DEÜ MÜHENDİSLİK FAKÜLTESİ
MÜHENDİSLİK BİLİMLERİ DERGİSİ
Cilt: 16 Sayı: 48 sh. 49-60 Eylül 2004

NİTELİK SEÇME PROBLEMİ İÇİN DİFERANSİYEL GELİŞİM ALGORİTMASI VE YAPAY ARI KOLONİSİ OPTİMİZASYON TEKNİĞİNİ KULLANAN MELEZ YÖNTEM

(HYBRID METHOD USING DIFFERENTIAL EVOLUTION ALGORITHM AND ARTIFICIAL BEE COLONY OPTIMIZATION TECHNIQUE FOR FEATURE SELECTION PROBLEM)

Ezgi ZORARPACI[1], Selma Ayşe ÖZEL[2]

ÖZET/*ABSTRACT*

Nitelik seçme işlemi ile özellik uzayı optimum şekilde daraltılarak veri kümesini en iyi şekilde temsil edebilecek niteliklerin bulunması amaçlanır. Bu çalışma sınıflandırma işlemleri üzerinde nitelik seçme problemi için Yapay Arı Kolonisi optimizasyon tekniği ve Diferansiyel Gelişim algoritmasını birleştirerek yeni bir melez yöntem önermektedir. Önerilen algoritma UCI veri kümeleri üzerinde karar ağacı sınıflandırıcısı (J48) kullanılarak test edilmiştir. Deneysel sonuçlar yeni melez yöntemin sınıflandırma işleminin doğruluğunu düşürmeden ya da en az seviyede düşürerek nitelik sayısını azalttığını ve dolayısıyla yeni örneklerin sınıflandırılması için gereken sürenin de azaldığını göstermiştir.

Aim of the feature selection process is to find the best features which can represent the dataset by narrowing the feature space optimally. This study proposes a new hybrid method which combines Artificial Bee Colony and Differential Evolution algorithms for feature selection problem of classification tasks. The proposed algorithm was tested using decision tree classifier (J48) on UCI datasets. The experimental results show that the new hybrid method reduces the number of features by not decreasing or least decreasing the classification performance and therefore the time which it takes for classification of new instances decreases as well.

ANAHTAR KELİMELER/*KEYWORDS*

Nitelik seçme, Hibrit optimizasyon, Yapay arı kolonisi, Diferansiyel gelişim
Feature selection, Hybrid optimization, Artificial bee colony, Differential evolution

[1] Çukurova Üniversitesi, Bilgisayar Mühendisliği Bölümü, ADANA, ezorarpaci@gmail.com
[2] Çukurova Üniversitesi, Bilgisayar Mühendisliği Bölümü, ADANA, saozel@cu.edu.tr

1. GİRİŞ

Nitelik boyutu küçültme olarak da bilinen nitelik seçimi, sınıflandırıcı modelin öğrenme aşaması için orijinal veri kümesindeki nitelikleri en iyi şekilde temsil edebilecek ilgili niteliklerin optimum bir alt kümesinin oluşturulması işlemidir. Nitelik seçme teknikleri sayesinde öğrenme ve test süresinin kısalması, öğrenici modelin ezberlenmesinin önlenerek ya da azaltılarak daha gelişmiş bir genelleme yapılması ve ilgisiz özelliklerin elenmesi ile daha yorumlanabilir bir veri kümesinin elde edilmesi mümkün olmaktadır.

Teorik olarak bir nitelik seçme yönteminin tüm nitelik alt kümelerini değerlendirerek bunlar arasından en iyi alt kümeyi seçmesi gerekir. Ancak bu işlem genellikle maliyetlidir ve kısıtlayıcı nedenlerden dolayı da uygulanması pek mümkün değildir. Bu nedenle optimum nitelik alt kümesi yerine sınıflandırma performansını düşürmeyen ya da en az şekilde düşüren (alt)optimum bir nitelik alt kümesinin bulunması kabul edilebilmektedir. Meta-sezgisel ve rastgele arama yöntemleri bu (alt)optimum nitelik kümelerini bulmak için kullanılabilir. Bu nedenle nitelik seçme problemi için Tabu Arama (TA), Tavlama Benzetimi (TB), Genetik Algoritma (GA), Parçacık Sürü Optimizasyonu (PSO), Karınca Kolonisi Optimizasyonu (KKO), Diferansiyel Gelişim (DG) ve Yapay Arı Kolonisi (YAK) meta-sezgisellerini de içeren birçok yöntem geliştirilmiştir (Yang ve Honavar, 1998; Chen vd., 2010; Prasartvit vd., 2013; Palanisamy ve Kanmani, 2012; Schiezaro ve Pedrini, 2013; Khushaba vd., 2011).

Meta-sezgisel yöntemler global ve lokal arama işlemlerini dengede tutabilmek için farklı stratejiler kullanırlar. Ancak meta-sezgisel yöntemlerde çoğunlukla ya global arama ya da lokal arama işlemi daha baskın olmaktadır. Bu nedenle iki ya da daha fazla algoritmanın güçlü özelliklerinin birleştirilmesiyle oluşan melez yöntemler kullanılarak daha iyi sonuçlara ulaşılabilmektedir.

Diferansiyel Gelişim (DG) algoritması Storn ve Price tarafından önerilmiş, iterasyonlar kullanarak belli bir uygunluk fonksiyonu ile çözümleri geliştiren bir meta-sezgisel yöntemdir (Storn ve Price, 1997). DG çok boyutlu reel değerli optimizasyon problemleri için kullanılmaktadır. DG'nin diğer optimizasyon tekniklerine göre hızlı çalışma, büyük boyutlardaki karmaşık problemlere uygulanabilme ve az sayıda kontrol parametresine ihtiyaç duyulması gibi avantajları mevcuttur. Bu avantajların yanında kararsız yakınsama ve lokal optimuma takılma gibi dezavantajları da bulunmaktadır.

Yapay Arı Kolonisi (YAK) Karaboğa tarafından önerilen bal arılarının beslenme davranışlarını taklit eden bir optimizasyon tekniğidir (Karaboğa ve Baştürk, 2008). YAK güçlü, uygulanması kolay bir meta-sezgiseldir ve ayrıca çok iyi lokal arama özelliğine sahiptir. Ancak bu lokal arama süreci ile algoritmanın yakınsama süresi bazı problemler için oldukça uzamakta ve bu gibi durumlarda algoritma gerçek performansını gösterememektedir.

Bu çalışma DG' nin global arama stratejisi ile YAK'ın güçlü lokal arama işlemini birleştirerek standart DG ve YAK' dan daha iyi bir performans elde etmeyi amaçlamaktadır.

2. ÖNCEKİ ÇALIŞMALAR

Sınıflandırma işlemleri için önemli problemlerden birisi büyük boyutlu nitelik uzayıdır. Bu durumda sadece ilgili nitelikleri sınıflandırma işlemine dahil etmek daha az ilgili ya da ilgisiz nitelikleri elemek gerekmektedir. Literatüre bakıldığında bu seçim işlemini gerçekleştirmek için nitelikleri puanlandırarak değerlendiren sıralamaya dayalı arama yöntemleri ile DG ve YAK gibi meta-sezgiselleri de içeren bir çok yöntem geliştirilmiştir (Yang ve Pedersen, 1997).

Khushaba ve arkadaşları DG tabanlı bir nitelik seçme algoritması geliştirmiştir (Khushaba vd., 2011). Bu çalışmada önceden belirlenen nitelik sayısı için DG operatörleri niteliklerin indis değerlerine uygulanarak hangi niteliklerin seçileceği belirlenmiştir. Geliştirilen yöntem 3 ve 70

arasında değişen nitelik sayıları için GA ve PSO ile karşılaştırılmıştır. EEG sinyal sınıflandırma veri kümesi kullanılarak test edilen çalışmada önerilen yöntem ile %92,4 kesinlik değeri elde edilirken GA ve PSO ile % 89,9 ve % 88,64 kesinlik değerleri elde edilmiştir (Khushaba vd., 2011).

Prasartvit ve arkadaşları nitelik seçimi için en yakın komşu (EYK) sınıflandırıcısını YAK içinde kullanarak nitelik seçme problemi için yeni bir yöntem geliştirmiştir. Önerilen yöntem gen ifade analizi ve otistik davranışlar alanlarındaki veri kümeleri üzerinde test edilmiştir. Otistik davranışlar veri kümesi için % 85 kesinlik değeri elde edilirken gen ifade analizi alanındaki Colon_Cancer, Acute_Leukemia, Hepatocellular_Carcinoma High-grade_Glioma ve Prostate_Cancer veri kümeleri için % 89,5 ve % 100 arasında değişen kesinlik değerleri elde edilmiştir (Prasartvit vd., 2013).

Palanisamy ve Kanmani uygunluk fonksiyon değerlerini karar ağacı (J48) ile seçtikleri YAK tabanlı yeni bir nitelik seçme yöntemi geliştirmiştir (Palanisamy ve Kanmani, 2012). Geliştirilen yöntem UCI veri kümeleri ile test edilmiş ve % 81,2 ve % 96,9 arasında kesinlik değerleri elde edilmiştir (Palanisamy ve Kanmani, 2012).

Schiezaro ve Pedrini geliştirdikleri YAK tabanlı nitelik seçme yöntemini UCI veri kümeleri üzerinde GA, KKO ve PSO ile karşılaştırmıştır (Schiezaro ve Pedrini, 2013). YAK tabanlı nitelik seçme yöntemi ile % 71,48 ve % 98,46 arasında değişen kesinlik değerleri elde edilirken GA ile % 69,2 ve % 96,6 arasında değişen kesinlik değerleri, KKO ile % 71,03 ve % 96,6 arasında değişen kesinlik değerleri ve PSO ile % 68,7 ile % 96,6 arasında değişen kesinlik değerleri elde edilmiştir (Schiezaro ve Pedrini, 2013).

Lokal arama ve global arama işlemleri arasındaki dengeyi sağlayabilmek amacıyla genellikle iki ya da daha fazla sezgisel yöntem birleştirilerek melez yöntemler geliştirilmiştir. Önceki çalışmalar incelendiğinde DG ve YAK algoritmasının farklı şekillerde birleştirilerek fonksiyon optimizasyonu, yeniden düzenlenebilir anten-dizisi optimizasyonu, portföy optimizasyonu ve biyolojik modeller için parametre tahmini problemleri için kullanıldığı görülmektedir (Alizadegan vd., 2012; Abraham vd., 2012; Zhang vd., 2012; Abdullah vd., 2012; Xu vd., 2013; Li ve Yin, 2012).

Shanthi ve Bhaskaran mamografi resim sınıflandırma veri kümesi için YAK'ın komşu çözüm üretme operatörüne DG'nin mutant vektör üretme formülünü ekleyerek yeni bir nitelik seçme yöntemi önermiştir (Shanti ve Bhaskaran, 2014). Sınıflandırma işlemi Öz-uyarlamalı Kaynak Tahsisi Ağı (Self-adaptive Resource Allocation Network) ile gerçekleştirilmiştir. Önerilen yöntem temel YAK, GA ve PSO ile karşılaştırılmıştır. Melez yöntem, temel YAK, GA ve PSO ile sırasıyla % 96,89, % 96,27, % 96,27, % 95,96 kesinlik değerleri elde edilmiştir (Shanti ve Bhaskaran, 2014).

Yusoff ve arkadaşları DG ve YAK algoritmalarını birleştirerek nitelik seçme problemi için yeni bir melez yöntem geliştirmiştir (Yusoff vd., 2014). Bu birleştirme işlemini gerçekleştirirken YAK'ın kaşif arı ile rastgele bir yiyecek kaynağı bulma süreci yerine terkedilebilecek dört yiyecek kaynağına DG'nin mutasyon ve çaprazlama işlemlerini uygulayarak rastgele bir yiyecek kaynağı yerine eldeki bu 4 mevcut yiyecek kaynağını kullanarak yeni bir yiyecek kaynağı üretme işlemini gerçekleştirmiştir. Geliştirdikleri nitelik seçme tekniğini PSO ve KKO ile yumurtalık kanseri ve biyo-işaretleyici analizi veri kümeleri üzerinde karşılaştırmıştır. Sonuçlar değerlendirildiğinde ise geliştirilen melez yöntem PSO ve KKO'a göre daha iyi performans göstermiştir (Yusoff vd., 2014).

3. YÖNTEM

3.1. Diferansiyel Gelişim Algoritması

DG, Price ve Storn tarafından 1995 yılında geliştirilen ve özellikle sürekli optimizasyon problemlerinin çözümünde kullanılan populasyon temelli sezgisel bir algoritmadır (Storn ve Price, 1997). Algoritmada iteratif olarak, operatörler aracılığıyla ve uygunluk fonksiyon değerleri yardımıyla problem için daha iyi çözümler araştırılmaktadır. Algoritmanın işleyişi için ilk olarak başlangıç popülasyonu yani bireyleri oluşturulmaktadır. Başlangıç popülasyonunun oluşturulmasından sonra belli bir durdurma kriteri sağlanıncaya kadar tüm bireylere sırasıyla mutasyon, çaprazlama ve seçim işlemleri uygulanmaktadır.

3.1.1. Başlangıç Popülasyonunun Oluşturulması

Probleme ait parametre sayısı her bir bireye ait gen (boyut) sayısını göstermektedir. Popülasyondaki birey sayısı mutasyon sırasında mevcut birey haricinde 3 farklı birey (*r1*,*r2*,*r3*) kullanılacağından dolayı her zaman üçten büyük olmalıdır. Popülasyondaki bireyleri oluşturmak için çoğunlukla Eşitlik 1 kullanılmaktadır.

$$X_i^j = X_{min}^j + rand(0,1).(X_{max}^j - X_{min}^j) \quad (1)$$

Burada X_i^j popülasyondaki *i.* bireyin *j.* parametresini, X_{min}^j ve X_{max}^j, *j.* parametre için en küçük ve en büyük sınır değerleri, $rand(0,1)$ ise 0 ve 1 arasında üretilmiş rastgele bir reel değeri ifade etmektedir.

3.1.2. Mutasyon

Mutasyon, mevcut birey kromozomun bir kısım genleri üzerinde, rastgele belirlenmiş miktarlarda değişiklikler yapmaktır. Diferansiyel gelişim algoritmasında mutasyon işlemi sonunda mutant birey oluşturulmaktadır. Bir birey (X_i) için mutant birey (V_i) oluşturlurken bu bireyden farklı 3 birey (X_{r1}, X_{r2}, X_{r3}) seçilmektedir. Bu 3 bireyden ikisinin farkı mutasyon faktörü (F) oranında üçüncü bireyin genlerine etki etmektedir. Mutant birey oluşturulurken kullanılan formül Eşitlik 2'de gösterilmektedir.

$$V_i^j = X_{r3}^j + F.(X_{r1}^j - X_{r2}^j) \quad (2)$$

3.1.3. Çaprazlama

Mutasyon sonucu elde edilen mutant birey (V_i) ve mevcut birey (X_i) arasında çaprazlama işlemi uygulanarak aday birey (U_i) oluşturulmaktadır. Bunun için aday bireye ait her bir gen çaprazlama oranı (ÇR) olasılıkla mutant bireyden 1-ÇR olasılıkla mevcut bireyden alınmaktadır. Bu işlemde 0 ile 1 arasında üretilen rastgele sayı ÇR'den küçükse gen mutant bireyden diğer durumlarda mevcut bireyden seçilmektedir (Keskintürk, 2006). Çaprazlama işleminin matematiksel ifadesi Eşitlik 3'te gösterilmektedir.

$$U_i^j = \begin{cases} V_i^j, & Eğer\ rand(0,1) \leq CR\ veya\ j = jrand \\ X_i^j, & Diğer\ durumlarda \end{cases} \quad (3)$$

Burada $j=jrand$ koşulu en az bir tane genin mutant bireyden alınmasını garanti etmek amacıyla kullanılmaktadır.

3.1.4. Seçim

Seçim işlemi ile mevcut birey ve üretilen yeni aday birey değerlendirilerek hangisinin yeni popülasyonda bulunacağına karar verilmektedir. Uygunluk fonksiyon değeri daha yüksek olan birey bir sonraki popülasyonun bireyi olarak atanmaktadır.

DG algoritmasıyla ilgili temel kavramlar açıklandıktan sonra algoritmanın temel adımları aşağıdaki gibi verilebilir:

Adım1. Başlangıç popülasyonu oluştur
Adım 2. Bireylerin uygunluk değerlerini hesapla
Adım 3. Her bir birey için;
Mutasyon işlemi uygula
Çaprazlama işlemi uygula
Seçim işlemi uygula
Adım 4. Durdurma kriteri sağlanmışsa Adım 5'e diğer durumlarda Adım 3'e git.
Adım 5. Popülasyondaki uygunluk değeri en yüksek olan bireyi döndür.

3.2. Yapay arı Kolonisi Optimizasyon Tekniği

YAK ilk olarak 2005 yılında Derviş Karaboğa tarafından tasarlanan, arı kolonilerinin beslenme davranışlarını taklit eden bir optimizasyon algoritmasıdır. Reel parametreli optimizasyon problemlerinin çözümü için önerilmiştir (Karaboğa ve Baştürk, 2008).

Bal arıları, iş bölümü yapabilme ve kendi kendine organize olabilme kabiliyetine sahiptir. Bir bal arısı kolonisinde arılar üç gruba ayrılmaktadır.

1- *İşçi arılar:* İşçi arılar daha fazla nektarın olduğu komşu yiyecek kaynaklarını araştırırlar. Her yiyecek kaynağında bir işçi arı bulunur ve dolayısıyla yiyecek kaynağı sayısı işçi arı sayısına eşit olmaktadır.
2- *Gözcü arılar:* Gözcü arılar kovanda bekler ve işçi arılardan dansla edinmiş oldukları yiyecek kaynağı bilgilerini değerlendirerek nektarın fazla olduğu yiyecek kaynağına yönelirler.
3- *Kaşif arılar:* Kaşif arılar rasgele olarak etrafı dolaşarak yeni yiyecek kaynakları aramaktadırlar.

Algoritmanın akışında ilk olarak işçi arılar yiyecek kaynaklarına gönderilerek bulunan komşu yiyecek kaynakları için nektar miktarları hesaplanır. Daha sonra işçi arılar tarafından edinilen yiyecek kaynak bilgileri dans aracılığıyla gözcü arılarla paylaşılarak gözcü arılar en iyi yiyecek kaynaklarına yönlendirilir ve bu yiyecek kaynakları etrafında daha iyi nektar miktarına sahip komşu yiyecek kaynaklarını ararlar. Son olarak ise rastgele olarak yeni yiyecek kaynakları bulması için kaşif arılar gönderilir. Durdurma kriteri sağlanana kadar bu 3 işlem tekrar edilir.

3.2.1. Başlangıç Yiyecek Kaynaklarının Belirlenmesi

Başlangıç yiyecek kaynakları problemin her bir parametresinin alt ve üst sınırlarından faydalanılarak rastgele geliştirilmektedir. Bunun için Eşitlik 1 ile ifade edilen denklem kullanılmaktadır.

3.2.2. İşçi Arıların Yiyecek Kaynaklarına Gönderilmesi

Bu işlemde her bir işçi (X_i) arı yeni bir komşu yiyecek kaynağı (U_i) belirleyerek bu kaynağın nektar miktarını (uygunluğu) değerlendirir. Her yeni komşu çözüm kaynağı Eşitlik 4 ile belirlenmektedir.

$$U_i^{jrand} = X_i^{jrand} + rand[-1,1].(X_i^{jrand} - X_k^{jrand}) \quad (4)$$

Eşitlik 4'te X_k mevcut yiyecek kaynakları arasından rastgele seçilen bir yiyecek kaynağını, $jrand$ problemin rastgele seçilen bir parametresini, $rand[-1,1]$ ise -1 ve 1 aralığında rastgele üretilen bir reel değeri temsil etmektedir.

3.2.3. Gözcü Arıların Yiyecek Kaynaklarına Gönderilmesi

Tüm işçi arılar komşu yiyecek kaynağı arama sürecini tamamladıktan sonra edindikleri kaynak bilgilerini "Waggle" dansı yardımıyla kovanda bekleyen gözcü arılar ile paylaşmaktadır. Bir gözcü arı için yiyecek kaynağı nektar miktarı ile orantılı bir olasılıkla seçilmektedir. Bunun için mevcut tüm yiyecek kaynakları için nektar miktarı orantılı olasılık değerleri (uygunluk olasılık değeri) belirlenmektedir. Her bir yiyecek kaynağı (X_i) için uygunluk olasılık değeri Eşitlik 5 kullanılarak hesaplanmaktadır.

$$p_i = \frac{uygunluk_i}{\sum_{n=1}^{SN} uygunluk_n} \quad (5)$$

Burada $uygunluk_i$ *i.* yiyecek kaynağının nektar miktarını (uygunluk), *SN* işçi arı sayısını temsil etmektedir. Bu uygunluk olasılık değeri hesaplaması işlemi ile bir yiyecek kaynağının nektar miktarı (uygunluk) ne kadar fazla ise o kadar çok gözcü arı tarafından seçilecektir.

Tüm yiyecek kaynaklarının uygunluk olasılık değerleri belirlendikten sonra her bir yiyecek kaynağı için [0,1] aralığında rasgele sayılar üretilir ve eğer yiyecek kaynağının uygunluk olasılık değeri bu üretilen değerden büyükse gözcü arı işçi arı gibi davranarak (Eşitlik 4 kullanılarak) yeni bir komşu yiyecek kaynağı araştırılır, yeni komşu yiyecek kaynağının nektar miktarı belirlenir ve mevcut yiyecek kaynağı ile karşılaştırılır. Yeni komşu yiyecek kaynağının uygunluk değeri daha yüksek ise mevcut yiyecek kaynağının yerini yeni komşu yiyecek kaynağı alır.

3.2.4. Kaşif Arı İle Rasgele Yeni Yiyecek Kaynağının Bulunması

Algoritmada yiyecek kaynaklarının nektar miktarlarının tükenip tükenmediği çözüm geliştirememe sayaçları ile kontrol edilmektedir. Bir yiyecek kaynağının çözüm geliştirememe sayacı belli bir sınır değerin üzerindeyse artık bu kaynağın işçi arısı kaşif arıya dönüşür ve eşitlik (1) ile belirtilen denklem kullanılarak rastgele yeni bir yiyecek kaynağı keşfedilir ve mevcut yiyecek kaynağı bırakılır yani mevcut yiyecek kaynağının yerini kaşif arı ile bulunan yeni yiyecek kaynağı almaktadır. Kaynağın tükendiğini belirleyen sınır değer "limit" olarak adlandırılmaktadır ve önemli bir kontrol parametresidir. Temel YAK algoritmasında her bir çevrimde sadece bir tane kaşif arı gönderilmektedir.

YAK algoritmasıyla ilgili temel kavram ve işlemler açıklandıktan sonra algoritmanın temel adımları aşağıdaki gibi verilebilir:

Adım1. Başlangıç yiyecek kaynaklarını rastgele olarak belirle

Adım 2. Yiyecek kaynaklarının nektar miktarlarını (uygunluk) belirle

Adım 3. Her bir işçi arı ile yeni komşu yiyecek kaynağı bul, bu komşu yiyecek kaynağının nektar miktarlarını belirle ve mevcut yiyecek kaynağı ile karşılaştır
Adım 4. Her bir yiyecek kaynağı için nektar miktar olasılığını (uygunluk olasılık değeri) belirle
Adım 5. Nektar miktarı olasılıklarını kullanarak gözcü arılar ile yeni komşu yiyecek kaynaklarını bul ve değerlendir
Adım 6. En iyi yiyecek kaynağını hafızaya al
Adım 7. Limit değeri aşan en büyük çözüm geliştirememe sayaçlı yiyecek kaynağı yerine kaşif arı tarafından rastgele bir yeni yiyecek kaynağı bul
Adım 7. Durdurma kriteri sağlanmışsa Adım 9'a diğer durumlarda Adım 3'e git.
Adım 9. En iyi yiyecek kaynağını döndür.

3.3. Melez Yöntem

Bu çalışmada DG ve YAK algoritmalarının güçlü özeliklerinin birleştirilmesinden oluşan yeni bir melez yöntem önerilmektedir.

DG algoritmasının başarısı kontrol parametrelerinin doğru ayarlanmasıyla çok yakından ilgilidir (Sá vd., 2008). Özellikle ÇR parametresi lokal arama ve global aramayı dengeleme açısından büyük öneme sahiptir. Düşük ÇR değerleri algoritmanın lokal arama özelliğini artırırken, ÇR' in yüksek değerleri ise global arama işlemini desteklemekte olup aynı zamanda da optimuma yakınsama sürecini kısaltmaktadır (Montgomery ve Chen, 2010). DG için bir diğer hayati parametre ise F, mutasyon ölçeği faktörüdür. Bu ölçek genel olarak [0..2] aralığında bir reel sayıdır. Bu ölçeğin yüksek değerleri global arama sürecini destekleyerek lokal optimuma takılma olasılığını azaltmaktadır (Malipeddi vd., 2011).

YAK gözcü arılar aşamasındaki çözüm geliştirme süreciyle lokal arama işlemini çok iyi bir şekilde yönetmektedir. Ancak bu durumda da algoritma için yakınsama süreci oldukça uzamaktadır ve algoritma daha düşük performans göstermektedir (Gao ve Liu, 2011).

Bu çalışma nitelik seçme problemi için yukarıda bahsedilen olumsuz durumları engellemek amacıyla DG algoritmasının global arama stratejisiyle YAK optimizasyon tekniğinin lokal arama özelliğini birleştirerek yeni bir melez çözüm yöntemi önermektedir. Önerilen metodun adımları aşağıdaki gibidir:

Adım 1. Başlangıç popülasyonun oluşturulması: Popülasyondaki her bir birey (X_i) aşağıdaki ifadeye göre oluşturulur.

$$X_i^j = rand(0,1) \tag{6}$$

Burada, X_i^j popülasyondaki *i.* bireyin *j.* parametresini, $rand(0,1)$ ise 0 ve 1 arasında üretilmiş rastgele bir reel değeri ifade etmektedir.

Adım 2. İkili vektörlerin oluşturulması: Popülasyondaki her bir birey vektör için eşitlik (7) kullanılarak bireylerin ikili vektörleri oluşturulur.

$$X\_ikili_i^j = \begin{cases} 1, & Eğer\ X_i^j > rand(0,1) \\ 0, & Diğer\ durumlarda \end{cases} \tag{7}$$

Adım 3. Birey vektörler için uygunluk değerlerinin hesaplanması: Her bir bireyin ikili vektöründe bulunan 0 değerine karşılık gelen indisteki nitelik veri kümesinden silinir. Geriye

kalan nitelikler ile veri sınıflandırılır. Sınıflandırma işleminin F-ölçeği bireyin uygunluk değeri olmaktadır.

Adım 4. Popülasyondaki bireylere mutasyon ve çaprazlama işlemlerinin uygulanması: Bir birey (X_i) için mutant vektör (V_i) oluşturulurken Eşitlik 2 kullanılır. Birey vektör için mutant vektör oluşturulduktan sonra çaprazlama işlemine geçilmektedir. Çaprazlama işleminde mutant vektör (V_i) ve birey vektörden belirtilen oranda genler alınarak aday vektör (U_i) olarak adlandırılan yeni bir birey üretilmektedir. Bu üretim işlemi için Eşitlik 3 kullanılır. Bu adım popülasyondaki tüm birey vektörler için gerçekleştirilir.

Adım 5. Oluşturulan aday vektörler için uygunluk değerlerinin hesaplanması: Bir aday vektör için ikili vektör Eşitlik 7 kullanılarak oluşturulur ve bu vektörde 0 değerini taşıyan nitelikler veri kümesinden silinerek geriye kalan nitelikler sınıflandırıcıya gönderilir, sınıflandırıcıdan dönen F-ölçeği değeri aday vektörün uygunluk değeri olmaktadır.

Adım 6. Seçim işleminin uygulanması: Bu adımda birey (X_i) ve aday (U_i) vektörlerden hangilerinin bir sonraki jenerasyon için popülasyonda bulunacağına karar verilir. Uygunluk değeri daha yüksek olan vektörler alınır.

Adım 7. Popülasyondaki bireyler için uygunluk olasılık değerlerinin bulunması: Popülasyondaki her bir birey vektör (X_i) için uygunluk olasılık değeri (p_i) Eşitlik 5'e göre hesaplanır.

Adım 8. Gözcü arı işleminin uygulanması: Bu adımda uygunluk olasılık değeri 0 ve 1 arasında üretilen sayıdan daha büyük olan bir birey vektörün (X_i) rastgele seçilmiş olan bir parametresi ($jrand$) üzerinde Eşitlik 8'e göre yeni aday vektör (U_i) oluşturulur. Daha sonra bu aday çözüm için Eşitlik 7 kullanılarak ikili vektörü oluşturulur ve bu ikili vektöre göre sınıflandırma yapılarak aday vektörün uygunluk değeri bulunur. Bu vektörün uygunluk değeri birey vektörden daha yüksek ise birey vektör aday vektör ile değiştirilir. Bu adım popülasyondaki birey vektör sayısı kadar tekrarlanır.

$$U_i^{jrand} = en\ iyi\ \text{çözüm}^{jrand} + F.(X_{r1}^{jrand} - X_{r2}^{jrand}) \quad (8)$$

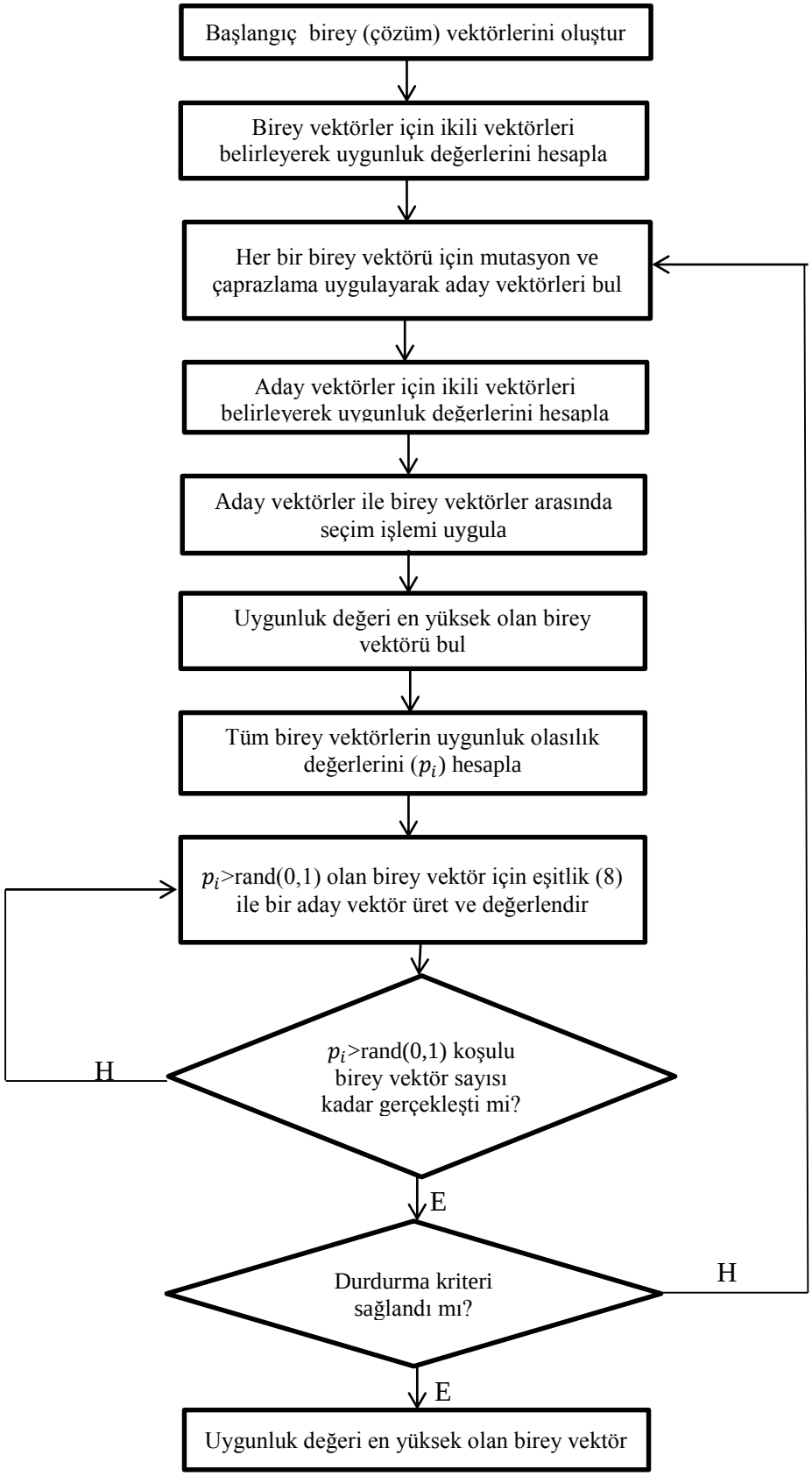
Algoritmanın sonlandırılabilmesi için adım 4-8 önceden belirlenen bir sonlandırma kriteri sağlanana kadar tekrarlanır. Daha iyi anlaşılması açısından önerilen melez yöntemin akış diyagramı Şekil 1'de gösterilmektedir.

Problem çözümünde DG algoritmasının global arama stratejisi kullanıldığından ÇR için [0,9..1] aralığında rastgele üretilen değerler alınmıştır. Böylelikle çözüm uzayında yeni arama alanlarının keşfedilmesi sağlanmıştır. F değeri olarak da DG için önerilen değer olan 0,5 kullanılmıştır.

4. TARTIŞMA

UCI veri kümeleri % 75 eğitim % 25 test kümeleri olarak ayrılmıştır. Veri kümelerindeki örnek ve sınıf sayıları Çizelge 1'de gösterilmektedir.

Eğitim kümeleri kullanılarak, önerilen melez algoritma ile en iyi nitelik kümeleri belirlenmiştir. Elde edilen en iyi nitelik kümeleri, test veri kümeleri üzerinde 10'lu çapraz doğrulama ile sınanmıştır. Algoritmalar her bir veri kümesi için aynı çekirdek değerleri, nitelik sayısı*4 boyutundaki popülasyon büyüklüğü ve 500 iterasyon ile 10 kez çalıştırılmış olup bu çalışmalar sonucunda elde edilen en iyi ve ortalama F-ölçeği değerleri Çizelge 2'de, 10 çalışma sonundaki en iyi çözüm için seçilen nitelik sayıları ve seçilen ortalama nitelik sayıları da Çizelge 3'de gösterilmiştir.



Şekil 1. Geliştirilen melez algoritma için akış diyagramı

Çizelge 1. Veri kümelerinin özellikleri

Veri Kümesi	Örnek Sayısı		Sınıf Sayısı		
	Eğitim	Test	Eğitim	Test	Toplam
Autos	121	38	6	6	6
Dermatology	267	91	6	6	6
Diabetes	576	192	2	2	2
T.Surgery	352	118	2	2	2
Glass	160	54	6	6	6
Heart-c	222	74	2	2	2
Lymph	112	36	4	4	4
Hepatitis	61	19	2	2	2
Vote	175	57	2	2	2
Zoo	76	25	7	7	7

Çizelge 2. Veri kümelerinin özellikleri

Veri Kümesi	En iyi/Ortalama F-ölçeği Değerleri			
	DG	YAK	Melez	Nitelik seçimi uygulanmadan
Autos	0,727/0,587	**0,747**/0,592	**0,747**/**0,624**	0,633
Dermatology	**0,900**/0,819	0,876/0,828	0,880/**0,841**	0,878
Diabetes	0,737/0,662	**0,746**/0,649	**0,746/0,687**	0,667
T.Surgery	**0,765/0,765**	**0,765/0,765**	**0,765/0,765**	0,765
Glass	0,721/0,626	0,713/0,592	**0,757**/**0,682**	0,663
Heart-c	0,756/0,695	**0,824**/**0,748**	0,756/0,708	0,703
Lymph	0,673/0,580	0,673/0,574	**0,684/0,594**	0,609
Hepatitis	**0,944**/0,855	**0,944**/0,817	**0,944/0,887**	0,833
Vote	**0,964**/0,889	**0,964/0,926**	**0,964**/0,921	0,964
Zoo	**0,772**/0,734	**0,772/0,735**	**0,772**/0,729	0,733

Sonuçlar incelendiğinde, F-ölçeği değerleri açısından melez yöntemin toplam 10 veri kümesinin 7'si üzerinde standart DG ve YAK algoritmalarından ortalama olarak daha iyi bir performans sergilediği görülmektedir. Bunun yanı sıra 10 çalışma sonucunda bulunan en iyi F-ölçeği değerleri için ise melez yöntem tüm veri kümeleri üzerinde sınıflandırma performansının artmasına katkı sağlamıştır.

Çalışma 4 GB RAM ve Intel Core i5-2430 M 2.4 GHz işlemci konfigürasyonu üzerinde gerçekleştirilmiştir. Sınıflandırıcı olarak C4.5 karar ağacı sınıflandırıcısının Weka veri madenciliği yazılımındaki karşılığı olan J48 sınıflandırıcısı kullanılmıştır. Bu sınıflandırıcının seçilmesinin nedeni; Saraç ve Özel tarafından yapılan çalışmada, J48 sınıflandırıcısının URL tabanlı Web sayfası sınıflandırma işleminde NaiveBayes, RBF Networks, Voted Perceptron, Threshold Selector ve VFI sınıflandırıcılarına göre daha iyi F-ölçeği değeri ile sınıflandırma yapmış olmasıdır (Saraç ve Özel, 2010). Gelecek çalışmalarda NaiveBayes, EYK, SVM gibi başka sınıflandırıcılarla bu çalışma yapılabilir ve sonuçlar karşılaştırılabilir.

Çizelge 3 değerlendirildiğinde, geliştirilen melez yöntem ile veri kümelerindeki nitelik sayıları gerçekleştirilen 10 çalışma sonunda, ortalama olarak yaklaşık % 50 oranında azaltılmıştır. Nitelik sayılarının azaltılması sayesinde de sınıflandırma işlemleri için gerekli olan toplam sürenin oldukça kısalması sağlanmıştır. Örneğin; "T.Surgery" veri kümesi için tüm nitelikler kullanılarak 10'lu çapraz doğrulama ile sınıflama işlemi 50 milisaniyede yapılırken, nitelik sayısı % 50 oranında azaldığında ise bu işlem toplam 5 milisaniyede gerçekleştirilmiştir.

Çizelge 3. Seçilen nitelik sayıları

Veri Kümesi	En iyi çözüm için/Ortalama Seçilen Özellik Sayısı			Toplam Nitelik Sayısı
	DG	YAK	Melez	
Autos	16/13	**9/12**	11/**12**	25
Dermatology	**16/16**	20/17	17/**16**	34
Diabetes	7/4	6/**3**	**5**/4	8
T.Surgery	**5**/**7**	7/8	**5**/8	16
Glass	4/**4**	6/**4**	**3**/5	9
Heart-c	10/7	**4/5**	7/6	13
Lymph	**6/8**	9/9	8/**8**	18
Hepatitis	7/9	9/9	**6/9**	19
Vote	7/**6**	7/7	**6**/7	16
Zoo	**8**/10	9/9	**8/9**	17

5. SONUÇ

Bu çalışmada, Diferansiyel Gelişim algoritması ve Yapay Arı Kolonisi optimizasyon tekniğini birleştirerek sınıflandırma işlemlerindeki nitelik seçme probleminin çözümü için yeni bir melez yöntem geliştirilmiştir. Geliştirilen yöntem, araştırmacılar tarafından sınıflandırma işlemlerinde sıklıkla kullanılan UCI veri kümeleri üzerinde sınanmıştır. Elde edilen sonuçlar değerlendirildiğinde geliştirilen melez yöntem test veri kümeleri üzerinde standart DG ve YAK algoritmalarından daha yüksek bir sınıflandırma performansı göstermiştir. Çalışmamızda genel olarak nitelik sayısı bakımından küçük ve orta boyutlu veri kümeleri kullanılmıştır. Ancak daha sonraki çalışmalarımızda, nitelik sayıları oldukça fazla olan veri kümeleri üzerinde farklı sınıflandırıcılar kullanılarak, literatürdeki diğer nitelik seçme yöntemleri ile bu çalışmada geliştirilen yöntem arasında bir performans değerlendirmesinin gerçekleştirilmesi planlanmaktadır.

KAYNAKLAR

Abdullah A., Deris S., Anwar S. (2011): "Hybrid Evolutionary Clonal Selection for Parameter Estimation of Biological Model", International Journal of Computer Applications in Engineering Sciences, Cilt 1, No. 3, s.313-319.

Abraham A., Jatoth R.K., Rajasekhar A. (2012): "Hybrid Differential Artificial Bee Colony Algorithm", Journal of Computational and Theoretical Nanoscience, Cilt 9, No. 2, s.249-257.

Alizadegan A., Meybodi M. R., Asady B. (2012): "A Novel Hybrid Artificial Bee Colony Algorithm and Differential Evolution for Unconstrained Optimization Problems", Advances in Computer Science and Engineering, Cilt 8, No. 1, s.45-56.

Chen Y., Miao D., Wang R. (2010): "A Rough Set Approach to Feature Selection Based on Ant Colony Optimization", Pattern Recognition Letters, Elsevier, s.226-233.

Gao W.F., Liu S. (2011): "Improved Artificial Bee Colony Algorithm for Global Optimization", Information Processing Letters, Elsevier, s.871-882.

Karaboğa D., Baştürk B. (2008): "On the Performance of Artificial Bee Colony (ABC) Algorithm", Applied Soft Computing, Elsevier, s.687-697.

Keskintürk T. (2006): "Diferansiyel Gelişim Algoritması", İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi, Cilt 1, s.85-99.

Khushaba R. N., Al-Ani A., Al-Jumaily A. (2011): "Feature Subset Selection Using Differential Evolution and A Statistical Repair Mechanism", Expert Systems with Applications, Elsevier, s.11515-11526.

Li X., Yin M. (2012): "Hybrid Differential Evolution with Artificial Bee Colony and Its Application for Design of A Reconfigurable Antenna Array with Discrete Phase Shifters", IET Microwaves Antennas & Propagation, Cilt 6, No. 14, s.1573–1582.

Mallipeddi R., Suganthan P. N., Pan Q. K., Tasgetiren M. F. (2011): "Differential Evolution Algorithm with Ensemble of Parameters and Mutation Strategies", Applied Soft Computing, Elsevier, s.1679-1696.

Montgomery J., Chen S. (2010): "An Analysis of the Operation of Differential Evolution at High and Low Crossover Rates", Evolutionary Computation (CEC), IEEE Congress, s.1-8.

Palanisamy S., Kanmani S. (2012): "Artificial Bee Colony Approach for Optimizing Feature Selection", International Journal of Computer Science Issues, Cilt 9, No. 3, s.432-438.

Prasartvit T., Banharnsakun A., Kaewkamnerdpong B., Achalakul T. (2013): "Reducing Bioinformatics Data Dimension with ABC-kNN", Neurocomputing, Elsevier, s.367-381.

Sá Â. A., Andrade A. O., Soares A. B. (2008): "Exploration vs. Exploitation in Differential Evolution", AISB Convention Communication, Interaction and Social Intelligence, Cilt 1, s.57-63.

Saraç E., Özel S.A. (2010): "URL Tabanlı Web Sayfası Sınıflandırma", ASYU Sempozyumu, s.13-17.

Schiezaro M.,Pedrini H. (2013): "Data Feature Selection Based on Artificial Bee Colony Algorithm", EURASIP Journal on Image and Video Processing, Springer US, s.1-8.

Shanthi S., Bhaskaran V.M. (2014): "Modified Artificial Bee Colony Based Feature Selection: A New Method in the Application of Mammogram Image Classification", International Journal of Science, Engineering and Technology Research, Cilt 3, No. 6, s.1664-1667.

Storn R., Price K. (1997): "Differential Evolution–A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces", Journal of Global Optimization, Springer US, s.341-359.

Xu Y., Fan P., Yuan L. (2013): "A Simple and Efficient Artificial Bee Colony Algorithm", Mathematical Problems in Engineering, Hindawi, s.1-9.

Yang J., Honavar V. (1998): "Feature Subset Selection Using A Genetic Algorithm", Feature Extraction, Construction and Selection, Springer US, s.117-136.

Yang Y., Pedersen J. O. (1997): "A Comparative Study on Feature Selection in TextCategorization", ICML, Cilt 97, s.412-420.

Yusoff S. A. M., Abdullah R., Venkat I. (2014): "Adapted Bio-inspired Artificial Bee Colony and Differential Evolution for Feature Selection in Biomarker Discovery Analysis", Recent Advances on Soft Computing and Data Mining, Springer International Publishing, s.111-120.

Zhang Y., Wu L., Wang S. (2011): "Magnetic Resonance Brain Image Classification by An Improved Artificial Bee Colony Algorithm", Progress in Electromagnetics Research, Cilt 116, s.65-79.