Postprint of Research on Multi-Distance Clustering Based on Multi-Objective Evolutionary Algorithms

Authors: Liu Cong, Wan Xiuhua

Date: 2018-05-20T00:00:00+00:00

Abstract

Traditional clustering algorithms are typically designed based on a single distance measure, and how to organically fuse multiple distance measures is a current challenge. This paper proposes a multiobjective evolutionary multiple distance measure clustering framework (MOMDC) based on multiobjective evolutionary algorithms, and utilizes Euclidean distance and Path distance to design the actual framework. The framework first pre-clusters the dataset separately using the two distance measures, and then merges the pre-clustering results to reduce the problem scale; second, it respectively calculates the two types of distance relationships between sub-clusters; finally, it employs a multiobjective evolutionary algorithm to cluster in parallel across the two distance spaces. In the design of the multiobjective evolutionary algorithm, a real-number-label encoding scheme is adopted to design chromosomes, and two fitness functions based on the two distance measures are designed to evaluate chromosomes. Finally, MOMDC is experimentally compared with several other classical algorithms on a large number of datasets. Experimental results demonstrate that the framework can achieve good results on datasets with different distributions.

Full Text

Research on Multiple Distance Clustering Based on Multi-Objective Evolutionary Algorithm

Liu Cong, Wan Xiuhua (School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China)

Abstract: Traditional clustering algorithms are typically designed based on a single distance metric, and integrating multiple distance metrics organically presents a key challenge. This paper proposes a Multi-Objective Evolutionary

Multiple Distance Measure Clustering (MOMDC) framework based on multi-objective evolutionary algorithms, implementing the actual framework using Euclidean distance and Path distance. The framework first pre-clusters the dataset using both distance measures separately, then merges the pre-clustering results to reduce problem scale. Second, it calculates the two distance relationships between sub-clusters. Finally, it employs a multi-objective evolutionary algorithm to perform parallel clustering in both distance spaces. In the multi-objective evolutionary algorithm design, chromosomes are encoded using a real-number-label approach, and two fitness functions based on the two distance measures are designed to evaluate chromosomes. The MOMDC framework is compared with several classical algorithms on numerous datasets. Experiments demonstrate that the framework achieves favorable results for datasets with different distributions.

Keywords: similarity measure; distance metric; multi-objective RMMEDA evolutionary algorithm; real-tag coding

---

1. Introduction

With the rapid development of information technology and computer science, data scales continue to expand and data types exhibit increasing diversification. Extracting implicit and valuable knowledge from massive datasets represents a primary challenge for researchers. This has led to the advancement of various data analysis techniques, including cluster analysis, correlation analysis, regression analysis, and variance analysis, among which cluster analysis is most widely applied. Clustering can partition a dataset into several subsets according to similarity rules, such that data within the same cluster exhibit high similarity while data from different clusters show significant differences [1]. As a data processing algorithm, cluster analysis finds extensive applications in statistics, pattern recognition, machine learning, data mining, and other fields [2].

Existing clustering algorithms can be categorized into hierarchical clustering, partition-based clustering, density-based clustering, grid-based clustering, and model-based clustering. In recent years, with deepening research, clustering methods based on genetic algorithms, fuzzy mathematics, and spectral segmentation have been proposed successively. Partition-based clustering is particularly prevalent in practical applications, with the K-means algorithm and FCM algorithm being the two most classical methods in this category. However, these algorithms have limitations: (a) they achieve good clustering results for spherical cluster structures but perform poorly on arbitrarily shaped structures; (b) both algorithms are prone to falling into local optima during the solution process. Density-based algorithms utilize data density properties to discover non-spherical clusters [3]. Hierarchical algorithms employ maximum inter-class distance, minimum inter-class distance, or other distance metrics to merge or split data in search of non-spherical clusters. Spectral clustering uses kernel

space distance to represent data in graph form [4], applying graph cut concepts for clustering. Path distance [5], manifold distance [6], Chebyshev distance, and kernel distance map arbitrarily shaped data into non-Euclidean spaces for clustering.

In summary, most clustering algorithms for arbitrarily shaped clusters either map data points into a partitionable distance space or employ new distance functions to measure similarity between points. This indicates that similarity measures play a crucial role in cluster analysis. The use of multiple distance metrics for clustering has attracted increasing attention from researchers, with numerous multi-distance metric clustering algorithms proposed in recent years. Reference [7] proposed a hard clustering algorithm that simultaneously considers multiple dissimilarity matrices for object partitioning, where matrices are generated from different variable sets and dissimilarity functions. Reference [8] presented a clustering method combining multi-objective distance metrics. However, these methods simply superimpose two distances with weights, making it extremely difficult to set appropriate weights. Therefore, designing an algorithm that can automatically select different similarity measures for different data structures represents a major current challenge.

Clustering problems can also be viewed as optimization problems, with evolutionary algorithms being widely applied in seeking global optima [9]. Evolutionary algorithms offer unparalleled advantages as population intelligence optimization algorithms that find global optimal solutions through selection, crossover, and mutation. In recent years, researchers have proposed many clustering methods based on evolutionary algorithms. Reference [10] addressed the sensitivity of K-modes clustering algorithm to initial cluster center selection and its tendency to fall into local optima by proposing a differential evolution-based K-modes clustering algorithm that achieved better results. Reference [11] introduced a differential evolution-based fuzzy C-means clustering algorithm that applies DE to FCM, mitigating FCM' s excessive dependence on initial values and sensitivity to noisy data. However, most of these algorithms are designed for single objective functions and single similarity measures. Due to the limitations of single-objective functions, researchers have also proposed clustering methods based on multi-objective evolutionary algorithms. The MOCK algorithm proposed in [12] is a classic multi-objective evolutionary clustering algorithm that simultaneously optimizes cluster compactness and connectivity. The MODEFC algorithm in [13] uses two objectives: the FCM algorithm and the XB index. Reference [14] proposed a multi-objective differential evolution algorithm based on spatial distance (SD-MODE) on the foundation of classical differential evolution. Nevertheless, existing multi-objective evolutionary clustering algorithms are typically designed based on single distance spaces, most commonly Euclidean distance, which performs well on hyper-spherical data but poorly on non-hyper-spherical data. Therefore, designing a clustering framework based on multi-distance measures and multi-objective evolutionary algorithms holds significant importance. Addressing these two issues, this paper proposes a multi-distance clustering algorithm based on multi-objective evolutionary algorithms

that uses multi-objective algorithms as the optimization method to largely avoid local optima and incorporates multiple distance measures as multiple objective functions into the algorithm framework, enabling parallel clustering for various data structures and handling both hyper-spherical and non-hyper-spherical data.

1.1 Clustering

Let $X = \{x_1, x_2, ..., x_n\}$ denote a dataset with n samples, where x_i is a sample vector containing d features. Clustering partitions X into k subsets $C = \{C_1, C_2, ..., C_k\}$ such that $C_1 \cup C_2 \cup ... \cup C_k = X$ and $C_i \cap C_j = \emptyset$ for $i, j = 1, ..., k$ and $i \neq j$, with $C_i \neq \emptyset$ for $i = 1, ..., k$.

The K-means algorithm is a classical clustering algorithm with the following basic procedure: a) Randomly select k data objects from n data samples as initial cluster centers, where k is the number of clusters; b) Calculate the distance between each data object and the cluster centers, and assign data points to different clusters based on the nearest distance; c) Recalculate the cluster center for each cluster; d) Repeat steps b) and c) until each cluster center no longer changes.

Due to its simplicity and effectiveness, this algorithm has been widely used by researchers, but it has several limitations: a) Since the initial cluster centers are randomly selected, different initial centers yield different clustering results, often causing the algorithm to fall into local optima; b) The algorithm uses Euclidean distance as the distance metric between data samples, which works well for hyper-spherical data but poorly for non-hyper-spherical data.

1.2 Multi-Objective Evolutionary Algorithms

A multi-objective evolutionary algorithm problem with n decision variables and m objectives can be defined as:

$$\min \quad F(x) = (f_1(x), f_2(x), ..., f_m(x)) \tag{1}$$

$$\text{s.t.} \quad x \in \Omega \tag{2}$$

where $x = \{x_1, x_2, ..., x_n\} \in \Omega$ represents n decision variables, and $f_i(x)$ is the i-th objective function. The objectives are generally conflicting—a solution may be optimal for one objective but worst for another. Typically, the solution to a multi-objective optimization problem is not a single solution but a set called the Pareto optimal solution set. Several important definitions are given below:

Definition 1 (Feasible Solution): Any set of decision variable values that satisfies all constraints (both previous and subsequent constraints) of a linear programming problem is called a feasible solution of that linear programming problem.

Definition 2 (Feasible Solution Set): The set composed of all feasible solutions.

Definition 3 (Pareto Dominance): Assume x_a and x_b are two solutions satisfying the constraint functions. We say x_a dominates x_b if and only if $f_i(x_a) \leq f_i(x_b)$ for all $i = 1, 2, ..., m$ and $f(x_a) \neq f(x_b)$.

Definition 4 (Pareto Optimal Solution): A solution x^* is called a Pareto optimal solution if and only if $\neg\exists x \in \Omega$ such that x dominates x^*.

Definition 5 (Pareto Optimal Solution Set): The Pareto optimal solution set is the collection of Pareto optimal solutions, defined as $X^* = \{x \in \Omega \mid \neg\exists x' \in \Omega, x' \text{ dominates } x\}$.

Definition 6 (Pareto Front): The surface in function space corresponding to the Pareto optimal solution set, i.e., $PF^* = \{f(x) = (f_1(x), f_2(x), ..., f_m(x)) \mid x \in X^*\}$.

The RMMEDA algorithm is a classical multi-objective optimization algorithm whose basic idea is to establish several $m-1$ dimensional manifold distribution probability models to approximate the entire Pareto set. Its basic process is: a) Set generation $g = 0$, randomly generate initial population $Pop(g)$ and calculate the fitness function F for each individual; b) Establish the probability distribution model $P(g)$, where N represents N solutions; c) Generate a new solution set R according to the probability model and calculate the fitness function for R; d) Select NP individuals from R and $Pop(g)$, and generate offspring $Pop(g+1)$ using non-dominated sorting strategy; e) Check if stopping criteria are met; if so, proceed to step 6, otherwise set $g = g + 1$ and repeat steps b) through e); f) Return the non-dominated solution set of $Pop(g)$ to form the PS front.

For specific algorithm details, please refer to [15].

2. Algorithm Model

To address the shortcomings of traditional clustering algorithms in handling non-hyper-spherical clusters, this paper proposes a multi-distance clustering framework based on multi-objective evolutionary algorithms. This framework can organically integrate multiple distances into a single algorithmic framework, yielding clustering results containing two distance measures.

The MOMDC algorithm flowchart is shown in [Figure 1: see original paper]. The algorithm consists of three main components: data preprocessing, evolutionary algorithm clustering, and final clustering result display. The clustering preprocessing stage performs pre-classification and merging of the dataset, reducing algorithmic time complexity by decreasing the scale of data points. The evolutionary algorithm clustering stage is the core of the algorithm, which finds optimal clustering through chromosome encoding, evolutionary operator iteration, and objective function evaluation. Finally, the optimal clustering result is selected.

2.2 Data Preprocessing

This section primarily divides data into multiple small sub-clusters to reduce algorithmic time complexity, requiring pre-clustering of the data. Since this paper uses two distances for clustering, pre-clustering must also be performed using both distances. First, clustering is performed separately in the two distance spaces, then sample pairs belonging to the same class are placed into the same sub-cluster. The two distances used here are Euclidean distance and Path distance [4]. Euclidean distance can effectively discover hyper-spherical clusters in data, while Path distance can uncover irregularly shaped clusters.

2.2.1 Euclidean Space Pre-Clustering First, pre-clustering is performed using Euclidean distance. The FCM algorithm is used as the clustering algorithm, as shown in Equation (2). Since the number of pre-clustering categories needs to be predetermined, to demonstrate the adaptability of this algorithm, the CS index is used to automatically detect the optimal number of clusters, with the CS index shown in Equation (3).

$$J_{fcm} = \sum_{i=1}^{m_1} \sum_{j=1}^{n} u_{ij}^{m} d^2(c_i, x_j) \tag{2}$$

where m_1 represents the number of clusters, n represents the number of sample points, m represents a membership factor (generally set to 2), C_i represents all data points in the i-th class, c_i represents the center of the i-th class, and $d(\cdot, \cdot)$ is the distance measure. After this module, a set of sub-clusters $C = \{C_r, r = 1, ..., m_1\}$ is obtained.

2.2.2 Path Space Pre-Clustering Path distance is used for pre-clustering of data. Since Path distance describes the relationship between any two samples, it is necessary to calculate the Path distance between any two points to form a distance matrix, then use the NCUT algorithm [4] to pre-cluster the Path distance matrix. The optimal number of pre-clustering categories also needs to be automatically determined, using Equation (4) to detect the optimal number of clusters.

$$CS = \frac{\sum_{i=1}^{k} \sum_{x_j \in C_i} d(x_j, c_i)}{\min_{i \neq j} d(c_i, c_j)} \tag{3}$$

where P_i represents the data points contained in the i-th class, X represents all data, $L(P_i, P_i)$ represents the distance between elements within the same class, $L(X, X)$ represents the distance between all elements, and $L(P_i, X)$ represents the distance between data and other elements in different classes. $L(P_i, P_i)/L(X, X)$ represents the correlation between data and other samples in the same class, while $L(P_i, X)/L(X, X)$ represents the correlation between

points in a class and all other samples. After this module, a set of sub-clusters $P = \{P_k, k = 1, ..., m_2\}$ is obtained.

2.2.3 Merging Pre-Clustering Results The main task of this module is to merge the clustering results obtained from the two similarity measures. The merging principle is: if two data points belong to the same class in both the FCM clustering result and the spectral clustering result, then these two data points are of the same class; otherwise, they belong to different classes. If the pre-clustering numbers are m_1 and m_2, the merged clustering number is $M = \{M_q, q = 1, ..., r\}$. The pseudo-code for the merging algorithm is as follows:

Input: X, C, P
Output: Merged clustering results
Initialization: Dataset $X = \{x_i, i = 1, ...n\}$; Euclidean distance pre-clustering set $C = \{C_r, r = 1, ..., m_1\}$; Path distance pre-clustering set $P = \{P_k, k = 1, ..., m_2\}$; $Q = \emptyset$; $M_1 = \{x_1\}$; $q = 1$; $M = \{M_q, q = 1, ..., r\}$

```
1. for i=1 to n do
2.   for j=i+1 to n do
3.     if (x_i   Q   x_j   Q) then
4.       q = q+1; M_q = {x_i}  // Place x_i into a new M_q class first
5.     if (x_i   Q   x_j   Q) then
6.       if ( s,t: x_i, x_j   C_s   r,k: x_i, x_j   P_r) then
7.         M_q = M_q   {x_j}; Q = Q   M_q  // Place x_j into the sub-class M_i where x_i resi
8.     else if (x_i   Q   x_j   Q) then
9.       if ( s,t: x_i, x_j   C_s   r,k: x_i, x_j   P_r) then
10.        Find the sub-class M_p where x_i resides
11.        M_p = M_p   {x_j}
12.    else if (x_j   Q   x_i   Q) then
13.      if ( s,t: x_i, x_j   C_s   r,k: x_i, x_j   P_r) then
14.        Find the sub-class M_p where x_j resides
15.        M_p = M_p   {x_i}
16.    j = j+1
17.  i = i+1
```

2.3 Multi-Objective Evolutionary Clustering Algorithm

This module performs parallel clustering in both Euclidean distance and Path distance simultaneously through the multi-objective evolutionary algorithm framework. It requires consideration of chromosome encoding, objective function design, evolutionary operator settings, and selection of optimal Pareto points, with the first two being the primary concerns.

2.3.1 Chromosome Encoding After preprocessing and merging, a set of pre-clustering results $M = \{M_q, q = 1, ..., r\}$ is obtained. These r sub-classes need to be partitioned into k classes. Real-number encoding is used for chromosomes, which can be represented as $R = \{R_1, R_2, ..., R_r\}$, where $R_i \in (0, 1]$ for $i =$

$1, 2, ..., r$. After decoding, each class center mc_i is represented as shown in Equation (6).

$$mc_i = \frac{1}{|M_i|} \sum_{x \in M_i} x \tag{6}$$

2.3.2 Objective Function Design This module primarily defines two fitness functions. The first fitness function f_1 is designed based on Euclidean distance, and the second fitness function f_2 is designed based on Path distance.

For f_1, we first calculate the center of each sub-class $\{m_1, m_2, ..., m_r\}$ as shown in Equation (6). Next, operations are performed on the class centers to partition $\{m_1, m_2, ..., m_r\}$ into k classes. Each class C_i^* can be represented as in Equation (7).

$$C_i^* = \{M_j \mid R_j \in [(i-1)/k, i/k]\} \tag{7}$$

After decoding, the class center mc_i of each class is represented as:

$$mc_i = \frac{1}{|C_i^*|} \sum_{x \in C_i^*} x \tag{8}$$

Then the objective function f_1 is:

$$f_1 = \sum_{i=1}^{k} \sum_{M_j \in C_i^*} \|mc_i - mc_j\| \tag{9}$$

For the objective function f_2, the distance between two sub-classes is calculated using Path distance. Since Path distance is a path connectivity distance, the shortest distance between two sub-classes is used as the inter-class distance, as shown in [Figure 2: see original paper].

In [Figure 2: see original paper], d represents the shortest distance between all points of class M_1 and class M_2, which is used as the inter-class distance. The objective function f_2 is shown in Equation (10).

$$f_2 = \sum_{i=1}^{k} L_P(C_i^*, C_i^*) \tag{10}$$

Since objective function f_1 is designed based on Euclidean distance compactness and objective function f_2 is designed based on Path distance compactness, simultaneously optimizing these two objective functions considers both the aggregation in Euclidean space and the aggregation in Path space.

2.3.3 Evolutionary Operators This paper uses the evolutionary operators from the RMMEDA algorithm, which will not be detailed here.

2.3.4 Selecting the Best Clustering Result In the final generated Pareto set, selecting the best solution among multiple solutions is also a problem. For massive result sets, manual selection is impossible. However, the Pareto set obtained in this experiment is not large, so the corresponding clustering results are displayed through decoded plotting, and the best classification result is manually identified from all solution sets. Although this reduces efficiency, it successfully solves the problem.

3. Experimental Results and Analysis

To demonstrate the effectiveness of the proposed algorithm, this section compares it with several existing algorithms. The comparison algorithms include K-means, FCM, Euclidean distance-based NCUT algorithm (NCUTE) [4], Path distance-based NCUT algorithm (NCUTP), and density-based DBSCAN algorithm [3].

There are many evaluation metrics for clustering effectiveness, such as NMI index, F-score index, and Rand index. This experiment uses the Rand index to evaluate clustering accuracy. A higher Rand value indicates better clustering performance, with Rand = 1 indicating that all samples are classified into the correct categories.

3.1 Algorithm Parameter Settings

First, we briefly explain the parameters used in the algorithms. For MOMDC, the population size is 600 and the number of iterations is 2500. For the DBSCAN algorithm, the scanning radius eps is 0.9534 and the minimum number of points (minPts) is 5.

3.2 Test Data Settings

This paper uses six test datasets to evaluate the proposed algorithm. Since the advantage of the proposed algorithm lies in considering both Euclidean distance and Path distance, enabling clustering of both spherical and irregularly shaped data, the simulated test sets include both spherical cluster data (XOR and ARC) and non-spherical cluster data (LFF and LTS). Additionally, two UCI datasets, IRIS and WINE, are included in the test data. Partial test datasets are shown in Figure 3: see original paper~(d).

The attributes of all test datasets are shown in , where DimenN represents dimensionality, DataN represents the number of sample points, and ClusterN represents the correct number of clusters.

3.3 MOMDC Clustering Demonstration

This section selects the XOR, ARC, and LTS datasets to demonstrate the Pareto set and corresponding clustering results. In each Pareto plot, f_1 and f_2 represent Euclidean distance clustering and Path distance clustering, respectively, and Gen represents the number of iterations.

The Pareto plot and corresponding clustering results for XOR are shown in [Figure 4: see original paper]. From [Figure 4: see original paper], we can see that the Pareto plot contains only one solution, indicating that this dataset can achieve good results in both distance spaces. The result values are shown in .

The Pareto plot and corresponding clustering results for ARC are shown in [Figure 5: see original paper]. In Figure 5: see original paper, the Pareto set has 4 points, and two representative points are selected for analysis. Figure 5: see original paper shows the clustering result of the first point, while Figure 5: see original paper shows the clustering result of the third point. The clustering result in Figure 5: see original paper is not completely correct, with only 92% of points correctly clustered, whereas Figure 5: see original paper shows completely correct clustering. The result values are shown in . This demonstrates that the proposed algorithm can achieve good clustering results on hyper-spherical data.

The Pareto plot and corresponding results for LTS are shown in [Figure 6: see original paper]. In Figure 6: see original paper, there are 3 Pareto points. Figure 6: see original paper shows the clustering result of the second point, where all data are assigned to the same class with only 50% accuracy. Figure 6: see original paper shows the clustering result of the first point with 50.5% accuracy. Figure 6: see original paper shows the result of the third point. The specific result values are shown in . This test data demonstrates that MOMDC can achieve good clustering results on non-hyper-spherical data.

3.4 Comparative Analysis Results

This section compares the proposed algorithm with K-means, FCM, NCUT, and DBSCAN algorithms from on six datasets, using the Rand index as the evaluation metric. The test results are shown in .

The experimental results show that K-means, FCM, and NCUTE–three Euclidean space-based algorithms–achieve good clustering results on the XOR and ARC spherical cluster datasets but perform poorly on the LFF and LTS non-spherical cluster datasets. Their clustering results on the two UCI datasets are also not satisfactory, primarily because these three algorithms are based on Euclidean distance and thus work well for hyper-spherical data. Conversely, NCUTP and DBSCAN achieve ideal results on non-hyper-spherical datasets because these two algorithms use distance metrics suitable for data distribution, yielding good clustering effects on non-hyper-spherical data. MOMDC demonstrates good clustering performance on both hyper-spherical and non-hyper-spherical data. Although it cannot achieve completely correct clustering

on UCI data, its clustering accuracy is better than the other five algorithms. The main reasons are: (1) the proposed algorithm integrates Euclidean distance and Path distance within the same clustering framework, enabling automatic selection of appropriate distance measures based on different data distributions; (2) the proposed algorithm is a population optimization algorithm, providing better stability when seeking global optima.

4. Conclusion

This paper proposes a multi-similarity multi-objective evolutionary clustering algorithm. The main challenge lies in how to fully utilize Euclidean distance and Path distance and integrate them organically. This paper first uses the FCM algorithm and spectral clustering algorithm for pre-classification based on Euclidean distance and Path distance, respectively, to reduce time complexity by decreasing data volume and obtain the best pre-clustering number. Then, a multi-distance fusion framework based on multi-objective evolutionary algorithms is designed to iteratively obtain the best Pareto set. This method compensates for the shortcomings of most clustering algorithms that easily fall into local optima and can only handle either hyper-spherical or non-hyper-spherical data. The algorithm can process both spherical and non-spherical cluster datasets. However, while improving the generality of the clustering algorithm, it also increases time complexity.

The framework is still in its preliminary research stage and has considerable room for improvement, mainly including: (a) how to reduce algorithmic time complexity; (b) how to automatically select appropriate clustering results from the obtained Pareto set; (c) how to automatically determine the number of clusters—these are the focuses of future research.

References

[1] He Ling, Wu Lingda, Cai Yichao. Survey of clustering algorithms in data mining [J]. Computer Application Research, 2007, 24(1): 10-13.

[2] Mei J P, Chen L. Fuzzy clustering with weighted medoids for relational data [J]. Pattern Recognition, 2010, 43(5): 1964-1974.

[3] Syzhou Fud. FDBSCAN: a fast DBSCAN algorithm [J]. Journal of Software, 2000, 11(6): 735-744.

[4] Shi J, Malik J. Normalized cuts and image segmentation [J]. IEEE Trans on Pattern Analysis & Machine Intelligence, 2002, 22(8): 888-905.

[5] Fischer B, Buhmann J M. Path-based clustering for grouping of smooth curves and texture segmentation [J]. IEEE Trans on Pattern Analysis & Machine Intelligence, 2003, 25(4): 513-518.

[6] Gong Maoguo, Wang Shuang, Ma Meng, et al. Two-stage clustering algorithm for complex distribution data [J]. Journal of Software, 2011, 22(11): 2760-2772.

[7] De Carvalho F D A T, Lechevallier Y, De Melo F M. Partitioning hard clustering algorithms based on multiple dissimilarity matrices [J]. Pattern Recognition, 2012, 45(1): 447-464.

[8] Rao A S, Ramakrishna S, Babu P C. MODC: multi-objective distance based optimal document clustering by GA [J]. Indian Journal of Science and Technology, 2016, 9(28): 1-8.

[9] Chung H S, Alonso J. Multiobjective optimization using approximation model-based genetic algorithms [C]//Proc of the 10th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization. 2006: 3-4.

[10] Wang Hongbo. Research on clustering algorithm based on differential evolution computation [D]. Jinan: Shandong Normal University, 2011.

[11] Yang Yang. Research on fuzzy C-means clustering algorithm based on differential evolution [D]. Chengdu: University of Electronic Science and Technology, 2015.

[12] Handl J, Knowles J. Exploiting the Trade-off: the benefits of multiple objectives in data clustering [C]//Lecture Notes in Computer Science. 2005.

[13] Saha I, Maulik U, Plewczynski D. A new multi-objective technique for differential fuzzy clustering [J]. Applied Soft Computing, 2011, 11(2): 2765-2773.

[14] Zeng Yinglan, Wu Jun, Zheng Jinhua. Multi-objective differential evolution algorithm based on spatial distance [J]. Computer Application Research, 2009, 26(2): 57-60.

[15] Zhang Q, Zhou A, Jin Y. RM-MEDA: a regularity model-based multiobjective estimation of distribution algorithm [J]. IEEE Trans on Evolutionary Computations, 2008, 12(1): 41-63.

Note: Figure translations are in progress. See original paper for figures.