Applied Soft Computing 24 (2014) 679–691

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

A multi-objective genetic algorithm with fuzzy *c*-means for automatic data clustering

Siripen Wikaisuksakul

Department of Mathematics and Computer Science, Faculty of Science and Technology, Prince of Songkla University, Pattani Campus, 181 Charoenpradit Road, Muang, Pattani 94000, Thailand

ARTICLE INFO

Article history:
Received 28 September 2012
Received in revised form 2 May 2014
Accepted 15 August 2014
Available online 23 August 2014

Keywords:
Clustering
Multiobjective optimization
Fuzzy clustering
Genetic algorithms

ABSTRACT

This article presents a multi-objective genetic algorithm which considers the problem of data clustering. A given dataset is automatically assigned into a number of groups in appropriate fuzzy partitions through the fuzzy *c*-means method. This work has tried to exploit the advantage of fuzzy properties which provide capability to handle overlapping clusters. However, most fuzzy methods are based on compactness and/or separation measures which use only centroid information. The calculation from centroid information only may not be sufficient to differentiate the geometric structures of clusters. The overlap-separation measure using an aggregation operation of fuzzy membership degrees is better equipped to handle this drawback. For another key consideration, we need a mechanism to identify appropriate fuzzy clusters without prior knowledge on the number of clusters. From this requirement, an optimization with single criterion may not be feasible for different cluster shapes. A multi-objective genetic algorithm is therefore appropriate to search for fuzzy partitions in this situation. Apart from the overlap-separation measure, the well-known fuzzy J_m index is also optimized through genetic operations. The algorithm simultaneously optimizes the two criteria to search for optimal clustering solutions. A string of real-coded values is encoded to represent cluster centers. A number of strings with different lengths varied over a range correspond to variable numbers of clusters. These real-coded values are optimized and the Pareto solutions corresponding to a tradeoff between the two objectives are finally produced. As shown in the experiments, the approach provides promising solutions in well-separated, hyperspherical and overlapping clusters from synthetic and real-life data sets. This is demonstrated by the comparison with existing single-objective and multi-objective clustering techniques.

1. Introduction

Clustering is an important mechanism in data analysis to define or organize a group of patterns or objects into clusters. The objects in the same cluster share common properties and those in different cluster s have distinct dissimilarity [1,2]. This basic exploratory analysis provides meaningful information for many disciplines such as pattern classification, image segmentation, document retrieval, biology, marketing research, health psychology, etc. Most data in these disciplines are in multi-dimensional space which inherently creates difficulty for humans to capture or perceive information. Therefore, the ultimate goal of data clustering is to achieve unsupervised classification of complex data where there is little or no prior knowledge on those data.

Consider a set of input patterns $X = \{x_1, x_2, \ldots, x_M\}$, where $x_i = (x_{i1}, x_{i2}, \ldots, x_{in})^T \in R^n$. Each x_{ij} is defined as a feature, attribute, dimension or variable. The aim is to partition the patterns into disjoint c subsets, $C = \{C_1, \ldots, C_c\}$, such that $C_i \neq \emptyset$, $\bigcup_{i=1}^{c} C_i = X$, and $C_i \cap C_j = \emptyset$ for $i \neq j$. This partitioning problem can be considered as an NP-hard optimization problem [3].

The most widely used algorithms to solve this problem are *k*-means [4] and fuzzy *c*-means (FCM) [5]. Both algorithms are centroid-based and require a fixed number of clusters beforehand. k and c denote the number of clusters and may be used interchangeably. The *k*-means algorithm starts from randomizing k centroids, one for each cluster. Each pattern is assigned into a cluster based on its nearest centroid, and then positions of the centroids are iteratively adjusted according to their corresponding members in order to minimize the sum of the squared error. This clustering method is considered as *hard* partitioning since k disjoint clusters are obtained. In FCM, *soft* partitioning is defined since every pattern is considered as a member of every cluster but different degrees of membership are assigned for different clusters. The aim of FCM

E-mail addresses: siripen@bunga.pn.psu.ac.th, wsiripen@gmail.com, swikai@gmail.com

http://dx.doi.org/10.1016/j.asoc.2014.08.036
1568-4946/

is to minimize the generalized least-squares objective function, which the degree of membership plays an important role to optimize the data partitioning problem. However, the disadvantages of both algorithms are well known: a correct number of clusters is required beforehand and the algorithms are quite sensitive to centroid initialization [3,6]. In practice, many problems have no knowledge on the number of clusters a priori. Most research papers have proposed solutions for such problems by running an algorithm repeatedly with different fixed values of k and with different initializations [3]. However, this may not be feasible with large data sets and large k. Furthermore, the algorithm running only with a limited number of k may be inefficient or not attractive since a solution depends on a limited set of initializations. This is known as the "number of clusters dependency" problem [7]. In a problem like this, the optimization may get stuck in local minima which the search process identifies, rather than a global optimal solution.

From the above reasons, evolutionary algorithms are shown to be alternative optimization methods using stochastic principles to evolve clustering solutions. In other words, they are based on probabilistic rules to search for a near-optimal solution from a global search space. Some evolutionary algorithms to optimize the number of clusters in data partitioning problems have been proposed in [8–17]. The approaches in [8,9] are based on swarm intelligence. In [8], simple and robust Artificial Bee Colony (ABC) algorithm which simulates the intelligent foraging behavior of honey bee swarms has been applied. Another proposed method in [9] has applied a hybrid of fuzzy c-means and PSO also known as FCM-FPSO.

In [10–12], genetic algorithms (GAs) have been applied for automatic determination of the number of clusters, where variable-length strings have an important role to tackle the problem of the number of clusters. The proposed method in [10] apply a GA with the split-and-merge operator and the technique in [10] uses the k-means algorithm. In [13], the k-means clustering has been enhanced by a GA, which considers the affect of isolated points. The k-means based on GA has also been applied in [14]. The immune GA and dynamic chromosome coding has been proposed to improve the search and to increase the convergence. FCM is also incorporated in the evolutionary algorithm in [16] and its application for remote sensing imagery in [17]. To optimize the number of clusters in appropriate data partitioning, suitable cluster validities or objective functions must be identified. The aforementioned methods have deployed the objective function based on cluster compactness which is commonly applied in most clustering algorithms. In complex data sets where clusters are partitioned with different sizes or different geometric shapes, optimization with only a single criterion may not be applicable to solve these characteristics simultaneously [7,15,18].

Multi-objective evolutionary algorithms (MOEAs) have been shown to deliver promising solutions for such problems with effective search performance over single-objective clustering algorithms [15]. Two or more conflicting (or complementary) objective functions are deployed in the evolutionary process. MOEAs for clustering have been proposed in [15,19]. In [15], the algorithm called MOCK is based on PESA-II with locus based chromosome encoding. It has shown to outperform single-objective clustering algorithms and ensemble techniques. However, it performs well for hyperspherical shaped or well-separated clusters but provides low performance on overlapping clusters [19]. Another disadvantage on the locus based encoding is the length of the string, which increases with the size of the data set. This imposes expensive computation when a large data set is analyzed.

VAMOSA, developed in [19], is based on multi-objective simulated annealing with center-based encoding and the newly developed point symmetry based distance from [20]. The approach has been compared with MOCK and other algorithms in artificial and real-life data sets of varying shapes, sizes or convexity. It successfully determines the appropriate number of clusters and provides overall performance better than other algorithms. However, it fails to detect a cluster having non-symmetrical shapes.

This article presents a multi-objective evolutionary approach to optimize data clustering with no requirement of the number of clusters. The clustering method considers fuzzy clusters, where data associating with degrees of membership contain more information than hard clusters. Fuzzy clustering is more naturally feasible in many real situations since data elements are not assigned to exactly one class. The discrete nature of hard clustering is not favorable if clusters are overlapped. On the other hand, fuzzy clustering has the potential to overcome this limitation. In our approach, the fuzzy functionality is used to capture the overall cluster structures through the simultaneous optimization of two objectives: the well-known fuzzy J_m index [5] and the overlap-separation measure [21,22]. This approach called FCM-NSGA applies the non-dominated sorting genetic algorithm-II (NSGA-II) [23] as an underlying multi-objective optimization framework. The fuzzy c-means algorithm is utilized in the allocation of data points to fuzzy clusters. The problem encoding is represented by variable-length strings with real-coded values of cluster centers. This encoding enables searching for a proper number of clusters.

In the remainder of this article, FCM-NSGA is described in Section 2 providing details for each part of the algorithm. Section 3 presents evaluation of solutions and how to select the most appropriate solution. Data sets applied in the experiment are detailed in Section 4. Thereafter, Section 5 describes the results of FCM-NSGA compared with those of other approaches. Section 6 provides details of time complexity of the algorithm and Section 7 concludes the paper.

2. FCM-NSGA

This section presents the overall mechanism of FCM-NSGA starting from a brief background of genetic algorithms (GAs) and NSGA-II. A center-based string encoding suitable for the clustering problem is then presented. The functionality of FCM is incorporated in the approach and two objective functions are deployed in the multi-objective optimization. In the final part, genetic operators responsible for the search process are detailed.

2.1. An evolutionary approach

Genetic algorithms (GAs) are search and optimization techniques based on the stochastic approach enhanced by the principles of biological evolution in nature [24]. They were invented to mimic natural evolution, using the idea of a chromosome which encodes the genetic information of an individual. The genetic information is decoded to determine the individual's fitness which depends on its interaction with an environment. In simulated evolution, individuals represent encodings of solutions in GAs' problem space. The chromosome's performance, known as fitness, is then interpreted by decoding its representation according to an objective function (i.e., evaluation function, fitness function) for a particular problem. In the evolutionary process, reproduction, crossover and mutation operators play significant roles to produce better and better individuals until an optimal solution is obtained.

A GA has been applied in the non-dominated sorting genetic algorithm (NSGA), which was originally proposed by Srinivas and Deb [23] to solve multi-objective optimization problems. Due to the evaluation by multiple objectives, a vector of decision variables is given, which produces a set of solutions called Pareto-optimal solutions or non-dominated solutions. The main mechanism of NSGA is a non-dominated sorting process to construct a non-dominated

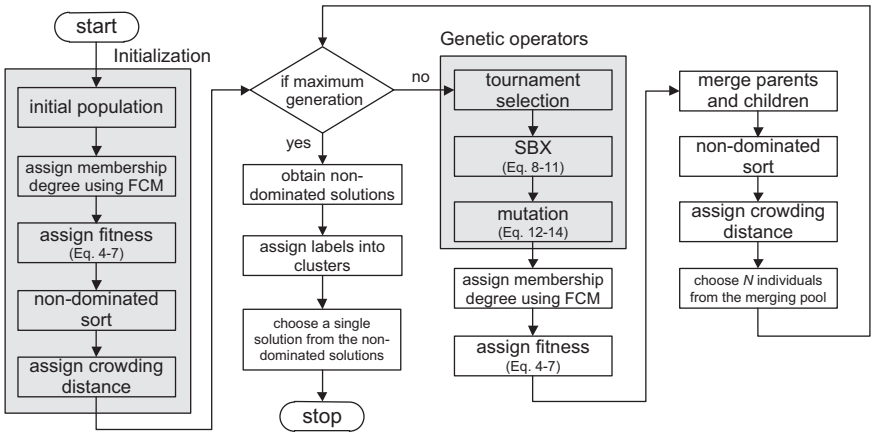

Fig. 1. Flowchart of FCM-NSGA.

front. However, NSGA has high computational complexity with $O(MN^3)$, where M is the number of objectives and N is the population size. NSGA-II [25] was therefore developed from the first version to reduce the complexity from $O(MN^3)$ to $O(MN^2)$. NSGA-II uses a fast non-dominated sorting to search for a non-dominated front and a crowding distance to maintain diversity in population.

In our work, NSGA-II has been applied as a framework for multi-objective optimization of appropriate fuzzy clusters. The overall procedure of our algorithm is presented in Fig. 1. In the first generation, the random operator initializes N individuals for an initial population by randomly choosing cluster center points from M data patterns. The initial numbers of clusters are generated from a uniform distribution over the range 2 to $\sqrt{M}$ which is recommended by [19]. The assignment of membership degree and the update of center values are based on the FCM algorithm further described in Section 2.3. The values of fitness are then computed according to the two objectives detailed in Section 2.4. The fast non-dominated sorting and crowding distance assignment of NSGA-II are then performed.

After the first generation, the GA operators then produce children solutions in the evolutionary process through simulated binary crossover (SBX) and mutation operators. The parent and children solutions are then merged to $2N$ solutions but only N individuals are chosen after the non-dominated sorting and crowded comparison procedures have finished. The process runs and searches for non-dominated front solutions until it meets the termination criterion (e.g., maximum number of generations). The non-dominated solutions are finally obtained. On each solution, each data point is assigned into a cluster corresponding to its maximum degree of membership. Further details for each module of the algorithm are described in Sections 2.2–2.5.

2.2. String representation

The chromosome in the FCM-NSGA algorithm is represented by a string of real-coded values defining cluster centers in n-dimensional space. Fig. 2 shows an example of a chromosome

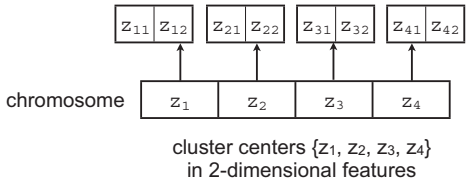

Fig. 2. String representation.

comprising four centers $\{z_1, z_2, z_3, z_4\}$ in two dimensions. For example, a string representation can be {(6.01, 2.79), (4.99, 3.40), (5.49, 3.25), (6.88, 3.08)}. The number of clusters is therefore represented by the length of the string. Each chromosome in an initial population has different lengths varied over a range of 2 to $\sqrt{M}$.

2.3. Fuzzy c-means (FCM)

The main procedures of the FCM algorithm are the calculation of membership degree and the update of cluster centers. The membership degree is used to indicate the extent to which each data point belongs to each cluster, and this information is also used to update the values of cluster centers.

Let $X = \{x_1, x_2, \ldots, x_M\}$ be data points of M patterns, where each pattern x_k is a vector of features in R^n (n-dimensional space). C is the number of clusters. A distance from a data point x_k to a cluster center v_i is calculated using the squared Euclidean distance as follows:

$$d_{ik}^2 = \sum_{j=1}^{n} (x_{kj} - v_{ij})^2, \quad 1 \le k \le M, 1 \le i \le C \tag{1}$$

d_{ik}^2 denotes the squared Euclidean distance calculated in n-dimensional space. Thereafter, the distance is used in the calculation of membership degree in Eq. (2).

$$u_{ik} = \frac{1}{\sum_{j=1}^{C} \left(\frac{d_{ik}}{d_{jk}}\right)^{2/(m-1)}}, \quad 1 \le k \le M, 1 \le i \le C \tag{2}$$

u_{ik} denotes a degree of membership of x_k in the ith cluster. $m > 1$ is a parameter which controls a degree of fuzziness. This means that each data pattern has a degree of membership in every cluster.

The values of centroids are then updated according to Eq. (3). Finally, the membership degree of each point is calculated once again using Eq. (2) taking the new centroid values.

$$v_i = \frac{\sum_{k=1}^{M} (u_{ik})^m x_k}{\sum_{k=1}^{M} (u_{ik})^m}, \quad 1 \le i \le C \tag{3}$$

2.4. Objective functions

Most objective functions considered in the clustering problem have usually based on two indexes: compactness and separation. The compactness indicates variation between data within a cluster or between data and cluster centroids, and it must be kept small. The separation measures the isolation of clusters, which is preferred to be large.

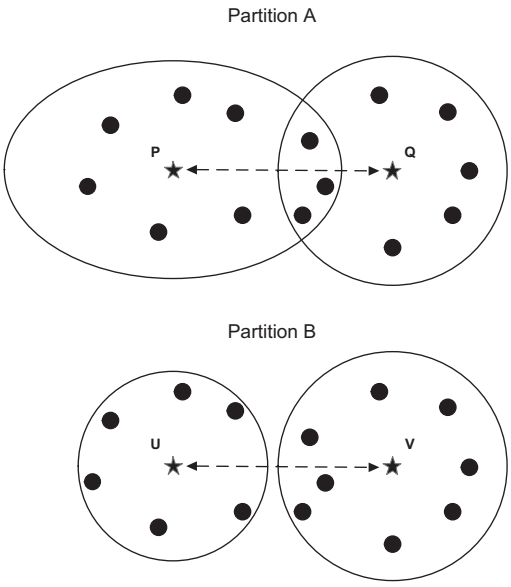

Fig. 3. Two partitions having equal distance between centroids from [26].

If only compactness or separation is considered in a single-objective optimization, some limitations may degrade the performance of the optimization. The drawback of compactness is well known that it suffers from a monotonic decrease with increasing number of clusters [26,22]. This is apparent from the experimental results in Section 5.3. For the traditional separation, its disadvantage has been raised in [26]. Separation considering inter-distance measurement between cluster centroids cannot correctly detect geometric structures. As demonstrated in Fig. 3, the distance from centroids P to Q in the partition A compared to the distance between U and V in the partition B are equal. In terms of the traditional measure of inter-cluster distance, these two partitions have the same property of separation but intuitively partition B is shown to have more separation. It can be seen that the measurement of centroid distance only can misjudge the separation of clusters because the overall shape is not considered. This leads to limited information about cluster structures.

In order to solve the aforementioned problems, two objectives are optimized simultaneously in the multi-objective optimization of FCM-NSGA. These objectives are based on compactness together with on overlap and separation measure. The compactness is formulated by the objective function J_m proposed by Bezdek [5] as shown in Eq. (4).

$$J_m(U,V) = \sum_{k=1}^{M}\sum_{i=1}^{C}(u_{ik})^m d_{ik}^2 \tag{4}$$

u_{ik} and m are defined in accordance with Eqs. (2)–(3). The sum of the squared error is measured by the squared Euclidean distance from a pattern to each centroid with the weight $(u_{ik})^m$ attached. The aim is to minimize J_m to optimize compactness taking into account distance and degree of membership.

Overlap and separation measure (overlap-separation for short) is the second objective that has a functionality to tackle the overlapping problem and to solve the problem of centroid separation as shown in Fig. 3. This measure is based on an aggregation operation of fuzzy membership degrees. There are two parts for this index. The first part is the overlap measure proposed by [21,22], which computes an inter-cluster overlap using fuzzy degrees as shown in Eq. (5).

$$O_{\perp}(u_k(x_k), C) = \underset{l=2,C}{\overset{1}{\perp}}\left(\underset{i=1,C}{\overset{l}{\perp}} u_{ik}\right) \tag{5}$$

A pattern x_k has membership vector $u_k(x_k) = (u_{1k}, \ldots, u_{ck})$. The ambiguity measurement of several membership values requires an aggregation operator (AO). The AO applied here is based on triangular norms (t-norms) for l-order ambiguity measurement. In this aggregation, the standard t-norm and t-conorm are used. The standard t-norm has the basic property that $a \top b = \min(a, b)$ and for the standard t-conorm $a \perp b = \max(a, b)$. With a single value $\underset{}{\overset{l}{\perp}} u_k \in [0, 1]$, the l-order fuzzy-OR operator (fOR-l) which is the combination of a dual couple $(\top\, \perp)$ [27] is associated with u_k, defined by

$$\underset{i=1,C}{\overset{l}{\perp}} u_{ik} = \underset{A\in\mathcal{P}_{l-1}}{\top}\left(\underset{j\in C\setminus A}{\perp} u_j\right) \tag{6}$$

$\mathcal{P}$ denotes the power set of $C = \{1, 2, \ldots, c\}$ and $\mathcal{P}_l = \{A \in \mathcal{P} : |A| = l\}$, where $|A|$ is cardinality of the subset A. From Eq. (6), the sorting in decreasing order $(u_1 \geq \cdots \geq u_C)$ is obtained, and then, the lth highest value is chosen. We apply $l = 2$ (i.e., ambiguity measures between two classes) so that the second largest element of u_k can be used.

For the separation measure [26], the maximum degree of $\max_{i=1,C} u_{ik}$ is taken into account. Therefore, the overall overlap-separation (OS) measure for M patterns is defined as follows:

$$OS = \frac{1}{M}\sum_{k=1}^{M}\frac{O_{\perp}(u_k(x_k), C)}{\max_{i=1,C} u_{ik}} \tag{7}$$

2.5. Genetic operators

In the FCM-NSGA clustering method, solutions have to be optimized in continuous search space according to the real-coded string representation of cluster centers. The stochastic search for the real-coded string is made possible by the binary tournament selection [28], the simulated binary crossover (SBX) operator [28] and the polynomial mutation operator in [29].

2.5.1. Binary tournament selection

In NSGA-II, the binary tournament selection is used to select parents to create the new generation. Two individuals are randomly chosen to play a tournament and a winner is chosen by the crowded comparison operator ($\prec_n$). This operator considers two attributes which are non-domination rank (i_{rank}) and crowding distance (i_{dist}). Let two individuals be i and j, the crowded comparison operator $\prec_n$ is defined as:

$$i \prec_n j \text{if}(i_{rank} < j_{rank}) \text{or}((i_{rank} = j_{rank}) \text{and}(i_{dist} > j_{dist}))$$

The lower rank is preferred if two individuals are in different ranks. If both individuals are in the same front (same rank), the solution with lesser crowded region is chosen.

2.5.2. SBX operator

The SBX operator [28] performs similarly to the search power of a single-point crossover on binary strings and maintains the interval schemata processing in continuous variables instead of discrete variables. To control how children are different from their parents, a spread factor β is defined under the probability distribution function:

$$\mathcal{C}(\beta) = \begin{cases} 0.5(\eta_c + 1)\beta^{\eta_c}, & \text{if } \beta \leq 1 \\ 0.5(\eta_c + 1)\dfrac{1}{\beta^{\eta_c+2}}, & \text{otherwise} \end{cases} \tag{8}$$

The value of the distribution index η_c which is any nonnegative real number has an impact on the spread of children solutions from parent solutions. A large value of η_c gives a high probability of obtaining children solutions near to parent solutions whereas a small value of η_c allows children far from their parents. From the probability distribution in Eq. (8), $\overline{\beta}$ is a random variable which makes the area under the probability curve equal to a uniform random number $u(0, 1)$, as follows:

$$\overline{\beta} = \begin{cases} (2u)^{\frac{1}{\eta_c+1}}, & \text{if } u \le 0.5 \\ \left(\frac{1}{2(1-u)}\right)^{\frac{1}{\eta_c+1}}, & \text{otherwise} \end{cases} \quad (9)$$

To obtain children solutions, two parent solutions P and Q are selected from a mating pool by the binary tournament selection. Thereafter, a random number u is generated and $\overline{\beta}$ is calculated from Eq. (9). Consider $P=(p_1, \ldots, p_n)$ and $Q=(q_1, \ldots, q_n)$, where n is the length of strings. Two children c_i^1 and c_i^2 are calculated in Eqs. (10) and (11).

$$c_i^1 = 0.5[(1+\overline{\beta})p_i + (1-\overline{\beta})q_i] \quad (10)$$

$$c_i^2 = 0.5[(1-\overline{\beta})p_i + (1+\overline{\beta})q_i] \quad (11)$$

Since each chromosome in the population has different lengths, the chromosomes of two parents may have equal or different lengths. In case their lengths are equal, the crossover operation can be illustrated in Fig. 4. Let two parents $P=\{p_1, p_2, p_3, p_4\}$ and $Q=\{q_1, q_2, q_3, q_4\}$ denote parent solutions with four cluster centers, where each p_i and q_i is a vector of features. Two children A and B are created. In uniform crossover, the decision to perform crossover on each pair of centers from parents is with a probability 0.5 [28]. In this example, the crossover operation does not performs on position 3, the value of p_3 and q_3 are directly copied to position 3 of strings A and B, respectively. Positions 1, 2 and 4 have new values a_1, a_2, a_4 and b_1, b_2, b_4 on A and B, respectively.

The main loop to control the crossover operation between two strings of parents is described as follows:

```
string length n
parent P, P = (p1, p2, ..., pn)
parent Q, Q = (q1, q2, ..., qn)
for i = 1 to n
    if random(0,1) <= 0.5
        ai = pi
        bi = qi
        SBX(ai, bi)
    else
        ai = pi
        bi = qi
    end if
end for
```

The new values of centers for two children are calculated from Eqs. (10) and (11) by the SBX() procedure as follows:

```
SBX(a, b)
u = random(0,1) // uniform random number between 0 and 1
if u <= 0.5
    beta = (2u)^(1/(eta_c+1))
else
    beta = (1/(2*(1-u)))^(1/(eta_c+1))
end if
for j = 1 to maximum dimensions
    if aj < bj
        y1 = aj
        y2 = bj
    else
        y1 = bj
        y2 = aj
    end if
    cj1 = 0.5*((1+beta)*y1 + (1-beta)*y2)
    cj2 = 0.5*((1-beta)*y1 + (1+beta)*y2)
end for
if random(0,1) <= 0.5
    a = c1
    b = c2
else
    a = c2
    b = c1
end if
```

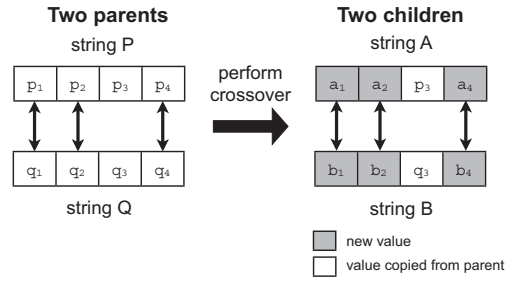

Fig. 4. Crossover operation for two chromosomes with same length.

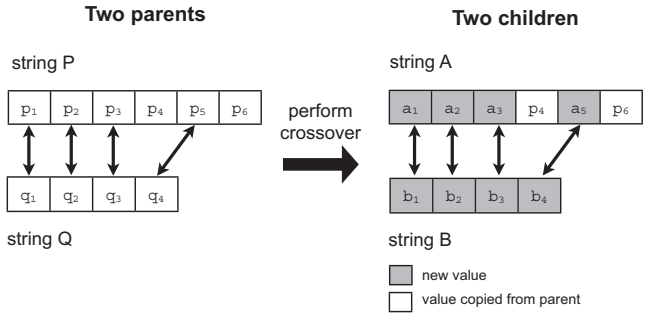

Fig. 5. Crossover operation for two chromosomes with different length.

In case string lengths of two parents are different, an example of this operation is illustrated in Fig. 5. The centroids considered to be crossed in the longer string are randomly chosen. p_1, p_2, p_3, p_5 are chosen and crossed with q_1, q_2, q_3, q_4, respectively. The values of the omitted centers (p_4, p_6) are directly copied to a child. The calculation of the new values of children then follows the same procedures as in the case of equal length.

2.5.3. Mutation operator

The polynomial mutation operator defined in [29,30] is applied with a low probability to perturb a solution for the new population. The result of mutation is controlled by a probability distribution:

$$\mathcal{P}(\delta) = 0.5(\eta_m + 1)(1 - |\delta|)^{\eta_m} \tag{12}$$

From the above distribution, the perturbation factor $\overline{\delta}$ can be calculated according to a random number r_i in the range (0, 1) and the distribution index η_m, as follows:

$$\overline{\delta_i} = \begin{cases} (2r_i)^{1/(\eta_m+1)} - 1, & \text{if } r_i < 0.5 \\ 1 - [2(1 - r_i)]^{1/(\eta_m+1)}, & \text{if } r_i \geq 0.5 \end{cases} \tag{13}$$

One parent is chosen as x_i which is the value of the ith cluster. x_i^L and x_i^U are the lower and upper bound of x_i, respectively. The mutated value y_i is thereafter calculated by

$$y_i = x_i + (x_i^U - x_i^L)\overline{\delta_i} \tag{14}$$

3. Selection and evaluation of the solution set

In the final generation, the FCM-NSGA algorithm produces a set of non-dominated solutions whose number varies according to the population size. All the solutions are considered to be equal in terms of fitness values compromised by the two objectives. In most real-world problems, a single solution must be chosen out of this set.

We deploy the selection mechanism presented in [19] where a semi-supervised method has been used. The class label of 10% of the whole data set is assumed to be known. The remaining 90% of the sample has no class label information provided and FCM-NSGA executes on these unknown label samples called *test patterns*. After the clustering procedure in the multi-objective optimization has finished, patterns are grouped in their corresponding clusters but class labels of clusters have not been defined. The class labels are later assigned by the following procedure. In [19], the assignment of class labels is based on the nearest center criterion. In FCM-NSGA, the fuzzy degree of membership is combined with the center-based criterion in the class label assignment. First, each known label sample is mapped to a cluster corresponding to the maximum degree of membership. Second, a frequency table is built by the frequency of the samples that fall in their mapped clusters. In the final step, the label of the cluster is chosen from the known label of the samples having the maximum frequency. If frequencies between the groups are equal, the cluster label is assigned by the label of the sample that has the maximum fuzzy degree with the corresponding cluster.

After the label assignment procedure, the *Minkowski score* (MS) according to [19] is computed to measure the amount of misclassification as shown in Eq. (15). Consider the *true* solution set T and the solution set to be measured S. The measure is defined by the number of point-pairs assignments of data items between T and S. n_{11} denotes the number of point-pairs that are in the same class in both S and T. n_{01} and n_{10} represents the mismatched number of point-pairs between the two sets. n_{01} denotes the number of point-pairs that are in S only, and n_{10} denotes the number of point-pairs that belong to T only. A lower score indicates a better solution. After the scores of all non-dominated solutions have been calculated, the solution with the minimum score is chosen as the best solution.

$$MS(T, S) = \sqrt{\frac{n_{01} + n_{10}}{n_{11} + n_{10}}} \tag{15}$$

4. Data sets

For the purpose of comparison, data sets applied in [19] have been used in our study. There are two groups of data sets, artificial and real-life data sets. The five artificial data sets are *AD_5_2* from [31], *AD_10_2* from [32], *Square1*, *Square4* and *Sizes5* from [18]. The five real-life data sets which are *Iris*, *BreastCancer*, *Newthyroid*, *Wine* and *LiverDisorders* are obtained from the UCI repository [33]. These data sets are summarized in Table 1 and have some characteristics as follows:

Table 1
Summary of data sets, where D denotes the number of features, K is the number of clusters, M is sample size, and M_i gives the number of members in cluster i.

DataSet	D	K	M	M_i
AD_5_2	2	5	250	5×50
AD_10_2	2	10	500	10×50
Square1	2	4	1000	4×250
Square4	2	4	1000	4×250
Sizes5	2	4	1000	769, 77, 77, 77
Iris	4	3	150	50, 50, 50
BreastCancer	9	2	683	444, 239
Newthyroid	5	3	215	150, 35, 30
Wine	13	3	178	59, 71, 48
LiverDisorders	6	2	341	142, 199

(1) *AD_5_2* contains four squared clusters with highly overlapped data.
(2) *AD_10_2* has equal size of clusters and contains some overlapped clusters.
(3) *Square1* consists of four separated clusters with equal size.
(4) *Square4* contains the same number of data samples and the size of clusters as *Square1* but the distance between individual clusters is closer. This increases overlap between clusters.
(5) *Size5* has four clusters with unequal size.
(6) *Iris* comprises 50 instances from each of three species of Iris, namely Setosa, Versicolor and Virginica. Four attributes are collected from the length and the width of sepals and petals in centimeters. Two species, Versicolor and Virginica, are highly overlapped whereas Setosa is obviously separable from others.
(7) *BreastCancer* originated from the Wisconsin breast cancer database. Nine features are represented by clump thickness, uniformity of cell size, uniformity of cell shape, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli and mitoses. Each sample is categorized within two classes: benign or malignant, and these two classes are linearly separated.
(8) *Newthyroid* includes five laboratory tests which can identify whether patient's thyroid gland functions are in normal condition, hypothyroidism or hyperthyroidism.
(9) *Wine* data are obtained from a chemical analysis determining the quantities of 13 constituents found in three types of wines in the same region but from three different cultivars in Italy.
(10) *LiverDisorders* has six attributes including five blood tests and the amount of daily drinks. The data are divided into two sets which indicate whether an individual suffers from alcoholism.

5. Experimental results

In our experiment, ten data sets detailed in Section 4 have been applied with the parameter settings summarized in Table 2. In MOEAs, multiple solutions with different numbers of clusters can be evolved and compared all together. Thus, a single run of FCM-NSGA is adequate and a set of the non-dominated solutions is obtained. The best solution is then identified from these non-dominated solutions by the minimum value of MS.

Table 2
Summary of parameter settings for the experiment.

Parameter	Setting
Population size	100
Number of generations	40
Probability of crossover	0.8
Probability of mutation	0.2
η_c	2.0
η_m	5.0
β	2.0
k_{min}	2
k_{max}	$\sqrt{M}$, M is sample size

5.1. Comparison of results with other multi-objective and single-objective approaches

FCM-NSGA is compared with other multi-objective approaches: VAMOSA [19] and MOCK [15], and with a single-objective clustering VGAPS [17]. The comparison is considered in terms of the results of MS and the number of clusters as shown in Table 3, where the results of VAMOSA, MOCK and VGAPS are obtained from [19]. The results can be described as follows:

(1) For the *AD_5_2* data set, VAMOSA and VGAPS provide the minimum MS value which shows the best partitioning. Regarding the estimation of the number of clusters, both VAMOSA, VGAPS and FCM-NSGA are able to detect the correct number whereas MOCK overestimates the number of clusters.
(2) The MS value for the *AD_10_2* data set shows that FCM-NSGA achieves best partitioning over the other three techniques. Both FCM-NSGA and VAMOSA can detect the appropriate number of clusters but MOCK and VGAPS fail to do so.
(3) Regarding the MS result from the *Square1* data set, FCM-NSGA performs slightly better than the other three approaches. All four techniques are able to identify the correct number of clusters.
(4) For the *Square4* data set, FCM-NSGA provides the minimum MS value. All techniques except VGAPS can determine the correct number of clusters. MOCK provides the hightest MS value, which means poor partitioning is obtained.
(5) The result from the *Sizes5* data set shows best partitioning from VAMOSA which gains the minimum MS value. The MS value of FCM-NSGA is slightly higher than that of VAMOSA. Both VAMOSA and FCM-NSGA are able to detect the proper number of clusters whereas MOCK underestimates and VGAPS overestimates the number of clusters.
(6) With respect to the first real-life data, the best result of partitioning the *Iris* data set is obtained by FCM-NSGA with the minimum value of MS. FCM-NSGA and VGAPS are able to identify the number of species from the *Iris* data set comprising Setosa, Versicolor and Virginica. VAMOSA and MOCK cannot distinguish the high overlap between Versicolor and Virginica. Consequently, the two species are combined into the same group and the data is then partitioned into two clusters.
(7) Regarding the *BreastCancer* data set, all four approaches are able to determine the appropriate number of clusters but the minimum value of MS is obtained by VAMOSA.
(8) From the *Newthyroid* data set, FCM-NSGA provides the minimum MS value. FCM-NSGA and VGAPS are able to determine the correct number of clusters whereas VAMOSA overestimates and MOCK underestimates the number of clusters.
(9) For the *Wine* data set, all techniques except VGAPS can identify the appropriate number of clusters and MOCK gives the best MS value.
(10) In the last partitioning of the *LiverDisorders* data set, FCM-NSGA, VAMOSA and VGAPS are able to determine the correct number of clusters. All four approaches provide high values of MS, which means they all have difficulty in partitioning the structure of this data set.

From the above results, the FCM-NSGA algorithm provides appropriate data partitioning from most of the data sets. It achieves the detection of the number of clusters of the ten data sets. VGAPS performs well on separated and compact clusters but fails for overlapping clusters. This shows that a single optimization of *Sym*-index in VGAPS cannot capture different characteristics of data. VAMOSA also uses *Sym*-index and applies another criterion, *XB* index, where these two indexes consider compactness and separation, respectively. Good solutions are obtained by VAMOSA and the overlapping structure of *AD_5_2* and *AD_10_2* can be detected. However, VAMOSA does not show a consistent performance for the highly overlapped data as shown in the result from the *Iris* data. This can assume that compactness and separation measures in VAMOSA have a problem to deal with highly overlapping. Regarding MOCK's performance, it provides poor results on overlapping clusters (*AD_5_2*, *AD_10_2* and *Iris* data sets). This is reflected from the limitation of MOCK's two objectives considering compactness and connectivity. A tradeoff between the two criteria fails to distinguish the structure of overlapping data. Overall, FCM-NSGA outperforms VAMOSA, MOCK and VGAPS. In some cases that FCM-NSGA is beaten, the difference is on a very small margin. All four techniques are able to handle well-separated and compact clusters. However, in highly overlapping data, FCM-NSGA is superior to the other three techniques as obviously shown in the results from *Square4* and *Iris* which are highly overlapped.

5.2. Comparison of results with MOCK

More experiments have been conducted to compare FCM-NSGA with MOCK [15]. In this comparison, three data sets (*Square1*,

Table 3
Minkowski score (*MS*) and the number of clusters (*K*) obtained by three multi-objective optimization approaches: FCM-NSGA, VAMOSA and MOCK, and by a single objective clustering technique, VGAPS.

DataSet	Actual *K*	FCM-NSGA		VAMOSA		MOCK		VGAPS	
		K	*MS*	*K*	*MS*	*K*	*MS*	*K*	*MS*
AD_5_2	5	5	0.29	**5**	**0.25**	6	0.39	**5**	**0.25**
AD_10_2	10	**10**	**0.13**	10	0.43	6	1.01	7	0.84
Square1	4	**4**	**0.18**	4	0.19	4	0.19	4	0.20
Square4	4	**4**	**0.49**	4	0.51	4	0.60	5	0.52
Sizes5	4	4	0.19	**4**	**0.14**	2	0.64	5	0.22
Iris	3	**3**	**0.57**	2	0.80	2	0.82	3	0.62
BreastCancer	2	2	0.36	**2**	**0.32**	2	0.39	2	0.37
Newthyroid	3	**3**	**0.52**	5	0.57	2	0.82	3	0.58
Wine	3	3	0.93	3	0.97	**3**	**0.90**	6	0.97
LiverDisorders	2	**2**	**0.97**	2	0.98	3	0.98	2	0.98

The values indicate the best performance.

Table 4
Adjusted Rand index (*ARI*) and the number of clusters (*K*) obtained by FCM-NSGA and MOCK.

DataSet	Actual *K*	FCM-NSGA		MOCK	
		K	*ARI*	*K*	*ARI*
Square1	4	4	0.9763	4.22	0.9622
Square4	4	4	0.8388	4.32	0.7729
Size5	4	4	0.9517	4.2	0.9760
2d-4c	4	4	0.9724	4.12	0.9893
2d-10c	10	10	0.7291	10.33	0.9451
10d-4c	4	4	0.9730	4.07	0.9962
10d-10c	10	13	0.8217	12.68	0.9673

Square4 and *Sizes5*) from the previous experiment and simulated data sets including *2d-4c, 2d-10c, 10d-4c* and *10d-10c* from [15] are applied. The quality of clusters is evaluated using the adjusted Rand index (ARI) [34] which is to be maximized. Table 4 summarizes the results obtained from FCM-NSGA and the results of MOCK taken from [15]. On *Sizes5*, *2d-4c* and *10d-4c*, the ARI values of FCM-NSGA are slightly lower than those of MOCK. FCM-NSGA provides the ARI value slightly higher than that of MOCK on *Square1*. This shows comparable performance between these two algorithms on those data sets. The different results are shown in the highly overlapped clusters of *Square4* and the elliptical Gaussian clusters of *2d-10c* and *10d-10c*. FCM-NSGA provides better performance to cope with the highly overlapped clusters but it performs worse in dealing with the elliptical clusters.

This is due to the property of these two data sets which contain clusters of elongated ellipsoid with not clearly separated shapes. This property causes difficulty for FCM-NSGA which utilizes the fuzzy *c*-means method. With the intrinsic property of the centroid-based mechanism, the fuzzy *c*-means algorithm is applicable to allocate clusters of spherical shapes but has a drawback to split elongated ellipsoidal clusters. On the other hand, the connectivity property of MOCK is feasible to handle elongated shapes. From this investigation, it can be concluded that the strength of FCM-NSGA is well-suited for highly overlapped clusters and its capability is needed to be enhanced to deliver better partitioning on elongated elliptical cluster shapes.

5.3. Results from multi-objective and single-objective optimizations

The following study is to explore the improvement of multi-objective optimization over single-objective optimization from the corresponding objective functions. Apart from FCM-NSGA, two single-objective clustering techniques have been implemented. Both use the same framework of the genetic algorithm and fuzzy *c*-means embeded in FCM-NSGA but consider only a single objective function. In other words, the NSGA-II mechanism has been excluded. The first single-objective approach deploys the overlap-separation measure (Eq. (7)), called FCM-GA-OS. Another approach, called FCM-GA-Jm, implements the J_m index (Eq. (4)) as the objective function.

In this study, all data sets and parameter settings are the same as those applied in Section 5.1. The results of MS value and the obtained number of clusters from all three clustering techniques are shown in Table 5. The performance ranks are also indicated within parentheses.

Considering overall results from the two single-objective techniques, FCM-GA-OS provides better results of MS and the number of clusters than those of FCM-GA-Jm. FCM-GA-Jm overestimates the number of clusters from all data sets. FCM-GA-OS is not able to detect the proper number of clusters in *Sizes5*, *Iris*, *Newthyroid* and *Wine* data sets. The unequally-sized clusters of *Sizes5* and *Newthyroid* as well as the higher number of features from *Wine* cause a problem for FCM-GA-OS. Overlapping data in *Iris* is another important characteristic that FCM-GA-OS fails to detect although it is able to detect the appropriate number of clusters of *AD_5_2* and *Square4* which also have the overlapping property.

Regarding the inferior result from FCM-GA-Jm, the obtained number of clusters tends to be overestimated. This corresponds to the aforementioned problem of J_m because data partitioning with an increasing number of clusters tends to give a decreasing value of J_m. This objective function therefore fails to determine the number of clusters.

After the overlap-separation measure and J_m have been incorporated in the multi-objective optimization in FCM-NSGA, both objectives can balance the measurement for chromosomes' fitness to capture data structures. The performance improvement is clearly seen in FCM-NSGA which is superior to the two single-objective techniques, as shown from the MS value and the result of the number of clusters in most data sets.

Figs. 6–10 compare the results from the five synthetic data sets of data partitioning obtained by FCM-NSGA with center markings and the true data partitioning. FCM-NSGA performs well under the well-separated structures of *AD_10_2* (Fig. 7) and *Square1* (Fig. 8) as well as the unequally-sized clusters of *Sizes5* (Fig. 10). The higher values of MS are obtained from the partitioning for the highly-overlapped shape in *AD_5_2* (Fig. 6) and *Square4* (Fig. 9). The overlapping characteristic causes more misclassification of the data points which are at the borderline between clusters.

It can be seen that the overlap-separation objective performs well to identify overlapping clusters. However, this objective cannot succeed for diverse cluster shapes when it works alone in the single-objective optimization. This is because the number of clusters is kept dynamic in the optimization. The J_m index is capable of balancing well against the overlap-separation objective for this dynamic problem. Therefore, the performance of the algorithm is obviously improved when the two criteria are optimized simultaneously.

Table 5
Minkowski score (*MS*) and the number of clusters (*K*) from the multi-objective optimization FCM-NSGA and two single-objective approaches: FCM-GA-OS and FCM-GA-Jm.

DataSet	Actual *K*	FCM-NSGA		FCM-GA-OS		FCM-GA-Jm	
		K	*MS*	*K*	*MS*	*K*	*MS*
AD_5_2	5	**5**	**0.29(1)**	5	**0.29(1)**	13	0.81(3)
AD_10_2	10	**10**	**0.13(1)**	10	0.15(2)	18	0.66(3)
Square1	4	**4**	**0.18(1)**	4	0.20(2)	28	0.92(3)
Square4	4	**4**	**0.49(1)**	4	0.50(2)	29	0.94(3)
Sizes5	4	**4**	**0.19(1)**	3	0.34(2)	27	0.96(3)
Iris	3	**3**	**0.57(1)**	2	0.82(2)	8	0.79(3)
BreastCancer	2	**2**	**0.36(1)**	2	**0.36(1)**	20	0.93(3)
Newthyroid	3	**3**	**0.52(1)**	2	0.74(2)	11	0.93(3)
Wine	3	**3**	**0.93(1)**	2	1.06(2)	9	0.96(3)
LiverDisorders	2	**2**	**0.97(1)**	**2**	**0.97(1)**	14	0.99(3)

The values indicate the best performance.

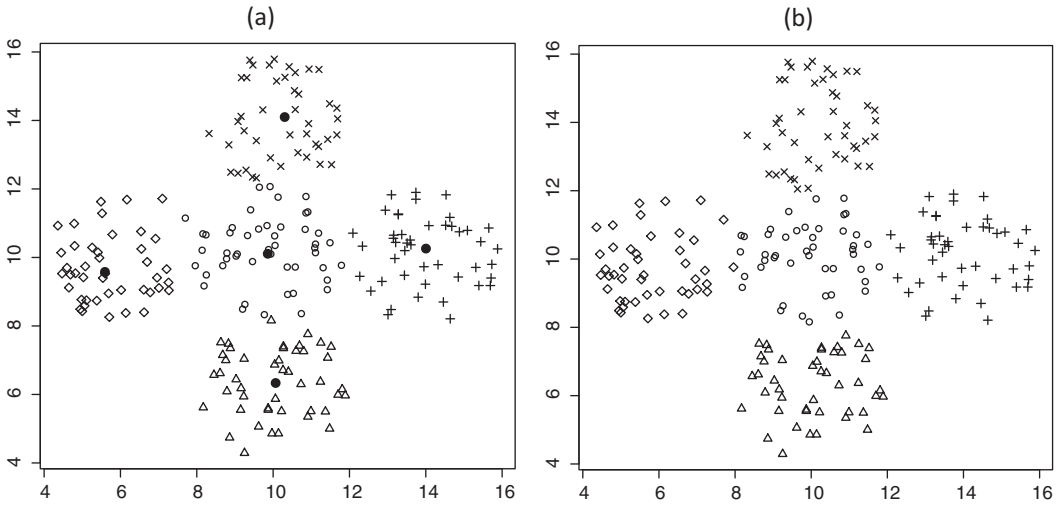

Fig. 6. *AD_5_2* ($K = 5$): (a) data partition using FCM-NSGA (MS = 0.29) and (b) true solution.

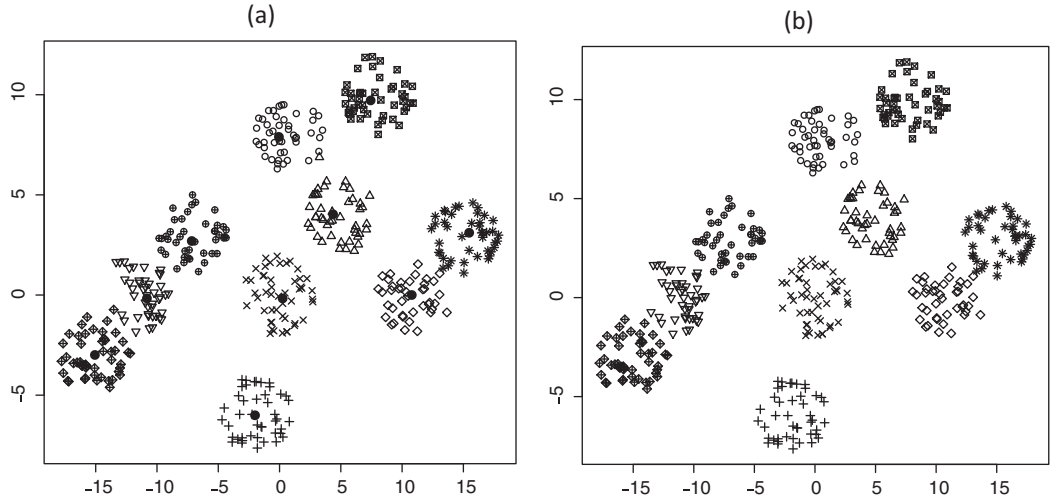

Fig. 7. *AD_10_2* ($K = 10$): (a) data partition using FCM-NSGA (MS = 0.13) and (b) true solution.

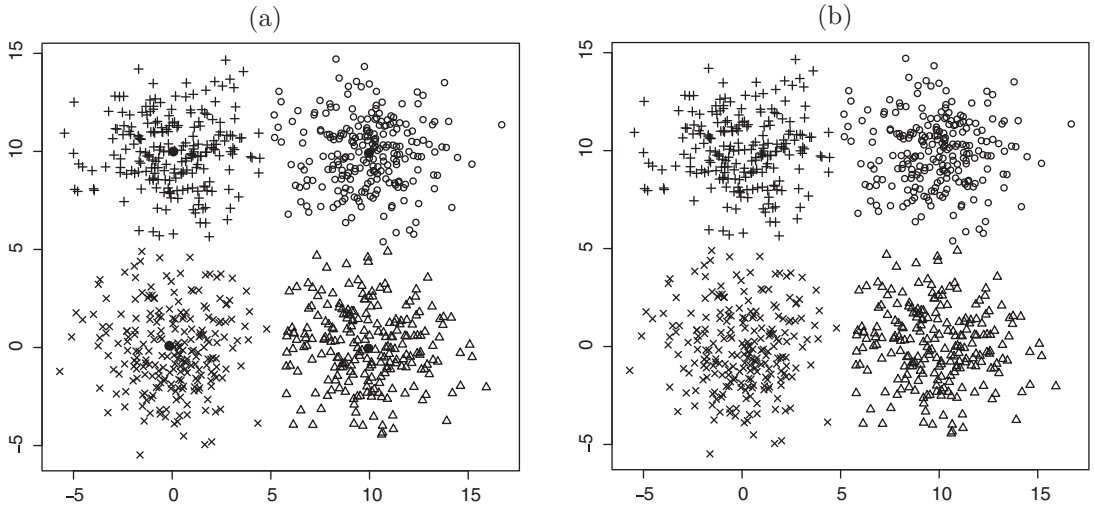

Fig. 8. *Square1* ($K = 4$): (a) data partition using by FCM-NSGA (MS = 0.18) and (b) true solution.

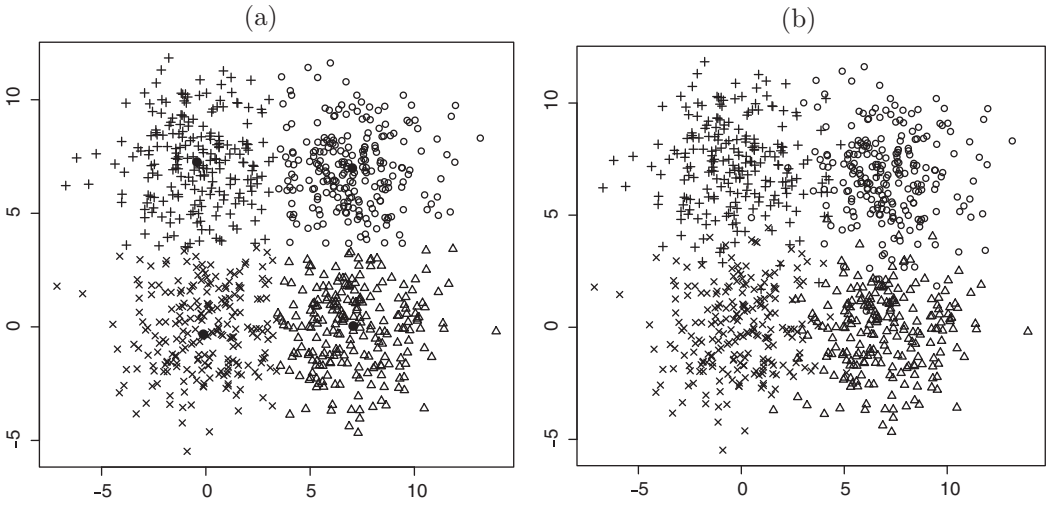

Fig. 9. *Square4* ($K = 4$): (a) data partition using by FCM-NSGA (MS = 0.49) and (b) true solution.

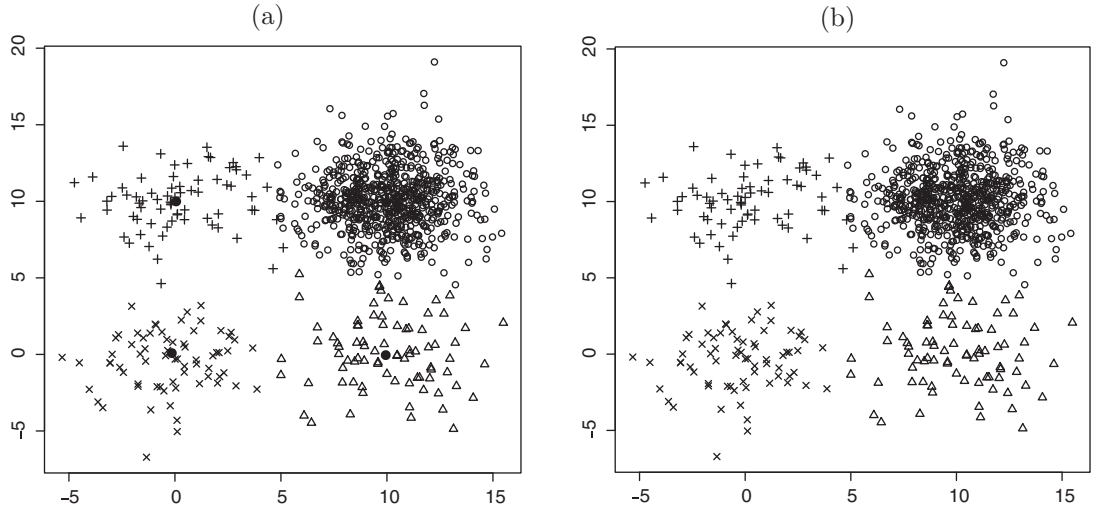

Fig. 10. *Sizes5* ($K = 4$): (a) data partition using FCM-NSGA (MS = 0.19) and (b) true solution.

5.4. Results of non-dominated solutions

The final non-dominated solutions obtained by FCM-NSGA on the real-life data sets, *Iris*, *BreastCancer*, *Newthyroid*, *Wine* and *LiverDisorders* are illustrated from Figs. 11–15, respectively. The solutions are plotted in the two-objective space and grouped according to K, the number of clusters. FCM-NSGA provides solutions with a range of different numbers of clusters varying from 2 to 8 on *Iris*, 2 to 21 on *BreastCancer*, 2 to 11 on *Newthyroid*, 2 to 9 on *Wine* and 2 to 14 on *LiverDisorders*. Some numbers of K are eliminated. This can be assumed that the solutions with skipped numbers of K are dominated by better solutions according to a tradeoff between their fitness values from the two objectives.

All results of the Pareto-optimal front are accordingly shown as the value of overlap-separation increases whereas J_m decreases with an increasing number of K. These solutions are approximately ordered by K increasing from left to right. This shows the conflict between the two objectives when a range of numbers of clusters is considered. To partition data for the appropriate number of clusters, a tradeoff between the J_m index and overlap-separation is required. The preservation of diversity in the population is another key to maintaining solutions with different numbers of clusters. From the reasonable spread of the solutions, it shows that

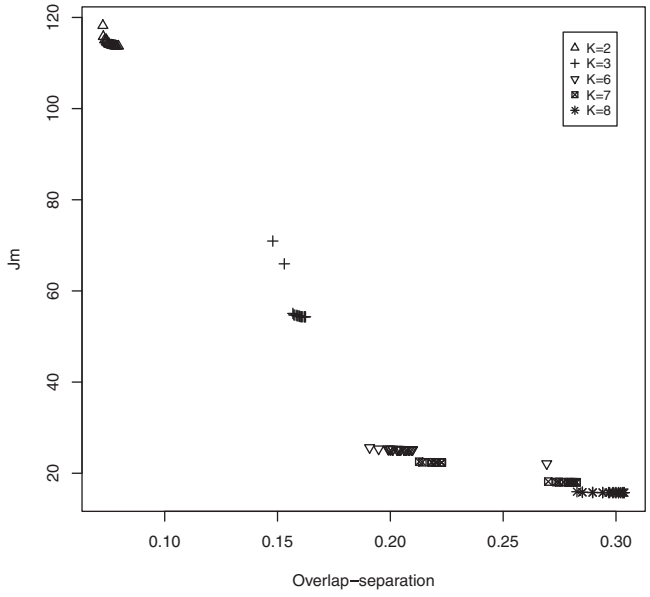

Fig. 11. Non-dominated solutions obtained by FCM-NSGA on the *Iris* data set, where *K* varies along the Pareto front.

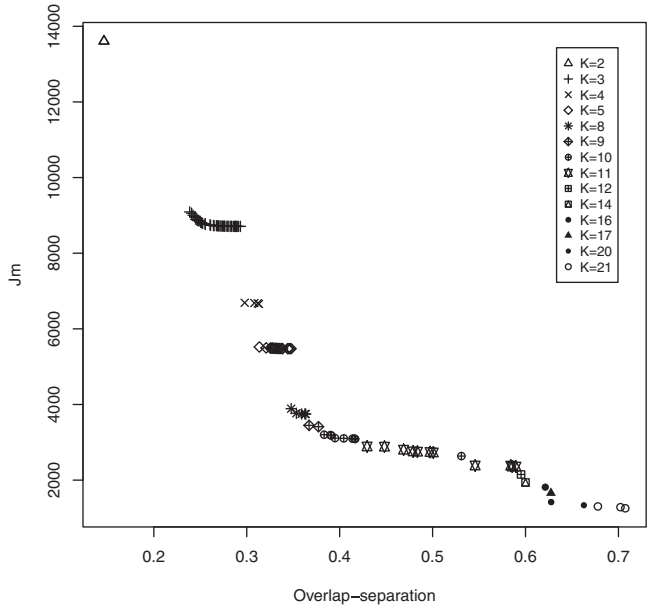

Fig. 12. Non-dominated solutions obtained by FCM-NSGA on the *BreastCancer* data set, where *K* varies along the Pareto front.

FCM-NSGA through the NSGA-II mechanism is able to handle such preservation.

5.5. Results from different parameter settings

In the evolutionary process of SBX, η_c in crossover operation and η_m in mutation are the distribution indexes to control how offspring are similar to their parents. The experiment in this section shows the sensitivity of these parameters in two population sizes. η_c and η_m are set equal as 2, 5 and 20 in small population size (*PopSize* = 40) and larger population size (*PopSize* = 100). Additionally, other parameters are kept constant as follows: termination criterion: 40 generations; probability of crossover: 0.8; and probability of mutation: 0.2. Different population sizes and distribution indexes are set:

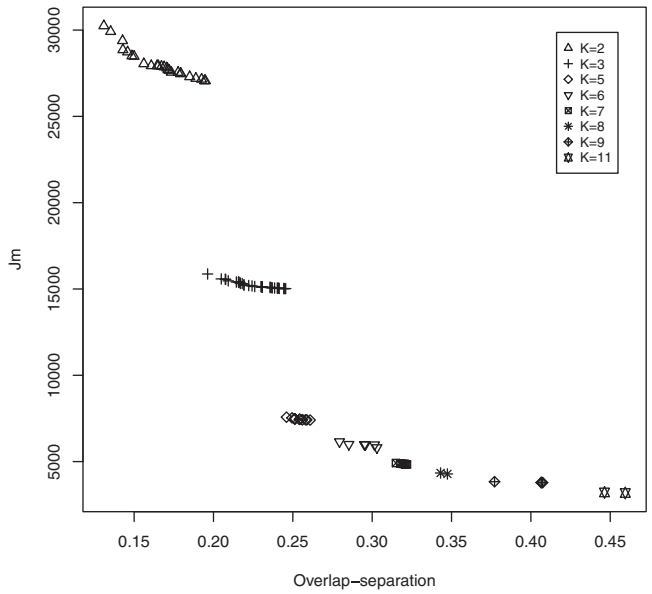

Fig. 13. Non-dominated solutions obtained by FCM-NSGA on the *Newthyroid* data set, where *K* varies along the Pareto front.

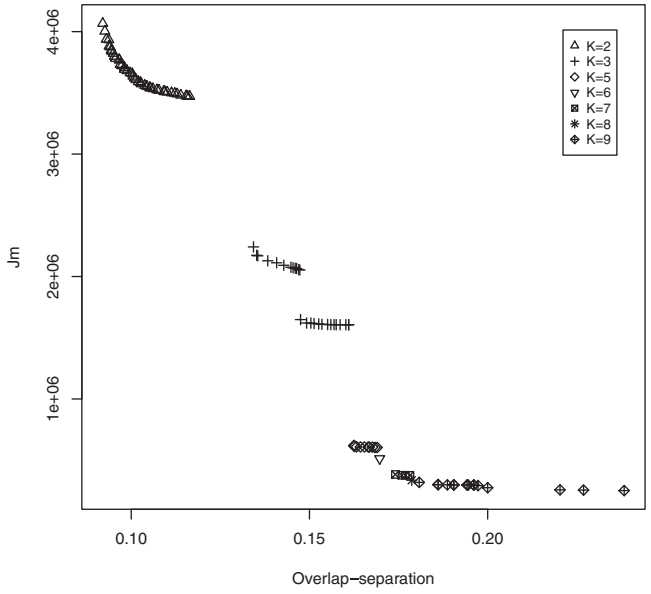

Fig. 14. Non-dominated solutions obtained by FCM-NSGA on the *Wine* data set, where *K* varies along the Pareto front.

(1) Setting 1: *PopSize* = 40, $\eta = 2$
(2) Setting 2: *PopSize* = 40, $\eta = 5$
(3) Setting 3: *PopSize* = 40, $\eta = 20$
(4) Setting 4: *PopSize* = 100, $\eta = 2$
(5) Setting 5: *PopSize* = 100, $\eta = 5$
(6) Setting 6: *PopSize* = 100, $\eta = 20$

These six settings have been tested for three data sets: *Square4*, *BreastCancer* and *Iris*. The results of *Minkowski score* (MS) and the number of clusters (K) are shown in Table 6. In Setting 1-3 using *PopSize* = 40, the MS values of *BreastCancer* and *Iris* are slightly different in different η values. The different η values obviously affect the optimization for *Square4* which the algorithm cannot determine K and produces high values of MS. Once larger population size is used in Setting 4-6, FCM-NSGA with different η values produces the similar results of MS and K for each data set. This means

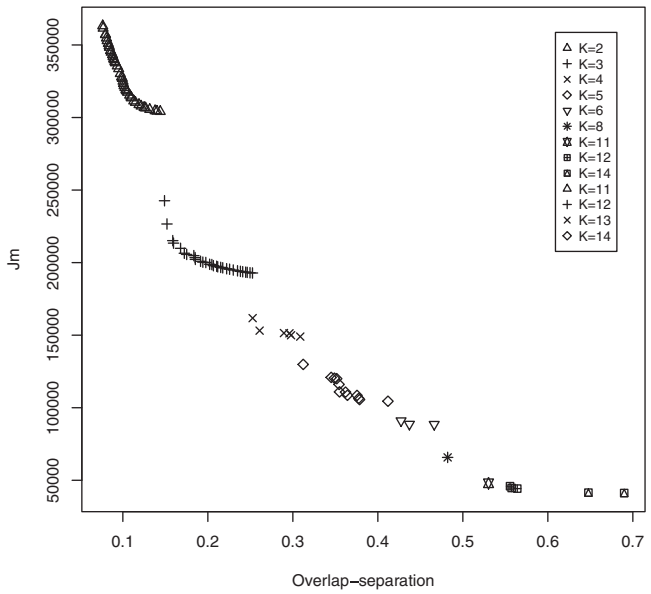

Fig. 15. Non-dominated solutions obtained by FCM-NSGA on the *LiverDisorders* data set, where *K* varies along the Pareto front.

Table 6
Minkowski score (*MS*) and the number of clusters (*K*) from the multi-objective optimization FCM-NSGA using different settings for three data sets: Square4, BreastCancer and Iris.

Settings	Square4		BreastCancer		Iris	
	K	*MS*	*K*	*MS*	*K*	*MS*
Setting 1	8	0.72	2	0.37	3	0.57
Setting 2	6	0.63	2	0.38	3	0.59
Setting 3	6	0.65	2	0.38	3	0.57
Setting 4	4	0.50	2	0.37	3	0.57
Setting 5	4	0.49	2	0.37	3	0.57
Setting 6	4	0.50	2	0.37	3	0.57

different η values do not affect the performance of the algorithm in a proper population size.

For the results of *Square4*, the correlation between data size and population size is considered. Since the size of *Square4* is larger than those of *BreastCancer* and *Iris*, varying η values affect only on *Square4*. Once data size is large, a range of the number of clusters in a random population is also wide. Small population size cannot cope with this situation and may make many errors in the search process. This can be solved when larger population size is applied. However, the question is raised if a large data set with large K is applied, how the population size should be. This issue is remained to be explored in further research.

6. Time complexity

The FCM-NSGA has a worse-case $O(GPNk_{max}d)$ time complexity, where G denotes the number of generations, P is population size, N is the size of data, k_{max} is maximum number of clusters and d is data dimensions.

(1) In an initial population, strings are created $k_{max}d$ time and each string is created in d-dimensional features until the population size P is full. Therefore, this construction requires $O(Pk_{max}d)$.
(2) In FCM clustering for each individual, both procedures of membership assignment and updating of center values take $Nk_{max}d$ time. This requires $O(PNk_{max}d)$ for all individuals in the population.
(3) The first fitness assignment of the overlap-separation objective needs $k_{max}\log(k_{max})$ to sort membership degrees of all clusters. $Nk_{max}\log(k_{max})$ time is executed for all data points. In another fitness assignment of J_m index, $Nk_{max}d$ is required to calculate distance from data points to cluster centers. The overlap-separation and J_m index take $O(PNk_{max}\log(k_{max}))$ and $O(PNk_{max}d))$, respectively.
(4) Mutation and crossover operators execute $O(Pk_{max}d)$ time each.
(5) The non-dominated sorting in NSGA-II needs MP time for each solution to compare with every other solution to find if it is dominated. M is the number of objectives and a maximum number of the non-dominated solutions equals the population size P. The comparison for all population members therefore requires $O(MP^2)$ time, where $M = 2$.
(6) In the label assignment for each non-dominated solution, $Nk_{max}d$ time is required to assign label for every data point. To select the best solution from P non-dominated solutions, this yields $O(PNk_{max}d)$ time.

The computation is dominated by the operations of FCM and fitness assignment of J_m index. This worse-case time complexity is $O(PNk_{max}d)$ per generation. It therefore becomes $O(GPNk_{max}d)$ for G number of generations.

7. Conclusion and discussion

In this paper, FCM-NSGA has been developed using NSGA-II and the FCM clustering technique to solve the data clustering problem. NSGA-II is used to control the multi-objective optimization considering two criteria: J_m and overlap-separation. The FCM algorithm is applied when patterns are assigned into different clusters with different degrees of membership in soft partitioning. The algorithm finally produces a set of non-dominated solutions. The final solution is chosen from this set by a semi-supervised method with the FCM concept for label assignment. FCM-NSGA has been compared with a single-objective technique: VGAPS [17] and two multi-objective techniques: VAMOSA [19] and MOCK [15] for given data sets including five synthetic and five real-life data sets. More data sets with elliptical shapes are also included for the comparison with MOCK. FCM-NSGA is able to handle well-separated, compact and overlapped clusters, but has a poor performance to detect elongated ellipsoidal clusters.

Apart from the fundamental frameworks, another key success in FCM-NSGA is from the chosen criteria. J_m is used to indicate compactness, and overlap-separation is responsible for the detection of overlapping characteristics of clusters. It is clear that the optimization of either J_m or overlap-separation one at a time is not effective enough for various cluster shapes. Once these two objectives are optimized simultaneously, the performance of the algorithm is improved. This shows the multi-objective approach has a performance advantage and can achieve quality solutions.

Despite its good performance, FCM-NSGA has some inherent drawbacks. Since this multi-objective technique provides an approximation to the real (unknown) Pareto front only, the optimal cluster solution cannot be guaranteed. Also, FCM-NSGA provides a set of admissible solutions with different numbers of clusters. The choice of the number of clusters cannot be done by the algorithm itself. It requires ten percent of known labels of data patterns used by the semi-supervised fuzzy decision for label assignment. To select the final solution, all data points with true labels are used to calculate the Minkowski score. In this case, this post process is needed to be improved, once labels of data set are not known a priori and not to be included in the decision maker.

From the high computation of the J_m index, it is possible that other criteria may be more suitable to be incorporated with the overlap-separation criteria. Other criteria may be investigated further. In addition, this multi-objective technique inherently requires high computation. It is therefore not suitable for some applications with heavy time or memory constraints.

References

[1] G. Gan, C. Ma, J. Wu, Data Clustering: Theory, Algorithms, and Applications, SIAM, 2007.
[2] R. Xu, Survey of clustering algorithms, IEEE Trans. Neural Netw. 16 (2005) 645–678.
[3] E.R. Hruschka, R.J.G.B. Campello, A.A. Freitas, P. Leon, A survey of evolutionary algorithms for clustering, IEEE Trans. Syst. Man Cybern. C: Appl. Rev. 39 (2009) 133–155.
[4] J.B. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, University of California Press, 1967, pp. 281–297.
[5] J.C. Bezdek, R. Ehrlich, W. Full, FCM: the fuzzy c-means clustering algorithm, Comp. Geosci. 10 (1984) 191–203.
[6] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, 2nd edition, Wiley, 2001.
[7] K.S.N. Ripon, C.-H. Tsang, S. Kwong, Multi-objective data clustering using variable-length real jumping genes genetic algorithm and local search method, in: International Joint Conference on Neural Networks, IEEE, 2006, pp. 3609–3616.
[8] D. Karaboga, C. Ozturk, A novel clustering approach: Artificial Bee Colony (ABC) algorithm, Appl. Soft Comp. 11 (2011) 652–657.
[9] H. Izakian, A. Abraham, Fuzzy c-means and fuzzy swarm for fuzzy clustering problem, Expert Syst. Appl. 38 (2011) 1835–1838.

[10] G. Garai, B.B. Chaudhuri, A novel genetic algorithm for automatic clustering, Pattern Recognit. Lett. 25 (2004) 173–187.
[11] U. Maulik, S.B. S. Bandyopadhyay, Genetic algorithm-based clustering technique, Pattern Recognit. 33 (2000) 1455–1465.
[12] D.-X. Chang, X.-D. Zhang, C.-W. Zheng, A genetic algorithm with gene rearrangement for *k*-means clustering, Pattern Recognit. 42 (2009) 1210–1222.
[13] W. Min, Y. Siqing, Improved *k*-means clustering based on genetic algorithm, in: International Conference on Computer Application and System Modeling (ICCASM), IEEE, 2010, pp. V6-636–V6-639.
[14] L. Cai, X. Yao, Z. He, X. Liang, *K*-means clustering analysis based on immune genetic algorithm, in: World Automation Congress (WAC), IEEE, 2010, pp. 413–418.
[15] J. Handl, J. Knowles, Evolutionary multiobjective clustering, in: Proceedings of the Eighth International Conference on Parallel Problem Solving from Nature, Springer, 2004, pp. 1081–1091.
[16] S. Sriparna, S. Bandyopadhyay, A fuzzy genetic clustering technique using a new symmetry based distance for automatic evolution of clusters, in: International Conference on Computing: Theory and Applications (ICCTA), 2007, pp. 304–309.
[17] S. Bandyopadhyay, S. Saha, A point symmetry based clustering technique for automatic evolution of clusters, IEEE Trans. Knowl. Data Eng. 20 (2008) 1–17.
[18] J. Handl, J. Knowles, An evolutionary approach to multiobjective clustering, IEEE Trans. Evol. Comp. 11 (2007) 56–76.
[19] S. Saha, S. Bandyopadhyay, A symmetry based multiobjective clustering technique for automatic evolution of clusters, Pattern Recognit. 43 (2010) 738–751.
[20] S. Bandyopadhyay, S. Saha, GAPS: a clustering method using a new point symmetry-based distance measure, Pattern Recognit. 40 (2007) 3430–3451.
[21] H.L. Capitaine, C. Frélicot, A family of cluster validity indexes based on a *l*-order fuzzy or operator, Lect. Notes Comp. Sci. 5342 (2008) 622–631.
[22] H.L. Capitaine, C. Frélicot, A cluster-validity index combining an overlap measure and a separation measure based on fuzzy-aggregation operators, IEEE Trans. Fuzzy Syst. 19 (2011) 580–588.
[23] N. Srinivas, K. Deb, Multiobjective optimization using nondominated sorting in genetic algorithms, Evol. Comp. 2 (1994) 221–248.
[24] D.E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, 1st edition, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.
[25] K. Deb, A. Pratap, S. Agarwal, T. Meyariva, A fast elitist non-dominated sorting genetic algorithm for multi-objective, IEEE Trans. Evol. Comp. 6 (2002) 182–197.
[26] D.-W. Kim, K.H. Lee, D. Lee, On cluster validity index for estimation of the optimal number of fuzzy clusters, Pattern Recognit. 37 (2004) 2009–2025.
[27] L. Mascarilla, M. Berthier, C. Frélicot, A *k*-order fuzzy or operator for pattern classification with *k*-order ambiguity rejection, Fuzzy Sets Syst. 159 (2008) 2011–2029.
[28] K. Deb, R.B. Agrawal, Simulated binary crossover for continuous search space, Complex Syst. 9 (2002) 115–148.
[29] K. Deb, M. Goyal, A combined genetic adaptive search (GeneAS) for engineering design, Comp. Sci. Inf. 26 (1996) 30–45.
[30] K.S.N. Ripon, S. Kwong, K.F. Man, A real-coding jumping gene genetic algorithm (RJGGA) for multiobjective optimization, Inf. Sci. 177 (2007) 632–654.
[31] S. Bandyopadhyay, U. Maulik, Genetic clustering for automatic evolution of clusters and application to image classification, Pattern Recognit. 35 (2002) 1197–1208.
[32] S. Bandyopadhyay, S. Pal, Classification and Learning Using Genetic Algorithms: Applications in Bioinformatics and Web Intelligence, Springer, 2007.
[33] A. Frank, A. Asuncion, UCI Machine Learning Repository, 2010.
[34] J. Santos, M. Embrechts, On the use of the adjusted rand index as a metric for evaluating supervised classification, Artif. Neural Netw.-ICANN (2009) 175–184.