g-BSAFCM :Yeni Bir Hibrit Kümeleme Algoritması

g-BSAFCM :A New Hybrid Clustering Algorithm

Güliz Toz[1], Pakize Erdoğmuş[1]
[1] Elektrik ve Elektronik ve Bilgisayar Mühendisliği Bölümü, Düzce Üniversitesi, Düzce, Türkiye
glz.toz@gmail.com
[1]Bilgisayar Mühendisliği Bölümü, Düzce Üniversitesi, Düzce Türkiye
pakizeerdogmus@duzce.edu.tr

***Özetçe*—Kümeleme bir veri setindeki verilerin kendi arasında benzer özelliklere sahip alt kümelere ayrılmasıdır. Bu çalışmada veri kümeleme amacıyla bulanık tabanlı kümeleme algoritması (FCM) ile yeni bir evrimsel optimizasyon algoritması olan Backtracking Search (BSA) algoritması birleştirilerek yeni bir hibrit kümeleme algoritması olan BSAFCM önerilmiştir. Ayrıca bu algoritmanın yerel arama yeteneklerini arttırmak için bir iyileştirme yapılmış ve yeni algoritmaya g-BSAFCM ismi verilmiştir. Geliştirilen bu algoritmalar kullanılarak UCI Machine Learning Repository veri tabanında yer alan üç ayrı veri seti kümelenmiştir. Elde edilen sonuçlara göre g-BSAFCM algoritması FCM ve BSAFCM algoritmasına göre daha iyi sonuçlar elde etmiştir.**

Anahtar Kelimeler — kümeleme;BSA;FCM; anahtar kelimeler.

***Abstract*— Clustering is dividing a dataset into subsets that has similar characteristics. In this study, fuzzy c-means clustering algorithm (FCM) and a new evolutionary optimization algorithm, Backtracking Search (BSA) algorithm, were combined and a new hybrid clustering algorithm (BSAFCM) was proposed. Moreover, the local search abilities of the new algorithm was improved and the new algorithm was named as g-BSAFCM. Three benchmark datasets from UCI Machine Learning Repository database were clustered by using the developed algorithms and FCM. According to the results g-BSAFCM has achieved better results than FCM and BSAFCM.**

Keywords — clustering; BSA; FCM; key words.

I. GİRİŞ

Kümeleme, bir veri setindeki değerleri, sonlu sayıda ve kendi içinde benzer alt gruplara bölmek olarak tanımlanabilmektedir[1]. Kümeleme algoritmaları pazar araştırmaları[2], görüntü işleme[3], veri madenciliği[4] ve biyoenformatik[5-7] gibi farklı alanlarda kullanılmaktadır. Kümeleme teknikleri farklı şekillerde sınıflandırılabilmektedir. Bu sınıflamalardan biride bulanık kümeleme yöntemidir. Bulanık kümeleme yönteminde veri tek bir kümeye ait olmak yerine bütün kümeler için farklı üyelik değerleri alır[1]. FCM algoritması (Fuzzy C-Means) da bulanık kümeleme yöntemiyle hareket eden bir algoritmadır. FCM algoritması Dunn[8] tarafından ortaya konmuş ve Bezdek[9] tarafından geliştirilmiş bir algoritmadır. FCM algoritmasının yerel minimuma takılma ve başlangıç değerlerinin rastgele seçilmesi gibi iki önemli dezavantajı bulunmaktadır. Bu sorunların üstesinden gelmek amacıyla pek çok araştırmacı FCM algoritmasını farklı optimizasyon yöntemleriyle birleştirme yoluna gitmiştir[10-13]. Bu çalışmada da Çivicioğlu tarafından geliştirilen evrimsel BSA(Backtracking Search Algorithm) optimizasyon algoritması ile FCM algoritması birleştirilerek BSAFCM isimli yeni bir hibrit kümeleme algoritması önerilmiştir. Ayrıca BSA algoritmasında iyileştirme yapılarak g-BSAFCM algoritması geliştirilmiştir. Geliştirilen algoritmalar ve FCM algoritması UCI Machine Learning Repository Veri tabanından seçilen Iris, Breast Cancer ve Lung Cancer veri setleri için test edilmiştir[17]. Elde edilen sonuçlar ile gerçek değerler arasındaki benzerlik Rand Indeks ile hesaplanmış ve g-BSAFCM algoritmasının kümelemede çok daha iyi sonuçlar elde ettiği görülmüştür[16]. Çalışmanın ikinci bölümünde çalışmada kullanılan algoritmalar hakkında bilgiler verilmiş, üçüncü bölümünde deneysel çalışmalar ve dördüncü bölümde de çalışmanın kısa bir özeti verilmiştir.

II. ÇALIŞMADA KULLANILAN ALGORİTMALAR

A. FCM Algoritması

FCM algoritması Dunn[8] tarafından bulunmuş ve Bezdek[9] tarafından geliştirilmiş bir veri kümeleme algoritmasıdır. Kümeleme işlemi, eldeki verilerin benzer özelliklere sahip alt gruplara ayrılması olarak ifade edilebilir. FCM algoritması bu işlem için bulanık mantık yöntemini kullanır. Yani bir elemanın sadece bir kümeye ait olması yerine aynı elemanın tüm kümelerde bir değer

978-1-5090-1679-2/16/$31.00 ©2016 IEEE

ile temsil edilmesi sağlanır. Bu doğrultuda kümeleme işlemi bir *V*, veri setinde yer alan *n* tane elemanın *c* tane kümeye ayrılması olarak tanımlanabilir ve aşağıdaki şekilde ifade edilir [14].

$$V=\left(v_1,v_2,v_3,...,v_n\right) \quad (1)$$

Tüm üyelik değerlerinin bir arada tutulduğu matrise de üyelik matrisi denir. Bu durum aşağıdaki gibi gösterilebilir [14].

$$U=u_{i,j}\in \left[0,1\right]_{cxn} \quad (2)$$

(2)'de U tüm eleman üyelik değerlerinin tutulduğu üyelik matrisi, $u_{i,j}$ ise veri setindeki j. elemanın i. kümede yer alan üyelik değeridir. Üyelik matrisi (3)'te verilen kriterleri sağlamalıdır [14]. FCM algoritması kümeleme işlemi yaparken (4)'te verilen amaç fonksiyonu minimize eder.

$$\sum_{i=1}^{c}u_{i,j}=1,\ 0\le\sum_{j=1}^{n}u_{i,j}<n,\ 0\le u_{i,j}\le 1 \quad (3)$$

$$O=\sum_{i=1}^{c}\sum_{j=1}^{n}(u_{i,j})^m\left(d_{i,j}\right)^2 \quad (4)$$

(4)'te O amaç fonksiyon, m bulanık mantık sabiti ve d ise j 'nci elemanın i'nci küme merkezine olan uzaklığıdır. Bu uzaklık Öklid uzaklığıdır ve aşağıdaki gibi hesaplanmaktadır[14].

$$d_{i,j}=\left\|h_i,v_j\right\| \quad (5)$$

(5)'de h_i , i 'nci kümenin merkezidir ve j 'nci elemanın i 'nci küme merkezine olan uzaklığıdır. FCM algoritması (1-5) te yer alan işlemleri belirli bir durdurma kriteri sağlanana kadar tekrar eder.

B. BSA Algoritması

BSA son yıllarda geliştirilmiş bir evrimsel optimizasyon algoritmasıdır. BSA'da iki çaprazlama ve bir mutasyon oparatörü ve arama-yönü matrisi şeklinde bir matris kullanılmaktadır. BSA'nın çalışması aşağıdaki gibi sıralanabilir[15].

Başlangıç: Bu aşamada optimizasyon için başlangıç nüfusu aşağıdaki denklemle belirlenir [15].

$$S_{i,j}\sim R\left(\min_j,\max_j\right) \quad (6)$$

(6)'da , $S_{i,j}$ = (i=1,2,3,….,n ve j=1,2,3,…,d) başlangıç nüfusunun (S 'in) i 'nci elemanının j 'inci boyuttaki elemanını ifade etmektedir . Denklemde geçen n ve d sayıları, sırasıyla başlangıç nüfusunun eleman sayısı ve her bir elemanın boyut sayısıdır. Son olarak R ise normal dağılımı ifade etmektedir. Başlangıç aşamasında BSA ayrıca S için amaç fonksiyon değerlerini de belirlemektedir.

$$fitness=ObjectFunc\left(S\right) \quad (7)$$

(7)'de $fitness\ nx1$ lik bir vektördür ve nüfusun her bir elemanı için hesaplanan amaç fonksiyon değerlerini içerir. $ObjectFunc$ise amaç fonksiyondur.

Seçim I: Bu bölümde *oldP* matrisi üretilir. İlk aşamada *oldP* matrisi başlangıç nüfusu gibi rastgele üretilir [15].

$$oldP_{i,j}\sim R\left(\min_j,\max_i\right) \quad (8)$$

oldP, diğer döngülerde rastgele seçilmez. Diğer bölümlerde *oldP* (9) 'daki formülle hesaplanır. [15].

$$Eğer\ r_1<r_2\ \text{ise}\ oldP=S\quad ;\quad r_1,r_2\sim R\left(0,1\right) \quad (9)$$

BSA hesaplanırken kullanılan $permutting$ matris elemanlarını rastgele yer değiştiren bir fonksiyondur.

$$oldP=permutting\left(oldP\right) \quad (10)$$

Mutasyon: BSA'nın mutasyon işleminde T matrisi adı verilen bir matris elde edilmektedir. Bu matris (11)'de verilen formülle hesaplanır.

$$T=S+F\left(oldP-S\right) \quad (11)$$

Çaprazlama: BSA çaprazlama yaparken T matrisini, *mixrate* adı verilen bir parametreyi ve n ve d sayılarını kullanmaktadır. Bu işlemin sonucunda elde edilen matrise ise *Mutant* matris adı verilmektedir. Buna göre çaprazlama yapılırken ilk olarak olarak nxd boyutunda birler matrisi olan *map* matrisi tanımlanmaktadır. Ardından iki farklı seçim stratejisi kullanılarak T matrisinin bazı elemanları ile S in aynı sıradaki elemanları yer değiştirmektedirler [15].

$$\begin{aligned}&Eğer\ r_1<r_2\ ise\ map_{i,u(1:[mixrate.rand.d])}=0;\\&Değilse\ map_{i,randi(d)}=0\end{aligned} \quad (12)$$

(12)'de $u=permutting\left(1,2,3\ldots,d\right)$, $rand$ $\left[0,1\right]$ aralığında *randi* ise $\left[0,d\right]$aralığında rastgele değer üreten fonksiyonlardır, *mixrate* ise kullanıcının belirleyeceği bir parametredir. (12)'de birlerden oluşan *map* matrisinin bazı elemanları iki farklı yöntemle seçilmekte ve seçilen elemanlar yerine 0 atanmaktadır. Dikkat edilirse eğer $r_1<r_2$ ise birden fazla eleman seçilirken tersi durumda sadece bir eleman seçilmektedir. *map* matrisi oluşturulduktan sonra bu matrisin elemanlarından değeri 1 olanlara karşılık gelen T matrisi elamanlarının yerine S in aynı sıradaki elemanları gelmektedir [12].

$$\begin{aligned}&Eğer\ map_{i,j}=1\ ise\ T_{i,j}=S_{i,j}\\&\left(i=1,2,3,\ldots,n\ ;\ j=1,2,3,\ldots,d\right)\end{aligned} \quad (13)$$

$$Mutant=T \quad (14)$$

Mutant matrisinin bazı elemanları arama uzayı sınırlarını aşabilir bu durumda sınırı aşanlar için (6)'da verildiği gibi rastgele seçilen elemanlar atanmaktadır.

Seçim II: Bu bölümde BSA elde edilen *Mutant* matrisi için amaç fonksiyon değerini hesaplamakta ve buna göre *fitness* vektörü ve S matrisi güncellenmektedir [15].

$$fitnessM=ObjectFunc\left(Mutant\right) \quad (15)$$

$$\begin{aligned}&Eğer\ fitnessM_{i,j}<fitness_{i,j}\ ise\\&fitness_{i,j}=fitnessM_{i,j}\ ve\ S_{i,j}=Mutant_{i,j}\end{aligned} \quad (16)$$

Başlangıç kısmı hariç son dört aşama BSA nın durdurma kriteri sağlanana kadar bir döngü içerisinde devam etmektedir.

978-1-5090-1679-2/16/$31.00 ©2016 IEEE

C. BSAFCM Algoritması

FCM algoritması, kümeleme algoritması olarak sıklıkla kullanılan bir algoritmadır. Fakat FCM algoritmasında iki sorun karşımıza çıkmaktadır. Bunlardan biri yerel minimum takılma, diğeri ise FCM'nin başlangıç küme merkezlerinin rastgele belirlemesinin performansını etkilemesidir. Bu olumsuzlukları ortadan kaldırmak amacıyla literatürde pek çok araştırmacı FCM ile farklı algoritmaları birleştirme yoluna giderek iyileşme sağlamaktadır. Bu çalışmada da FCM algoritmasındaki bu sorunları aşabilmek amacıyla BSA algoritması ile FCM algoritması birleştirilmiştir. Bunun için ilk olarak genel nüfus yapısı (17)' deki gibi tanımlanır.

$$S = \begin{bmatrix} S_{1,1} & \cdots & S_{1,c} \\ \vdots & \cdots & \vdots \\ S_{n,1} & \cdots & S_{n,c} \end{bmatrix} \quad (17)$$

(17)'ye göre başlangıç nüfusu *nxc* lik bir matris şeklinde tanımlanmıştır. Matristeki her bir satırda FCM'nin küme sayısınca eleman bulunur. Her bir satır kümeleme işlemi için gerekli aday küme merkezlerinden oluşmaktadır. Birleştirme işlemi FCM algoritmasının amaç fonksiyonunun BSA ile minimize edilmesiyle gerçekleşmektedir. Böylelikle FCM algoritması veri setindeki üyelerin küme merkezlerine uzaklığını, üyelik değerlerini hesaplarken, BSA algoritması da amaç fonksiyonu hesaplayarak küme merkezlerinin güncellenmesini sağlamaktadır. Bu şekilde iki algoritma birleştirilerek hibrit BSAFCM algoritması önerilmiştir.

C.1. g-BSAFCM Algoritması

Pınar Çivicioğlu tarafından geliştirilen BSA algoritmasında (11)'de T matris elde edilirken kullanılan F değeri (18)'deki gibi bulunmaktadır.

$$F = 3r \ , (r \sim R(0,1) \quad (18)$$

g-BSAFCM algoritmasında ise bu değer

$$F = 3r(\frac{1}{e^{k/l}})); (k = 1,2,3,\ldots,l), (r \sim R(0,1) \quad (19)$$

formülüyle hesaplanır. l değeri iterasyon sayısını, k değeri anlık iterasyonu göstermektedir. Bu parametre ile tüm optimizasyon algoritmalarının hedefi olan algoritmanın başlarında çözüm uzayının iyi taranması, sonlarına doğru ise optimum çözüme yoğunlaşılması hedefi gerçekleştirilmeye çalışılmıştır.

D. Rand Index

Rand Indeks iki küme arasındaki benzerlik oranını bulmak için kullanılır. n_s, n_d sırasıyla aynı / farklı kümelere atanmış nokta çiftlerinin sayısı ve N verilen veri setinde tüm noktaların çiftlerinin sayısını göstermek üzere aşağıdaki formülle hesaplanmaktadır[16].

$$RI = (n_s + n_d)/N \quad (20)$$

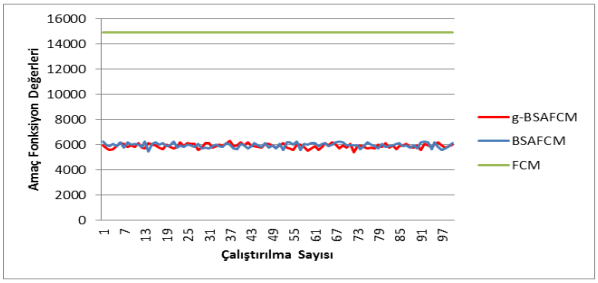
Rand indeks 0 ile 1 arasında değerler alır ve iki küme tamamen tutarlı olduğunda, rand indeks 1 değerini alır[16].

III. DENEYSEL ÇALIŞMALAR

Bu çalışmada geliştirilen BSAFCM algoritmasının performansını test edebilmek amacıyla UCI Machine Learning Repository veri tabanından Breast Cancer, Lung Cancer ve Iris veri setleri kullanılmıştır[17]. Deneysel çalışmaların hepsi aynı şartlarda Intel I-5 3.1 GHz işlemci ve 4 GB Ram özelliklerine sahip bir bilgisayarda gerçekleştirilmiştir. Bu çalışmada kullanılan tüm algoritmalarda ortak kullanılan parametreler; döngü sayısı 100, popülasyon büyüklüğü 40 ve m=2 dir. Elde edilen tüm veriler ile gerçek değerler arasındaki ilişki Rand indeks ile karşılaştırılmıştır[16].Tablo 1.'de de görüldüğü gibi amaç fonksiyonun minimize edilmesi konusunda g-BSAFCM algoritması üç veri seti için de diğer algoritmalardan daha iyi sonuçlar üretmiştir. Kümeleme kalitesini gösteren Rand Index açısından bakıldığında Breast Cancer Dataset için 0.95, Iris Dataset için 0.96 benzerlik değeriyle oldukça başarılı kümeleme gerçekleştirdiği söylenebilmektedir. Lung Cancer Dataset içinde 0.67 benzerlik değeri elde edilmiştir. Buradaki benzerlik değeri sınıflama yapılacak özellik sayısındaki artışa bağlı olarak azalabilmektedir. Ayrıca, her üç veri seti için 100 çalışmada elde edilen en iyi amaç fonksiyon değerleri de Şekil 1-3 de verilmiştir. Şekil.1-3.'e göre g-BSAFCM algoritmasının tüm veri setleri için diğer iki algoritmadan daha iyi amaç fonksiyon değerleri elde ettiği açıkça görülmektedir. Ayrıca FCM algoritmasının yerel minimuma takıldığı da görülmektedir.

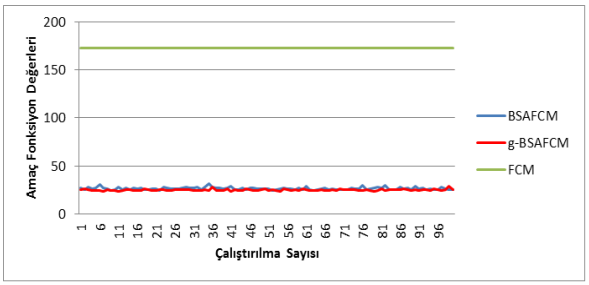
Temel İstatistiksel Sonuçlar		Amaç Fonksiyon (Ortalama)	Amaç Fonksiyon (Minimum)	Rand Index (Ortalama)	Rand Index (Maximum)
Breast Cancer Dataset	FCM	14916,68	14916,68	0,915	0,915
	BSAFCM	5936,49	5449,69	0,931	0,954178
	g-BSAFCM	**5875,42**	**5402,95**	**0,933**	**0,954178**
Lung Cancer Dataset	FCM	172,76	172,76	0,529	0,641026
	BSAFCM	26,32	23,94	0,553	0,663817
	g-BSAFCM	**24,80**	**23,35**	**0,563**	**0,675214**
Iris Dataset	FCM	6050,57	6050,57	0,879	0,879732
	BSAFCM	1561,80	1466,80	0,877	0,957494
	g-BSAFCM	**1512,17**	**1456,36**	**0,880**	**0,965638**

Tablo 1. 3 Farklı Dataset için Elde Edilen Amaç Fonksiyon ve Rand Indeks Değerlerinin İstatistiksel Sonuçları

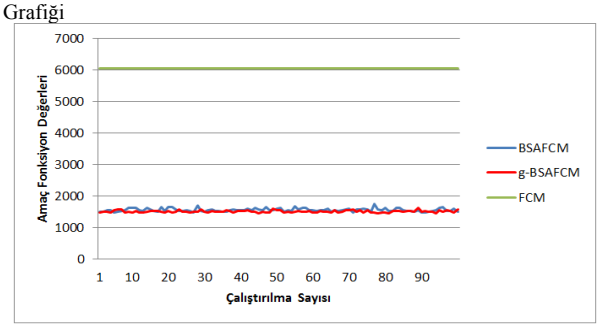


Şekil 1. Breast Cancer Dataset için Elde Edilen Amaç Fonksiyon Grafiği

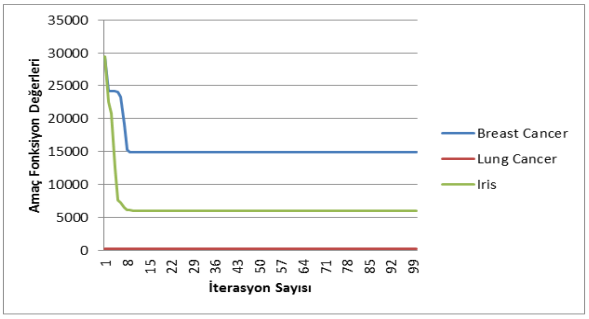
978-1-5090-1679-2/16/$31.00 ©2016 IEEE



Şekil 2. Lung Cancer Dataset için Elde Edilen Amaç Fonksiyon Grafiği



Şekil 3. Iris Dataset için Elde Edilen Amaç Fonksiyon Grafiği



Şekil 4. Breast Cancer,Lung Cancer ve Iris Dataset için FCM Algoritmasının Bir Çalıştırılma Sonucu Amaç Fonksiyon Değerleri

FCM algoritmasının yerel minimuma takılma örneği için üç veri setinin bir çalıştırılma sonucunda elde ettikleri amaç fonksiyon değer grafiği Şekil 4.'de verilmiştir.

IV. SONUÇLAR

Bu çalışmada evrimsel bir optimizasyon algoritması olan BSA algoritması ile FCM algoritması birleştirilerek BSAFCM isimli yeni bir hibrit kümeleme algoritması önerilmiştir. Ayrıca BSA algoritmasının yerel arama yeteneği iyileştirilmiş ve bu algoritmaya g-BSAFCM adı verilmiştir. Geliştirilen algoritmalar ve FCM algoritması UCI Machine Learning Repository Veri tabanından seçilen Iris, Breast Cancer ve Lung Cancer Datasetleri için test edilmiştir[17]. Elde edilen sonuçlar ile gerçek değerler arasındaki benzerlik Rand Indeks[16] ile hesaplanmış ve geliştirilen algoritmanın kümelemede diğer iki algoritmaya göre daha iyi sonuçlar elde ettiği görülmüştür.

KAYNAKÇA

[1] Hruschka, E.R.; Campello, R.J.G.B.; Freitas, A.A.; de Carvalho, A.C.P.L.F., "A Survey of Evolutionary Algorithms for Clustering," in *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* , vol.39, no.2, pp.133-155, March 2009

[2] Baoying Wang; Qin Ding; Rahal, I., "Parallel Hierarchical Clustering on Market Basket Data," in *Data Mining Workshops, 2008. ICDMW '08. IEEE International Conference on* , vol., no., pp.526-532, 15-19 Dec. 2008

[3] Ahmed, N., "Recent review on image clustering," in *Image Processing, IET* , vol.9, no.11, pp.1020-1032, 11 2015

[4] Shah, C.; Jivani, A., "Comparison of data mining clustering algorithms," in *Engineering (NUiCONE), 2013 Nirma University International Conference on* , vol., no., pp.1-4, 28-30 Nov. 2013

[5] L. Fu and E. Medico, FLAME: a novel fuzzy clustering method for the analysis of DNA microarray data, *BMC Bioinformatics*, 8:3, 2007.

[6] D. Dembele and P. Kanstner, Fuzzy C-means method for clustering microarray data. *Bioinformatics*, 19: 973-980, 2003.

[7] S.Y. Kim, J.W. Lee and J.S. Bae, Effect of data normalization on fuzzy clustering of DNA microarray data. *BMC Bioinformatics*,7:134, 2006.

[8] Dunn, J. C. (1973). "A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters." Journal of Cybernetics 3(3): 32-57.

[9] Bezdek J C. Pattern recognition with fuzzy objective function algorithms [M]. New York;Plenum Press. 1981. 95-107.

[10] Yifang Yang; Guoqiang Chen; Yanchun Guo, "SVM Combined with FCM and PSO for Fuzzy Clustering," in *Computational Intelligence and Security (CIS), 2011 Seventh International Conference on* , vol., no., pp.1370-1373, 3-4 Dec. 2011

[11] Xu Yong-Feng; Zhang Shu-Ling, "Fuzzy Particle Swarm Clustering of Infrared Images," in *Information and Computing Science, 2009. ICIC '09. Second International Conference on* , vol.2, no., pp.122-124, 21-22 May 2009

[12] Yunguang Gao; Shicheng Wang; Shunbo Liu, "Automatic Clustering Based on GA-FCM for Pattern Recognition," in *Computational Intelligence and Design, 2009. ISCID '09. Second International Symposium on* , vol.2, no., pp.146-149, 12-14 Dec. 2009

[13] Karthigayan, M.; Rizon, M.; Yaacob, S.; Nagarajan, R.; Sugisaka, M.; Rozailan Mamat, M.; Desa, H., "Fuzzy clustering for genetic algorithm based optimized ellipse data in classifying face emotion," *in Control, Automation and Systems, 2007. ICCAS '07. International Conference on* , vol., no., pp.1-5, 17-20 Oct. 2007

[14] Xia K, Wu Y, Ren X, (2013). "Research in Clustering Algorithm for Diseases Analysis." *Journal of Networks*; Vol 8, No 7 (2013).

[15] Civicioglu, P., "Backtracking Search Optimization Algorithm for numerical optimization problems." *Applied Mathematics and Computation* c. 219(15), s.8121-8144, 2013.

[16] Zhang, Y. and Chen, Y.W., *Hierarchical Clustering with Proximity Metric Derived from Approximate Reflectional Symmetry*, Springer,Berlin,2006.

[17] Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science

978-1-5090-1679-2/16/$31.00 ©2016 IEEE