Journal of the Faculty of Engineering and Architecture of Gazi University 32:1 (2017) 101-118

Mühendislik Mimarlık Fakültesi Dergisi
Journal of The Faculty of Engineering and Architecture of Gazi University

Elektronik / Online ISSN : 1304 - 4915
Basılı / Printed ISSN : 1300 - 1884

Genelleştirilmiş ilginç sınıflandırma kurallarının yapay kimyasal reaksiyon optimizasyon algoritması ile keşfi

Bilal Alataş[1*], A. Bedri Özer[2]

[1]Fırat Üniversitesi, Mühendislik Fakültesi, Yazılım Mühendisliği Bölümü, 23100, Elazığ, Türkiye
[2]Fırat Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, 23100, Elazığ, Türkiye

Ö N E Ç I K A N L A R

- Genelleştirilmiş kural keşfi problemi için yeni bir yaklaşım
- Yapay kimyasal reaksiyon optimizasyon algoritması'nın veri madenciliği alanındaki performansı
- İlginç kuralların keşfi için esnek amaç fonksiyonu

Makale Bilgileri
Geliş: 09.01.2015
Kabul: 22.01.2017

DOI:
10.17341/gazimmfd.300600

Anahtar Kelimeler:
Veri madenciliği, sınıflandırma, metasezgisel optimizasyon, yapay kimyasal reaksiyon optimizasyon algoritması, performans

ÖZET

Sınıflandırma kural madenciliği en çok çalışılan veri madenciliği problemlerindendir. Bu makalede birden fazla sınıflama etiketi bulunan ve tahmin edilecek birden fazla hedef niteliğin olduğu veritabanlarında, sınıflandırma kural madenciliğinin daha genel bir hali olan kompleks ve fazla çalışılmamış genelleştirilmiş kural keşfi problemi için ilk kez kimya tabanlı Yapay Kimyasal Reaksiyon Optimizasyon Algoritması (YAKROA) kullanılmıştır. İlginçlik kriteri de eklenerek; algoritmanın keşfedeceği kuralların sadece doğru ve anlaşılabilir değil aynı zamanda ilginç, beklenmedik ve sürpriz olması için de gerekli düzenlemeler yapılmıştır. Farklı amaçlar doğrultusunda değişik veritabanlarında farklı kurallar kümesi, esnek bir şekilde algoritmadaki temsil biçimi ve amaç fonksiyonunun düzenlenmesiyle bulunmuştur. Farklı özellikte halka açık gerçek veritabanlarında YAKROA'nın sınıflandırma kural keşfi problemindeki performansı genetik algoritma, parçacık sürü optimizasyon algoritması ve karınca koloni optimizasyon algoritması ile karşılaştırılmıştır. Deneysel sonuçlardan, YAKROA'nın veri madenciliğinin bu özel alanındaki performansının umut verici olduğu görülmüştür. YAKROA'nın farklı veri madenciliği problemleri; özellikle birliktelik kurallarının keşfi, kümeleme kurallarının keşfi, ardışık örüntü keşfi vb. için etkili bir çözüm yöntemi olarak kullanılabileceği öngörülmektedir.

Mining of generalized interesting classification rules with artificial chemical reaction optimization algorithm

H I G H L I G H T S

- A novel approach for generalized rules mining
- Performance of artificial chemical reaction optimization algorithm within data mining field
- Flexible objective function for interesting rule mining

Article Info
Received: 09.01.2015
Accepted: 22.01.2017

DOI:
10.17341/gazimmfd.300600

Keywords:
Data mining, classification, metaheuristic optimization, artificial chemical reaction optimization algorithm, performance

ABSTRACT

Classification rules mining is one of the most studied data mining problems. In this article, chemistry based Artificial Chemical Reaction Optimization Algorithm (ACROA) has been for the first time used for mining of generalized classification rules, which is a complex and no well researched generalized variant of classification rules mining where there is more than one goal attribute to be predicted. Furthermore, interestingness measure has been added by performing the adaptations to the algorithm in order to make the rules mined by the algorithm not only accurate and comprehensible but also interesting, surprising, and unexpected. Different rule sets within different databases satisfying different objectives have been flexibly mined by adapting the representation scheme and objective function. Performance of ACROA in classification rules mining within different real public databases have been compared that of genetic algorithm, particle swarm optimization algorithm, and ant colony optimization algorithm. It has shown that performance of ACROA within this special task of data mining is promising. ACROA can be an efficient solution method for different data mining tasks such as association rules mining, clustering rules mining, sequential pattern mining, and etc.

[*] Sorumlu Yazar/Corresponding author: balatas@firat.edu.tr/ Tel: +90 0 424 237 0000-5599

1. GİRİŞ (INTRODUCTION)

Sınıflandırma kurallarının madenciliği, en çok kullanılan ve insan düşünce yapısına en yatkın veri madenciliği tekniklerinden biridir. Bu teknik ile bir veri tabanından kullanıcıların çok kolay anlayacağı kurallar bulunur [1]. Sınıflandırmada yeni gelen her bir örnek, önceden sınıflandırılmış bir takım veriler üzerinde yapılan bir eğitim neticesinde ortaya çıkan bir modele göre, daha önce belirlenmiş olan bir sınıfa atanmaktadır. Bu bağlamda kullanılan belki de en önemli değerlendirme kriterleri, tahmini doğruluk ve anlaşılabilirliktir. Tahmini doğruluk, genelleme olarak ta bilinir ve oluşturulan modelin daha önce görülmemiş örnekleri sınıflandırmadaki performansını gösterir. Anlaşılabilirlik ise, oluşturulan modelin kullanıcılar tarafından anlaşılabilirliğini ölçer [2]. Sınıflandırma için çeşitli yöntemler ve algoritmalar bulunmaktadır. Karar ağaçları, sınıflandırma için güçlü bir modeldir. Bunlar, C4.5 [3] ve CART [4] gibi tekniklerle oluşturulur ve 'böl-ve-yönet' stratejisini uygular. Veri, ayrı alt kümelere ayrılır ve algoritma her kümeye tekrarlı olarak uygulanır [5]. Karar ağaçlarının en önemli avantajı, türetilen bir model olarak karar verme işlemine açık bir şekilde hâkim olmasıdır. Çok sayıda işlem yapmaya gerek duymadan sınıflandırma işlemini gerçekleştirebilir. Ancak bunlar, oluşturulmaları sırasında eğitim verisinde örneklerin saf alt kümelerini belirleme eğilimindedir. Bu da yanlış ya da tutarsız olan örneklere aşırı uymaya neden olabilir ve böylece son modelin genelleme gücünü azaltır. Bu problemin üstesinden gelmek için kural budama ve benzeri yardımcı yordamlar kullanılmaktadır [6].

Ayrıca karar ağaçları, tahmin için kullanıldığında, tahmin edilecek değişkenin sürekli değerler alması durumunda uygun sonuçlar üretememektedir [7]. Karar listeleri de eğitim verisinden çıkarılan bilginin açık bir temsilini belirli şekilde göstermesiyle karar ağaçlarına benzer. Ancak bunlar 'ayır-ve-yönet' yaklaşımını kullanır ve bir kural, eğitim verisinin bir alt kümesini kapsamak için oluşturulur ve sonra daha fazla kural, kalan örnekleri tekrarlı olarak kapsamak için üretilir. Bu strateji, ilk olarak AQ [8] ailesinde uygulanmıştır ve daha sonra CN2 [8] gibi algoritmalara temel teşkil etmiştir. Algoritmanın sonunda sıralı EĞER-O ZAMAN (IF-THEN) kural listesi elde edilir ve yeni bir örneğin sınıflandırılmasında sırasıyla uygulanır. Eğer listedeki ilk kural örneği kapsamıyorsa, o zaman bir sonraki denenir. İkincisi de çalışmazsa, listedeki üçüncü kural denenir ve işlem böylece devam eder. Bir örnek bir kural tarafından sınıflandırılırsa, daha fazla kural denenmez. Eğer kuralların hiçbiri örneği kapsamıyorsa, o zaman kural listesinin en altındaki varsayılan bir kural işletilir. Yani varsayılan kurala ulaşan tüm sınıflandırılmamış örnekler bu kuralın sınıf etiketiyle işaretlenir [6]. Sıralı listelerin bir dezavantajı, bireysel kuralların kendilerinin anlaşılabilirlik bakımından zor olabilmesidir. Bir listedeki bir kural, önceki tüm kurallar bağlamında ele alınmalıdır. Karar ağaçları gibi, karar listeleri de gürültülü eğitim verisine aşırı uyma problemiyle karşı karşıyadır ve bu yüzden genellikle kural budama işlemi uygulanır. Evrimsel hesaplama, özellikle genetik algoritma ve genetik programlama da etkili şekilde sınıflandırma kural madenciliğinde kullanılmıştır. Bu amaçla, dağıtık genetik algoritma [9], melez evrimsel algoritma [10] ve performansı arttırılmış genetik algoritma [11] önerilmiştir. Bu yaklaşımlarda arama uzayı üzerinde global bir arama yapılır ve kaba seçim algoritmalarına göre nitelik etkileşimiyle daha iyi baş edilebilir. Ayrıca açıklanabilir sonuçlar üretirler ve çok değişik tiplerdeki verileri işleme özelliğine sahiptirler. Ancak, optimal sonucun üretildiğine dair bir garanti bulunmamaktadır ve bazen ağır işlem yükü gerektirebilir. Sürü zekâsı tekniklerinden karınca koloni optimizasyonu (KKO) ile sınıflandırma kurallarının keşfi yapılmıştır [6].

Yang ve arkadaşları da KKO ile sınıflandırma kurallarının bulunması için, KKO'ya erken yakınsama ve lokal çözüme takılıp kalmanın engellenmesi amacıyla feromen çekim ve dışlama stratejisi eklemiş, ayrıca kural etkileşim problemini gidermiştir [12]. Başarımın arttırılması amacıyla Liang ve arkadaşları [13] ile Asadi ve Shahrabi [14] KKO'yu uyarlamıştır. Parçacık sürüsü optimizasyonu [15], arı algoritması [16] ve yapay arı koloni algoritmaları [17] ile de sınıflandırma kural madenciliği çalışmaları yapılmıştır. Yapay bağışıklık sistemlerinden klonal seçim algoritması da sınıflandırma kurallarının keşfi için kullanılmıştır [18, 19]. Yakın zamanda güncel sürü zekâsı yöntemlerinden kedi sürüsü optimizasyonu da anlaşılabilir nümerik sınıflandırma kurallarının keşfi için kullanılmış ve etkili sonuçlar elde edilmiştir [20]. Sınıflandırmaya ayrıca örnek-tabanlı öğrenme, yapay sinir ağları, lojistik gerileme ve Bayesian ağları yaklaşımları da vardır. Bu metotların çoğunun temel dezavantajı, tahmini doğrulukları bazı durumlarda iyi olmasına rağmen, açıklayıcı güçlerinin eksikliğidir [6]. Literatürdeki bu yöntemlere bazen bulanık mantık ta eklenerek bulanık kurallar üretilmiştir. Bulanık mantık; yapay bağışıklık sistemi algoritmalarından klonal seçim [21], karınca koloni algoritması [22], genetik algoritma [23] ile birleştirilip bulanık sınıflandırma kuralları elde edilmiştir. Kural budama gürültülü eğitim verisine aşırı uymadan kaçınmak için gerekli bir işlemdir. Karar listelerinde kural budama için iki temel strateji vardır. Birincisi komple bir kural kümesi oluşturulur ve sonra nitelikler kurallardan elimine edilerek ya da bireysel kurallar silinerek kural kümesi basitleştirilir. Bu, kural kümesinin önceden tanımlı bazı budama kriterlerine bağlı olarak global şekilde optimize edilmesiyle yapılır. İkinci strateji ise artımsal budama olarak adlandırılır. Çünkü her kural, algoritmayla oluşturulduktan hemen sonra basitleştirilir [6].

Kısaca sınıflandırma için karar ağaçları, sinirsel ağ, Bayes sınıflandırma, karar destek makineleri vb. birçok model önerilmiştir. Ancak bu modellerin çoğu kara-kutu modellerdir; açıklayıcı ve anlaşılabilir özellikleri bulunmamaktadır. Ayrıca esnek şekilde ilginç, sürpriz vb. farklı amaçlar doğrultusunda farklı kural kümesi sunmazlar.

Metasezgisel algoritmalar basit bir yaklaşımı, arama ya da optimizasyon problemlerinin çözümünde kullanırlar. Bu algoritmalar son yıllarda gittikçe güçlenmekte ve her geçen gün biraz daha popüler olmaktadır [24]. Bunların nedeni şu şekilde özetlenebilir:

a. Eş zamanlı olarak, farklı tipte karar değişkenleri, amaç fonksiyonları ve sınırlayıcıların olması durumunda probleme uygulanabilecek genel çözüm stratejileri sunmaktadırlar. Çözüm stratejileri, amaç fonksiyonu ve sınırlayıcıların tipine ve problemi modellemede kullanılan değişkenlerin tipine bağlı değildir.

b. Çözüm uzayı tipine, karar değişken sayısına ve sınırlayıcı sayısına bağlı değildir.

c. Sistemin modeli ve amaç fonksiyonu için kurulması zor olan ve bazen de kurulup da çözüm zamanı maliyeti çok yüksek olduğundan kullanılamayan çok iyi tanımlanmış matematiksel modellere ihtiyaç duymamaktadır.

d. Hesaplama güçleri iyidir, yani aşırı derecede hesaplama zamanına ihtiyaç duymazlar.

e. Dönüşümleri ve uyarlanmaları kolaydır.

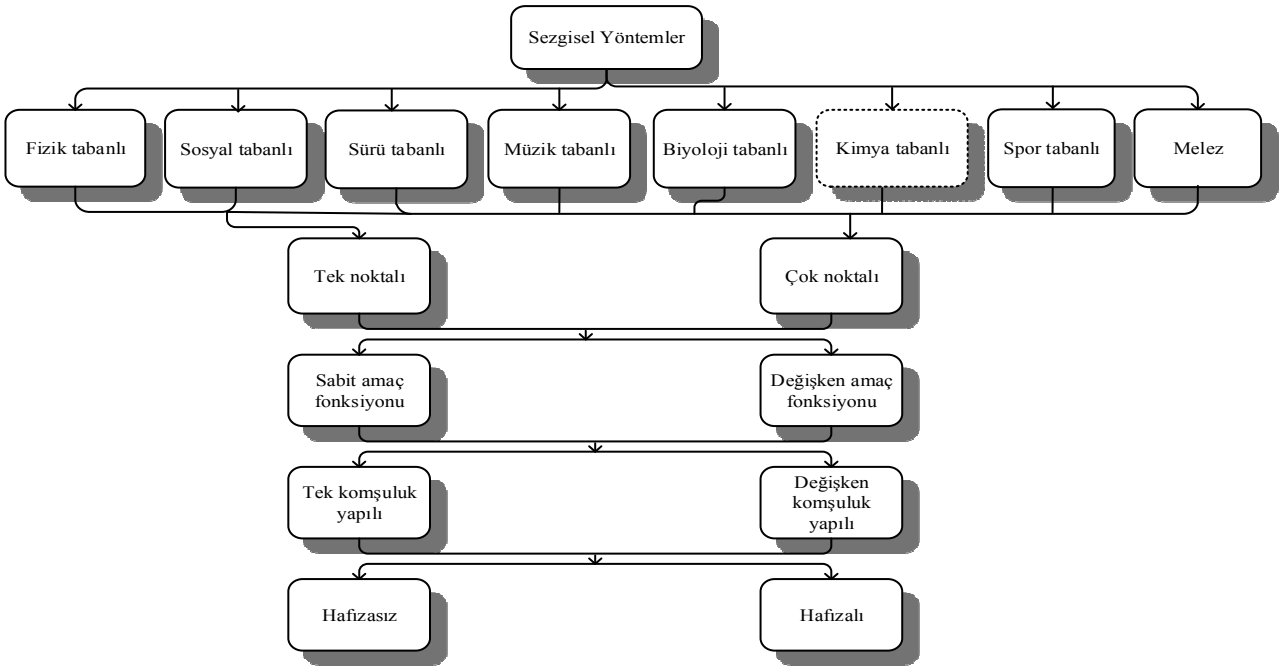
f. Büyük ölçekli kombinasyonsal ve doğrusal olmayan problemlerde etkili sonuçlar vermektedir.

g. Klasik algoritmalardaki gibi verilen bir probleme bir çözüm algoritması uyarlamada geçerliliğinin onaylanması zor olabilen bazı varsayımlar, ihmaller, doğrusallaştırmalar gerektirmemektedir. Klasik algoritmalardaki gibi ilgilenilen problem üzerinde değişiklik gerektirmemektedir. Genel amaçlı bu algoritmalar farklı türdeki problemleri çözmek için kendilerini uyarlar.

Bu avantajlarından dolayı metasezgisel algoritmalar; yönetim bilimi, mühendislik, bilgisayar gibi farklı birçok alanda yoğun olarak kullanılmakta ve bu algoritmaların yeni versiyonları önerilmektedir [24].

Genel amaçlı metasezgisel yöntemler; biyolojik tabanlı (evrimsel algoritmalar, karınca koloni algoritması, arı koloni algoritması, yapay bağışıklık algoritmaları, ateşböceği algoritması, enzim algoritması, fidan gelişim algoritması, istilacı yabani ot optimizasyonu, maymun arama algoritması, bakteriyel yem arama algoritması), fizik tabanlı (çok noktalı ısıl işlem algoritması, elektromanyetizma algoritması, parçacık çarpışma algoritması, büyük patlama - büyük çökme algoritması), sürü tabanlı (parçacık sürü optimizasyonu, karınca koloni optimizasyonu, arı koloni optimizasyonu), sosyal tabanlı (çok noktalı tabu araştırma algoritması, emperyalist yarışmacı algoritma, parlamenter optimizasyon algoritması), müzik tabanlı (armoni arama), spor tabanlı (lig şampiyonası algoritması) ve bu makale kapsamında kullanılan kimya tabanlı yöntemler olmak üzere yedi farklı grupta değerlendirilmektedir (Şekil 1) [25]. Ayrıca bunların birleşimi olan melez yöntemler de vardır [26].

Her ne kadar literatüre kazandırılmış çok başarılı algoritmalar ve teknikler geliştirilmiş olsa da; bilimsel alanda sürekli iyileşme ve daima daha iyiyi arama felsefesi altında yeni tekniklerin tasarlanması, geliştirilmesi ve uygulanması önemli bir görevdir. Ayrıca tüm problemler



Şekil 1. Metasezgisel yöntemler (Metaheuristic methods)

için en iyi sonucu veren algoritma henüz tasarlanmadığından sürekli, yeni metasezgisel algoritmalar önerilmekte ya da var olanların daha etkili çalışması için öneriler sunulmaktadır [27]. Son yıllarda bu bilinçle araştırmacılar, yeni metasezgisel yöntemleri başarılı bir şekilde literatüre kazandırmışlar ve başarılı uygulamalar gerçekleştirmişlerdir [28].

Bu makale kapsamında, sınıflandırma kural madenciliği bir optimizasyon problemi olarak ele alınmış ve doğru, anlaşılabilir ve ilginç kural listesi kimyasal reaksiyonların tipleri ve oluş şekilleri göz önüne alınarak literatürdeki optimizasyon algoritmalarından daha gürbüz ve daha az parametreli yeni bir hesapsal yöntem olan Yapay Kimyasal Reaksiyon Algoritması (YAKROA) [29] ile ilk kez elde edilmeye çalışılmıştır. Bu yöntemde, yoğun işlem gerektiren budama işlemine gerek duyulmamakta ve bu iş kural keşif aşamasında direkt halledilmektedir. Ayrıca bu yöntem 'ayır-ve-yönet' stratejini kullanmaz. Bu şekilde diğer önerilen yöntemlerde ortaya çıkabilecek kurallar arasında beklenmedik etkileşimler ortadan kalkacaktır. Bu etkileşimler, bir örnek farklı sınıfların birkaç kuralı tarafından kapsandığı zaman ortaya çıkabilir. Diğer önerilen algoritmalarda parçalanma problemi ortaya çıkabilir. Kapsama algoritmaları bir kural üretildiğinde tüm eğitim verisindeki kapsanan örnekleri çıkarır ve iterasyonlardan sonra eğitim örneklerinin sayısını azaltır ve lokal olarak önemli ancak global olarak önemsiz kuralların üretilmesine yol açar. YAKROA ile, bu global olarak önemli kurallar bulunmuştur.

YAKROA birden fazla sınıflama etiketi bulunan ve tahmin edilecek birden fazla hedef niteliğin olduğu veritabanlarında, sınıflandırma kural madenciliğinin daha genelleştirilmiş bir hali olan, genelleştirilmiş kural çıkarma problemi için de çözüm yöntemi olarak ele alınmıştır. İlginçlik kriteri de eklenerek; algoritmanın keşfedeceği kuralların ilginç, beklenmedik ve sürpriz olması için gerekli düzenlemeler yapılmış ve birden fazla hedef niteliğe sahip olma potansiyeli bulunan veritabanlarında ilk kez kural bulma algoritması olarak çalıştırılmıştır.

2. YAPAY KİMYASAL REAKSİYON OPTİMİZASYON ALGORİTMASI (ARTIFICIAL CHEMICAL REACTION OPTIMIZATION ALGORITHM)

Kimyasal reaksiyonların meydana geldiği ortam, enerji ve madde alışverişlerindeki ilişkilere göre izole, kapalı ve açık olmak üzere üç farklı şekilde olabilir. İzole sistemlerde çevre ile enerji ve madde alışverişi olmaz. Kapalı sistemlerde çevre ile enerji alışverişi olur, fakat madde alışverişi olmaz. Açık sistemlerde ise çevre ile hem enerji hem madde alışverişi olur. İlgilenilen optimizasyon problemine göre uygun sistem seçilecek ve reaksiyona tabi tutulacak ilk moleküller düzenli bir şekilde algoritmanın başlangıç aşaması olarak üretilmiştir. Moleküllerin ortama düzgün şekilde dağıtılması bir algoritma olarak ele alınmıştır. Kimyasal tepkimeler gerçekleşirken bazı maddeler arasında bağlar koparken, bazı maddeler arasında yeni bağlar ve böylece kimyasal değişimler olur. Bir reaksiyon sonucunda reaksiyona giren reaktantın veya reaktantların yapısı ve enerjisi değişir. Bir reaksiyon sonunda oluşan moleküller, yeni farklı reaksiyonlar için birer reaktant olarak düşünebilir. Ayrıca iki yönlü reaksiyonlarda, oluşan moleküller tekrar ilk reaktantlarına dönüşebilir. Böylece kimyasal reaksiyonların tipleri, oluş şekilleri göz önüne alınarak literatürdeki algoritmalardan daha gürbüz ve daha az parametre gerektiren yeni bir hesapsal yöntem geliştirilmiştir ve bu yeni yönteme YAKROA adı verilmiştir [29, 30].

Bir molekül atom tipi, bağ uzunluğu, açısı ve bükülmeleri ile karakterize edilen birkaç atomdan oluşmaktadır ve biri diğerinden farklı atom ya da farklı sayıda atom içermesiyle ayrılmaktadır. Herhangi bir kimyasal reaksiyonda başlangıç reaktantlar tek-çift biçimlenme, tek-çift yıkım ve kimyasal bağların kayması yoluyla yeni ürünlere dönüşür. Algoritmanın ilk adımı olan reaktantların oluşturulması aşamasında düzenli popülasyon yöntemi kullanılmıştır. Reaktantlar normal olarak temel adımlar süresince geçiş durumlarında bir dizi ara reaktantlara dönüşür. Moleküller de YAKROA'da optimizasyon problemleri için uygun kodlama tekniği ile kodlanır. Bir minimizasyon probleminde amaç fonksiyonu olarak ta entalpi, maksimizasyon probleminde ise entropi kullanılabilir [30].

Akış diyagramı Şekil 2'de ve sözde kodu Şekil 3'te verilen ve her türlü kodlama çeşidi için çalışabilen global optimizasyon için geliştirilen YAKROA, toplam beş aşamadan oluşmaktadır [29, 30]:

Adım 1: Problem ve algoritma parametrelerinin ayarlanması

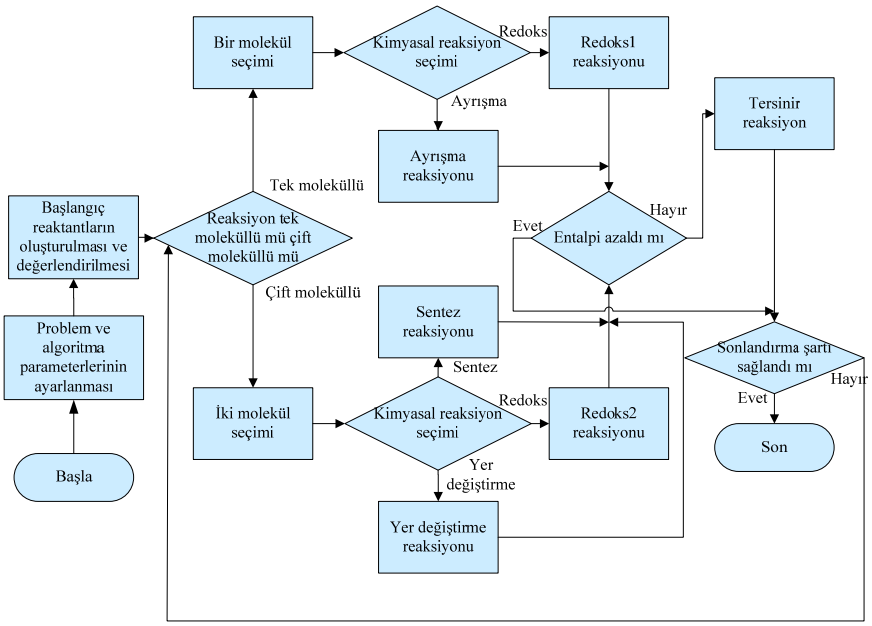
Adım 2: Başlangıç reaktantların hazırlanması ve değerlendirilmesi

Adım 3: Kimyasal reaksiyon operatörlerinin uygulanması

Adım 4: Reaktantların güncellenmesi

Adım 5: Sonlandırma şartı kontrolünün yapılması

İlk adımda ilgili uygun parametreler belirlendikten sonra ikinci adımda reaktantlar uygun arama uzayında düzgün şekilde oluşturulur. Başlangıç reaktantlarının oluşturulması için düzenli popülasyon yöntemi kullanılmıştır [31, 32]. Düzenli popülasyon ile çözüm adayı olan reaktantlar arama uzayına mümkün olduğunca düzgün olarak dağıtılır [33]. Bu şekilde daha kısa sürede en iyi çözümlere yakınsamak ve daha iyi sonuçlar elde etmek amaçlanmaktadır [34]. Diğer algoritmalarda bazen lokal minimum veya maksimumlara takılmalar olabilmektedir [34, 35]. Başlangıç reaktantların oluşturulması için kullanılan düzenli popülasyon ile bu problemler ortadan kaldırılmıştır. Başlangıç reaktantların düzenli bir şekilde oluşturulması adımları Şekil 4'te gösterilmiştir.



Şekil 2. YAKROA'nın akış diyagramı (Flowchart of ACROA)

Giriş: Problem bağımlı bilgiler (amaç fonksiyonu, karar değişken sayısı gibi) ve algoritma bağımlı bilgi, yani başlangıç reaktant sayısı (*ReakSay*)

ReakSay sayısında m_i molekülleri düzenli popülasyon yöntemiyle üret
for $i \leftarrow 1$ to *ReakSay* **do**
  Entalpiyi hesapla $e(m_i)$
end for
while sonlandırma şartı sağlanmadı **do**
  for $i \leftarrow 1$ to *ReakSay* **do**
    [0, 1] aralığında rassal r_1 sayısı üret
    if $r_1 \leq 0{,}5$ **then**
      [0, 1] aralığında rassal r_2 sayısı üret
      if $r_2 \leq 0{,}5$ **then**
        Ayrisma(m_i)
      end if
      else
        Redoks1(m_i)
      end if
    else
      Başka bir molekül m_j ($m_i \neq m_j$) daha seç
      [0, 1] aralığında rassal r_3 sayısı üret
        if $0 \leq r_3 \leq 0{,}33$ **then**
          Sentez(m_i, m_j)
        end if
        if $0{,}33 < r_3 \leq 0{,}66$ **then**
          Yerdegistirme(m_i, m_j)
        end if
        else
          Redoks2(m_i, m_j)
        end if
    end if
    Artan entalpi için tersinir reaksiyon uygula
  end for
end while
Çıkış minimum çözüm ve amaç fonksiyon değeri

Şekil 3. YAKROA'nın sözde kodu (Pseudo-code of ACROA)

```
// R reaktant kümesi; I indis kümesi ve Ie genişletilmiş indis kümesidir.
1-  Değişkenler için biri tüm üst sınırları içeren R[1] ve tüm alt sınırları içeren R[2] şeklinde iki reaktant üret.
2-  Indeks ← 3
3-  k ← 2
4-  while R doymadığı müddetçe do
    ie, Ie'in bir elemanı olsun ve her ie bit değeriyle genişletilsin ve bit değeri parçanın karşılığı olsun.
    i ← 1
    while R doymadığı müddetçe ve tüm reaktantlar belirli k değeri için üretilmediği müddetçe (ve I ≤ 2^k-2) do
      i k-bit bir sayıdır ve ie i'nin genişletilmiş değerine tekabül etmektedir. i'nin her biti R[0] ve R[1]'in ilgili parçasının
      uzunluğuna genişletilir.
      for j ← 1 to n do
        if ie'nin j. biti 1 then
          R[Indeks]'in j. değeri R[1]* r olur
        else
          R[Indeks]'in j. değeri R[2]* r olur
        end if
        r [0,1] aralığında rassal reel bir sayıdır
      end for
    Indeks ← Indeks+1
    i ← i + 1
  end while
  k ← k + 1
end while
```

Şekil 4. Başlangıç reaktantların oluşturulması adımları (Steps of generating initial reactants)

Üçüncü adımdaki kimyasal reaksiyonlar, tek moleküllü reaksiyonlar ve iki moleküllü reaksiyonlar olarak ikiye ayrılmaktadır. Tek moleküllü reaksiyonlar; Ayrışma reaksiyonu, Redoks1 reaksiyonu; iki moleküllü reaksiyonlar ise Sentez reaksiyonu, Yer değiştirme reaksiyonu, Redoks2 reaksiyonu olarak düşünülüp birer operatör olarak yöntemde kullanılmıştır.

O anki popülasyona bütün operatörler uygulandıktan sonra elde edilen yeni çözümlerin değerlendirmesi yapılıp, daha sonra yeni reaktantlar ile var olan reaktantlar arasından bir sonraki nesil seçilirse, bu YAKROA1 olarak adlandırılmıştır. Yöntem farklı bir şekilde daha uygulanabilir. Var olan popülasyona operatörler sırayla uygulanır ve iyiler seçilip bir sonraki operatör uygulaması yapılırsa, bu da YAKROA2 olarak adlandırılmıştır. YAKROA'nın bu versiyonlarının sözde kodları, Şekil 5 ve Şekil 6'da verilmiştir.

3. YAKROA'NIN TEORİSİ (THEORY OF ACROA)

YAKROA stokastik bir yöntem olup, amaç fonksiyon değerini kılavuz olarak kullanan tepe tırmanma işlemini gerçekleştiren bir algoritma olarak düşünülebilir. Başlangıç reaktantları, birden çok noktadan araştırma imkânı sağlamaktadır ve tepe tırmanma işlemi sadece o anki çözümün hatırlandığı anlamına gelmektedir. Bitim şartı, iterasyon sayısı olarak verilebildiği gibi en iyi amaç değerinin belli bir süre değişmemesi olarak ta verilebilir. YAKROA için bu bölümde verilen tanımlar [34, 36] kaynağındaki tanımlardan esinlenerek düzenlenmiştir.

Tanım 3.1. Hesapsal metot HM = (*Q, I, O, f*) ve
Q: Hesaplama durumları kümesi
I: Algoritma için girişler kümesi
O: Algoritma için çıkışlar kümesi
f: Hesapsal kurallar

```
1. t ← 0 // başlangıç zamanı
2. MolekülleriUret(R(t)) // t zaman parametresi
3. AmaçDegeriniHesapla(R(t))
4. While bitim şartı sağlanmadıysa do
 4.1. R1(t) ← Ayrışma(R(t))
 4.2. R2(t) ← Redoks1(R(t))
 4.3. R3(t) ← Sentez(R(t))
 4.4. R4(t) ← Yerdeğiştirme(R(t))
 4.5. R5(t) ← Redoks2(R(t))
 4.6. AmaçDegeriniHesapla(R1(t) ∪ R2(t) ∪ R3(t) ∪ R4(t) ∪ R5(t))
 4.5. G(t+1) ← Seçme(R(t) ∪ R1(t) ∪ R2(t) ∪ R3(t) ∪ R4(t) ∪ R5(t))
 4.6. t ← t + 1
```

Şekil 5. YAKROA1 sözde kodu (Pseudo-code of ACROA1)

1. $t \leftarrow 0$ // başlangıç zamanı
2. MolekülleriUret(R(t)) // t zaman parametresi
3. AmaçDegeriniHesapla(R(t))
4. **While** bitim şartı sağlanmadıysa **do**
4.1. $R_1(t) \leftarrow$ Ayrışma (R(t))
4.2. AmaçDegeriniHesapla($R_1(t)$)
4.3. $R_m(t) \leftarrow$ Seçme($R_1(t) \cup R(t)$)
4.4. $R_2(t) \leftarrow$ Redoks1 (R_m(t))
4.5. AmaçDegeriniHesapla($R_2(t)$)
4.6. $R_b(t) \leftarrow$ Seçme($G_2(t) \cup R_m(t)$)
4.7. $R_3(t) \leftarrow$ Sentez ($R_b(t)$)
4.8. AmaçDegeriniHesapla($R_3(t)$)
4.9. $R_k(t) \leftarrow$ Seçme(R_3(t) $\cup$ $R_b(t)$)
4.10. $R_4(t) \leftarrow$ Yerdeğiştirme($R_k(t)$)
4.11. AmaçDegeriniHesapla($R_4(t)$)
4.12. $R_l(t) \leftarrow$ Seçme($R_4(t) \cup R_k(t)$)
4.13. $R_5(t) \leftarrow$ Redoks2($R_l(t)$)
4.14. AmaçDegeriniHesapla($R_5(t)$)
4.15. R(t+1) $\leftarrow$ Seçme(R_5(t) $\cup$ $R_l(t)$)
4.16. $t \leftarrow t + 1$

Şekil 6. YAKROA2 sözde kodu (Pseudo-code of ACROA2)

Q kümesi I ve O kümelerinin bir alt kümesidir. f: $Q \rightarrow Q$ ve O kümesinin elemanları için bir birim fonksiyondur. I kümesinin elemanlarının x_0, x_1, …, olduğu kabul edilsin ve $x_0 = x$ ve $x_{k+1} = f(x_k)$, $k \geq 0$.

Eğer minimum k değeri için $x_k \in O$ ise, hesapsal dizi biter. Bu durumda x_k verisi x verisinden elde edilir [32, 34].

Tanım 3.2. YAKROA, ayrık zamanda t tekrarlı bağıntı olarak çalışabilir. Ayrık zaman adımların her birine nesil denirse, t = 0, 1, 2, ... : R(t + 1) = Seç(R2(Y(S(R1(A(R(t))))))). R(t) o anki reaktantlar ve R(t+1) ise elde edilen yeni reaktantlardır ve bu reaktantlar; Ayrışma(A), Redoks1(R1), Sentez(S), Yerdeğiştirme(Y), Redoks2(R2) ve Seçim operatörlerinin uygulanması ile elde edilir. R(t) ve R(t+1) kümeleri mümkün olan bütün reaktantlar kümesi olan M kümesinin elemanlarıdır. R(0) $\in$ M reaktantları, başlangıç reaktantları, $R_f \in M$ reaktantları en az bir tane optimum veya optimuma yakın çözüm içeren reaktantlardır ve bu reaktant bitim şartını sağlamaktadır. Arzu edilen bitim şartı amaç fonksiyonunun f(R[t]) optimum durumudur.

Tanım 3.3. YAKROA bir hesapsal metot olup YAKROA = (Q, I, O, f) ve Q nesiller dizisi; I üretilen reaktantlar kümesi; O reaktantlar kümesi ve f ise her nesilde uygulanan kurallardır ve f = Seç(R2(Y(S(R1(A(…)))))))

Bilginin temsili çok değişik şekillerde yapılabilir. Kodlama probleme ve modelleme durumuna bağlı olarak değişir. Temsil operatörlerin uygulanma sırasını ve düzenini belirler. Bu operatörlerin sırası değişebilir. YAKROA her nesilde reaktantlar üzerinde çalışır ve operatörler bu reaktantlara uygulanır. YAKROA için Turing makinesi tanımı yapılabilir. YAKROA için tanımlanacak Turing Makinesi aslında bir Turing Makineleri serisidir ve bu makineler reaktantlar üzerinde çalışırlar. Geçiş fonksiyonu her nesilde YAKROA'yı temsil eder. Bu makine serisi (YAKROA-M) aşağıdaki gibi tanımlanabilir.

Tanım 3.4. YAKROA-M mümkün olan sonsuz tane Turing makinesi serisinden oluşur ve YAKROA1 ve YAKROA2 için Turing makinesi tanımlanabilir.

a) Şekil 4'te verilen algoritma ile R(0) oluşturulur ve bu algoritma TM(0) ile temsil edilebilir. Bu makinenin geçiş fonksiyonu R(0) reaktantları oluşturma kurallarıdır.

b) Her neslin çıktısı bir ikili olup (TM(t+1), R(t+1), TM(t+1) bir Turing makinesi ve R(t+1) ise reaktantlardır. Bu reaktantlar f = Seç(R2(Y(S(R1(A(…)))))) kurallarının uygulanması ile elde edilir.

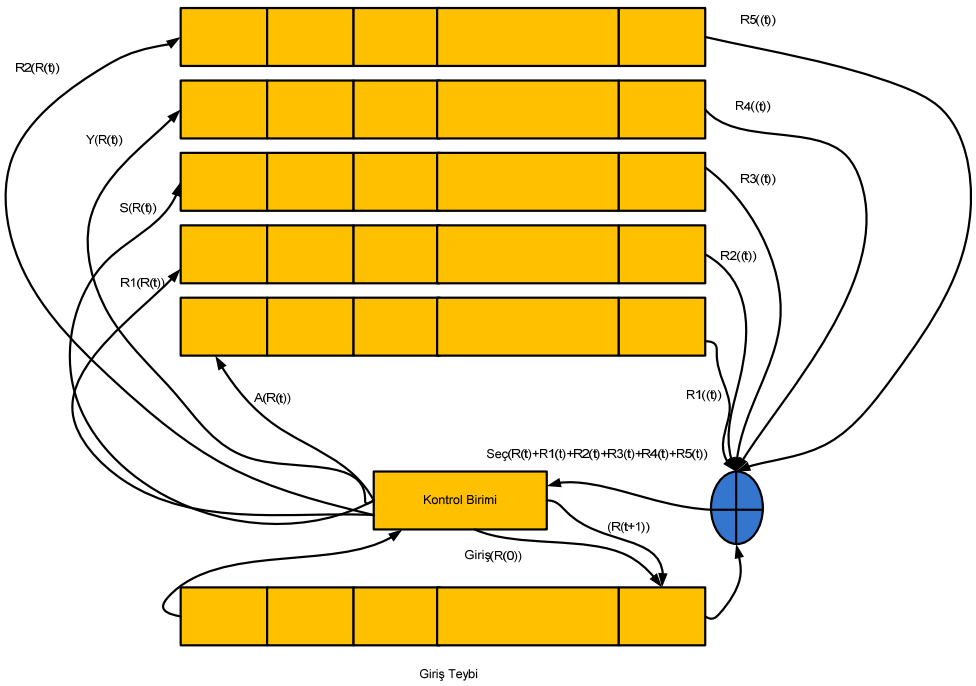
c) Optimum veya optimuma yakın amaç fonksiyonu elde edildiğinde (M(t), R(t)) bitim şartını sağlayan ikilidir.

d) Amaç değeri hesaplanırken f(TM(t), R(t)) = (TM(t), f(R(t))) ve f(.) fonksiyonu probleme özel bir fonksiyondur [34, 36].

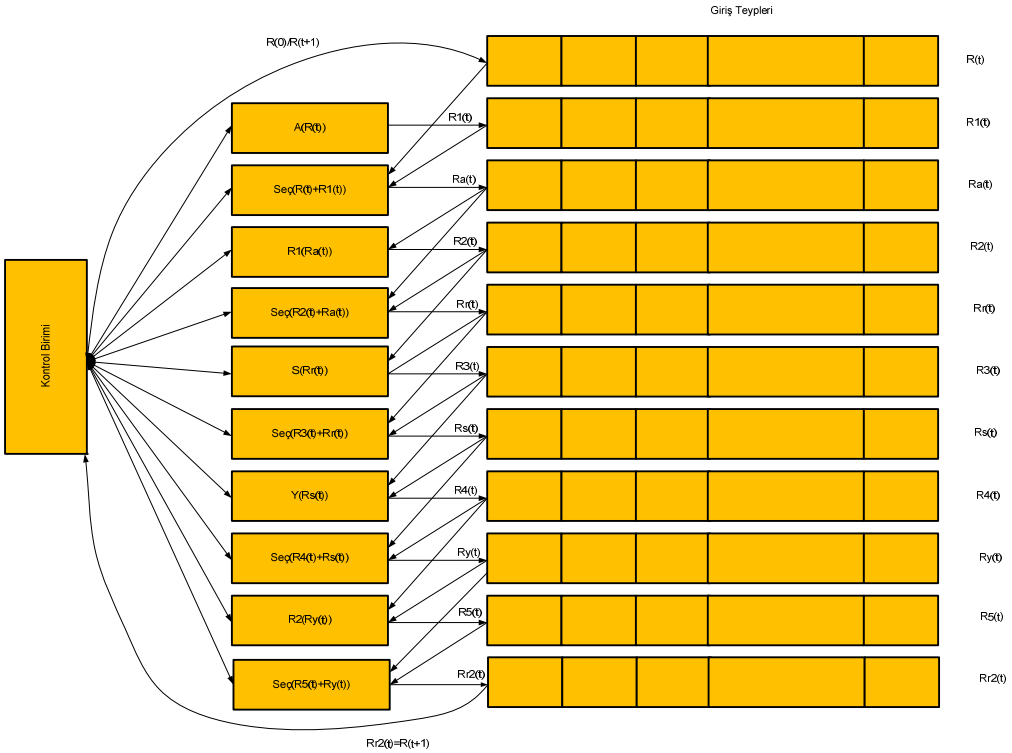
YAKROA-M bir Turing makineleri serisidir ve serideki ilk Turing Makinesi diğerlerinden farklıdır, çünkü ilk Turing makinesi Şekil 4'te verilen algoritmaya karşılık gelecek; diğerleri ise, operatörlerin uygulanması aşamasına denk gelecektir. Şekil 7'de YAKROA1-M için Turing makinesi ana hatları ile görülmektedir. YAKROA2-M için Turing makinesi ise ana hatları ile Şekil 8'de gösterilmiştir

4. GENELLEŞTİRİLMİŞ İLGİNÇ KURALLARIN YAKROA İLE KEŞFİ (MINING OF GENERALIZED INTERESTING RULES WITH ACROA)

Farklı veri tipleri, eksik veri ve gürültülü veri içeren veritabanlarında kural keşfi klasik yöntemlerle yapılamamaktadır.



Şekil 7. YAKROA1-M için Turing makinesi (Turing machine for ACROA1-M)



Şekil 8. YAKROA2-M için Turing makinesi (Turing machine for ACROA2-M)

Yapılmak istense bile veri üzerinde ön işlemler gerekmekte, bu da veri üzerinde değişiklik (dolayısıyla problem üzerinde değişiklik) getirmekte, bazı istenmeyen durumların ortaya çıkmasına neden olmaktadır. Ayrıca klasik yöntemlerle kural keşfi sonrası da bazı ek işlemler (budama vb.) yapılmaktadır. Başka bir problem ise diğer yöntemlerle bulunacak kuralların arasında beklenmedik etkileşimlerin ortaya çıkmasıdır. Kapsama problemi ve lokal olarak önemli ancak global olarak önemsiz kuralların bulunması problemi de diğer yöntemlerdeki dezavantajlardır. YAKROA ile bu dezavantajlar giderilmiş; daha doğru, anlaşılabilir ve ilginç kuralların otomatik keşfi amaçlanmıştır. Ayrıca esnek amaç fonksiyonu sayesinde, farklı amaçları karşılayabilecek kurallar kolaylıkla bulunabilir.

4.1. Doğru ve Anlaşılabilir Sınıflandırma Kurallarının Keşfi
(Mining of Accurate and Comprehensible Classification Rules)

Veritabanları arama uzayı, YAKROA da arama yöntemi olarak tasarlanmış; gerekli kodlamalar ve amaç fonksiyonu belirlenmiş ve halka açık iki gerçek veritabanında uygulamalar yapılmıştır. YAKROA'nın doğru ve anlaşılabilir kural keşfi yapan algoritma olarak uyarlanabilmesi için kullanılan reaktant temsili Şekil 9'da gösterilmiştir. *ns* sayıda nitelik bulunan veritabanında *ns* atom bulunmakta ve her atomda iki alan bulunmaktadır: *Bayrak* (B_i) ve *Değer* (D_i). Reaktant kuralın IF kısmını kodlamaktadır ve her atom da bu IF kısmındaki kuralın bir konumunu temsil etmektedir. Sınıf, bu temsil içinde kodlanmaz. Algoritmanın çalıştırılması esnasında reaktantlar aynı sınıfın kuralı için arama yapar. *Bayrak* ikili değer alır. Eğer değeri 1 ise o zaman *i*. nitelik kuralın ata kısmında yer alacak, aksi durumda yer almayacaktır. *Değer* alanı ise ilgili niteliğin uygun sınırlardaki değerini temsil eder.

$Atom_1$				$Atom_{ns}$	
B_1	D_1	...	...	B_{ns}	D_{ns}

Şekil 9. Reaktantta kural temsili (Rule representation in reactant)

Keşfedilen kuralların yüksek tahmini doğruluk ve anlaşılabilirliğe sahip olması gerekmektedir. Bu yüzden amaç fonksiyonu için tahmini doğruluk ve anlaşılabilirlik ölçüleri tanımlanmalıdır. Doğruluk için Eş. 1 kullanılmıştır.

$$Tahmini\ Doğruluk = \frac{|A\ \&\ S| - 1/2}{|A|} \quad (1)$$

$|A\&S|$ kuralın hem ata (A) hem de sonuç (S) kısmını karşılayan kayıt sayısıdır. $|A|$ ise sadece kuralın ata kısmındaki şartları sağlayan kayıtların sayısıdır. Anlaşılabilirlik için ise Eş. 2 kullanılmıştır.

$$Anlaşılabilirlik = 1 - \frac{|B_i = 1|}{|B_i|} \quad (2)$$

Anlaşılabilirlik ölçüsünde $|B_i|$, reaktantta kodlanan şartların (karar niteliği) sayısıdır ve benzer olarak $|B_i=1|$ de reaktantın bayrak kısmında 1'e eşit olan şartların sayısıdır [30].

YAKROA, Zoo [37] ve Nursery [37] veritabanında uygulanmıştır. Başlangıç reaktant sayısı 50 olarak seçilmiş ve 10 iterasyon boyunca en iyi reaktantta gelişme olmadığı anda algoritma sonlandırılmıştır. Veritabanının 1/3'ü eğitim, 2/3'ü test olarak kullanılmıştır. YAKROA ile birlikte Genetik Algoritma (GA), Parçacık Sürü Optimizasyonu (PSO) ve Karınca Koloni Optimizasyonu (KKO) da aynı veritabanlarında her sınıflandırma kuralı için 3 kez çalıştırılmış ve kurallar iterasyon sayısı kaydedilmiştir. Reaktant sayısı kadar GA için kromozom, PSO için parçacık ve KKO için karınca kullanılmıştır. Zoo veritabanı için 60, Nursery veritabanı için yaklaşık 200 iterasyonla sonuçlar bulunmuştur. GA, PSO ve KKO için kullanılan parametreler Tablo 4'te verilmiştir.

GA, PSO ve KKO'da; Zoo veritabanı için yaklaşık 100 Nursery veritabanı için yaklaşık 450 iterasyon kullanmıştır. Zoo veritabanı için YAKROA, GA, PSO ve KKO ile elde edilen sonuçlar sırasıyla Tablo 5, Tablo 6, Tablo 7 ve Tablo 8'de gösterilmiştir. YAKROA, yeni bir yöntem olmasına rağmen bu veritabanında daha hızlı bir şekilde, diğer kullanılan yöntemler ile elde edilen sonuçlardan daha iyi ya da onlara eşdeğer düzeyde sonuçlar bulmuştur.

Tablo 4. GA, PSO ve KKO için parametre değerleri (Parameter values for GA, PSO, and ACO)

GA		PSO		KKO	
Çaprazlama Oranı	0,8	Atalet ağırlığı	0,9-0,4 azalan		
		Minimum hız	0		
Mutasyon Oranı	0,03	Maksimum hız	1	Alfa parametresi	1
		Minimum pozisyon	0		
		Maksimum pozisyon	1		
Seçim	10 boyutlu turnuva	Bilişsel öğrenme katsayısı	2	Beta parametresi	1
		Sosyal öğrenme katsayısı	2		

Tablo 5. Zoo veritabanından YAKROA ile elde edilen kurallar (Obtained rules via ACROA from Zoo data set)

Sınıf	Kural	Tahmini doğruluk	Anlaşılabilirlik
1	If (eggs = 0) ∧ (venomous = 0) ∧ (domestic = 0)	0,91	0,81
2	If (hair = 0) ∧ (feathers = 1) ∧ (venomous = 0) ∧ (legs = 2) ∧ (domestic = 0)	1,00	0,69
3	If (eggs = 1) ∧ (predator = 1) ∧ (toothed = 1) ∧ (catsize = 0)	1,00	0,75
4	If (aquatic = 1) ∧ (breathes = 0) ∧ (tail = 1)	0,80	0,81
5	If (airborne = 0) ∧ (aquatic = 1) ∧ (toothed = 1) ∧ (catsize = 0)	1,00	0,75
6	If (airborne = 1) ∧ (fins = 0) ∧ (tail = 0)	0,83	0,81
7	If (predator = 1) ∧ (breathes = 0) ∧ (tail = 0) ∧ (domestic = 0)	0,88	0,75

Tablo 6. Zoo veritabanından GA ile elde edilen kurallar (Obtained rules via GA from Zoo data set)

Sınıf	Kural	Tahmini doğruluk	Anlaşılabilirlik
1	If (eggs = 0) ∧ (venomous = 0) ∧ (domestic = 0)	0,91	0,81
2	If (feathers = 1) ∧ (breathes = 1) ∧ (domestic = 0)	0,93	0,81
3	If (eggs = 1) ∧ (predator = 1) ∧ (catsize = 0)	0,99	0,81
4	If (aquatic = l) ∧ (breathes = 0) ∧ (tail = l)	0,80	0,87
5	If (airborne = 0) ∧ (aquatic = 1) ∧ (toothed = 1) ∧ (catsize = 0)	1,00	0,75
6	If (airborne = 1) ∧ (fins=0) ∧ (tail = 0)	0,83	0,81
7	If (predator = 1) ∧ (breathes = 0) ∧ (domestic = 0)	0,88	0,81

Tablo 7. Zoo veritabanından PSO ile elde edilen kurallar (Obtained rules via PSO from Zoo data set)

Sınıf	Kural	Tahmini doğruluk	Anlaşılabilirlik
1	If (milk =1)	0,90	0,94
2	If (feathers = 1) ∧ (toothed = 0)	0,97	0,88
3	If (hair = 0) ∧ (feathers = 0) ∧ (aquatic = 0) ∧ (backbone = 1)	0,88	0,75
4	If (milk = 0) ∧ (fins = 1)	0,96	0,88
5	If (milk = 0) ∧ (aquatic = 1) ∧ (breathes = 1) ∧ (catsize = 0)	0,88	0,75
6	If (aquatic = 0) ∧ (legs = 6)	0,94	0,88
7	If (airborne = 0) ∧ (backbone = 0)	0,79	0,88

Tablo 8. Zoo veritabanından KKO ile elde edilen kurallar (Obtained rules via ACO from Zoo data set)

Sınıf	Kural	Tahmini doğruluk	Anlaşılabilirlik
1	If (hair = 1) ∧ (domestic = 0) ∧ (venomous = 0) ∧ (eggs = 0)	0,95	0,75
2	If (feathers = 1) ∧ (breathes = 1) ∧ (domestic =0)	0,93	0,81
3	If (hair = 0) ∧ (aquatic = 0) ∧ (predator = 1) ∧ (toothed = 1) ∧ (domestic = 0)	0,84	0,69
4	If (eggs = 1) ∧ (fins = 1) ∧ (domestic = 0)	0,94	0,81
5	If (feathers= 0) ∧ (airborne = 1) ∧ (aquatic = 1) ∧ (breathes = 1) ∧ (catsize = 0)	0,83	0,69
6	If (airborne = 1) ∧ (fins=0) ∧ (tail = 0)	0,83	0,81
7	If (airborne = 0) ∧ (backbone = 0)	0,79	0,88

Nursey veritabanı için YAKROA, GA, PSO ve KKO ile elde edilen sonuçlar sırasıyla Tablo 9, Tablo 10, Tablo 11 ve Tablo 12'de gösterilmiştir. Algoritmaların ortalama performansları ise Tablo 13'te verilmiştir. YAKROA, yeni bir yöntem olmasına rağmen bu veritabanında daha hızlı bir şekilde, diğer kullanılan yöntemler ile elde edilen sonuçlardan daha iyi ya da onlara eşdeğer düzeyde sonuç bulmuştur ve YAKROA'nın sınıflandırma kural madenciliğinde etkili bir yöntem olduğu görülmüştür.

10-kat çapraz geçerlilik testi sonuçları ise Tablo 14 ile Tablo 15'te verilmiştir.

Tablo 9. Nursery veritabanından YAKROA ile elde edilen kurallar (Mined rules via ACROA from Nursery data set)

Sınıf	Kural	Tahmini doğruluk	Anlaşılabilirlik
NR	If (parents = pretentious) ∧ (children = 3) ∧ (housing = convenient) ∧ (health = not_recom)	0,76	0,50
R	If (has_nurs = proper) ∧ (finance = convenient)	0,75	0,75
VR	If (housing = less_conv) ∧ (finance = inconv) ∧ (social = slightly_prob)	0,90	0,63
P	If (parents = usual) ∧ (housing = less_conv) ∧ (social = problematic)	0,78	0,63
SP	If (parents = usual) ∧ (has_nurs = very_crit) ∧ (form = more)	0,79	0,63

Tablo 10. Nursery veritabanından GA ile elde edilen kurallar (Mined rules via GA from Nursery data set)

Sınıf	Kural	Tahmini doğruluk	Anlaşılabilirlik
NR	If (parents = usual) ∧ (housing = less_conv) ∧ (social = slightly_prob) ∧ (health = not_recom)	0,63	0,5
R	If (has_nurs = proper) ∧ (finance = convenient) ∧ (health = recommended)	0,81	0,63
VR	If (housing = less_conv) ∧ (finance = inconv) ∧ (social = slightly_prob)	0,90	0,63
P	If (parents = great_pret) ∧ (children = 3) ∧ (social slightly_prob) ∧ (health = recommended)	0,69	0,50
SP	If (parents = usual) ∧ (has_nurs = very_crit) ∧ (form = more)	0,79	0,63

Tablo 11. Nursery veritabanından PSO ile elde edilen kurallar (Mined rules via PSO from Nursery data set)

Sınıf	Kural	Tahmini doğruluk	Anlaşılabilirlik
NR	If (has_nurs = proper) ∧ (finance = convenient) ∧ (social = non_prob) ∧ (health = not_recom)	0,98	0,5
R	If (parents = usual) ∧ (has_nurs = proper) ∧ (finance = convenient) ∧ (children = 1) ∧ (form = complete) ∧ (housing = convenient) ∧ (social = non_prob) ∧ (health = recommended)	0,50	0,0
VR	If (parents = usual) ∧ (has_nurs = less_proper) ∧ (housing = convenient) ∧ (finance = convenient) ∧ (social = slightly_prob) ∧ (health = recommended)	0,97	0,25
P	If (has_nurs = proper) ∧ (housing = convenient) ∧ (finance = convenient) ∧ (health = priority)	0,99	0,5
SP	If (parents = usual) ∧ (has_nurs = very_crit) ∧ (form = more)	0,79	0,63

Tablo 12. Nursery veritabanından KKO ile elde edilen kurallar (Mined rules via ACO from Nursery data set)

Sınıf	Kural	Tahmini doğruluk	Anlaşılabilirlik
NR	If (has_nurs = proper) ∧ (finance = convenient) ∧ (social = non_prob) ∧ (health = not_recom)	0,98	0,5
R	If (parents = usual) ∧ (has_nurs = proper) ∧ (finance = convenient) ∧ (children = 1) ∧ (form = complete) ∧ (housing = convenient) ∧ (social = non_prob) ∧ (health = recommended)	0,50	0,0
VR	If (parents = usual) ∧ (has_nurs = less_proper) ∧ (housing = convenient) ∧ (finance = convenient) ∧ (social = slightly_prob) ∧ (health = recommended)	0,97	0,25
P	If (parents = great_pret) ∧ (children = 3) ∧ (social slightly_prob) ∧ (health = recommended)	0,69	0,50
SP	If (has_nurs = very_crit) ∧ (finance = inconvenient) ∧ (social = problematic) ∧ (health = recommended)	0,99	0,5

Tablo 13. Ortalama performans (Mean performance)

Veritabanı / Algoritma	Zoo		Nursery	
	Tahmini doğruluk	Anlaşılabilirlik	Tahmini doğruluk	Anlaşılabilirlik
YAKROA	0,92	0,82	0,80	0,63
GA	0,91	0,86	0,76	0,58
PSO	0,90	0,85	0,84	0,38
KKO	0,87	0,78	0,82	0,35

Tablo 14. Zoo veritabanında tahmini doğruluk (%) (Predictive accuracy in Zoo data set) (%)

Sınıf	YAKROA	GA	PSO	KKO
1	100∓0,0	100∓0,0	99∓9,2	99∓8,1
2	100∓0,0	100∓0,0	98∓11,2	97∓1,9
3	92∓12,8	0,0∓0,0	83∓0,0	87∓0,9
4	100∓0,0	100∓0,0	96∓0,4	96∓0,8
5	100∓0,0	100∓0,0	88∓0,8	87∓0,0
6	90∓10,0	90∓10,0	89∓0,9	89∓0,9
7	83,9∓10,2	85,5∓0,0	89∓0,2	89∓0,2

Tablo 15. Nursery veritabanında tahmini doğruluk (%) (Predictive accuracy in Nursery data set) (%)

Sınıf	YAKROA	GA	PSO	KKO
NR	33,8∓10,2	12,8∓9,8	81∓0,8	80∓5,5
R	0,0∓0,0	0,0∓0,0	0,0∓0,0	0,0∓0,0
VR	100∓0,0	100∓0,0	0,0∓0,0	0,0∓0,0
P	0,0∓0,0	0,0∓0,0	39∓8,1	45∓2,5
SP	100∓0,0	100∓0,0	60∓1,9	59∓11,0

Farklı kodlama çeşidi ve amaç fonksiyonuyla yapılan bir diğer uygulamada ise Breast Cancer [37] ve Dermatology [37] veritabanları seçilmiş ve sınıflandırma kuralları bulunmaya çalışılmıştır. Bu uygulamada kullanılan pozisyonal kodlamada *n* nitelik sayısını göstermektedir ve reaktant *n* atomdan oluşmaktadır. Her atom da üç parça şeklinde düşünülmüştür: Bayrak (B_i), ilişkisel operatör ($İO_i$) ve Değer (D_i) (Şekil 10).
Bu gösterim kuralın IF (ata) kısmını temsil eder ve sonuç kısmı için YAKROA en az bir kez çalıştırılır. Bayrak ikili bir değerdir ve ilgili niteliğin kuralda olup olmadığını gösterir. İlişkisel operatör nitelik kategorik ise "=" ve "≠" değerlerini; nitelik sürekli ise "<" ve "≥" değerlerini alır. Değer ise niteliğin değerini gösterir.

Reaktantların amaç fonksiyon değerleri için Eş. 3 kullanılmıştır:

$$Amac = \frac{DP}{DP + YN} \times \frac{DN}{DN + YP} \quad (3)$$

- *DP* doğru pozitiflerdir ve kuralla aynı sınıf etiketine sahip olan, kural tarafından kapsanan örneklerin sayısıdır.
- *YP* yanlış pozitiflerdir ve kuraldan farklı sınıf etiketine sahip olan, kural tarafından kapsanan örneklerin sayısıdır.
- *YN* yanlış negatiflerdir ve kural tarafından kapsanmayan fakat kuralla aynı sınıf etiketine sahip örneklerin sayısıdır.
- *DN* ise doğru negatiflerdir ve kural tarafından kapsanmayan ve kuralla da aynı etikete sahip olmayan örneklerin sayısıdır.

54 başlangıç reaktantı ile YAKROA kural aramaya başlamış ve 10 iterasyon boyunca amaç fonksiyon değerinde gelişme olmadığında sonlandırılmıştır. Verilerin 2/3'si eğitim kümesi kalanı da test kümesi olarak ayrılmış ve her kümede her sınıfa ait örnek sayısı aynı tutulmuştur. Elde edilen kurallar ve amaç fonksiyon değerleri (eğitim ve test verilerinden) Tablo 16 ve Tablo 17'de gösterilmiştir.

$Atom_1$			$Atom_2$			…	$Atom_n$		
B_1	$İO_1$	D_1	B_2	$İO_2$	D_2	…	B_n	$İO_n$	D_n

Şekil 10. Reaktant gösterimi (Reactant representation

Tablo 16. Breast Cancer veritabanında bulunan kurallar (Mined rules within Breast Cancer data set)

Sınıf	Kural	Amaç fonksiyon değeri
1	If (Inv-nodes < 12-14) ∧ (node-caps ≠ yes) ∧ (deg-malig < 3) ∧ (irradiat = no)	0,518-0,246
2	If (Age < 70-79) ∧ (tumor-size ≥ 20-24) ∧ (inv-nodes < 21-23) ∧ (degalig ≥ 2)	0,426-0,262

Tablo 17. Dermatology veritabanında bulunan kurallar (Mined rules within Dermatology data set)

Sınıf	Kural	Amaç fonksiyon değeri
1	If (Clubbing of the rete ridges ≥ 1) ∧ (elongation of the rete ridges ≥ 1) ∧ (spongiosis < 1)	0,973-0,973
2	If (Koebner phenomenon ≠ 1) ∧ (follicular papules = 0) ∧ (brosis of the papillary dermis < 1) ∧ (thinning of the suprapapillary epidermis < 2) ∧ (munro microabcess < 1) ∧ (saw-tooth appearance of retes < 1)	0,881-0,881
3	If (Elongation of the rete ridges < 1) ∧ (inflammatory monoluclear inflitrate ≥ 1) ∧ (band-like in_ltrate ≥ 2)	1-1
4	If (Itching < 3) ∧ (polygonal papules < 1) ∧ (knee and elbow involvement < 1) ∧ (PNL in_ltrate < 2) ∧ (elongation of the rete ridges < 1) ∧ (spongiosis ≥ 0) ∧ (band-like in_ltrate < 1)	0,884-0,884
5	If (Fibrosis of the papillary dermiş ≥ 1) ∧ (sawtoothappearence < 2)	1-1
6	If (Itching ≥ 0) ∧ (follicular papules ≠ 0) ∧ (thinning of the suprapapillary epidermis < 2) ∧ (saw-tooth appearance of retes < 3) ∧ (perifollicular parakeratosis ≥ 1)	1-1

4.2. Genelleştirilmiş İlginç Kuralların Keşfi
(Generalized Interesting Rules Mining)

YAKROA, birden fazla sınıflama etiketi bulunan ve tahmin edilecek birden fazla hedef niteliğin olduğu veritabanlarında, sınıflandırma kural madenciliğinin daha genelleştirilmiş bir hali olan, genelleştirilmiş kural çıkarma problemi için de çözüm yöntemi olarak ele alınmıştır. İlginçlik kriteri de eklenerek; algoritmanın keşfedeceği kuralların ilginç, beklenmedik ve sürpriz olması için gerekli düzenlemeler yapılmış ve birden fazla hedef niteliğe sahip olan Zoo veritabanında kural bulma algoritması olarak çalıştırılmıştır.

Kodlama yine bir önceki çalışmaya benzemektedir. Ancak, amaç fonksiyonu çok farklıdır ve Eş. 4'te gösterilmiştir.

$$Amac = \frac{w_1 \times \frac{Ailginclik + Silginclik}{2} + w_2 \times TDogruluk}{w_1 + w_2} \quad (4)$$

w_1 ve w_2, kullanıcı tanımlı ağırlıklar olup sırasıyla 1 ve 2 olarak seçilmiştir. Eş. 5'te gösterilen *Ailginclik* kuralın ata kısmının ilginçliğini, Eş. 9'da gösterilen *Silginclik* ise sonuç kısmının ilginçliğini temsil etmektedir [38].

n ata kısımdaki nitelik sayısıdır. ($|dom(G_k)|$) sonuç kısımdaki hedef nitelik G_k'nın olası değer sayısıdır. Eş. 5'teki *InfoGain* Eş. 6'daki hesaplanır. Eş. 6'daki $Info(G_k)$ ve $Info(G_k|A_i)$ sırasıyla Eş. 7 ve Eş. 8'deki gibi hesaplanır.

$$Ailginclik = 1 - \left[\frac{\sum_{i=1}^{n} InfoGain\ (A_i)}{\log_2(|\ dom\ (G_k)\ |)} \right] \quad (5)$$

$$InfoGain\ (A_i) = Info(G_k) - Info(G_k|A_i) \quad (6)$$

$$Info(G_k) = -\sum_{j=1}^{m_k} (\Pr(V_{kj}) \log_2 (\Pr(V_{kj}))) \quad (7)$$

$$Info(G_k|A_i) = \sum_{z=1}^{n_i} \left(\Pr(V_{iz}) \left(-\sum_{j=1}^{m_k} \Pr(V_{kj}|V_{iz}) \log_2 (\Pr(V_{kj}|V_{iz})) \right) \right) \quad (8)$$

m_k G_k'nın olası değer sayısıdır. n_i, A_i niteliğinin olası değer sayısıdır. Pr(X) X'in olasılığını; Pr($X|Y$), Y verildiğinde X'in şartlı olasılığını gösterir [38].

$$Silginclik = (1 - \Pr(G_{kl}))^{1/\beta} \quad (9)$$

Pr(G_k) hedef nitelik G_k'nın bağıl frekansıdır. β *k*ullanıcı tanımlı bir değerdir ve simülasyonda 2 olarak seçilmiştir.

Tahmini doğruluk olarak Eş. 10 kullanılmıştır.

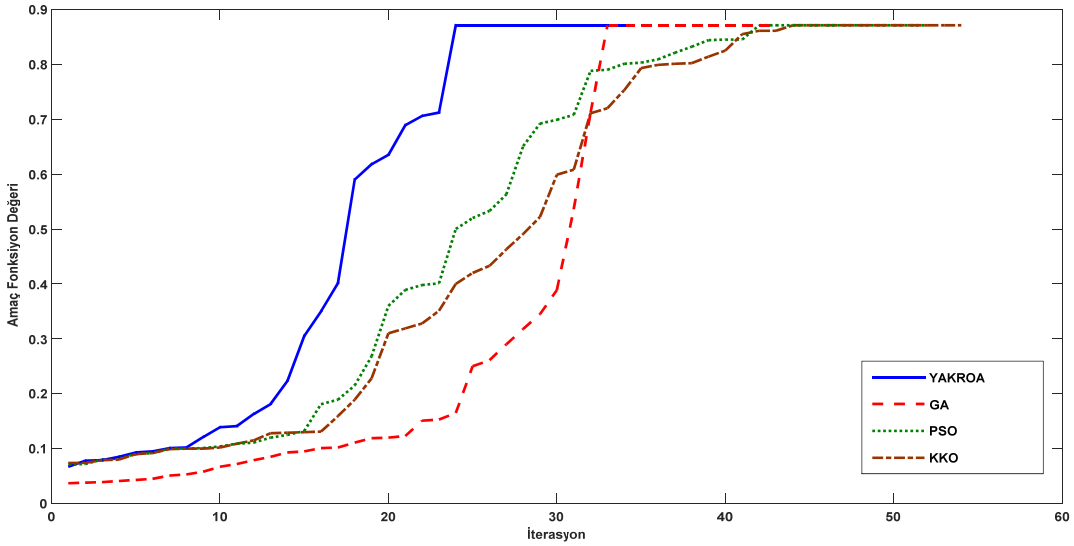
$$TDogruluk = \frac{|A \,\&\, S| - 1/2}{|A|} \quad (10)$$

$|A\&S|$ kuralın hem ata hem de sonuç kısmını karşılayan kayıt sayısıdır. $|A|$ ise sadece kuralın ata kısmındaki şartları sağlayan kayıtların sayısıdır. Veritabanı olarak Zoo seçilmiştir çünkü birden fazla potansiyel hedef nitelik içermektedir. 48 başlangıç aday çözümü amaç fonksiyon değeri 10 kez değişmeyinceye kadar YAKROA, GA, PSO ve KKO ile düzenlenmiştir. Şekil 11, Şekil 12, Şekil 13 ve Şekil 14 elde edilen sonuçları karşılaştırmalı olarak göstermektedir.

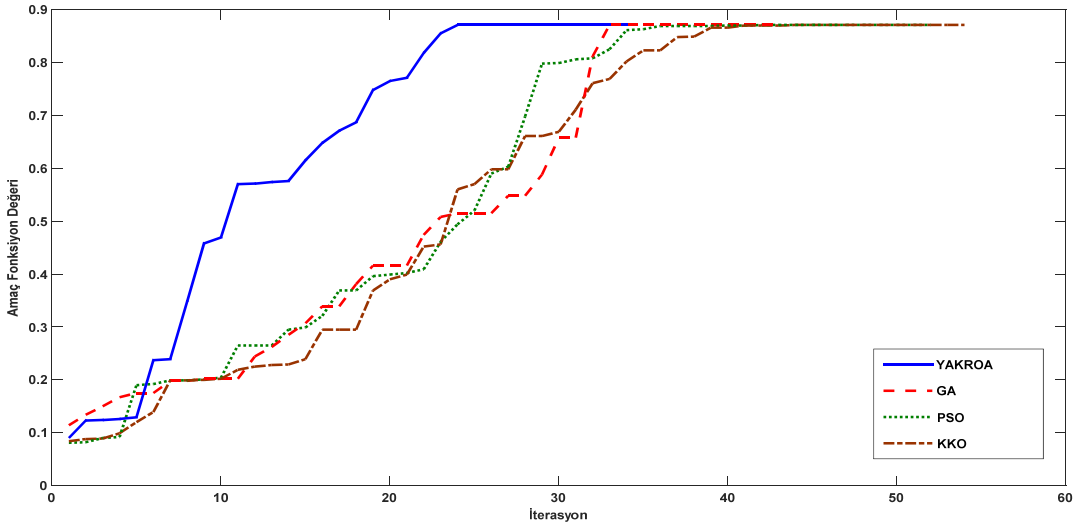
YAKROA, GA, PSO ve KKO ile elde edilen kurallar ve ilgili değerler sırasıyla Tablo 18, Tablo 19, Tablo 20 ve Tablo 21'de verilmiştir. Sonuçlardan; YAKROA'nın henüz yeni önerilmesine ve üzerinde fazla değişiklik ve eklenti yapılmamasına rağmen, diğer algoritmalardan daha iyi sonuçlar verdiği görülmektedir. Hem doğruluk oranı hem de ilginçlik ölçütü göz önüne alındığında PSO ve KKO'dan daha iyi sonuçlar vermektedir.

5. SONUÇLAR VE TARTIŞMALAR
(RESULTS AND DISCUSSIONS)

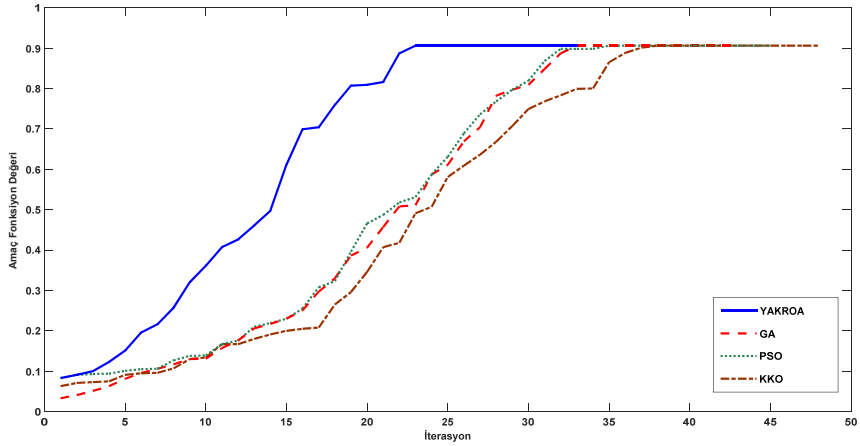
Bu çalışmada, sınıflandırma kural madenciliği bir optimizasyon problemi olarak ele alınmış ve halka açık gerçek veritabanlarından doğru, anlaşılabilir ve ilginç kural listesi kimyasal reaksiyonların tipleri ve oluş şekilleri göz önüne alınarak literatürdeki optimizasyon algoritmalarından daha gürbüz ve daha az parametreli yeni bir hesapsal yöntem olan Yapay Kimyasal Reaksiyon Algoritması (YAKROA) ile ilk kez elde edilmeye çalışılmıştır.



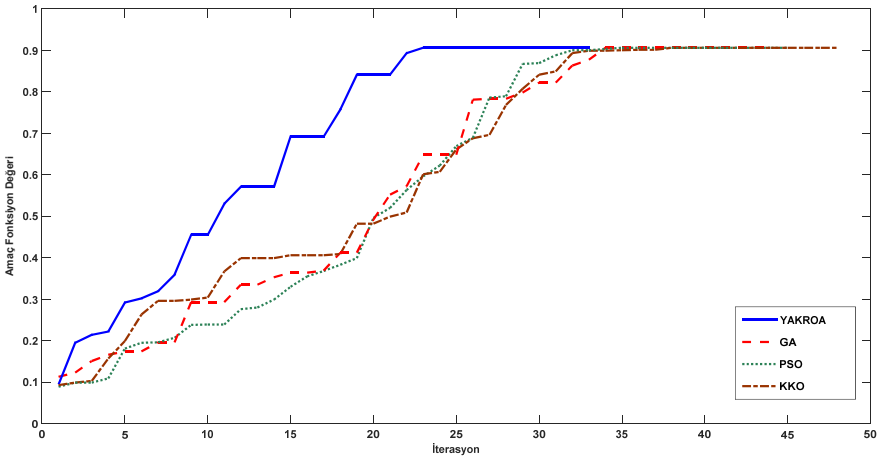
Şekil 11. Birinci kural için ortalama amaç fonksiyon değerleri (Mean objective function values for the first rule)



Şekil 12. Birinci kural için en iyi amaç fonksiyon değerleri (Best objective function values for the first rule)



Şekil 13. İkinci kural için ortalama amaç fonksiyon değerleri (Mean objective function values for the second rule)



Şekil 14. İkinci kural için en iyi amaç fonksiyon değerleri (Best objective function values for the second rule)

Tablo 18. Veri tabanından ACROA ile elde edilen sonuçlar (Obtained results with ACROA from data set)

<Hedef, Değer>	Kural	Ailginclik	Silginclik	Tahmini Doğruluk
<predator, false>	If (hair = 0) ∧ (eggs = 1) ∧ (milk = 0) ∧ (backbone = 1) ∧ (tail = 1) ∧ (domestic = 1)	0,984	0,745	0,875
<predator, true>	If (airborne = 0) ∧ (aquatic = 1) ∧ (backbone = 1) ∧ (catsize = 1)	0,952	0,667	0,955
<domestic, false>	If (eggs = 1) ∧ (airborne = 0) ∧ (predator = 1) ∧ (venomous = 0)	0,958	0,359	0,977
<domestic, true>	If (eggs = 0) ∧ (aquatic = 0) ∧ (tail = 0)	0,959	0,933	0,500
<type, 1>	If (eggs = 0) ∧ (venomous = 0) ∧ (domestic = 0)	0,950	0,770	0,984
<type, 2>	If (feathers = 1) ∧ (venomous = 0) ∧ (domestic = 0)	0,955	0,896	0,971
<type, 3>	If (eggs = 1) ∧ (aquatic = 0) ∧ (predator = 1) ∧ (toothed = 1) ∧ (fins = 0) ∧ (domestic = 0) ∧ (catsize = 0)	0,936	0,975	0,833
<type, 4>	If (aquatic = 1) ∧ (breathes = 0) ∧ (venomous = 0) ∧ (tail = 1)	0,939	0,973	0,958
<type, 5>	If (airborne = 0) ∧ (aquatic = 1) ∧ (toothed = 1) ∧ (breathes = 1)∧ (catsize=0)	0,921	0,980	0,875
<type, 6>	If (airborne = 1) ∧ (fins = 0) ∧ (tail = 0)	0,929	0,960	0,917
<type, 7>	If (predator = 1) ∧ (breathes = 0) ∧ (tail = 0) ∧ (domestic = 0)	0,953	0,949	0,923

Tablo 19. Veri tabanından GA ile elde edilen sonuçlar (Obtained results with PSO from data set)

<Hedef, Değer>	Kural	Ailginclik	Silginclik	Tahmini Doğruluk
<predator, false>	If (domestic = 1) ∧ (catsize = 0)	0,956	0,745	0,929
<predator, true>	If (airborne = 0) ∧ (aquatic = 1) ∧ (backbone = 1) ∧ (catsize = 1)	0,952	0,667	0,955
<domestic, false>	If (eggs = 1) ∧ (airborne = 0) ∧ (predator = 1) ∧ (venomous = 0)	0,958	0,359	0,977
<domestic, true>	If (eggs = 0) ∧ (venomous = 0) ∧ (tail = 0) ∧ (catsize = 0) ∧ (tail = 0)	0,996	0,933	0,500
<type, 1>	If (milk = 1) ∧ (venomous = 0)	0,880	0,770	0,988
<type, 2>	If (feathers = 1) ∧ (venomous = 0)	0,893	0,896	0,975
<type, 3>	If (hair = 0) ∧ (aquatic = 0) ∧ (predator = 1) ∧ (toothed = 1) ∧ (domestic = 0)	0,849	0,975	0,833
<type, 4>	If (eggs = 1) ∧ (fins = 1) ∧ (domestic = 0)	0,887	0,973	0,962
<type, 5>	If (feathers = 0) ∧ (airborne = 0) ∧ (aquatic = 1) ∧ (breathes = 1) ∧ (catsize = 0)	0,883	0,980	0,875
<type, 6>	If (airborne = 1) ∧ (fins = 0) ∧ (tail = 0)	0,856	0,960	0,833
<type, 7>	If (airborne = 0) ∧ (predator = 1)∧ (backbone = 0) ∧ (domestic = 0)	0,887	0,949	0,937

Tablo 20. Veri tabanından PSO ile elde edilen sonuçlar (Obtained results with PSO from data set)

<Hedef, Değer>	Kural	Ailginclik	Silginclik	Tahmini Doğruluk
<predator, false>	If (domestic = 1) ∧ (catsize = 0)	0,956	0,745	0,929
<predator, true>	If (airborne = 0) ∧ (aquatic = 1) ∧ (backbone = 1) ∧ (catsize = 1)	0,952	0,667	0,955
<domestic, false>	If (eggs = 1) ∧ (airborne = 0) ∧ (predator = 1) ∧ (venomous = 0)	0,958	0,359	0,977
<domestic, true>	If (eggs = 0) ∧ (venomous = 0) ∧ (tail = 0) ∧ (catsize = 0) ∧ (tail = 0)	0,996	0,933	0,500
<type, 1>	If (milk = 1) ∧ (venomous = 0)	0,880	0,770	0,988
<type, 2>	If (feathers = 1) ∧ (venomous = 0)	0,893	0,896	0,975
<type, 3>	If (hair = 0) ∧ (aquatic = 0) ∧ (predator = 1) ∧ (toothed = 1) ∧ (domestic = 0)	0,849	0,975	0,833
<type, 4>	If (eggs = 1) ∧ (fins = 1) ∧ (domestic = 0)	0,887	0,973	0,962
<type, 5>	If (feathers = 0) ∧ (airborne = 0) ∧ (aquatic = 1) ∧ (breathes = 1) ∧ (catsize = 0)	0,883	0,980	0,875
<type, 6>	If (airborne = 1) ∧ (fins = 0) ∧ (tail = 0)	0,856	0,960	0,833
<type, 7>	If (airborne = 0) ∧ (predator = 1) ∧ (backbone = 0) ∧ (domestic = 0)	0,887	0,949	0,937

YAKROA birden fazla sınıflama etiketi bulunan ve tahmin edilecek birden fazla hedef niteliğin olduğu veritabanlarında, sınıflandırma kural madenciliğinin daha genelleştirilmiş bir hali olan, daha kompleks ve üzerinde fazla çalışılmayan, genelleştirilmiş kural bulma problemi için de çözüm yöntemi olarak ele alınmıştır. İlginçlik kriteri de eklenerek; algoritmanın keşfedeceği kuralların ilginç, beklenmedik ve sürpriz olması için gerekli düzenlemeler yapılmış ve birden fazla hedef niteliğe sahip olma potansiyeli bulunan veritabanlarında ilk kez kural bulma algoritması olarak çalıştırılmıştır. Bu yöntemde, yoğun işlem gerektiren budama işlemine gerek duyulmamakta ve bu iş kural keşif aşamasında direkt halledilmektedir. Ayrıca bu yöntem 'ayır-ve-yönet' stratejini kullanmaz. Bu şekilde diğer önerilen yöntemlerde ortaya çıkabilecek kurallar arasında beklenmedik etkileşimler ortadan kalkmıştır. Bu etkileşimler, bir örnek farklı sınıfların birkaç kuralı tarafından kapsandığı zaman ortaya çıkabilir. Diğer önerilen algoritmalarda parçalanma problemi ortaya çıkabilir. Kapsama algoritmaları bir kural üretildiğinde tüm eğitim verisindeki kapsanan örnekleri çıkarır ve iterasyonlardan sonra eğitim örneklerinin sayısını azaltır ve lokal olarak önemli ancak global olarak önemsiz kuralların üretilmesine yol açar.

YAKROA ile, bu global olarak önemli kurallar bulunmuştur. YAKROA ilgilenilen her farklı tip problem için deneme yanılma ya da farklı deneylerle algoritma için uygun parametre değerlerinin belirlenmesi işlemi ile uğraşılmaz. Bu da geliştirilen yönteme etkinlik ve hız kazandırmaktadır. Kolayca kullanılabilir ve değiştirilebilirlik, etkili global ve lokal optimizasyon

Tablo 21. Veri tabanından KKO ile elde edilen sonuçlar (Obtained results with ACO from data set)

<Hedef, Değer>	Kural	Ailginclik	Silginclik	Tahmini Doğruluk
<predator, false>	If (domestic = 1) ∧ (catsize = 0)	0,956	0,745	0,929
<predator, true>	If (airborne = 0) ∧ (aquatic = 1) ∧ (catsize = 1)	0,983	0,667	0,958
<domestic, false>	If (hair = 0) ∧ (predator = 1)	0,990	0,359	0,986
<domestic, true>	If (eggs = 0) ∧ (venomous = 0) ∧ (tail = 0) ∧ (catsize = 0) ∧ (tail = 0)	0,996	0,933	0,500
<type, 1>	If (eggs = 0) ∧ (venomous = 0) ∧ (domestic = 0)	0,670	0,771	0,909
<type, 2>	If (feathers = 1) ∧ (breathes = 1) ∧ (domestic = 0)	0,563	0,896	0,933
<type, 3>	If (eggs = 1) ∧ (predator = 1) ∧ (toothed = 1) ∧ (catsize = 0)	0,563	0,975	1,000
<type, 4>	If (aquatic = 1) ∧ (breathes = 0) ∧ (tail = 1)	0,823	0,973	0,800
<type, 5>	If (airborne = 0) ∧ (aquatic = 1) ∧ (toothed = 1) ∧ (breathes = 1) ∧ (catsize = 0)	0,810	0,980	1,000
<type, 6>	If (airborne = 1) ∧ (fins = 0) ∧ (tail = 0)	0,856	0,960	0,833
<type, 7>	If (predator = 1) ∧ (breathes = 0) ∧ (tail = 0) ∧ (domestic = 0)	0,911	0,950	0,875

kabiliyeti, farklı tipte karar değişkenleri ve sınırlayıcılar altında etkili genel bir çözüm stratejisi sunma, karmaşık matematiksel modellere ihtiyaç duymama, çözüm için ekstra fonksiyon ya da bağıntıya ihtiyaç duymama, çok amaçlı optimizasyon problemleri için uygun bir doğaya sahip olma gibi özelliklerinden dolayı güçlü bir yöntem olarak görülmektedir. Kullanılan kimyasal operatörlerdeki kaotik yapı da, algoritmanın lokal çözüme takılıp kalmasını engelleyen bir özellik olarak görülmektedir.

Nitelik etkileşme probleminin bulunmaması, yoğun işlem gerektiren budama işlemine gerek duymaması, esnek amaç fonksiyonu sayesinde farklı amaçları karşılayan kural kümelerinin etkili şekilde bulunabilmesi gibi avantajlarla, YAKROA'nın kompleks veri madenciliği problemleri için etkili bir çözüm olduğu gösterilmiştir. Keşfedilen kuralların hem doğru, hem anlaşılabilir, hem de ilginç ve sürpriz olması için YAKROA'da gerekli düzenlemeler yapılmış ve umut verici sonuçlar elde edilmiştir. Çok amaçlı çözüm arama ve optimizasyon problemleri için de etkili ve alternatif bir hesapsal yöntem olarak görülen YAKROA farklı veri madenciliği problemleri; özellikle birliktelik kurallarının keşfi, kümeleme kurallarının keşfi, ardışık örüntü keşfi vb. için de etkili bir çözüm algoritması olarak kullanılabilir. Algoritmanın parametresi ayarlanmış paralel ve dağıtık versiyonlarıyla daha etkili çözümler vermesi beklenmektedir

TEŞEKKÜR (ACKNOWLEDGEMENT)

Bu makale TÜBİTAK 110M309 No'lu proje kapsamında desteklenip yürütülen bilimsel çalışmalar sonucunda gerçekleştirilmiştir. Yazarlar, sağladığı destekler için TÜBİTAK'a teşekkür eder.

REFERANSLAR (REFERENCES)

1. Han J., Kamber M., Pei J., Data Mining: Concepts and Techniques, The Morgan Kaufmann Series in Data Management Systems, 3, 744, 2011
2. Setiono R., Azcarraga A., Hayashi Y., Using Sample Selection to Improve Accuracy and Simplicity of Rules Extracted from Neural Networks for Credit Scoring Applications, International Journal of Computational Intelligence and Applications, 14 (4), 1550021, 2015.
3. Quinlan J.R., C4.5: Programs for Machine Learning, Morgan Kaufmann, 1992.
4. Breiman L., Friedman J.H., Olshen R.A., Stone C. J., Classification and Regression Trees, Wadsworth, 1984.
5. Berzal F., Cubero J.C., Sánchez D., Serrano J.M., Art: A Hybrid Classification Model, Machine Learning, 54 (1), 67-92, 2004.
6. Dai Q., Zhang C. Wu H., Research of Decision Tree Classification Algorithm in Data Mining, International Journal of Database Theory and Application, 9 (5), 1-8, 2016.
7. Alatas B., Akin E., Sınıflandırma Kurallarının Karınca Koloni Algoritmasıyla Keşfi, ASYU Akıllı Sistemlerde Yenilikler ve Uygulamaları Sempozyumu, Istanbul, 62-66, 2006.
8. Aggarwal C.C., Data Classification: Algorithms and Applications, CRC Press, 2014.
9. Sharma P., Discovery of Classification Rules Using Distributed Genetic Algorithm, Procedia Computer Science, 46, 276-284, 2015.
10. Panda M., Abraham A., Hybrid Evolutionary Algorithms for Classification Data Mining, Neural Computing and Applications, 26 (3), 507-523, 2015.

11. Gundogan K.K., Alatas B., Karci A., Mining Classification Rules by Using Genetic Algorithms with Non-Random Initial Population and Uniform Operator, Turkish Journal of Electrical Engineering & Computer Sciences, 12 (1), 43-52, 2004.

12. Yang L., Li K., Zhang W., Ke Z., Ant Colony Classification Mining Algorithm Based on Pheromone Attraction and Exclusion, Soft Computing, 1-13, 2016.

13. Liang Z., Sun J., Lin Q., Du Z., Chen J., Ming Z., A Novel Multiple Rule Sets Data Classification Algorithm Based on Ant Colony Algorithm, Applied Soft Computing, 38, 1000-1011, 2016.

14. Asadi S., Shahrabi J., ACORI: A Novel ACO Algorithm for Rule Induction, Knowledge-Based Systems, 97, 175-187, 2016.

15. Alatas B., Akin E., Multi-Objective Rule Mining Using a Chaotic Particle Swarm Optimization Algorithm, Knowledge-Based Systems, 22 (6), 455-460, 2009.

16. Tapkan P., Özbakır L., Kulluk S., Baykasoğlu, A., A Cost-Sensitive Classification Algorithm: BEE-Miner, Knowledge-Based Systems, 95, 99-113, 2016.

17. Celik M., Karaboga D., Koylu F., Artificial Bee Colony Data Miner (ABC-Miner), IEEE International Symposium on Innovations in Intelligent Systems and Applications (INISTA), 96-100, 2011.

18. Al-Sheshtawi K.A., Abdul-Kader H.M., Elsisi A.B., A Novel Artificial Immune Clonal Selection Classification and Rule Mining with Swarm Learning Model, Connection Science, 25 (2-3), 75-127, 2013.

19. Köklü M., Kahramanlı H., Allahverdi, N., A New Accurate and Efficient Approach to Extract Classification Rules, Journal of the Faculty of Engineering and Architecture of Gazi University, 29 (3), 477-486, 2014.

20. Akyol S., Alatas, B, Automatic mining of accurate and comprehensible numerical classification rules with cat swarm optimization algorithm, Journal of the Faculty of Engineering and Architecture of Gazi University, 31 (4), 839-857, 2016.

21. Alatas B., Akin E., Mining Fuzzy Classification Rules Using an Artificial Immune System with Boosting, Advances in Databases and Information Systems, 283-293, Springer Berlin Heidelberg, 2005.

22. Alatas B., Akin E., FCACO: Fuzzy Classification Rules Mining Algorithm with Ant Colony Optimization, Advances in Natural Computation, 787-797, Springer Berlin Heidelberg, 2005.

23. Pourpanah F., Lim C.P., Saleh J.M., A Hybrid Model of Fuzzy ARTMAP and Genetic Algorithm for Data Classification and Rule Extraction, Expert Systems with Applications, 49, 74-85, 2016.

24. Kar A.K., Bio Inspired Computing–A Review of Algorithms and Scope of Applications, Expert Systems with Applications, 59, 20-32, 2016.

25. Akyol S., Alatas B., Plant Intelligence Based Metaheuristic Optimization Algorithms, Artificial Intelligence Review, 1-46, 2016.

26. Blum C., Raidl G. R., Further Hybrids and Conclusions, Hybrid Metaheuristics, 127-136, Springer International Publishing, 2016.

27. Ozbay F.A., Alatas B., Review of Musics based Computational Intelligence Algorithms, 1st International Conference on Engeneering Technology and Applied Sciences, Afyon Kocatepe University, 663-669, 2016.

28. Bingol H., Alatas B., Chaotic League Championship Algorithms, Arabian Journal for Science and Engineering, 41 (12), 1-25, 2016.

29. Alatas B, ACROA: Artificial Chemical Reaction Optimization Algorithm for Global Optimization, Expert Systems with Applications, 38 (10), 13170-13180, 2011.

30. Alatas B., A Novel Chemistry Based Metaheuristic Optimization Method for Mining of Classification Rules**,** Expert Systems with Applications, 39 (12), 11080-11088, 2012.

31. Alatas B., Karci A., Genetik Algoritmalarda Düzenli Popülasyon ve Düzenli Operatör, Afyon Kocatepe Üniversitesi Fen Bilimleri Dergisi, 3 (1-2), 11-26, 2003.

32. Demir M., Karci A., Veri Kümelemede Fidan Gelişim Algoritmasının Kullanılması, 12. Elektrik, Elektronik, Bilgisayar, Biyomedikal Mühendisliği Ulusal Kongresi ve Fuarı, Eskisehir, 14-18 2007.

33. Karci A., Alatas A., Akin E., Fidan Gelişim Algoritması, ASYU Akıllı Sistemlerde Yenilikler ve Uygulamaları Sempozyumu, Istanbul, 57-61, 2006.

34. Karci A., Theory of Saplings Growing up Algorithm, Lecture Notes in Computer Science, 4431, 450-460, 2007.

35. Karci A., Alatas B., Thinking Capability of Saplings Growing Up Algorithm, Lecture Notes in Computer Science, 4224, 386-393, 2006.

36. Eberbach E., The Role of Completeness in Convergence of Evolutionary Algorithms, IEEE Congress on Evolutionary Computation, 2, 1706-1713, 2005.

37. Lichman M., UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science, Erişim tarihi 1 Temmuz 2014.

38. Freitas A.A., On Objective Measures of Rule Surprisingness, 2nd European Symposium on Principles of Data Mining and Knowledge Discovery (PKDD-98), Lecture Notes in Artificial Intelligence, 1510, 1-9, 1998.