

Research Article

An Adaptive Multiobjective Genetic Algorithm with Fuzzy c -Means for Automatic Data Clustering

Ze Dong, Hao Jia , and Miao Liu 

Hebei Engineering Research Center of Simulation & Optimized Control for Power Generation, North China Electric Power University, Baoding 071003, China

Correspondence should be addressed to Hao Jia; Tony.Jia@ncepu.edu.cn

Received 24 November 2017; Revised 27 March 2018; Accepted 5 April 2018; Published 13 May 2018

Academic Editor: Thomas Hanne

Copyright © 2018 Ze Dong et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents a fuzzy clustering method based on multiobjective genetic algorithm. The ADNPGA2-FCM algorithm was developed to solve the clustering problem by combining the fuzzy clustering algorithm (FCM) with the multiobjective genetic algorithm (NSGA-II) and introducing an adaptive mechanism. The algorithm does not need to give the number of clusters in advance. After the number of initial clusters and the center coordinates are given randomly, the optimal solution set is found by the multiobjective evolutionary algorithm. After determining the optimal number of clusters by majority vote method, the J_m value is continuously optimized through the combination of Canonical Genetic Algorithm and FCM, and finally the best clustering result is obtained. By using standard UCI dataset verification and comparing with existing single-objective and multiobjective clustering algorithms, the effectiveness of this method is proved.

1. Introduction

Clustering is a common unsupervised learning method in the field of machine learning. It has been widely used in many fields, such as data mining, pattern recognition, information retrieval, and other fields. In these fields, most of the data is unlabeled data with multiple attributes. In this high-dimensional space, it is very difficult for us to get the required information through the graph. Therefore, the ultimate goal of clustering is to achieve unsupervised classification of these few labeled or unlabeled complex data.

Fuzzy C -means algorithm (FCM) is a widely used clustering algorithm in the field of machine learning. It was proposed by Bezdek et al. in 1984 [1]. It is different from the traditional non-A or B hard clustering methods. By introducing the fuzzy membership matrix, the fuzzy C -means algorithm allows data points to belong to multiple classes according to their fuzzy membership degree. We choose the class with the highest value of the current data point in the fuzzy membership matrix as the final clustering result. This method solves the problem of clustering overlap of traditional hard clustering algorithms (such as k -means method [2]),

and more in line with the actual situation in data clustering. However, FCM has two serious shortcomings. Firstly, it easily falls into local minima. Secondly, it is necessary to specify the number of clusters and the algorithm is very sensitive to the initial center [3, 4].

To overcome the first shortcoming, some global optimization techniques have been introduced to deal with data clustering problems in the past years, for example, simulated annealing- (SA-) based [5], particle swarm optimization- (PSO-) based [6–8], genetic algorithms- (GA-) based [9–11], and quantum genetic algorithms- (QGA-) based techniques [12]. In recent years, genetic algorithms have been used to automatically determine the number of clusters by using variable-length strings [13, 14]. In [15], FCM combined with genetic algorithm, which has been successfully applied to remote sensing imagery in [16].

Most optimization based clustering algorithms are single-objective optimization algorithms, because only one validity measure of effectiveness is optimized. Noted that a single validity measure can only reflect some of the inherent segmentation attributes, for example, the compactness of clusters, the spatial separation between the clusters, and the

cluster's symmetry. If there are several classes of geometric shapes present in the same dataset, the clustering algorithms that use a single clustering validity index will not be able to process such datasets. Therefore, it is necessary to optimize several clustering validity indexes that can capture different data features at the same time. Based on this consideration, data clustering should be considered as a multiobjective optimization problem.

In recent years, multiobjective optimization problems have received extensive attention. Many scholars have conducted extensive researches on multiobjective evolutionary algorithms and have achieved extensive applications in feature extraction, data classification, and clustering.

In [17], a new method of classification feature extraction is proposed. A probability-based encoding technology and an effective hybrid operator, together with the ideas of the crowding distance, the external archive, and the Pareto domination relationship, are applied to PSO. By using this way to improve the search capability of the algorithm, the experimental comparison proves the effectiveness of the algorithm. In [18], a new multilabel feature selection algorithm is proposed to use an improved multiobjective particle swarm optimization (PSO), with the purpose of searching for a Pareto set of nondominated solutions (feature subsets). Two new operators are proposed to improve the performance of the proposed PSO-based algorithm. Finally, the effectiveness of the algorithm is verified by experiments.

Multiobjective evolutionary algorithms (MOEAs) have been proven to provide promising solutions to the problem of single-objective clustering algorithms that provide efficient search performance [19]. In [20], a multiobjective clustering technique, MOCK, is proposed to recognize the appropriate partitioning from the data sets that contain either hyperspherical shaped clusters or well-separated clusters. In [21], a multiobjective clustering technique is proposed, called VAMOS. The algorithm optimizes two clustering validity indices simultaneously, so that the algorithm can evolve proper partitioning from the clustering data set with any shape, size, or convexity. In [22], a fuzzy clustering algorithm named MOmoDEFC based on improved multi-objective differential evolution was proposed. By using XB Index [23] and FCM measure (J_m) as objective functions, the algorithm can optimize both the compactness and separation of clusters simultaneously. It also improved the clustering effect.

Based on the above consideration, we developed a fuzzy clustering algorithm by using the multiobjective optimization framework, combined the knowledge of FCM and general genetic algorithm. The algorithm is aimed to achieve the functions as follows: (1) automatically determine the number of clusters; (2) improve clustering performance.

The rest of this paper is arranged as follows. Section 2 introduces the related theories, including the FCM algorithm and the multiobjective genetic algorithm NSGA-II. Section 3 introduces the improved multiobjective optimization framework and the adaptive multiobjective dynamic fuzzy clustering algorithm ADNSGA2-FCM. In Section 4, experiments results were carried out by using some standard UCI datasets. The experimental results were compared with

many clustering algorithms in detail. Finally, conclusions are drawn in Section 5.

2. Theoretical Basis

2.1. Fuzzy C-Means Algorithm. Suppose that fuzzy c -means (FCM) partitions a set of n data objects $X = \{x_1, x_2, \dots, x_n\}_{n \times p}$ into c ($1 \leq c \leq n$) fuzzy clusters, where each object has p attributes. Let $V = \{v_1, v_2, \dots, v_c\}_{c \times p}$ be a set of cluster centers. Let $U = [u_{ik}]_{c \times n}$ be a $c \times n$ matrix of membership degrees in which u_{ik} is the membership degree of k th object to the c th cluster center. The matrix satisfies the conditions

$$\begin{aligned} \sum_{i=1}^c u_{ik} &= 1, \\ u_{ik} &\geq 0, \\ u_{ik} &\in [0, 1]. \end{aligned} \quad (1)$$

The FCM algorithm uses the objective function to solve the optimal clustering, which is a clear difference from the hard clustering algorithm. The objective function J_{FCM} can be defined as follows:

$$J_{\text{FCM}} = \sum_{i=1}^c \sum_{k=1}^n u_{ik}^m \|x_k - v_i\|. \quad (2)$$

In (2), m is the fuzzification coefficient, representing the fuzzy degree of clustering. Define $m = 2$. $\|x_k - v_i\|$ means the Euclidean distance between the k th data points and the first i th cluster centers, which represents the in-class similarity. A good clustering algorithm should ensure that the distance between similar points in the clustering result is as compact as possible. The standard FCM uses the J_{FCM} as a cost function to be minimized.

The minimization of J_{FCM} can be achieved by Lagrange multiplier method under constraint $\sum_{i=1}^c u_{ik} = 1$, $k = 1, 2, \dots, n$, while the membership degrees matrix U and cluster centers are updated according to

$$u_{ik} = \frac{\sum_{t=1}^c \left(\frac{\|x_k - v_i\|}{\|x_k - v_t\|} \right)^{-2/(m-1)}}{\sum_{t=1}^c \left(\frac{\|x_k - v_i\|}{\|x_k - v_t\|} \right)^{-2/(m-1)}}, \quad (3)$$

$$v_i = \frac{\sum_{k=1}^n (u_{ik}^m x_k)}{\sum_{k=1}^n u_{ik}^m}. \quad (4)$$

By iteration, the algorithm ends when condition $|J_{\text{FCM}}(t) - J_{\text{FCM}}(t+1)| < \varepsilon$ is satisfied, where ε is a small positive number representing the end of iteration threshold.

2.2. Multiobjective Optimization Based on Genetic Algorithm. Unlike single-objective optimization algorithms, the multi-objective optimization algorithm optimizes multiple objective functions simultaneously. Because it is necessary to optimize multiple conflicting objectives simultaneously, it is often difficult to find a solution to make all the objective functions reach the optimum simultaneously. For multiobjective optimization algorithms, each objective function is considered

equally important when the relative importance of the goals is unknown. Therefore, the multiobjective optimization problem is not to optimize one solution, but to optimize one solution set, which is characterized by improving any objective function without impairing other objective functions. We call this solution a nondominated solution or a Pareto optimal solution, which is defined as follows [24].

For minimizing the multiobjective problem, a vector of n target components f_i ($i = 1, \dots, n$) is

$$\bar{f}(\bar{X}) = (f_1(\bar{X}), f_2(\bar{X}), \dots, f_n(\bar{X})), \quad (5)$$

where $\bar{X}_u \in U$ is the decision variable. If \bar{X}_u is the Pareto optimal solution, it needs to be satisfied: only if $\bar{X}_v \in U$, there is no decision variable $v = f(\bar{X}_v) = (v_1, v_2, \dots, v_n)$, dominating $u = f(\bar{X}_u) = (u_1, u_2, \dots, u_n)$.

There are different approaches to solving multiobjective optimization problems [24, 25], for example, aggregating, population based non-Pareto and Pareto-based techniques. Vector evaluated genetic algorithm (VEGA) is a technique in the population based non-Pareto approach in which different subpopulations are used for the different objectives. Multiple objective GA (MOGA), nondominated sorting GA (NSGA), and niched Pareto GA (NPGA) constitute a number of techniques under the Pareto-based nonelitist approaches [25]. NSGA-II [26], SPEA [27], and SPEA2 [28] are some recently developed multiobjective elitist techniques.

As a multiobjective genetic algorithm, NSGA-II algorithm is a mature multiobjective elite selection algorithm. Compared with the NSGA, the NSGA-II has been improved in three aspects: (1) when constructing the Pareto optimal solution set, the time complexity of the algorithm is reduced from $O(mN^3)$ to $O(mN^2)$ by adopting a new rank-based fast nondominated sorting method. (2) The elitist reservation mechanism is proposed. After selection, offsprings from breeding individuals compete with their parents to produce the next generation. The new optimal individual reservation mechanism can not only improve the performance of multiobjective evolutionary algorithm (MOEA) but also effectively prevent the loss of the optimal solution and improve the overall evolutionary level of the population. (3) In order to calibrate the fitness values of different elements at the same level after rapid nondominated sorting and to make the individuals in the Pareto frontier extend to the front of the entire frontier Pareto, the crowded distance comparison operator is used instead of the original fitness sharing method.

The present paper uses NSGA-II as the underlying multiobjective algorithm for developing the proposed fuzzy clustering method.

3. Dynamic Fuzzy Clustering Method Based on Adaptive NSGA-II

3.1. Chromosome Representation. In general, there are two kinds of chromosome coding schemes to solve the clustering problem by using genetic algorithm: (1) numerical coding based on the clustering center; (2) encoding based on the partition matrix U [29]. Since the genetic operator in this

paper uses the variable chromosome length operation, the first chromosome coding scheme is adopted.

Definition Q_i denotes a chromosome that represents c_i ($1 < c_i < \sqrt{n}$) cluster centers with d dimensional attribute space. The coding form can be expressed as

$$Q_i = [g_1^1, g_2^1, \dots, g_d^1, g_1^2, g_2^2, \dots, g_d^2, \dots, g_1^{c_i}, g_2^{c_i}, \dots, g_d^{c_i}]. \quad (6)$$

Figure 1 shows an example of a chromosome comprising five centers $\{C1, C2, C3, C4, C5\}$ in two dimensions.

It uses the sequence form of real value to describe the chromosome, avoids the complex encoding form of binary form, and can display the practical significance of the representation more intuitively.

3.2. Population Initialization. The selection of initial cluster centers will have a great impact on the final clustering results. However, due to the crossover operator that dynamically changes the chromosome length, the fixed initial cluster centers are not conducive to maintaining the diversity of the population. Therefore, this paper uses the most common method of random given initial cluster centers to initialize the population.

Note that, for the sample datasets, the range of d attribute values may not be the same, which can have a significant impact on the calculation of the NSGA-II algorithm. Therefore, it is necessary to standardize the sample data set, Max-Min Normalization first needs to be performed on the sample dataset to reduce the possible error. The Max-Min Normalization is defined as follows:

$$x_{\text{normalization}} = \frac{x - \text{Min}}{\text{Max} - \text{Min}}. \quad (7)$$

3.3. Selection of Fitness Function. The performance of multiobjective optimization is highly dependent on the choice of objective function, which can produce good results by reasonably selecting the objective function. The selection of objective functions should be such so that they can balance each other critically and are possibly contradictory in nature. Contradiction in the objective functions is beneficial since it guides to global optimum solution. It also ensures that no single clustering objective is optimized leaving the other probable significant objectives unnoticed.

In this paper, two kinds of fitness functions, DB Index and Index I , are used as objective functions for NSGA-II algorithm. The two fitness functions are described in detail below.

3.3.1. Davies-Bouldin (DB) Index. DB Index [30] is a commonly used cluster validity index. This index is the ratio function of the sum of within-cluster scatter to between-cluster separation.

Define the scatter of the i th class as

$$S_i = \left\{ \frac{1}{|C_i|} \sum_{j=1}^{C_i} \|x_j - z_i\|^q \right\}^{1/q}, \quad (8)$$

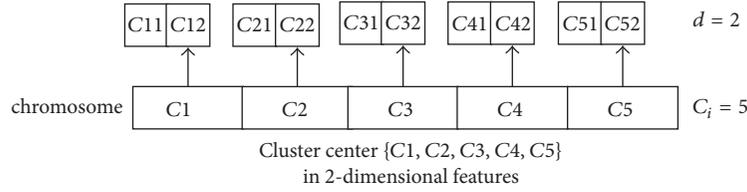


FIGURE 1: Chromosome representation.

where x_j denotes the data point in the i th class and z_i denotes the center of the i th class; $|C_i|$ represents the number of data points in the i th class; q is an index value. The distance between cluster center z_i and z_j , is defined as $M_{ij} = \|z_i - z_j\|$. The similarity between i th cluster and j th cluster is defined as

$$R_{ij} = \max_{i,i \neq j} \left\{ \frac{S_i + S_j}{M_{ij}} \right\}. \quad (9)$$

The Davies-Bouldin (DB) index is then defined as

$$DB = \frac{1}{N} \sum_{i=1}^N R_{ij}. \quad (10)$$

The objective is to minimize the DB index for achieving proper clustering.

3.3.2. Index I. Index I [31] is another commonly used cluster validity index.

$$I(c) = \left(\frac{1}{c} \times \frac{E_1}{E_c} \times D_c \right)^p, \quad (11)$$

where c is the number of clusters. Here, E_c stand for within-cluster scatter, defined as

$$E_c = \sum_{i=1}^c \sum_{j=1}^n u_{ij} \|x_j - z_i\|, \quad (12)$$

D_c stand for between-cluster separation, defined as

$$D_c = \max_{i,j=1}^c \|z_i - z_j\|, \quad (13)$$

E_1 and p are correlation coefficients. The power p is used to control the contrast between the different cluster configurations, in general, $p \geq 1$. In this article, we have taken $p = 1$. E_1 is a constant for a given dataset, normalized to avoid the minimum value of the indicator. The value of K for which $I(c)$ is maximized is considered to be the correct number of clusters.

The goal in this paper is to minimize DB and $1/I(c)$ simultaneously. At the same time, pay attention to adjust the correlation coefficient E_1 in $I(c)$. By adjusting the parameters, the values of DB and $1/I(c)$ are in the same order of magnitude, avoiding the selection error caused by too large a target value. At the same time, in the use of the algorithm, it can be found that, with the increase of the number of clusters c , the value of $1/I(c)$ begins to decrease, and the value of DB begins to increase, which conforms to the conflicting requirements of the two objective functions mentioned earlier.

3.4. Genetic Manipulation

3.4.1. Selection. The two individuals are randomly selected to play a tournament and the winner is selected by the crowded comparison operator. This operator takes into account two attributes of the nondominant rank and the crowded distance. If two individuals are at different levels, the lower level is preferred. If both individuals are at the same level, choose a solution that has less crowded region.

3.4.2. Crossover. After selection, the selected chromosomes are placed in the mating pool. The performance of crossover operator will determine the performance of genetic manipulation to a great extent. Because of the variable-length encoding used in chromosome coding, the conventional one-point crossover approach does not apply to the current situation. In this paper, the following two crossover methods are used to perform crossover operation with the same probability.

(1) *Based on the Nearest Neighbor Matching Crossover Operation.* Let two parents $S_1 = [a_1, a_2, \dots, a_{c_1}]$ and $S_2 = [b_1, b_2, \dots, b_{c_2}]$ denote parent solutions with c_1 and c_2 cluster centers. Assume that we are in the case of $c_1 \leq c_2$, select the gene string $a_i = (a_i^1, a_i^2, \dots, a_i^d)$ in turn, which represents each cluster center in S_1 , and select the nearest distance a_i string $b_i = (b_i^1, b_i^2, \dots, b_i^d)$ from S_2 to match them. Already paired gene strings are no longer involved in pairing. Reordering the previous c_1 gene strings in S_2 , choose a point randomly from within $0 \sim c_1 \times d$. For S_1 and S_2 , traditional crossover operations are used to generate new offspring S_3 and S_4 .

By using this crossover method, the offsprings maintain the same number of cluster centers as their parents and maintain the stability of the population. Using gene rearrangements before crossover can make the different chromosomes have the most similar cluster centers in the same position, avoid the generation of poor offspring when crossing and then lead to population degradation. The crossover operation can be illustrated in Figure 2.

(2) *Based on the Truncation and Stitching Cross Operation.* Different from the first method, the crossover operation based on truncation and stitching will produce the offsprings which are different from the number of the parent cluster centers, so as to maintain the diversity of the population. In this crossover operation, the string representing each cluster center is indivisible and can only be crossed at different gene strings.

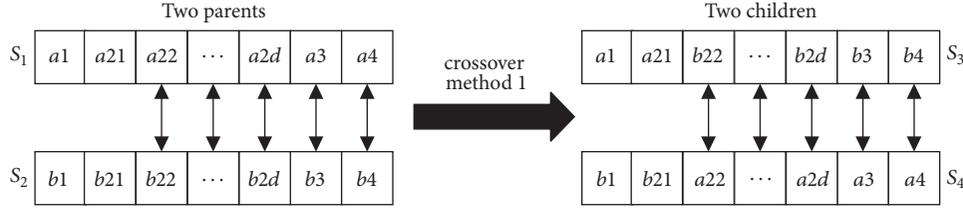


FIGURE 2: Crossover method 1.

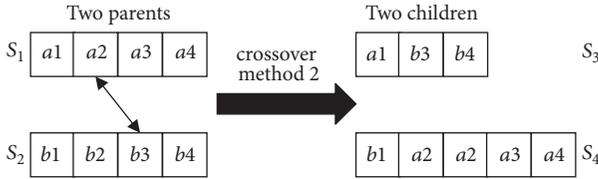


FIGURE 3: Crossover method 2.

The operation is described as follows: S_1 and S_2 are two parent individuals, where

$$\begin{aligned} S_1 (c_1 \times d) &= (a_1, a_2, \dots, a_{t_1} \mid a_{t_1+1}, \dots, a_{c_1}), \\ S_2 (c_2 \times d) &= (b_1, b_2, \dots, b_{t_2} \mid b_{t_2+1}, \dots, b_{c_2}). \end{aligned} \quad (14)$$

Suppose that the intersection points of S_1 and S_2 are t_1 and t_2 , respectively. The offsprings S_3 and S_4 generated after crossing can be expressed as

$$\begin{aligned} S_3 ((c_2 + t_1 - t_2) \times d) &= (a_1, a_2, \dots, a_{t_1} \mid b_{t_2+1}, \dots, b_{c_2}), \\ S_4 ((c_1 + t_2 - t_1) \times d) &= (b_1, b_2, \dots, b_{t_2} \mid a_{t_1+1}, \dots, a_{c_1}). \end{aligned} \quad (15)$$

The number of cluster centers represented by S_3 and S_4 is $(c_2 + t_1 - t_2)$ and $(c_1 + t_2 - t_1)$, respectively. The crossover operation can be illustrated in Figure 3.

3.4.3. Mutation. Individuals are mutated according to gene loci, and random variation is usually made according to the variation probability P_m . If the chromosome is selected for mutation, the location of the mutated gene will be selected randomly. After mutating, the floating point number at the gene site is replaced by another uniform random number.

3.4.4. Adaptive Operation. By using the adaptive strategy of crossover probability P_c and mutation probability P_m , the two parameters can be automatically changed according to the fitness of the current population. For the whole population, when the fitness value of the population tends to be consistent or tends to local optimum, the P_c and P_m increase appropriately; when the fitness value is dispersed, P_c and P_m are appropriately reduced. For an individual in a population, when its fitness is higher than the average fitness of the population, the lower P_c and P_m values make it more likely

to enter the next generation; when the current fitness value is lower than the average fitness value, the higher P_c and P_m values will be given to make it more likely to be eliminated. Thus, the adaptive strategy can provide the best P_c and P_m for the solution [32]. P_c and P_m are calculated as follows:

$$\begin{aligned} P_c &= \begin{cases} \frac{(p_{c1} - 0.6)(f_{\max} - f^*)}{f_{\max} - f_{\text{mean}}} & f^* \geq f_{\text{mean}} \\ p_{c1} & f^* < f_{\text{mean}} \end{cases} \\ P_m &= \begin{cases} \frac{(p_{m1} - 0.001)(f_{\max} - f)}{f_{\max} - f_{\text{mean}}} & f \geq f_{\text{mean}} \\ p_{m1} & f < f_{\text{mean}} \end{cases} \end{aligned} \quad (16)$$

where f^* is the larger fitness value of two individuals to be cross-operated, f is the fitness value of the current individual, f_{\max} is the maximum fitness value of the current generation, and f_{mean} is the average fitness of the current generation, $p_{c1} = 0.9$, $p_{m1} = 0.1$. It should be noted that the fitness value mentioned here is the sum of two objective function values. When an individual's fitness value is the maximum fitness value of a contemporary population, we set its P_c and P_m to 0.6 and 0.001, respectively.

3.4.5. Selecting a Solution from the Nondominated Set. In this paper, the majority voting method is used to determine the number of clusters c . That is to say, in the dominant set, the number of occurrences of a cluster in the whole dominating cluster is more than 50% of the total number of occurrences, and the same number continuously appears more than 5 generations; we think it is the optimal cluster number. If the algorithm still cannot choose the optimal cluster number at the specified maximum number of iterations, c corresponding to the best individual in the final generation is taken as the optimal cluster number.

3.4.6. Determine the Final Clustering Result. After the number of clusters is determined, all the individuals whose population number is equal to c are selected to form a new population for clustering. The method is to use a combination of Canonical Genetic Algorithm (CGA) and FCM algorithms. The crossover operation used here only uses the nearest neighbor matching cross operation mentioned above, so it will not change the number of clusters. By combining the global optimization algorithm with FCM, this can effectively overcome the problem that the FCM algorithm can only obtain the local optimal solution. Finally, the algorithm will terminate after the objective function value J_{FCM} no longer

changes obviously, and the obtained result is the optimal clustering result.

At this point, the relevant concepts of the algorithm have been described. Algorithm 1 shows the steps of the ADNSGA2-FCM algorithm.

3.5. Time Complexity. The ADNSGA2-FCM algorithm has a worse-case $O(GPNc_{\max}d)$ time complexity, where G denotes the number of generations, P is population size, N is the size of data, c_{\max} is maximum number of clusters, and d are data dimensions.

- (1) In the initial stage of population, the time required is $c_{\max}d$ and each string contains d dimensional features until the population size P is full. Therefore, this construction requires $O(Pc_{\max}d)$.
- (2) In FCM clustering for each individual, suppose the number of data in the current data set is N , both procedures of membership assignment and updating of center values take $Nc_{\max}d$ time. For the population, the time complexity is $O(PNc_{\max}d)$.
- (3) The time complexity of the two objective functions DB Index and Index I are both $O(N)$.
- (4) The time complexity of each execution of crossover and mutation operators is $O(Pc_{\max}d)$.
- (5) The nondominated 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 nondominated 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 nondominated solution, $Nc_{\max}d$ time is required to assign label for every data point. To select the best solution from P nondominated solutions, this yields $O(PNc_{\max}d)$ time.

It can be seen that the time complexity of the algorithm is worse, and the complexity of each generation in the worst case is $O(PNc_{\max}d)$. Assuming that the algorithm runs G generation, the time complexity is $O(GPNc_{\max}d)$.

4. Experiment Study

In this paper, for the purpose of verifying the performance of the method proposed in this paper (ADNSGA2-FCM), some clustering algorithms are chosen for extensive comparative analysis. There are two kinds of soft subspace clustering algorithms ESSC [33] and MOEASSC [34]. MOEASSC is a multiobjective method and ESSC is a single-objective one. There are three kernel-based attribute weighting algorithms, VKCM-K-LP [35], VKFCM-K-LP [36], and MOKCW [37]. The MOKCW method is a multiobjective method; VKCM-K-LP and VKFCM-K-LP are single-objective methods. The VKCM-K-LP method is crisp version of clustering method, and the VKFCM-K-LP method is a fuzzy clustering method. The NSGA-II-FCM method is the nonadaptive version of ADNSGA2-FCM and used fixed parameters.

TABLE 1: The characters of datasets.

Datasets	C	d	n
Square 1	4	2	1000
Square 4	4	2	1000
Sizes 5	4	2	1000
Iris	3	4	150
Wine	3	13	178
Newthyroid	3	5	215
Vertebral	3	6	310
Image	7	16	2310
Abalone	3	8	4177

TABLE 2: Parameter settings for the ADNSGA2-FCM algorithm.

Parameter	Setting
Population size	40
Max number of generations	300
C_{\min}	2
C_{\max}	\sqrt{n}
P_{c1}	0.9
P_{m1}	0.1

TABLE 3: Parameter settings for other algorithms.

Algorithm	Parameter setting
ESSC	$m = \frac{\min(n, D-1)}{\min(n, D-1)-2}, \gamma = 100, \eta = 0.1$
VKCM-K-LP	$m = 2$
VKFCM-K-LP	$m = 2$
MOEASSC	$P_c = 0.5, P_m = \frac{K}{D}, \sigma = 10^{-7}$
MOKCW	$P_m = 0.1, \alpha = 0.1$
NSGA-II-FCM	$P_c = 0.9, P_m = 0.05$

4.1. Datasets and Parameter Setting. For the purpose of comparison, there are two groups of data sets, artificial and real-life data sets. The three artificial data sets are Square 1, Square 4, and Sizes 5 from [20]. The six real-life data sets are obtained from the UCI Machine Learning Repository [38], namely, Iris, Wine, Newthyroid, Vertebral, Image, and Abalone. As shown in Table 1 the data sets considered are briefly described, where C is the true number of classes and d and n are, respectively, the number of features and objects. For most SSC algorithms, the experiments are conducted on the data sets standardized into the interval $[0, 1]$, which can alleviate the uneven impact of different attributes' ranges on updating the weights. Therefore, the standardization is based on the minimum and maximum values of each attribute.

The parameters of the ADNSGA2-FCM algorithm are set as shown in Table 2, the parameters of other algorithms are set as shown in Table 3.

4.2. Experiment Result and Analysis. In the first experiment, the above nine data sets (Square 1, Square 4, Sizes 5, Iris, Wine, Newthyroid, Vertebral, Image, and Abalone) and the

input: Dataset

- (1) Initialize parameters FCM and NSGA-II including population size Pop, itermax, ε , m , p_{c1} , p_{m1} , Tmax.
- (2) Random to select initial number of clusters c and random to generate initial cluster centers to create a initialize population $P(0)$.
- (3) Decode each individual to obtain the cluster centers, and calculate the membership degrees U using Eq. (3).
- (4) Calculate new cluster centers V of each individual using Eq. (4) based on U and V
- (5) Calculate the J_{FCM} of each individual using Eq. (2) based on U and V
- (6) Calculate fitness values f_1 and f_2 of each individual using Eq. (10) and (11). Calculate $f = f_1 + f_2$, store f_{max} and f_{mean} at each iteration.
- (7) Non-dominated sorting and crowding distance operation for population.
- (8) Using the crowded comparison operator to select.
- (9) Calculate P_c and P_m using Eq. (16).
- (10) Generate offspring using genetic operation.
- (11) Recombination current generation and offspring to select next generation using elitism operation.
- (12) Using majority voting technique to determine the number of cluster c .
- (13) If the number c satisfies the selection condition, go to step (14); else go to step (3).
- (14) Find all chromosomes whose cluster numbers are equal to c from the population in step (11). The new population is composed of these chromosomes.
- (15) Decode each individual to obtain the cluster centers, and calculate the membership degrees U using Eq. (3).
- (16) Calculate new cluster centers V of each individual using Eq. (4) based on U and V
- (17) Calculate the J_{FCM} of each individual using Eq. (2) based on U and V
- (18) Calculate fitness values f_1 and f_2 of each individual using Eq. (10) and (11). Calculate $f = f_1 + f_2$, store f_{max} and f_{mean} at each iteration.
- (19) Non-dominated sorting and crowding distance operation for population.
- (20) Using the crowded comparison operator to select.
- (21) Calculate P_c and P_m using Eq. (16).
- (22) Generate offspring using genetic operation.
- (23) Recombination current generation and offspring to select next generation using elitism operation.
- (24) If ADNSGA2-FCM has not met the stopping criterion ($|\overline{f(t+1)} - \overline{f(t)}| < \varepsilon$ and $t < Tmax$), else $t = t + 1$ and go to step (14).
- (25) **Return** the best individual ($\min J_{FCM}(t)$).

ALGORITHM 1: ADNSGA2-FCM.

adjusted Rand Index (ARI) [39] are used here to evaluate the clustering quality of ADNSGA2-FCM algorithm, which is to be maximized. Table 4 summarizes the result obtained from ADNSGA2-FCM. The real number of clusters can be obtained by the ADNSGA2-FCM algorithm in nine datasets. From the ARI value, the ADNSGA2-FCM algorithm has a big difference to the nine data sets. This is mainly because the ADNSGA2-FCM algorithm uses the fuzzy c -means (FCM) method as the clustering method, but the FCM algorithm is suitable for allocating the data of the spherical clusters. Therefore, for six datasets of Square 1, Square 4, Sizes 5, Iris, Wine, and Newthyroid, the effect is better, and the effect of the other three data sets is poor. It also can be observed from Table 4 that, in all data sets, the optimal clustering result is obtained by the algorithm.

Figures 4–6 compare the results from the three synthetic data sets (Square 1, Square 4, and Sizes 5) of data partitioning obtained by ADNSGA2-FCM with center markings and the true data partitioning. The algorithm performs well under the well-separated structures of Square 1 (Figure 4) and Square 4 (Figure 5) as well as the unequally sized clusters of Sizes 5

TABLE 4: Adjusted Rand index (ARI) and the number of the clusters (C) obtained by ADNSGA2-FCM.

Dataset	Actual C	ADNSGA2-FCM	
		C	ARI
Square 1	4	4	0.971
Square 4	4	4	0.830
Sizes 5	4	4	0.838
Iris	3	3	0.802
Wine	3	3	0.865
Newthyroid	3	3	0.843
Vertebral	3	3	0.316
Image	7	7	0.490
Abalone	3	3	0.160

(Figure 6). The overlapping and unequally sized characteristic causes more misclassification of the data points which are at the borderline between clusters.

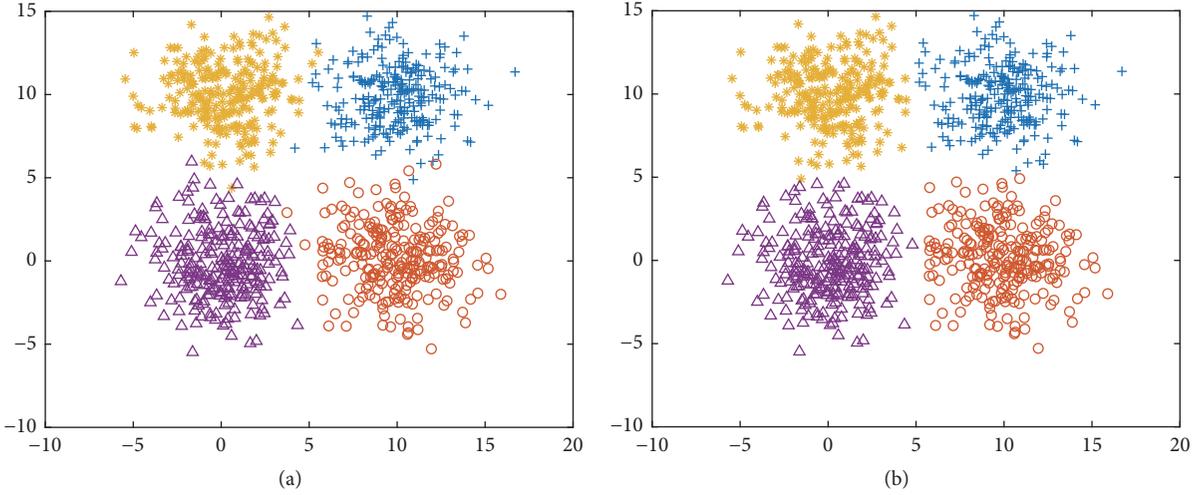


FIGURE 4: Square 1 ($C = 4$): (a) true solution and (b) data partition using ADNSGA2-FCM (ARI = 0.971).

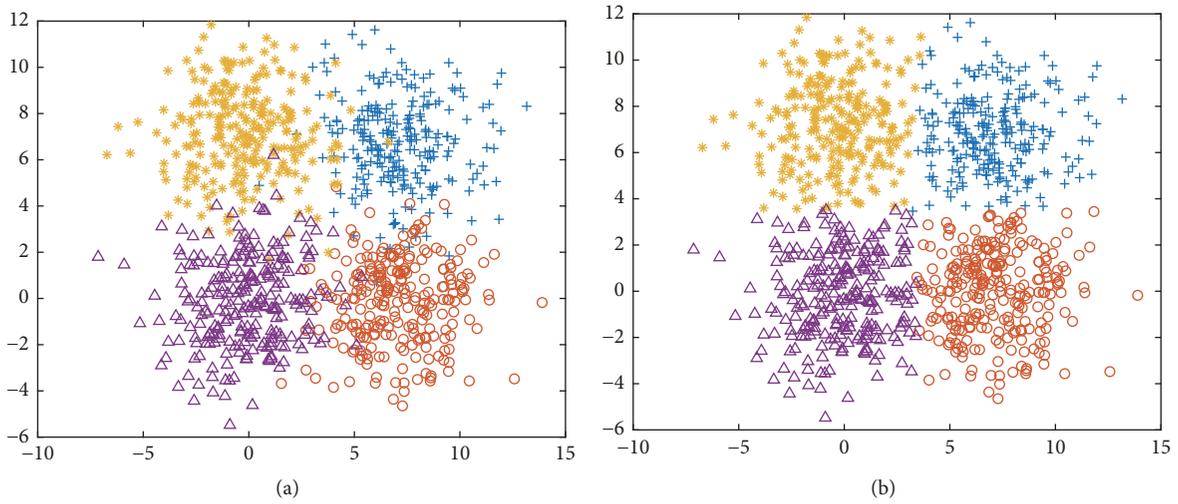


FIGURE 5: Square 4 ($C = 4$): (a) true solution and (b) data partition using ADNSGA2-FCM (ARI = 0.830).

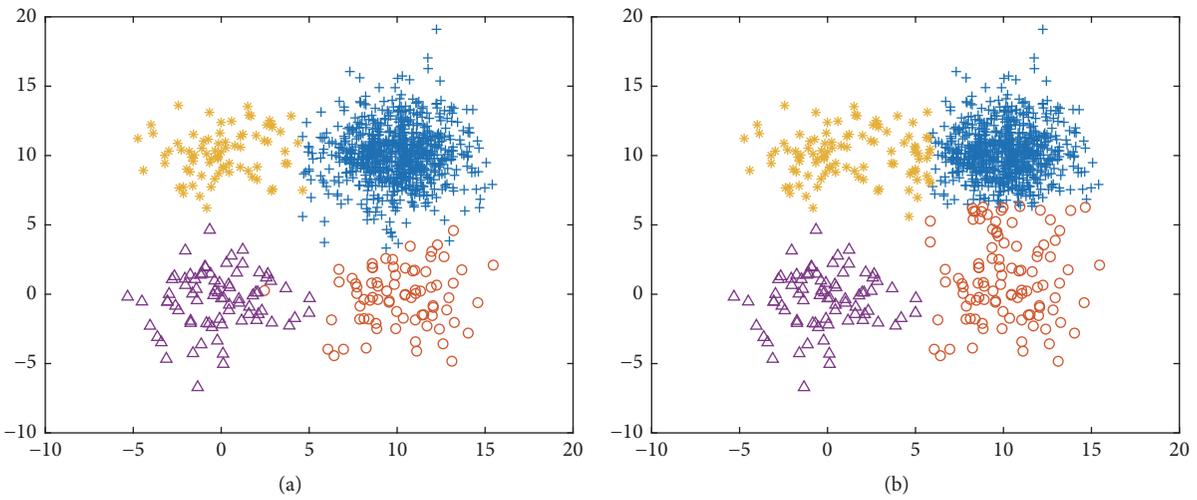


FIGURE 6: Sizes 5 ($C = 4$): (a) true solution and (b) data partition using ADNSGA2-FCM (ARI = 0.838).

In order to evaluate the performance of the clustering result of seven algorithms, three well-known external CVIs accuracy (Acc), rand index (RI) [40], and normalized mutual information (NMI) [33] are adopted here. They all take their values from the interval $[0, 1]$, in which 1 means the best match between the result and the true partition, whereas 0 means the worst result. In this experiment, all algorithms are executed 30 times independently, and their performances are compared in terms of the best case of Acc, RI, and NMI shown in Table 5. Among them, the best result is expressed in bold.

It can be firstly observed from Table 5 that, in all data sets, the optimal clustering result is obtained by the multiobjective algorithm. This result can prove that the multiobjective clustering algorithm has some advantages compared with the single-objective clustering algorithm. For most data sets, the ADNSGA-FCM algorithm proposed in this paper can obtain the best results.

For the data set Iris and Vertebral, the kernel-based multi-objective clustering algorithm MOKCW can achieve the best results. Compared with MOKCW algorithm and VKCM-K-LP algorithm, the two results are similar, and the result of ADNSGA2-FCM is worse. For the Vertebral data set, the ADNSGA2-FCM algorithm obtains the best Acc value. For the Vertebral data set, the ADNSGA2-FCM algorithm obtains the best Acc value. For the Wine data set, the MOEASSC algorithm can achieve the best effect, and the ADNSGA-FCM effect is very close to it. It shows that two kinds of multiobjective clustering methods based on evolutionary computation can obtain the best global results on Wine datasets. In the three datasets of Newthyroid, Image, and Abalone, ADNSGA2-FCM proposed in this paper has obvious advantages over other algorithms.

From Table 5 we can also see that the effect of the ADNSGA2-FCM algorithm using the adaptive mechanism is significantly better than the NSGA-II-FCM algorithm. Except that the two indicators are the same as ADNSGA2-FCM algorithm, the other indicators of NSGA-II-FCM are quite different from ADNSGA2-FCM algorithm. More obviously, in the Image dataset, the NSGA-II-FCM algorithm cannot obtain the correct number of clustering by 30 independent executions. Through careful analysis, we conclude that the adaptive mechanism makes the ADNSGA2-FCM algorithm finally find the correct number of clusters. This is mainly due to the fact that the adaptive mechanism effectively controls the speed of crossover and mutation of genetic algorithms. Because the NSGA-II-FCM algorithm does not adopt the adaptive mechanism, it leads to its premature convergence to the local optimal solution, which leads to the final clustering number being wrong. Looking at the other five data sets. Because the parameter values of NSGA-II-FCM are fixed, this leads to the fact that the algorithm does not make full use of data information in the optimization process, and the rate of convergence is too fast. Although the correct number of clusters was eventually found, the clustering effect was poor. With the increasing number of data and attributes in the data set, this trend is even more obvious. From this set of experiments, we can see that using an adaptive mechanism does improve the clustering effect.

From this result, it is easy to think that because the clustering problem lacks prior knowledge of the data set, and the genetic algorithm is also a random search algorithm, it is

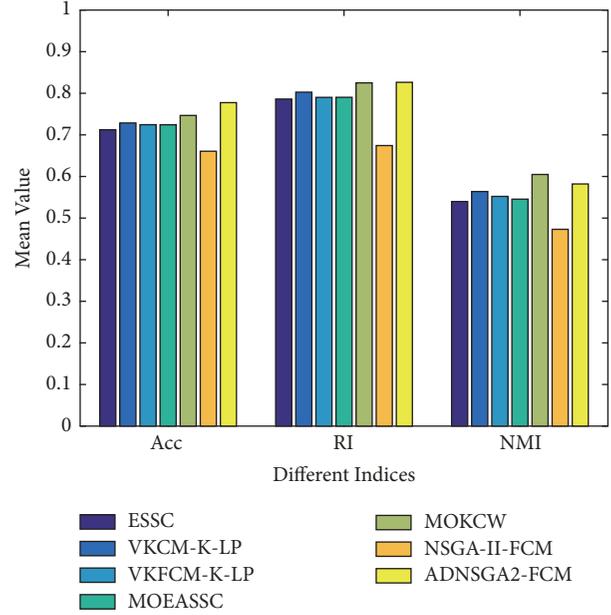


FIGURE 7: Mean values of Acc, RI, and NMI using different algorithms in the 6 real-life datasets.

difficult to give suitable crossover probability and mutation probability. However, adopting an adaptive mechanism here can avoid giving fixed global parameters directly.

From the above analysis, we believe that the adoption of an adaptive mechanism is effective.

Table 6 shows the average performance rankings of all algorithms on the 6 datasets regarding Acc, RI, and NMI computed from Table 5, making a more evident comparison. From Table 6, we can see that the ADNSGA2-FCM algorithm proposed in this paper ranks first in Acc and RI, ranking second on NMI, mainly due to the fact that NMI indicators are not consistent with Acc and RI indicators. On Acc, the ADNSGA2-FCM algorithm has a greater advantage than the second algorithm. On RI, the ADNSGA2-FCM algorithm performs slightly better than the second algorithm. In NMI, ADNSGA2-FCM algorithm is worse than MOKCW algorithm, but it is not much different. It shows that the ADNSGA2-FCM algorithm has some advantages over the other 6 algorithms on the three indexes of Acc, RI, and NMI, and better clustering results can be obtained.

Figure 7 shows the histogram of mean values of the three indices in comparison for different algorithms. As can be observed from Tables 5 and 6 and Figure 7, the performance of our proposed method has obvious advantages in the Acc index and has a slight advantage in the RI index, which is not as good as the MOKCW algorithm in the NMI index.

The final Pareto optimal front obtained by ADNSGA2-FCM clustering technique on the real-life data sets, Iris, Wine, Newthyroid, Vertebral, Image, and Abalone is illustrated in Figures 8–10, respectively.

5. Conclusion

This paper presents a fuzzy clustering method based on multi-objective genetic algorithm. The ADNSGA2-FCM algorithm

TABLE 5: The result of all algorithms on Acc, RI and NMI.

DataSet	ESSC	VKCM-K-LP	VKFCM-K-LP	MOEASSC	MOKCW	NSGA-II-FCM	ADNSGA2-FCM
Iris							
Acc	0.839	0.955	0.947	0.900	0.960	0.920	0.927
RI	0.844	0.944	0.934	0.886	0.950	0.906	0.912
NMI	0.716	0.858	0.832	0.778	0.864	0.777	0.790
Wine							
Acc	0.955	0.907	0.938	0.956	0.921	0.944	0.955
RI	0.940	0.883	0.919	0.942	0.897	0.925	0.940
NMI	0.847	0.733	0.796	0.854	0.768	0.823	0.847
Newthyroid							
Acc	0.893	0.864	0.809	0.888	0.946	0.953	0.954
RI	0.826	0.834	0.737	0.818	0.913	0.922	0.922
NMI	0.620	0.655	0.541	0.603	0.772	0.786	0.787
Vertebral							
Acc	0.474	0.510	0.517	0.479	0.506	0.655	0.655
RI	0.633	0.667	0.661	0.623	0.697	0.683	0.685
NMI	0.269	0.350	0.330	0.249	0.431	0.286	0.302
Image							
Acc	0.606	0.624	0.623	0.620	0.635	0	0.658
RI	0.862	0.869	0.871	0.863	0.874	0	0.874
NMI	0.624	0.625	0.649	0.628	0.631	0	0.598
Abalone							
Acc	0.507	0.512	0.513	0.503	0.512	0.492	0.516
RI	0.612	0.619	0.619	0.610	0.619	0.610	0.625
NMI	0.163	0.162	0.165	0.161	0.163	0.166	0.168

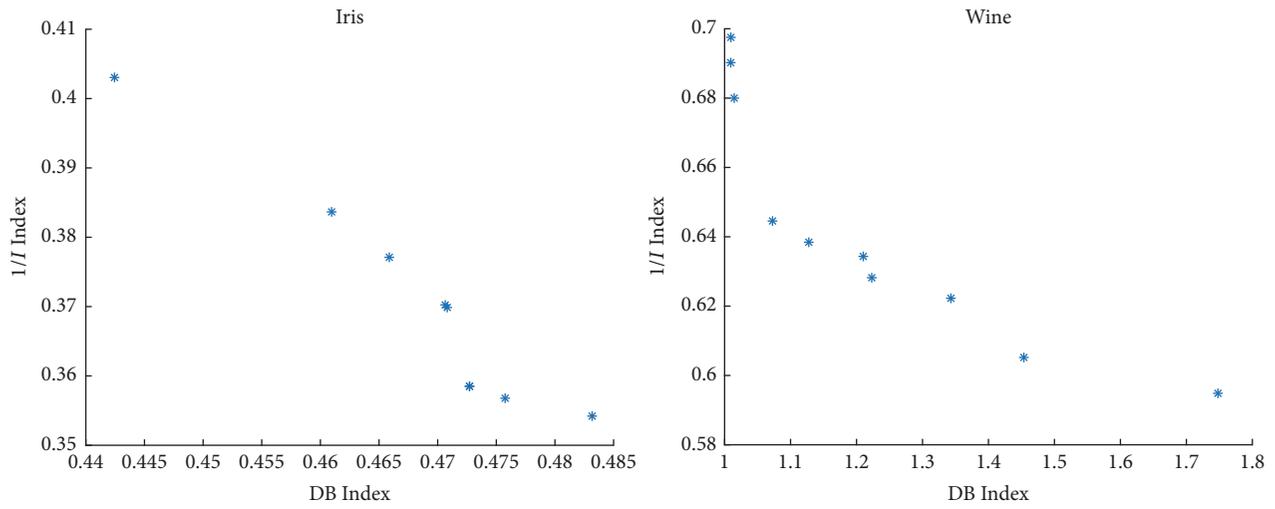


FIGURE 8: Pareto optimal front obtained by the proposed ADNSGA2-FCM algorithm for Iris data set and Wine data set.

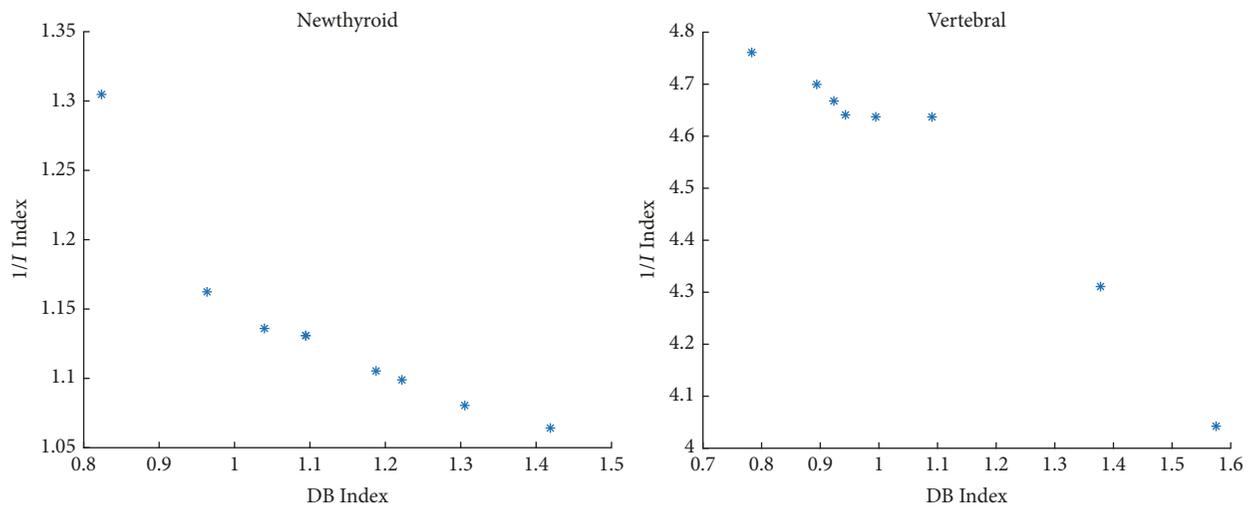


FIGURE 9: Pareto optimal front obtained by the proposed ADNSGA2-FCM algorithm for Newthyroid data set and Vertebral data set.

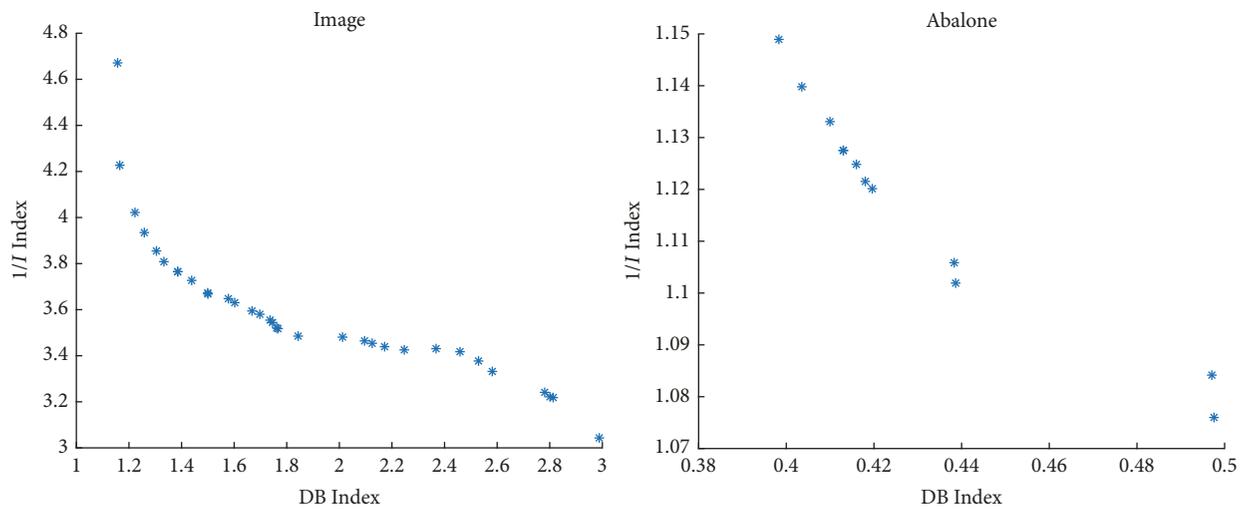


FIGURE 10: Pareto optimal front obtained by the proposed ADNSGA2-FCM algorithm for Image data set and Abalone data set.

TABLE 6: Average performance rankings of different algorithms on all datasets regarding Acc, RI and NMI.

Algorithms	Acc	RI	NMI
ESSC	5.2 (7)	5.2 (6)	4.8 (6)
VKCM-K-LP	4.2 (4)	3.8 (3)	4.2 (4)
VKFCM-K-LP	4.0 (3)	4.2 (4)	3.7 (3)
MOEASSC	4.8 (6)	5.2 (6)	4.8 (6)
MOKCW	3.3 (2)	2.3 (2)	2.8 (1)
NSGA-II-FCM	4.3 (5)	4.5 (5)	4.3 (5)
ADNSGA2-FCM	1.7 (1)	1.8 (1)	3.0 (2)

was developed to solve the clustering problem by combining the fuzzy clustering algorithm (FCM) with the multiobjective genetic algorithm (NSGA-II) and introducing an adaptive mechanism. In this paper, NSGA-II algorithm uses two cluster validity indexes of Index I and DB Index as its objective function, so as to control multiobjective optimization. The algorithm does not need to give the number of clusters in advance. After the number of initial clusters and the center coordinates are given randomly, the optimal solution set is found by the multiobjective evolutionary algorithm. After determining the optimal number of clusters by majority vote method, the J_m value is continuously optimized through the combination of Canonical Genetic Algorithm and FCM, and finally the best clustering result is obtained.

In addition to the basic framework of multiobjective genetic algorithm, the appropriate objective function is also one of the success factors of ADNSGA2-FCM algorithm. This paper does not use a single cluster evaluation index but uses two comprehensive evaluation indicators. These two indexes take into account both the within-cluster scatter and the between-cluster separation. The experimental results show that the multiobjective clustering method is better than the single-objective clustering method, and the better clustering results can be obtained by choosing a reasonable objective function.

Although the ADNSGA2-FCM algorithm performs well, it also has some inherent problems. Since the algorithm adopts the NSGA-II framework, the multiobjective genetic algorithm can only compromise among multiple objective functions, so the method can only approach the real Pareto front. Because the NAGA-II algorithm is a kind of genetic algorithm and there is strong randomness, we can find the optimal solution through the randomness, or we can not find the optimal solution through the randomness. So we cannot guarantee the optimal clustering solution is absolutely right.

In the following work, we hope to improve the selection and clustering accuracy of the optimal clustering results.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Fund of China (Grant no. 71471060) and the Fundamental Research

Funds for the Central Universities (Grant nos. 2017XS135, 2018QN096).

References

- [1] J. C. Bezdek, R. Ehrlich, and W. Full, "FCM: the fuzzy c-means clustering algorithm," *Computers & Geosciences*, vol. 10, no. 2-3, pp. 191-203, 1984.
- [2] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, p. 14, University of California Press, Berkeley, Calif, USA, 1967.
- [3] E. R. Hruschka, R. J. G. B. Campello, A. A. Freitas, and A. C. P. L. F. de Carvalho, "A survey of evolutionary algorithms for clustering," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 39, no. 2, pp. 133-155, 2009.
- [4] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, Wiley-Interscience, New York, NY, USA, 2nd edition, 2001.
- [5] S. Bandyopadhyay, "Simulated annealing using a Reversible Jump Markov Chain Monte Carlo algorithm for fuzzy clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 4, pp. 479-490, 2005.
- [6] A. Hatamlou and M. Hatamlou, "PSOHS: An efficient two-stage approach for data clustering," *Memetic Computing*, vol. 5, no. 2, pp. 155-161, 2013.
- [7] R. J. Kuo, Y. J. Syu, Z.-Y. Chen, and F. C. Tien, "Integration of particle swarm optimization and genetic algorithm for dynamic clustering," *Information Sciences*, vol. 195, pp. 124-140, 2012.
- [8] T. M. Silva Filho, B. A. Pimentel, R. M. C. R. Souza, and A. L. I. Oliveira, "Hybrid methods for fuzzy clustering based on fuzzy c-means and improved particle swarm optimization," *Expert Systems with Applications*, vol. 42, no. 17-18, pp. 6315-6328, 2015.
- [9] U. Maulik and S. Bandyopadhyay, "Genetic algorithm-based clustering technique," *Pattern Recognition*, vol. 33, no. 9, pp. 1455-1465, 2000.
- [10] U. Maulik and S. Bandyopadhyay, "Fuzzy partitioning using a real-coded variable-length genetic algorithm for pixel classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, no. 5, pp. 1075-1081, 2003.
- [11] C. D. Nguyen and K. J. Cios, "GAKREM: a novel hybrid clustering algorithm," *Information Sciences*, vol. 178, no. 22, pp. 4205-4227, 2008.
- [12] A. Ye and Y. Jin, "A Fuzzy C-Means Clustering Algorithm Based on Improved Quantum Genetic Algorithm," *Journal of Database Theory and Application*, vol. 9, no. 1, pp. 227-236, 2016.
- [13] G. Garai and B. B. Chaudhuri, "A novel genetic algorithm for automatic clustering," *Pattern Recognition Letters*, vol. 25, no. 2, pp. 173-187, 2004.
- [14] D. X. Chang, X. D. Zhang, and C. W. Zheng, "A genetic algorithm with gene rearrangement for K-means clustering," *Pattern Recognition*, vol. 42, no. 7, pp. 1210-1222, 2009.
- [15] S. Saha and S. Bandyopadhyay, "A fuzzy genetic clustering technique using a new symmetry based distance for automatic evolution of clusters," in *Proceedings of the International Conference on Computing: Theory and Applications, ICCTA 2007*, pp. 309-313, ind, March 2007.
- [16] S. Bandyopadhyay and S. Saha, "A point symmetry-based clustering technique for automatic evolution of clusters," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 11, pp. 1441-1457, 2008.

- [17] Y. Zhang, D.-W. Gong, and J. Cheng, "Multi-objective particle swarm optimization approach for cost-based feature selection in classification," *IEEE Transactions on Computational Biology and Bioinformatics*, vol. 14, no. 1, pp. 64–75, 2017.
- [18] Y. Zhang, D.-W. Gong, X.-Y. Sun, and Y.-N. Guo, "A PSO-based multi-objective multi-label feature selection method in classification," *Scientific Reports*, vol. 7, no. 1, article no. 376, 2017.
- [19] J. Handl and J. Knowles, "Evolutionary multiobjective clustering," in *Parallel Problem Solving from Nature—PPSN VIII*, vol. 3242 of *Lecture Notes in Computer Science*, pp. 1081–1091, Springer, Berlin, Germany, 2004.
- [20] J. Handl and J. Knowles, "An evolutionary approach to multiobjective clustering," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 1, pp. 56–76, 2007.
- [21] S. Saha and S. Bandyopadhyay, "A symmetry based multiobjective clustering technique for automatic evolution of clusters," *Pattern Recognition*, vol. 43, no. 3, pp. 738–751, 2010.
- [22] I. Saha, U. Maulik, and D. Plewczynski, "A new multi-objective technique for differential fuzzy clustering," *Applied Soft Computing*, vol. 11, no. 2, pp. 2765–2776, 2011.
- [23] X. L. Xie and G. Beni, "A validity measure for fuzzy clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 8, pp. 841–847, 1991.
- [24] C. A. C. Coello, "A comprehensive survey of evolutionary-based multiobjective optimization techniques," *Knowledge and Information Systems*, vol. 1, no. 3, pp. 269–308, 1999.
- [25] K. Deb, *Multiobjective Optimization Using Evolutionary Algorithms*, New York, NY, USA, Wiley, 2001.
- [26] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [27] E. Zitzler and L. Thiele, "An evolutionary algorithm for multiobjective optimization?: the strength pareto approach," *TIK-Report* 43, 1998.
- [28] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: improving the strength pareto evolutionary algorithm," *TIK-Report*, vol. 103, 2001.
- [29] L. Zhao, Y. Tsujimura, and M. Gen, "Genetic algorithm for fuzzy clustering," in *Proceedings of the IEEE International Conference on Evolutionary Computation*, pp. 716–719, Nagoya, Japan.
- [30] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, no. 2, pp. 224–227, 1978.
- [31] U. Maulik and S. Bandyopadhyay, "Performance evaluation of some clustering algorithms and validity indices," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 12, pp. 1650–1654, 2002.
- [32] M. Srinivas and L. M. Patnaik, "Adaptive probabilities of crossover and mutation in genetic algorithms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 24, no. 4, pp. 656–667, 1994.
- [33] Z. Deng, K.-S. Choi, F.-L. Chung, and S. Wang, "Enhanced soft subspace clustering integrating within-cluster and between-cluster information," *Pattern Recognition*, vol. 43, no. 3, pp. 767–781, 2010.
- [34] H. Xia, J. Zhuang, and D. Yu, "Novel soft subspace clustering with multi-objective evolutionary approach for high-dimensional data," *Pattern Recognition*, vol. 46, no. 9, pp. 2562–2575, 2013.
- [35] M. R. P. Ferreira, F. D. A. T. De Carvalho, and E. C. Simões, "Kernel-based hard clustering methods with kernelization of the metric and automatic weighting of the variables," *Pattern Recognition*, vol. 51, pp. 310–321, 2016.
- [36] M. R. Ferreira and F. d. de Carvalho, "Kernel fuzzy c-means with automatic variable weighting," *Fuzzy Sets and Systems*, vol. 237, pp. 1–46, 2014.
- [37] Z. Zhou and S. Zhu, "Kernel-based multiobjective clustering algorithm with automatic attribute weighting," *Soft Computing*, pp. 1–25, 2017.
- [38] K. Bache and M. Lichman, *UCI Machine Learning Repository*, University of irvine california (Schedule/Informatics) 2008, 2013.
- [39] J. M. Santos and M. Embrechts, "On the Use of the Adjusted Rand Index as a Metric for Evaluating Supervised Classification," in *Artificial Neural Networks – ICANN 2009*, vol. 5769 of *Lecture Notes in Computer Science*, pp. 175–184, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.
- [40] J. Z. Huang, M. K. Ng, H. Rong, and Z. Li, "Automated variable weighting in k-means type clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 5, pp. 657–668, 2005.



Hindawi

Submit your manuscripts at
www.hindawi.com

