Khalid Jebari*, Abdelaziz Elmoujahid and Aziz Ettouhami

Automatic Genetic Fuzzy c-Means

https://doi.org/10.1515/jisys-2018-0063
Received January 28, 2018; previously published online April 25, 2018.

Abstract: Fuzzy c-means is an efficient algorithm that is amply used for data clustering. Nonetheless, when using this algorithm, the designer faces two crucial choices: choosing the optimal number of clusters and initializing the cluster centers. The two choices have a direct impact on the clustering outcome. This paper presents an improved algorithm called automatic genetic fuzzy c-means that evolves the number of clusters and provides the initial centroids. The proposed algorithm uses a genetic algorithm with a new crossover operator, a new mutation operator, and modified tournament selection; further, it defines a new fitness function based on three cluster validity indices. Real data sets are used to demonstrate the effectiveness, in terms of quality, of the proposed algorithm.

Keywords: Genetic algorithms, unsupervised learning, fuzzy clustering, evolutionary algorithms, gravitational search, differential evolution.

1 Introduction

Clustering has been widely applied in various disciplines, including medical sciences [9], computer sciences [44], bioinformatics [20, 34, 48], bankruptcy forecasting [17], astronomy [3], and weather classification [35, 36]. Clustering can be divided into two different types: crisp and fuzzy. The former, which supposes the classes are clearly separated, is a traditional technique. It assigns each object to only one class [28]. By contrast, the latter does not make assumptions about the separation of the classes. In addition, instead of allocating the object to a unique class, fuzzy methods assign membership degrees to which the objects belong to classes [9]. Therefore, these models allow, generally, a better description of the real data where the borders between the classes are often inaccurately defined.

Fuzzy c-means (FCM) [7] is a dynamic method. Objects can change the degrees of membership during the process of class training. As this technique is iterative, the outcome is sensitive to initialization [5]. As a consequence, the improper selection of initial centroids will generally lead to undesirable clustering results [25]. A simulated annealing algorithm [47], a Tabu search algorithm [2, 39, 52], a genetic algorithm (GA) [4, 12, 23, 26, 37, 53], an ant colony [27, 30, 31, 49], and an artificial bee colony [32, 33] are examples of heuristic methods that have been used over the last two decades to overcome the problem of initialization. An exhaustive review of these algorithms can be found in Ref. [15].

Another handicap has been that FCM needs the number of clusters as input parameter. Viable methods to overcome this drawback can be found in Refs. [4, 11, 12, 23, 30, 38, 45, 46, 53]. In addition, other versions of FCM based on iterative or recursive techniques attempt remedying this encumbrance [13, 19]. However, these techniques present the problems of time complexity.

As clustering data sets can be viewed as an optimization problem [14], we propose a novel technique to cluster data based on a GA [24]. A relevant advantage of this algorithm is its ability to deal with local optima by maintaining, diversifying, and comparing several candidate solutions simultaneously. However, the GA challenge is to maintain the balance between exploitation and exploration.

***Corresponding author: Khalid Jebari,** Technologies and Sciences Faculty Tangier, Department of Computer Sciences, Tangier, Morocco, e-mail: khalid.jebari@gmail.com
Abdelaziz Elmoujahid: LCS Laboratory, Faculty of Sciences, Department of Physics, Mohamed V University, Rabat, Morocco
Aziz Ettouhami: LCS Laboratory, Faculty of Sciences, Department of Physics, Mohamed V University, Rabat, Morocco

We address the application of a hybrid GA, called automatic genetic FCM (AGFCM), based on gravitational mutation [43], differential crossover [41, 42, 51], modified tournament selection, and the FCM algorithm. We consider in AGFCM the balance between exploitation and exploration of candidate solutions using the new genetic operators mentioned previously.

The performance of AGFCM has been tested on three real data sets from the University of California at Irvine (UCI) repository [10], and the results have been compared using other techniques. The rest of this paper is organized as follows. Section 2 presents a brief description of the FCM algorithm. Section 3 describes our proposed algorithm to solve data clustering problems. Experimental results and comparison to other available methods are discussed in Section 4. Finally, conclusions and future work are highlighted in Section 5.

2 Fuzzy c-Means

Many clustering methods are introduced in the literature. These can be classified into two methods: hard and fuzzy. In hard clustering algorithms, which are based on classical set theory, the object belongs to one class. In fuzzy clustering algorithms, objects can belong to all classes with different degrees of membership. This is appropriate for real-world data where boundaries between clusters are not well defined. That is why fuzzy clustering presents the advantage of dealing with overlapping clusters.

Let $X=\{x_1, x_2, \ldots, x_n\} \subset \Re^p$ be a set of n objects with dimension p. Partitioning X in c clusters can be defined by a matrix $U=[u_i,_j] \in \Re^{n\times c}$, which satisfies the following three conditions:

$$0 \le u_{i,j} \le 1; \quad \forall \ 1 \le i \le n \ \text{ and } \ \forall \ 1 \le j \le c, \tag{1}$$

$$\sum_{j=1}^{c} u_{ij} = 1; \quad \forall \ 1 \le i \le n, \tag{2}$$

$$0 < \sum_{i=1}^{n} u_{ij} < n; \quad \forall \ 1 \le j \le c, \tag{3}$$

where u_{ij} is the membership degree of x_i for the j^{th} cluster.

The FCM algorithm optimizes the J_m criterion defined by

$$J_m(U, V) = \sum_{i=1}^{n}\sum_{j=1}^{c} (u_{ij})^m \| x_i - v_j \|^2, \tag{4}$$

where

- $V(v_1, v_2, \ldots, v_c) \in \Re^{c\times p}$ and v_j is the j^{th} prototype;
- m $(1 < m < \infty)$ is a parameter used to control the level of fuzziness in the resulting clusters;
- $\| \ \|$ is a norm to measure the distance between the j^{th} prototype and the i^{th} data point.

Bezdek showed that FCM always converges to a minimum of J_m under the following conditions [7]:

$$u_{ik} = \left(\sum_{j=1}^{c} \left(\frac{\| x_k - v_i \|}{\| x_k - v_j \|} \right)^{\frac{2}{m-1}} \right)^{-1}; \ 1 \le i \le c; \ 1 \le k \le n, \tag{5}$$

$$v_i = \frac{\sum_{k=1}^{n} (u_{ik})^m x_k}{\sum_{k=1}^{n} (u_{ik})^m} \ \text{with} \ 1 \le i \le c. \tag{6}$$

The pseudo-code of the FCM algorithm is given in Algorithm 1.

Algorithm 1: FCM pseudo-code.

Data: Vector of objects X: $(x_1, x_2, \ldots . x_n)$
Result: Prototypes $(v_1^*, v_2^*, \ldots, v_c^*)$

Choose:
– $1 < c < n$
– $m > 1$
– t_{max} maximum number of iterations
– ϵ tolerance threshold
– Norm for clustering criterion J_m
– Norm for calculating errors $E_t = || V_t - V_{t-1} ||$
Initialization:
– Prototypes V_0
– $t \leftarrow 0$
While ($E_t > \epsilon$ and $t < t_{max}$) **do**
  $t \leftarrow t + 1$
  Calculate U_t by using Eq. (5)
  Calculate V_t by using Eq. (6)

Algorithm 2: General description of AGFCM.

Data: Data set
Result: Best individual: number of clusters; prototypes
Initialization
for each individual i in the population **do**
  Choose $c_i \in \{c_{min}, \ldots, c_{max}\}$;
  In data set, **Choose** randomly c_i objects
end
while *not termination_condition* **do**
  Fitness evaluation;
  Modified enthusiasm selection MES();
  Differential crossover;
  Gravitational mutation;
end

3 Proposed Method

In this section, we describe the AGFCM clustering algorithm. This algorithm uses a GA that utilizes new operators and a new fitness function to evolve the number of clusters and to provide the initial centroids. The results of the GA phase are then used as an input in the FCM algorithm. The pseudo-code of the AGFCM algorithm is given in Algorithm 2. The AGFCM clustering algorithm is introduced in what follows.

3.1 Chromosome Representation

To encode a chromosome, we use a real-valued representation. The chromosomes represent the coordinates of the cluster center. If we consider P_i as the i^{th} candidate solution, $P_i = \{v_{i1}, v_{i2}, \ldots, v_{ic_i}\}$, where $v_{ij} = \{v_{ij}^1, \ldots, v_{ij}^p\}$ represent the j^{th} cluster center and c_i is a number of clusters. p is the dimensionality of the data set. A chromosome is thus a $p \times c_i$ one-dimensional array. As each chromosome P_i has a different c_i, the representation is of variable length.

3.2 Population Initialization

In the AGFCM clustering algorithm, an initial population is randomly generated. For each chromosome, a number of classes, c_i, is randomly chosen between $c_{\min}$ and $c_{\max}$. c_i points from the data set are randomly chosen to initialize the chromosome. In this paper, $c_{\min}$ is set to 2 and the value $\sqrt{n}$ is assigned to $c_{\max}$ [8].

3.3 Fitness Evaluation

The fitness function of a candidate solution indicates how relevant a solution is. In this paper, the fitness function is based on three well-known clustering validity measures:

1. Davies-Bouldin (DB) index: The DB index is a function of the ratio of the sum of within-cluster scatter to between-cluster separation [16]. This index is defined as

$$DB = \frac{1}{c}\sum_{i=1}^{c} \max_{i,j\neq i}\left\{\frac{(S_i + S_j)}{d_{ij}}\right\}, \tag{7}$$

where c is the number of clusters. S_i is the scatter within the i^{th} cluster and d_{ij} is the distance between cluster centers v_i and v_j. S_i is defined as

$$S_i = \frac{1}{|X_i|}\sum_{x\in X_i} ||x - v_i||^2. \tag{8}$$

X_i is the i^{th} cluster and d_{ij} is defined as

$$d_{ij} = ||v_i - v_j||^2. \tag{9}$$

2. Xie and Beni (XB) index: Xie and Beni [54] introduced a validity measure and defined it as

$$XB = \frac{\sum_{k=1}^{c}\sum_{j=1}^{n} u_{kj}^2 ||x_j - v_k||^2}{n(\min_{i\neq j}\{||v_i - v_j||^2\})}, \tag{10}$$

where u_{kj} is the membership of the j^{th} point to the k^{th} cluster.
In Eq. (10), the numerator measures the compactness of fuzzy partition. The denominator measures the separation between clusters.

3. Partition entropy (V_{PE}) index: partition entropy is a function of U proposed by Bezdek [6]. V_{PE} is formulated as

$$V_{PE} = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{c}[u_{ij}\log_a(u_{ij})], \tag{11}$$

where $U = [u_{i,j}]$ is the matrix of membership degrees.
The fitness function consists of summing the three indices outlined using a weighting coefficient for each of them. The function has the following form:

$$f(x) = \sum_{i=1}^{3} w_i f_i(x), \tag{12}$$

where f_i is the index of validity expressed above.

In this study, we chose the weighting coefficients as $w_1 = w_2 = w_3 = \frac{1}{3}$. As the values of the three indices of validity may be different for the maximum and the minimum, f may be dominated by the index validity with large values. Therefore, each index of validity is normalized according to

$$f_i = \frac{f_i - f_i^{\min}}{f_i^{\max} - f_i^{\min}}, \tag{13}$$

where $i \in \{1, 2, 3\}$ and $f_i^{\min}$ and $f_i^{\max}$ are, respectively, the minimum and the maximum value of the component validity index recorded so far in the evolution of the algorithm.

3.4 Selection Method

The modified enthusiasm selection (MES) [1, 29] is used as a selection operator. The MES is based on the tournament selection method. It is a technique that gives another chance to worst individuals in a population to compete with the best individuals in the evolving process. With a view to increase the fitness of those individuals, an enthusiasm coefficient λ is multiplied (mathematical meaning) with the old fitness value. After the enthusiasm individual has been selected, the MES put it to its raw fitness. MES also guards in each iteration the best individual [29]. The pseudo code of the selection method is given in Algorithm 3. In Algorithm 3, TabS

Algorithm 3: MES().

```
Data: Array: (TabR(), TabS() )
Result: Array: (TabW())
Initialization;
k←c;
l←0;
for i←0 to k do
  Shuffle TabR();
  j←0
  while j < n do
    C1←TabR(j);
    for m←1 to k do
      C2←TabR(j + m);
      if f(C1) < f(C2) then
        C1←C2
      end
      if m > 1 then
        aux←m;
        for m←1 to k do
          TabT(m)←f(I_{j+m});
          f(I_{j+m}) = λf((I_{j+m});
        end
        m←aux;
      end
    end
    for m←1 to k do
      f(I_{j+m})←TabT(m);
    end
    j←j + k + 1;
    TabW(l)←C1;
    TabW(l + 1)←TabS(l);
    l←l + 2;
  end
end
```

represents an array of n individuals' indices in the current population. TabR is an array holding the indices of individuals in a random order. TabT represents an array of $k-1$ individual fitness, where k is the tournament size. TabW, an array of individual indices, is the result of the selection operation.

3.5 Crossover Operator

A crossover operator is a probabilistic process to combine selected parents to create new offsprings. The crossover operator implemented in this study is a two-points crossover; it is a two-parents-two-offsprings schema following the steps below:

1. Firstly, two individuals P_1 and P_2 are randomly selected, and two crossing points are chosen.
2. The crossover occurs between P_1 and P_2 using simulated binary crossover and generates two intermediate children, C_1 and C_2. Note that crossover points are restricted to fall on the same location within each cluster description.
3. Differential crossover is used. It combines two strategies into one including their entire advantages [21, 42, 50]. The first strategy uses the values of the objective function to determine a "good" direction. The second strategy uses the best individual. Notice that the introduction of information such as the "good" direction and the best individual reduces the search space exploration capabilities. The new children are generated by the differential crossover according to

$$C_1' = P_1 + \lambda \cdot (x_{\text{best}} - P_1) + F_1 \cdot (C_1 - C_2), \tag{14}$$

and

$$C_2' = P_2 + \lambda \cdot (x_{\text{best}} - P_2) + F_2 \cdot (C_1 - C_2). \tag{15}$$

λ is used to enhance the crossover when incorporating the current best vector x_{best}. F_1 and F_2, which control the amplification of the differential variation of the offsprings C_1 and C_2, are real and constant factors. In this paper, we set $\lambda = F_1 = F_2 = 0.9$.

3.6 Mutation Operator

The mutation operator diversifies the population and avoids the creation of a set of homogeneous population elements. It should change the solution adequately to leave the attraction basin of the local optimum, but it should also avoid changing the solution too much and destroying already promising structures.

A novel mutation operator named *gravitational mutation* is used in this paper. The proposed operator is based on the gravitational search algorithm (GSA). GSA, a population-based search algorithm, is based on the law of gravity and interaction between masses [43]. To mutate an individual P_i, Eq. (16) is used:

$$P_i = P_i + s_i, \tag{16}$$

where $i \in \{1, \ldots. N\}$, N is the population size, and s_i is the next velocity of P_i computed as

$$s_i = (rand_i \times s_i) + a_i. \tag{17}$$

$rand_i$ is a random number in [0,1]. a_i is the acceleration of P_i computed as

$$a_i = \frac{F_i}{M_i}. \tag{18}$$

Here, F_i is the total force acting on P_i, and it is calculated as follows:

$$F_i = \sum_{j=1, j\neq i}^{N} rand_j G \frac{M_j M_i}{R_{ij} + \varepsilon}(P_j - P_i), \tag{19}$$

where $rand_j$ is a random number in [0,1], G is the gravitational constant, R_{ij} is the Euclidian distance between P_i and P_j, ϵ is a small constant to avoid division by zero, and M_i is the mass of P_i calculated as follows:

$$M_i = \frac{fit_i - worst}{\sum_{j=1}^{N}(fit_j - worst)}, \tag{20}$$

where fit_i is the fitness value of P_i and *worst* is defined as follows (for a maximization problem):

$$worst = \min(fit_j), \tag{21}$$

and $j \in \{1, \ldots, N\}$.

4 Experimental Results

To evaluate the relevance of AGFCM, experiments were conducted on three real-world data sets from the UCI Machine Learning Repository [10]: the Iris data set, Glass data set, Wine data set, and Breast Cancer Wisconsin data set.

- The Iris data set [22]: This may be one of the most used data sets in clustering problems. It consists of three classes that represent three species of iris plants: *Iris setosa*, *Iris virginica*, and *Iris versicolor*. Fifty observations belong to each class. Each observation consists of four characteristics: length and width of the sepal and petal of the flower [10].
- Glass data set: It consists of six classes that represent six different types of glass – building windows float processed, building windows non-float processed, vehicle windows float processed, containers, tableware, and headlamps. Each observation consists of nine numeric attributes: refractive index, sodium, magnesium, aluminum, silicon, potassium, calcium, barium, and iron [10].
- Breast Cancer Wisconsin: It consists of two classes that represent benign (239 objects) or malignant (444 objects) tumors. Each observation consists of nine features: clump thickness, uniformity of cell size, uniformity of cell shape, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli, and mitoses [10].
- The Wine dataset is the result of a chemical analysis of wines grown in three different cultivars in the same region in Italy [10]. There were 178 samples: 59 in the first class, 71 in the second class, and 48 observations belong to the third class. Each observation consists of 13 numeric features: alcohol, malic acid, ash, alkalinity of ash, magnesium, total phenols, flavanoids, non-flavanoid phenols, proanthocyanins, color intensity, hue, OD280/OD315 of diluted wines, and proline [10].

The following criteria were used to compare the clustering results:

- The number of classes found.
- Inter-cluster distance: the distance between the centroids of the clusters.
- Intra-cluster distance: the distance between data vectors within a cluster.
- Correct ratio R_c: the correct ratio of cluster numbers is defined by

$$R_c = \frac{TCNC}{RT} \times 100\%, \tag{22}$$

where *TCNC* represents the number of times when the correct number of clusters was obtained by the algorithm and *RT* is the number of times where the algorithm was run.

AGFCM was compared with dynamic clustering particle swarm optimization (PSO) [40] and the standard GA (SGA). As the three algorithms used for comparison are stochastic in nature, we have undertaken 100 independent runs. The results have been stated in terms of the mean values for each couple (algorithm, data set).

The algorithms discussed in this section have been developed in a C++ language on a Core i7 PC, with a 2-GB Debian OS environment.

In Table 1, we report the mean number of classes found by the compared algorithms. The inter-cluster distance and intra-cluster distance obtained for the three algorithms are given in Tables 2 and 3.

The results in Tables 1–3 indicate that the AGFCM algorithm succeeds in obtaining the most appropriate number of classes over 100 runs. It has a good performance in terms of the inter-cluster distance and the intra-cluster distance. AGFCM obtains a better clustering of the data.

Table 4 gives the comparative data based on the correct ratio. It can be obviously deduced that the AGFCM algorithm remains clearly and consistently superior to its competitors.

Table 1: Mean Number of Classes.

Data set	SGA	PSO	AGFCM
Iris	2.23 ± 0.07	2.50 ± 0.08	3.04 ± 0.01
Glass	4.71 ± 0.03	5.68 ± 0.04	6.05 ± 0.02
Wine	3.71 ± 0.07	2.68 ± 0.04	3.05 ± 0.05
Cancer	2.22 ± 0.06	3.01 ± 0.04	2.04 ± 0.04

Table 2: Inter-cluster Distance.

Data set	FCM	SGA	PSO	AGFCM
Iris	2.05 ± 0.05	2.105 ± 0.08	2.412 ± 0.09	2.598 ± 0.15
Glass	840.20 ± 6.15	898.20 ± 9.15	869.42 ± 8.01	853.12 ± 3.08
Wine	2.15 ± 0.15	2.20 ± 0.15	2.42 ± 0.06	3.12 ± 0.04
Cancer	2.15 ± 0.15	2.121 ± 0.09	2.621 ± 0.08	3.251 ± 0.06

Table 3: Intra-cluster Distance.

Data set	FCM	SGA	PSO	AGFCM
Iris	3.890 ± 0.15	3.662 ± 0.15	3.967 ± 0.12	3.114 ± 0.07
Glass	670.80 ± 5.454	663.30 ± 4.34	661.12 ± 3.15	563.12 ± 2.19
Wine	6.15 ± 1.2	5.95 ± 1.9	5.13 ± 0.10	4.12 ± 0.06
Cancer	5.01 ± 0.25	4.984 ± 0.25	4.538 ± 0.10	4.037 ± 0.08

Table 4: Correct Ratio (%) of Cluster Numbers for Different Methods.

Data set	SGA	PSO	AGFCM
Iris	48	73	97
Glass	61	94	94
Wine	64	80	88
Cancer	44	81	96

Table 5: T-test Results of Comparing the Different Algorithms.

T-test results	τ	Population size			
		50	100	200	500
AGFCM-SGA		≈	+	+	++
AGFCM-PSO	10	≈	+	+	+
AGFCM-SGA		+	+	++	++
AGFCM-PSO	50	≈	++	++	++
AGFCM-SGA		+	+	++	++
AGFCM-PSO	100	+	++	++	++
AGFCM-SGA		+	++	++	++
AGFCM-PSO	200	+	+	++	++

The statistical results [18] of comparing AGFCM with SGA and PSO are given in Table 5. We used the one-tailed t-test with 58 degrees of freedom at a 0.05 level of significance, the changes in every τ generations. The notation used in Table 5 to compare each pair of algorithms is "+," "++," or "≈," when the first algorithm is better than, significantly better than, or statistically equivalent to the second algorithm, respectively. Table 5 clearly demonstrates that the performance of AGFCM surpasses that of the other algorithms. When the change of population size N is small, the difference between the algorithms is not very meaningful. However, when the population size is large, AGFCM outperforms the other algorithms.

5 Conclusions

In this paper, we presented a hybrid GA for setting the appropriate number of clusters and determining initial cluster centroids for FCM. AGFCM uses two heuristic approaches, namely differential evolution and GSA, as a basis for genetic operators so as to exploit the best research areas and to explore other ones. The use of the MES, as a selection operator, controls the selection pressure. The experimental results on three real data sets indicate that the proposed algorithm improves the outcomes and outperforms two state-of-the-art clustering techniques. Future research may focus on integrating the automatic clustering scheme with the AGFCM algorithm for other metaheuristics such as PSO or differential evolution.

Bibliography

[1] A. Agrawal and I. Mitchell, Selection enthusiasm, in: *Proceedings of the 6th International Conference on Simulated Evolution and Learning*, pp. 449–456, Springer-Verlag, Berlin, 2006.
[2] K. S. Al Sultan, A Tabu search approach to the clustering problem, *Pattern Recogn.* **28** (1995), 1443–1451.
[3] G. J. Babu and E. D. Feigelson, *Statistical Challenges in Modern Astronomy II*, vol. 1, Springer, New York, 1997.
[4] S. Bandyopadhyay and U. Maulik, Genetic clustering for automatic evolution of clusters and application to image classification, *Pattern Recogn.* **35** (2002), 1197–1208.
[5] A. M. Bensaid, L. O. Hall, J. C. Bezdek and L. P. Clarke, Partially supervised clustering for image segmentation, *Pattern Recogn.* **29** (1996), 859–871.
[6] J. C. Bezdek, Mathematical models for systematics and taxonomy, in: *Proceedings of Eighth International Conference on Numerical Taxonomy*, vol. 3, pp. 143–166, W.H. Freeman, San Francisco, 1975.
[7] J. C. Bezdek, *Pattern recognition with fuzzy objective function algorithms*, Kluwer Academic Publishers, Norwell, MA, USA, 1981.
[8] J. C. Bezdek and N. R. Pal, Some new indexes of cluster validity, *IEEE Trans. Syst. Man Cybern. B Cybern.* **28** (1998), 301–315.
[9] J. C. Bezdek, J. Keller, R. Krisnapuram and N. Pal, Fuzzy models and algorithms for pattern recognition and image processing, *The Handbooks of Fuzzy Sets Series*, vol. 4, Springer US, New York, NY, USA, 1999.
[10] C. Blake, E. Keogh and C. J. Merz, UCI repository of machine learning databases (http://www.ics.uci.edu/mlearn/MLRepository.html), 1998. Accessed 27 January 2018.
[11] A. Bouroumi and A. Essaïdi, Unsupervised fuzzy learning and cluster seeking, *Intell. Data Anal.* **4** (2000), 241–253.

[12] D. -X. Chang, X. -D. Zhang, C. -W. Zheng and D. -M. Zhang, A robust dynamic niching genetic algorithm with niche migration for automatic clustering problem, *Pattern Recogn.* **43** (2010), 1346–1360.
[13] R. Cucchiara, C. Grana, S. Seidenari and G. Pellacani, Exploiting color and topological features for region segmentation with recursive fuzzy c-means, *Mach. Graphics Vis.* **11** (2002), 169–182.
[14] S. Das, A. Abraham and A. Konar, Automatic clustering using an improved differential evolution algorithm, *IEEE Trans. Syst. Man. Cybern. A Syst. Hum.* **38** (2008), 218–237.
[15] S. Das, A. Abraham and A. Konar, Metaheuristic pattern clustering – an overview, in: *Metaheuristic Clustering*, *Studies in Computational Intelligence*, vol. 178, pp. 1–62, Springer, Berlin, 2009.
[16] D. L. Davies and D. W. Bouldin, A cluster separation measure, *IEEE Trans. Pattern Anal. Mach. Intell.* **1** (1979), 224–227.
[17] J. De Andrés, P. Lorca, F. J. de Cos Juez and F. Sánchez-Lasheras, Bankruptcy forecasting: a hybrid approach using fuzzy c-means clustering and multivariate adaptive regression splines (MARS), *Expert Syst. Appl.* **38** (2011), 1866–1875.
[18] J. Derrac, S. Garca, D. Molina and F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, *Swarm Evol. Comput.* **1** (2011), 3–18.
[19] D. Dovžan and I. Škrjanc, Recursive fuzzy c-means clustering for recursive fuzzy identification of time-varying processes, *ISA Trans.* **50** (2011), 159–169.
[20] M. B. Eisen, P. T. Spellman, P. O. Brown and D. Botstein, Cluster analysis and display of genome-wide expression patterns, *Proc. Natl. Acad. Sci.* **95** (1998), 14863–14868.
[21] V. Feoktistov, *Differential Evolution: In Search of Solutions*, *Springer Optimization and Its Applications*, vol. 5, Springer Science + Business Media, LLC, Boston, MA, 2006.
[22] R. A. Fisher, The use of multiple measurements in taxonomic problems, *Ann. Eugen.* **7** (1936), 179–188.
[23] G. Garai and B. Chaudhuri, A novel genetic algorithm for automatic clustering, *Pattern Recogn. Lett.* **25** (2004), 173–187.
[24] D. E. Goldberg, *Genetic algorithms in search, optimization and machine learning*, 1st ed., Addison-Wesley Longman Publishing Co., Inc. Boston, MA, 1989.
[25] L. O. Hall, A. M. Bensaid, L. P. Clarke, R. P. Velthuizen, M. S. Silbiger and J. C. Bezdek, A comparison of neural network and fuzzy clustering techniques in segmenting magnetic resonance images of the brain, *IEEE Trans. Neural Netw.* **3** (1992), 672–682.
[26] L. O. Hall, I. B. Ozyurt and J. C. Bezdek, Clustering with a genetically optimized approach, *IEEE Trans. Evol. Comput.* **3** (1999), 103–112.
[27] Y. Han and P. Shi, An improved ant colony algorithm for fuzzy clustering in image segmentation, *Neurocomputing* **70** (2007), 665–671.
[28] J. A. Hartigan, *Clustering algorithms*, John Wiley & Sons, Inc., New York, USA, 1975.
[29] K. Jebari, A. Bouroumi and A. Ettouhami, Parameters control in gas for dynamic optimization, *Int. J. Comput. Intell. Syst.* **6** (2013), 47–63.
[30] P. M. Kanade and L. O. Hall, Fuzzy ants as a clustering concept, in: *22nd International Conference of the North American Fuzzy Information Processing Society, 2003, NAFIPS 2003*, pp. 227–232, IEEE, Chicago, IL, USA, 2003.
[31] P. M. Kanade and L. O. Hall, Fuzzy ants and clustering, *IEEE Trans. Syst. Man Cybern. A Syst. Hum.* **37** (2007), 758–769.
[32] D. Karaboga and C. Ozturk, Fuzzy clustering with artificial bee colony algorithm, *Sci. Res. Essays* **5** (2010), 1899–1902.
[33] D. Karaboga and C. Ozturk, A novel clustering approach: artificial bee colony (ABC) algorithm, *Appl. Soft Comput.* **11** (2011), 652–657.
[34] M. K. Kerr and G. A. Churchill, Bootstrapping cluster analysis: assessing the reliability of conclusions from microarray experiments, *Proc. Natl. Acad. Sci.* **98** (2001), 8961–8965.
[35] T. Littmann, An empirical classification of weather types in the Mediterranean basin and their interrelation with rainfall, *Theor. Appl. Climatol.* **66** (2000), 161–171.
[36] Z. Liu and R. George, Mining weather data using fuzzy cluster analysis, in: *Fuzzy Modeling with Spatial Information for Geographic Problems*, F. E. Petry, V. B. Robinson, M. A. Cobb, eds., pp. 105–119, Springer, Berlin, Heidelberg, 2005.
[37] U. Maulik and S. Bandyopadhyay, Genetic algorithm-based clustering technique, *Pattern Recogn.* **33** (2000), 1455–1465.
[38] U. Maulik and I. Saha, Automatic fuzzy clustering using modified differential evolution for image classification, *IEEE Trans. Geosci. Remote Sens.* **48** (2010), 3503–3510.
[39] M. K. Ng and J. C. Wong, Clustering categorical data sets using tabu search techniques, *Pattern Recogn.* **35** (2002), 2783–2790.
[40] M. Omran, A. Salman and A. Engelbrecht, Dynamic clustering using particle swarm optimization with application in unsupervised image classification, in: *Fifth World Enformatika Conference (ICCI 2005), Prague, Czech Republic*, pp. 199–204, Citeseer, 2005.
[41] S. Paterlini and T. Krink, Differential evolution and particle swarm optimisation in partitional clustering, *Comput. Stat. Data Anal.* **50** (2006), 1220–1247.
[42] K. V. Price, R. M. Storn and J. A. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*, Verlag Berlin Heidelberg, Germany, 2005.
[43] E. Rashedi, H. Nezamabadi-Pour and S. Saryazdi, GSA: a gravitational search algorithm, *Inform. Sci.* **179** (2009), 2232–2248.
[44] X. Rui and D. C. Wunsch, *Clustering*, IEEE Press, USA, 2009.

[45] S. Saha and S. Bandyopadhyay, A new point symmetry based fuzzy genetic clustering technique for automatic evolution of clusters, *Inform. Sci.* **179** (2009), 3230–3246.
[46] S. Saha and S. Bandyopadhyay, A symmetry based multiobjective clustering technique for automatic evolution of clusters, *Pattern Recogn.* **43** (2010), 738–751.
[47] S. Z. Selim and K. Alsultan, A simulated annealing algorithm for the clustering problem, *Pattern Recogn.* **24** (1991), 1003–1008.
[48] S. Selinski and K. Ickstadt, Cluster analysis of genetic and epidemiological data in molecular epidemiology, *J. Toxicol. Environ. Health Pt. A* **71** (2008), 835–844.
[49] P. Shelokar, V. K. Jayaraman and B. D. Kulkarni, An ant colony approach for clustering, *Anal. Chim. Acta* **509** (2004), 187–195.
[50] R. Storn, On the usage of differential evolution for function optimization, in: *1996 Biennial Conference of the North American Fuzzy Information Processing Society, 1996. NAFIPS*, pp. 519–523, IEEE, USA, 1996.
[51] R. Storn and K. Price, Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces, *J. Glob. Optim.* **11** (1997), 341–359.
[52] C. S. Sung and H. W. Jin, A tabu-search-based heuristic for clustering, *Pattern Recogn.* **33** (2000), 849–858.
[53] L. Y. Tseng and S. Bien Yang, A genetic approach to the automatic clustering problem, *Pattern Recogn.* **34** (2001), 415–424.
[54] X. L. Xie and G. Beni, A validity measure for fuzzy clustering, *IEEE Trans. Pattern Anal. Mach. Intell.* **13** (1991), 841–847.