

Gereksiz Kuralları Temizeleme Aşaması: Bu aşamada (adım 12) kurallar birbirleri içerisinde kontrol edilerek tekrar eden kurallar kural kümesinden çıkarılır.

Sonuç olarak algoritma belirlenen şarları sağlayan ve en iyi sınıflandırmayı yapan kural listesini çıktı olarak verir.

4. BULGULAR

Bu bölümde paralel ABC algoritmasının, ABC-tabanlı sınıflandırma algoritmasının ve işbirlikçi ABC-tabanlı sınıflandırma algoritmasının performans deneyleri verilmiştir.

4.1. Paralel ABC Algoritması Performans Deneyleri

Paralel ABC algoritmasının değerlendirilmesi amacıyla yapılan performans deneyleri, bir ve birden fazla işlemci üzerinde çalıştırılan yapay arı koloni algoritmasının hız ölçümleri ve bulunan küresel minimum noktaları şeklindedir. Test sonuçlarının yorumlanabilmesi için aşagıdaki tanımlamalar yapılmıştır. Deneylerde minimum noktası aranan fonksiyon olarak rastrigin fonksiyonu kullanılmıştır.

- CS - Koloni büyüklüğü (Employed Bees + Onlooker Bees)
- Yiyecek_miktarı – CS / 2
- Limit –İyileştirilemeyen yiyecek kaynaklarının baştan oluşturulması için deneme eşik değeri
- MDöngü – Toplam iterasyon sayısı
- D – Fonksiyona ait optimize edilecek parametre sayısı

Paralel ABC algoritması C/C++ ortamında geliştirilmiştir ve OpenMP paralel yaklaşımı kullanılmıştır. Yapılan ölçümlerde CS ve Yiyecek_miktarı dışındaki bütün parametreler sabit tutulmuş, CS'nin 500, 250, 200 ve 126 değerleri için ölçümler yapılmıştır. Elde edilen sonuçlar aşağıda listelenmiştir.

Parametre Ayarları 1: Aşağıdaki parameter değerleri için alınan sonuçlar tablo 1'de verilmiştir.

CS : 500

Yiyecek_Miktarı : 250

limit : 100

MDöngü : 25000

Tablo 1: Parametre ayarları 1 için paralel ABC algoritmasının performansı

Çekirdek Sayısı(N)	Bağıl Hızlanma Oranı (Tek İşlemciye Göre)	Bağıl Hızlanma Oranı (N-1'e Göre)	Süre (Saniye)	Global Minimum Değeri
1	1.000	1.000	189.54	1.421085e-014
2	1.325	0.325	142.99	3.552714e-015
3	1.489	0.164	127.23	1.598721e-014
4	1.580	0.091	119.94	1.065814e-014

Parametre Ayarları 2: Aşağıdaki parameter değerleri için alınan sonuçlar tablo 2'de verilmiştir.

CS : 250

Yiyecek_Miktarı : 125

limit : 100

maxCycle : 25000

Tablo 2: Parametre ayarları 2 için paralel ABC algoritmasının performansı

İşlemci Sayısı(N)	Bağıl Hızlanma Oranı (Tek İşlemciye Göre)	Bağıl Hızlanma Oranı(N-1'e Göre)	Süre (Saniye)	Global Minimum Değeri
1	1.000	1.000	94.57	1.598721e-014
2	1.326	0.326	71.30	1.198541e-014
3	1.485	0.159	63.66	1.598721e-014
4	1.570	0.085	60.20	2.486900e-014

Parametre Ayarları 3: Aşağıdaki parameter değerleri için alınan sonuçlar tablo 3'de verilmiştir.

CS : 200

Yiyecek_Miktarı : 100

limit : 100

maxCycle : 25000

Tablo 3: Parametre ayarları 3 için paralel ABC algoritmasının performansı

İşlemci Sayısı(N)	Bağıl Hızlanma Oranı (Tek İşlemciye Göre)	Bağıl Hızlanma Oranı(N-1'e Göre)	Süre (Saniye)	Global Minimum Değeri
1	1.000	1.000	75.76	1.953993e-014
2	1.327	0.327	57.09	3.197442e-014
3	1.485	0.158	50.99	3.019807e-014
4	1.561	0.076	48.51	: 2.309264e-014

Parametre Ayarları 4: Aşağıdaki parameter değerleri için alınan sonuçlar tablo 4'de verilmiştir.

CS : 126

Yiyecek_Miktarı : 63

limit : 100

maxCycle : 25000

Tablo 4: Parametre ayarları 4 için paralel ABC algoritmasının performansı

İşlemci sayısı(N)	Bağıl Hızlanma Oranı (Tek İşlemciye Göre)	Bağıl Hızlanma Oranı(N-1'e Göre)	Süre (Saniye)	Global Minimum Değeri
1	1.000	1.000	46.25	: 5.506706e-014
2	1.319	0.319	35.05	1.243450e-014
3	1.479	0.160	31.26	2.842171e-014
4	1.510	0.031	30.62	1.243450e-014

Farklı parametre ayarları için algoritmalar için elde edilen sonuçlar incelendiğinde arı ve yiyecek miktarı değişse bile aynı işlemci sayısı için algoritmanın aynı bağıl hızlanma oranını verdiği görülmektedir. Bu durumda geliştirilen algoritmanın ölçeklenebilir olabileceğini işareti olarak kabul edilebilir.

4.2. ABC-tabanlı Sınıflandırma Algoritmasının Performans Deneyleri

ABC-tabanlı sınıflandırma algoritmasının performansı sezgisel teknik olan PSO (Parçacık Sürü Optimizasyonu) algoritması [25] ile klasik teknik olan C4.5 algoritması [21] performansları ile karşılaştırılmıştır. Deneylerde 2 sınıfı bulunan Breast verikümesi ile 7 sınıfı bulunan Zoo veri kümesi kullanılmıştır [26].

Breast veri kümesi bir medikal merkez'deki gögüs kanseri hastalarının verilerini içermektedir. Verikümesinde 282 kayıt vardır, gögüs kanserinin tekrarlanıp tekrarlanmadığını ifade eden iki sınıf vardır ve herbir hastanın 9 parametresi (yaş, menapoz durumu, tümör boyutu gibi) kaydedilmiştir. Parametre değerleri veritabanında kategorik olarak kaydedilmiştir [26].

Zoo veritabanıda ise hayvanlar hakkında çeşitli bilgiler bulunmaktadır. Veritabanında 101 hayvanın kaydı vardır ve hayvanlar hakkında 16 parametre (saç, tüy, yumurta, süt, zehirli, gibi) boolean değer olarak tutulmuştur [26].

ABC-tabanlı sınıflandırma algoritmasının arı sayısı, yiyecek limiti, maksimum keşfedilmemiş veri sayısı, maksimum değer ve maksimum döngü giriş parametrelerine sırasıyla , 20, 100, 0.1, 0.1 ve 500 değerleri verilmiştir.

PSO algoritmasının parçacık sayısı, üst ağırlık limiti, kurulum katsayısı, maksimum keşfedilmemiş veri sayısı, farklılık eşik değeri, yakınsama yarıçağı ve yakınsama platform genişliği giriş parametrelerine sırasıyla 25, 2.05,0.73, 0.1, 0.1, 0.1 ve 30 değerleri verilmiştir

Deneylerde, sezgisel algoritmalar olan ABC-tabanlı sınıflandırma ve PSO algoritmaları 50 kez çalıştırılmıştır. Klasik teknik olan C4.5 her zaman aynı sonucu verdiği için bir kez çalışıtırılmıştır. Tablo ccc 'da sınıflandırma doğrulukları ve ürettikleri kural sayıları verilmiştir.

Breast verisi kullanılarak elde edilen sonuçlar incelendiğinde en iyi ortalama doğruluğu C4.5 algoritmasını verdiği görülmektedir (Tablo 5). ABC-tabanlı sınıflandırma algoritması ise PSO algoritmasından daha iyi sonuç vermiştir. Her üç algoritmanın kural sayısı 4 civarındadır ve

herbir kuralda bulunan ortalama şart sayısı PSO algoritmasında 4 civarında iken ABC-tabanlı sınıflandırma ve C4.5 algoritmalarında 7 civarındadır (Tablo 5).

Tablo 5: Breast veri kümesi için elde edilen algoritma sonuçları

Algoritma	Ortalama Doğruluk	Maksimum Doğruluk	Minimum Doğruluk	Standart Sapma	Kural Sayısı	Ortalama Şart Sayısı
ABC sınıflandırma	%74.92	%75.79	%73.66	0.0082	4.2	6.2
PSO	%73.6	%75.44	%72.14	0.0167	3.48	4.04
C4.5*	%75.09	-	-	-	4	7

*Bu algoritma kesin doğruluk sonucunu verir ve dolayısı ile maksimum ve minimum doğrulukları ve standart sapma değeri yoktur.

Zoo verisi kullanılarak elde edilen sonuçlar incelendiğinde ise en iyi ortalama doğruluğu C4.5 algoritmasını verdiği görülmektedir (Tablo 6). ABC-Madencisi ise PSO algoritmasından daha iyi sonuç vermiştir. ABC-Madencisi ve PSO algoritması kural sayıları 5 civarındayken C4.5'in ürettiği kural sayısı 13 olmuştur. Algoritmaların ürettikleri kuralların şart sayılarına bakıldıklarında ise ABC-Madencisi ve PSO algoritmalarının daha az şart ile sınıflandırmayı yaptıkları görülmektedir. C4.5 algoritması ise 62 şarta ihtiyaç duymaktadır.

Tablo 6: Zoo veri kümesi için elde edilen algoritma sonuçları

Algoritma	Ortalama Doğruluk	Maksimum Doğruluk	Minimum Doğruluk	Standart Sapma	Kural Sayısı	Ortalama Şart Sayısı
ABC Sınıflandırma	%90,49	%92,09	%88,09	0,0182	5,04	6,64
PSO	%87,85	%95	%89,01	0,0658	5,92	4,86
C4.5*	%93,07	-	-	-	13	62

*Bu algoritma kesin doğruluk sonucunu verir ve dolayısı ile maksimum ve minimum doğrulukları ve standart sapma değeri yoktur.

Sonuç olarak ABC-tabanlı sınıflandırma algoritması diğer segizel teknik olan PSO tabanlı algoritmadan daha iyi sonuç vermiştir.

4.3. İşbirlikçi ABC-tabanlı Sınıflandırma Algoritmasının Performans Deneyleri

İşbirlikçi ABC-tabanlı sınıflandırma algoritmasının değerlendirilmesi için UCI veritabanından alınan veriler kullanılmıştır [2]. Algoritmanın performansı daha önce geliştirilen ABC-Madencisi, Hider [27] ve CORE [28] algoritmaları ile karşılaştırılmıştır. Çalışmada Breast ve Zoo verileri kullanılmıştır.

ABC-tabanlı sınıflandırma algoritmasının arı sayısı, yiyecek limiti, maksimum keşfedilmemiş veri sayısı, maksimum değer ve maksimum döngü giriş parametrelerine sırasıyla , 20, 100, 0.1, 0.1 ve 1000 değerleri verilmiştir. HIDER algoritmasının populasyon büyüklüğü, çaprazlama katsayısı, mutasyon ihtimali, eleme faktörü ve hata katsayısı giriş paramatrelerine sırası ile 100, 0.8, 0.5, 0.05, ve 0 değerleri verilmiştir. CORE algoritmasının populasyon büyüklüğü, çaprazlama katsayısı, mutasyon ihtimali ve işbirlikçi populasyon büyüklüğü giriş paramatrelerine sırası ile 100, 0.9, 0.3 ve 50 değerleri verilmiştir.

Breast verisi kullanılarak elde edilen sonuçlar incelendiğinde en iyi ortalama eğitim ve test doğruluğu işbirlikçi ABC sınıflandırma algoritmasının veridiği görülmektedir (Tablo 7). Algoritmanın standart sapması CORE'den sonra gelmektedir. Diğerlerinden farklı olarak işbirlikçi ABC sınıflandırma algoritmasının ortalama şart sayısı 64 civarındadır.

Tablo 7 : Breast veri kümesi için elde edilen algoritma sonuçları

Algoritma	Ortalama Eğitim Doğruluk	Eğitim Standart Sapma	Kural Sayısı	Ortalama Şart Sayısı	Test Doğruluk	Test Standart Sapma
İşbirlikçi ABC Sınıflandırma	%81.31	0.0080	18.3	63.68	%75.86	0.0669
ABC Sınıflandırma	%77.79	0.0155	5.5	7.08	%75.11	0.0496
Hider	%75.93	0.0188	2.7	9.5	%74.69	0.08
CORE	%80.53	0.0062	8.2	19.32	%73.60	0.0590
C4.5	%74.57	0.0123	12.5	40.8	%71.42	0.0736

Zoo verisi kullanılarak elde edilen sonuçlar incelendiğinde ise en iyi ortalama eğitim ve test doğruluğu işbirlikçi ABC sınıflandırma algoritmasının veridiği söylenebilir (Tablo 8). Algoritmanın standart sapması sıfırdır Diğerlerinden farklı olarak işbirlikçi ABC sınıflandırma algoritmasının ortalama şart sayısı 29 civarındadır.

Tablo 8 : Zoo verikümesi için elde edilen algoritma sonuçları

Algoritma	Ortalama Eğitim Doğruluk	Eğitim Standart Sapma	Kural Sayısı	Ortalama Şart Sayısı	Test Doğruluk	Test Standart Sapma
İşbirlikçi ABC Sınıflandırma	%100	0.00	7.1	28.9	%93.82	0.083
ABC Sınıflandırma	%94.19	0.0083	6.0	6.72	%90.89	0.0804
Hider	%96.04	0.0178	6.2	9.46	%90.27	0.0655
CORE	%99.78	0.0054	8.1	43.1	%93.44	0.0716
C4.5	%98.57	0.0053	7.7	18.2	%94.00	0.0699

Sonuç olarak işbirlikçi ABC sınıflandırma algoritması literatürdeki sezgisel algoritmalardan saha iyi sınıflandırma performansı sağlamıştır.

5. TARTIŞMA VE SONUÇ

Proje kapsamında yapay arı koloni algoritmasının (ABC) analizi yapılmış, paralel ABC algoritması, ABC-tabanlı sınıflandırma algoritması ve işbirlikçi ABC-tabanlı sınıflandırma algoritması çalışılmıştır.

Paralel ABC algoritmasının seri ABC algoritmasına göre ölçeklenebilir bir performans sergilemiştir. Farklı parametre ayarları için algoritmalar için elde edilen sonuçlar incelendiğinde arı ve yiyecek miktarı değişse bile aynı işlemci sayısı için algoritmanın aynı bağıl hızlanma oranını verdiği görülmektedir. Bu durumda geliştirilen algoritmanın ölçeklenebilir olabileceğini işareti olarak kabul edilebilir.

ABC-tabanlı sınıflandırma algoritması literatürde var olan diğer sezgisel teknikler ile karşılaştırılmıştır. Elde edilen sonuçların ABC-tabanlı sınıflandırma algoritmasının sınıflandırma kurallarını bulmada başarılı olduğunu ortaya koymuştur.

İşbirlikçi ABC-tabanlı sınıflandırma algoritması literatürde var olan diğer algoritmalar ile karşılaştırılmıştır. İşbirlikçi ABC-tabanlı sınıflandırma algoritması diğer algoritmalara göre başarılı sonuçlar vermiştir.

KAYNAKLAR

[1] Bahamish, H.A.A., Protein Conformational Search Using Honey Bee Colony Optimization, In Proc. The 2nd IMT-GT Regional Conf. On Mathematics, Statistics and Applications, Malasia June 13-15, 2006.

[2] Tereshko, V., Reaction-Diffusion Model Of A Honeybee Colony's Foraging Behaviour, M. Schoenauer, et al, Eds., Parallel Problem Solving from Nature VI (Lecture Notes in Computer Science, Vol. 1917) Springer-Verlag: Berlin, p. 807-816, 2000.

[3] Tereshko, V. and Lee, T., How Information Mapping Patterns Determine Foraging Behaviour Of A Honey Bee Colony, Open Systems and Information Dynamics, 9, 181-193, 2002.

[4] Tereshko, V. and Loengarov, A., Collective Decision-Making in Honey Bee Foraging Dynamics, Computing and Information Systems Journal, ISSN 1352-9404, 9, No 3, October 2005.

[5] Abbass, H. A., MBO: Marriage in Honey Bees Optimization-a Haplometrosis Polygynous Swarming Approach, Proceedings of the 2001 Congress on Evolutionary Computation, 1, Page(s):207 – 214, 27-30 May 2001.

[6] Yang, X., Engineering Optimizations via Nature-Inspired Virtual Bee Algorithms, Lecture Notes in Computer Science, Springer-Verlag GmbH, 3562, pp. 317, 2005.

[7] Karaboga, D., An Idea Based On Honey Bee Swarm For Numerical Optimization, Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.

[8] Basturk, B., Karaboga, D., An Artificial Bee Colony (ABC) Algorithm for Numeric function Optimization, IEEE Swarm Intelligence Symposium 2006, May 12-14, 2006, Indianapolis, Indiana, USA.

[9] Karaboga D, Basturk B, Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems, IFSA 2007, Cancun, Mexico, June, 18-21, 2007.

[10] Karaboga D, Basturk B., A Powerful And Efficient Algorithm For Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm, Journal of Global Optimization, In Press.

[11] Storn, R. and Price, K., Differential Evolution -- A Simple And Efficient Heuristic For Global Optimization Over Continuous Spaces, Journal of Global Optimization, 11, 341-359, 1997.

[12] Kennedy, J. and Eberhart, R. C., Particle Swarm Optimization, 1995 IEEE International Conference on Neural Networks, 4, 1942-1948, 1995.

[13] Wong, L. P., Low, M. Y. H., Chong, C. S., Bee Colony Optimzation with Local Search for Traveling Salesman Problem, in Proc. IEEE Int'l. Conf. On Industrial Informatichs, Korea, July 13-16, 2008.

[14] Pham, D T, Eldukhr, E E i and Soroka, A J, Using The Bees Algorithm To Tune A Fuzzy Logic Controller For A Robot Gymnast, Proceedings of the 3rd Virtual International Conference on Intelligent Production Machines and Systems, ISBN: 978-1904445-52-4, 2007.

[15] Karaboga, D., Akay, B., and Ozturk, C., Artificial Bee Colony (ABC) Optimization Algorithm for Training Feed-Forward Neural Networks, LNCS: Modeling Decisions for Artificial Intelligence, Vol: 4617/2007, pp:318-319, Springer-Verlag, 2007, MDAI 2007.

[16] Karaboga, D., Akay, B., Artificial Bee Colony Algorithm on Training Artificial Neural Networks, Signal Processing and Communications Applications, 2007. SIU 2007, IEEE 15th. 11-13 June 2007, Page(s):1 - 4, doi: 10.1109/SIU.2007.4298679.

[17] Holmes, J.H., Lanzi, P.L., Stolzmann, W. and Wilson, S.W., Learning Classifier Systems: New Models, Successfull Applications, Information Processing Letters, 82(1), Pages:23-30, 2002.

[18] Giordana, A. and Neri, F., Search-Intensive Concept Induction, Evolutionary Computation, 3(4), Pages: 375-416, 1995.

[19] Corcoran, A.L. and Sen, S., Using Real-valued Genetic Algorithms to Evolve Rule Sets for Classification, Proceedings of the First IEEE Conference on Evolutionary Computation, 120-124, Orlando, 1994.

[20] Sousa, T., Silva A., and Neves, A., Particle Swarm Based Data Mining Algorithms for Classification Tasks, Parallel Computing, 30(5-6), Pages 767-783, 2004.

[21] Quinlan, J.R., C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers, 1993.

[22] Celik, M., Karaboga, D., and Koylu, F., Artificial Bee Colony Data Miner (ABC-Miner), International Symposium On Innovations in Intelligent Systems and Applications, INISTA'11, Pages: 96-100, İstanbul, Türkiye.

[23] Parpinelli, R., Lopes, H., and Freitas, A., Data Mining with and Ant Colony Optimization Algorithms, IEEE Transactions on Evolutionary Computation, 6(4), Pages 321-332, 2002.

[24] Celik, M., Koylu, F., and Karaboga, D., CoABCMiner: An Algorithm for Cooperative Rule Classification System Based on Artificial Bee Colony, International Journal On Artificial Intelligence Tools, 24, Pages:1-40, 2015.

[25] Alcala-Fdez, J., Sanchez, L., Garcia, S., Del Jesus, M. J., Ventura, S., Garrell, J. M., Otera, J., Romero, C., Bacardit, J., Rivas, V.M., Fernandez, J.C., and Herrera, F., KEEL: A Software Tool to Assess Evolutionary Algorithms to Data Mining Problems, Soft Computing, 13, 307-318, Springerling, 2009.

[26] UCLA ML Veri Deposu, https://archive.ics.uci.edu/ml/datasets.html

[27] Aguilar-Ruiz, J. S., Riquelme, J. C., and Toro, M., Evolutionary Learning of Hierarchical Decision Rules, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 33(2), 324–331, 2003.

[28] Tan, K. C. , Yu, Q. and Ang, J. H. , A coevolutionary Algorithm for Rules Discovery in Data Mining, International Journal of Systems Science 37(12), 835–864, 2006.