Multi-objective Differential Automatic Clustering Algorithm Using Cluster Center Density Strategy (Postprint)

Authors: Shen Xiaoning, Sun Yi, Xue Yunyong, Sun Shuai

Date: 2018-08-13T00:00:00+00:00

Abstract

To address the defect in clustering where the randomness of centroid selection leads to chosen centroids deviating from the dataset or centroids being overly concentrated, resulting in erroneous clustering, the proposed algorithm performs a two-stage screening of centroid selection. Specifically, it screens out centroids with excessively low density and centroids with pairwise distances that are too small, preventing them from participating in clustering; thereafter, the algorithm performs clustering on the remaining centroids after screening. To enable the algorithm to obtain optimal centroids more efficiently, an improved clustering criterion function is proposed, which dynamically penalizes the number of clusters. To evaluate the application performance of the proposed algorithm on clustering problems, simulation experiments were conducted on two different types of datasets. Comparison results with three existing automatic clustering algorithms demonstrate that the proposed algorithm can achieve better clustering results, thereby validating the effectiveness of the proposed strategy.

Full Text

Preamble

Title: Multi-Objective Differential Evolution Automatic Clustering Algorithm Based on Class-Center Density Strategy

Authors: Shen Xiaoning, Sun Yi, Xue Yunyong, Sun Shuai
(School of Information & Control, Nanjing University of Information Science & Technology, Nanjing 210044, China)

Abstract: In clustering processes, the randomness of class-center selection often leads to selected centers deviating from the dataset or becoming overly concentrated, resulting in erroneous clustering. To address this defect, the proposed

algorithm performs two-stage screening of class centers: those with excessively low density and those with excessively small pairwise distances are filtered out and excluded from clustering. The algorithm then proceeds to cluster the remaining centers. To enable rapid convergence to optimal class centers, an improved clustering criterion function is proposed that dynamically penalizes the number of clusters. To evaluate the proposed algorithm' s performance on clustering problems, simulation experiments were conducted on two different types of datasets. Comparative results with three existing automatic clustering algorithms demonstrate that the proposed algorithm achieves superior clustering results, thereby validating the effectiveness of the proposed strategies.

Keywords: automatic clustering; class-center density strategy; class-center screening; multi-objective optimization; differential evolution

0 Introduction

Many practical problems in scientific research and engineering design can be formulated as parameter optimization problems. In reality, these optimization problems often involve multiple design objectives that conflict with and constrain each other, where optimizing one objective typically degrades the performance of at least one other objective. Such problems are generally referred to as multi-objective optimization problems [1].

Clustering analysis, as a data analysis tool and methodology, finds broad applications across numerous research and application domains. However, most real-world data lack prior knowledge, making it impossible to predetermine the number of clusters. These problems can be categorized as automatic clustering [3], which aims to correctly cluster data without pre-specifying the number of clusters.

Differential evolution (DE), proposed by Storn and Price in 1997, is a population-based heuristic parallel search method [2]. The DE algorithm comprises initialization, mutation, crossover, and selection operations. Unlike other optimization algorithms, DE' s evolutionary perturbation of individuals is realized through differential information from multiple individuals. DE offers advantages including fast convergence, few control parameters, and stable optimization results, leading increasing researchers to apply it to multi-objective optimization problems.

In recent years, scholars have proposed numerous automatic clustering algorithms. Das et al. introduced the automatic clustering algorithm ACDE based on improved differential evolution in 2008 [4], which modified the mutation and crossover factors in basic DE and employed real-valued, fixed-length chromosome encoding to enable automatic clustering. Maulik et al. subsequently proposed two improved ACDE variants, ADEFC [5] and MoDEAFC [6], in 2009 and 2010, respectively. ADEFC incorporated fuzzy partition measures and up-

dated cluster centers using fuzzy C-means clustering (FCM), while MoDEAFC improved the mutation operation of ADEFC. The limitation of these algorithms lies in their use of a single metric as the clustering criterion function, resulting in varying clustering effects across different datasets and poor algorithm robustness.

In 2014, Rodriguez et al. proposed the density peak clustering algorithm (RLCLu) in *Science* [7]. This density-based clustering algorithm consists of two main steps: manually selecting density peaks (i.e., class centers) through a "decision graph," and assigning remaining data points to obtain clustering results. While capable of discovering non-spherical clusters, RLCLu cannot automatically determine class centers, and manual selection via decision graphs is prone to errors, particularly for special datasets. To address this limitation, Li Tao proposed an automatic density peak clustering algorithm (ADPC) [8] that automatically determines class centers through three steps: (a) calculating each data point's local density and its minimum distance to points with higher density; (b) automatically determining class centers via ranking graphs; and (c) assigning each remaining data point to the same class as its nearest higher-density neighbor while identifying noise points based on boundary density. However, ADPC struggles with complex manifold-structured datasets lacking density peaks or containing multiple density peaks within clusters, often failing to achieve ideal clustering despite automatic center determination.

Ye et al. [9] also improved RLCLu by modifying the measure "if a point' s distance deviation multiplied by its local density yields the maximum value, select it as a cluster center." They proposed calculating absolute differences between this measure and those of remaining points to amplify the distinction between cluster centers and other points, laying the foundation for automatic clustering. However, due to the randomness of center selection, this approach still suffers from the problem of being able to automatically determine centers yet failing to obtain ideal clustering results.

To address the defect that random class-center selection easily leads to erroneous clustering, and to enable accurate automatic clustering when the number of clusters is unknown, this paper proposes a Multi-Objective Differential Evolution Automatic Clustering algorithm based on Class-Center Density (MODEAC-CD). Building upon multi-objective differential evolution and incorporating improved class-center density strategies with clustering validity indices, MODEAC-CD first performs two-stage screening to exclude centers with excessively low density or excessively small pairwise distances. The remaining centers are then clustered by assigning each data point to its nearest center. This strategy is combined with multi-objective differential evolution using within-class distance and improved between-class distance as two criterion functions for evaluating clustering quality, effectively enhancing clustering accuracy and convergence speed.

1.1 Individual Encoding Scheme

Unlike traditional evolutionary algorithm-based clustering encoding methods, this paper adopts a real-valued, fixed-length chromosome encoding scheme [4]. For a dataset with n points, each having d dimensions, let $K_{max} = \sqrt{n}$ represent the maximum possible number of clusters per individual. An individual can be represented as:

$$V = (T_1, \cdots, T_i, \cdots, T_{K_{max}}, C_1, \cdots, C_i, \cdots, C_{K_{max}}) \quad (1)$$

where $C_i = (C_{i1}, C_{i2}, \cdots, C_{id})$ is the i-th class center, and T_i represents the label-bit threshold corresponding one-to-one with C_i, determining whether C_i participates in clustering. T_i is a real number with $T_i \in [0, 1]$.

In this encoding scheme, cluster center activation follows specific rules: when $T_i > 0.5$, its corresponding center C_i is activated for clustering. Each individual comprises two parts: label-bit thresholds and class centers, yielding a total length of $K_{max} + K_{max} \times d$. The first K_{max} positions represent T_i values, while the subsequent $K_{max} \times d$ positions represent class centers.

1.2 Improvement for Deviation of Cluster Centers from Dataset

Literature [7] states that a data point should satisfy two basic conditions to be considered a cluster center: it must be surrounded by neighboring points with relatively lower density, and its distance to other higher-density points should be relatively large. Based on this principle, this section proposes an improvement strategy for centers deviating from the dataset, with Section 1.3 addressing overly concentrated centers to select better clustering centers.

Let the population size be N. As previously described, each individual V contains K_{max} class centers, resulting in $N \times K_{max}$ total centers (each center' s dimensions are randomly generated within the feature ranges of the dataset). The improvement strategy for deviating centers proceeds as follows:

a) Treat the $N \times K_{max}$ centers as data to be clustered and compute pairwise Euclidean distances to obtain a distance matrix $M_{(N \times K_{max}) \times (N \times K_{max})}$;

b) From matrix M, compute the mean distance between each center and all other centers, yielding an average distance set $\{R_q\}$, where $q = 1, 2, \cdots, (N) \times K_{max}$. This set effectively reflects the distribution of centers in the dataset. Generally, a smaller scalar mean in $\{R_q\}$ indicates a higher probability that its corresponding center lies in a dense region of the dataset;

c) Compute the mean of $\{R_q\}$ to obtain a scalar threshold R_2. Based on the principle that a dataset' s mean reflects its central tendency, R_2 serves as a threshold for counting near neighbors across all centers;

d) For the q-th center ($q = 1, 2, \cdots, (N) \times K_{max}$), count the number of other centers within distance R_2, recorded in array $\{D_q\}$, where $q = 1, 2, \cdots, (N) \times K_{max}$. This count represents the q-th center' s density;

e) Compute the mean of array $\{D_q\}$ to obtain scalar threshold R_5. Identify centers with density below R_5 and modify their paired label-bit thresholds to a real number between 0 and 0.5 to exclude them from clustering. Conversely, modify label-bit thresholds for centers with density above R_5 to a real number between 0.5 and 1 to include them in clustering.

To ensure each individual V has at least 2 centers participating in clustering, the following measures are taken: when only 1 center participates, select the densest additional center from V, modify its paired label-bit threshold to a value between 0.5 and 1, and include it in clustering. If no centers participate, select the two densest centers, modify their paired thresholds to values between 0.5 and 1, and include them.

[Figure 1: see original paper] illustrates the center screening process of MODEAC-CD on the synthetic dataset long1 [10], where x and y axes represent center coordinates in 2D space. Points numbered 1 and 2 lie on class boundaries and will be screened out due to their densities being far below threshold R_5, effectively preventing randomly selected centers from deviating from the dataset.

1.3 Improvement for Overly Concentrated Class Centers

The above operations mitigate center deviation but may cause selected centers to become overly concentrated, complicating clustering. Therefore, further improvements address center concentration as follows:

a) To determine if centers are overly concentrated, this paper adopts a distance threshold R_q^{17} from literature [11], defined as the mean distance between the q-th center and all other centers ($q = 1, 2, \cdots, (N) \times K_{max}$). Here, $R_q^{17} = \frac{R_q + R_2}{2}$;

b) For individual V, identify the participating center with maximum density, denoted q_1. Compute Euclidean distances between q_1 and other participating centers in V, screening out centers with distances smaller than threshold R_{q1}^{17}. These centers are considered too close to q_1 and are excluded by modifying their paired label-bit thresholds to values between 0 and 0.5;

c) From remaining participating centers, identify the center with second-highest density, denoted q_2. Similarly screen out centers within distance

R_{q2}^{17} of q_2 (with q_1 excluded from this screening), modifying their thresholds to exclude them from clustering;

d) Repeat this process sequentially until the last participating center in the individual is processed.

As shown in [Figure 1: see original paper], points numbered 3 and 4 are the actual two centers of dataset long1, while points like 5 and 6 are potential centers around true class centers. If point 3 is the densest participating center in V and point 5' s distance to point 3 is smaller than R_3^{17}, point 5 will be eliminated. Subsequently, if point 4 is the second-densest participating center and point 6' s distance to point 4 is smaller than R_4^{17}, point 6 will also be eliminated. This effectively prevents erroneous clustering caused by overly concentrated centers.

Afterward, verify the number of participating centers in V. If fewer than 2 (i.e., only the densest center remains), find the center farthest from the densest center, include it in clustering, and modify its paired label-bit threshold to a random number between 0.5 and 1.

2.1 Clustering Criterion Functions

The proposed multi-objective differential evolution automatic clustering algorithm simultaneously optimizes two clustering criterion functions: the sum-of-squared-errors criterion function G_c and the improved between-class distance sum criterion function G_b.

2.1.1 Sum-of-Squared-Errors Criterion Function

The sum-of-squared-errors criterion function describes within-class distance [12] and is defined as:

$$G_c = \sum_{j=1}^{c} \sum_{k=1}^{n_j} \|x_k^{(j)} - C_j\|^2 \quad (2)$$

where c is the number of participating centers in individual V, $C_j(j = 1, 2, \cdots, c)$ is the center of the j-th class, n_j is the number of data points assigned to class j, $x_k^{(j)}$ is the k-th data point in class j, and G_c represents the total sum-of-squared-errors when n data points are clustered into c classes. A smaller G_c value indicates better clustering quality.

2.1.2 Between-Class Distance Sum Criterion Function

The between-class distance sum criterion function describes inter-class separation [13] and is defined as:

$$G_{b1} = \sum_{j=1}^{c} p_j (m_j - m)^T (m_j - m) \quad (3)$$

where m_j is the vector of mean values across each dimension for all data points in class j, m is the vector of mean values across each dimension for all data points, and p_j is the prior probability of data points in class j, describing the ratio of class j' s data points to the total number of points. G_{b1} describes the separation degree among different class centers; larger values indicate higher clustering quality.

Equation (3)' s G_{b1} tends to find more centers during early evolution stages to maximize its value. To enable rapid convergence to the optimal number of centers and obtain effective clustering partitions, the proposed algorithm improves G_{b1} by dynamically penalizing the number of centers c. The improved between-class distance sum criterion function G_b is described as:

$$G_b = \frac{1}{c'} \times G_{b1} \quad (4)$$

where G_{b1} is computed via equation (3), $c' = c^{2k}$, $k = 1 - 2 \times \frac{t}{t_{max}}$ when $t < 0.5 \times t_{max}$, t is the current iteration number, and t_{max} is the maximum iteration number. When $t < 0.5 \times t_{max}$, c is dynamically penalized (smaller t yields smaller $\frac{1}{c'}$). When $t \geq 0.5 \times t_{max}$, no penalty is applied ($c' = c$). This improved G_b prevents the algorithm from finding too many centers during early evolution, effectively improving evolutionary efficiency.

2.2 Solution Selection Strategy

Multi-objective clustering algorithms ultimately yield not a single clustering solution but a set of Pareto-optimal clustering solutions representing different trade-offs among objectives. However, clustering problems require a specific optimal clustering scheme. Therefore, after obtaining a Pareto-optimal solution set, the proposed multi-objective differential evolution automatic clustering algorithm requires an optimal solution selection process. Common selection methods include Gap Statistic [14] for evaluating cluster numbers and using the MS (Minkowski Score) index [15] to select the solution with minimum MS as optimal. This algorithm selects the solution with highest accuracy [10] from the Pareto-optimal set as the final optimal solution, where accuracy measures the ratio of correctly predicted data to total predictions.

2.3 Proposed Algorithm MODEAC-CD Flow

a) Initialization: The differential evolution algorithm's mutation and crossover operators each contain one parameter: mutation factor F and crossover factor CR. Set values for F and CR (see Section 3.3 for specific settings), specify maximum objective evaluations nmb_obj_max, and set $nArchive$ as the maximum capacity of external archive Archive. Generate initial parent population P of size N using the individual representation from equation (1).

b) For each individual in parent population P: - (a) Screen centers according to Sections 1.2 and 1.3; - (b) After updating centers, cluster the dataset by assigning each data point to the class of its nearest center, then compute individual objective values using equations (2) and (4). Set objective evaluation counter $nmb\_obj = N$; - (c) Identify the non-dominated solution set of P and store it in external archive Archive.

c) Perform differential evolution on parent population P **to generate offspring population** Q: - (a) Apply $DE/rand/2$ mutation strategy and binomial crossover strategy to population P to generate offspring population NPOP1. The $DE/rand/2$ mutation and binomial crossover expressions are given in equations (5) and (6):

$$v_i = x_{r1} + F(x_{r2} - x_{r3} + x_{r4} - x_{r5}) \quad (5)$$

$$u_{i,j} = \begin{cases} v_{i,j} & \text{if } (rand_j[0,1] \le CR) \text{ or } (j = j_{rand}) \\ x_{i,j} & \text{otherwise} \end{cases} \quad (6)$$

where three random individuals x_{r1}, x_{r3}, x_{r5} are taken from population P, and two random individuals x_{r4}, x_{r2} are taken from external archive Archive, with $x_{r1} \neq x_{r2} \neq x_{r3} \neq x_{r4} \neq x_{r5}$. Mutation factor F is a random number uniformly generated in $[0.8, 0.9]$, v_i is the i-th intermediate individual after mutation, $rand_j[0,1]$ is a uniformly distributed random number in $[0,1]$, j_{rand} is a uniformly random integer from $\{1, 2, \cdots, d\}$, d is the decision variable dimension, $x_{i,j}$ is the j-th dimension of current individual x_i, $v_{i,j}$ is the j-th dimension of intermediate individual v_i, and $u_{i,j}$ is the j-th dimension of offspring individual u_i; - (b) Apply $DE/current-to-best/1$ mutation strategy and binomial crossover to P to generate offspring population NPOP2. The $DE/current-to-best/1$ mutation strategy is given in equation (7):

$$v_i = x_i + F_1(x_{best} - x_i) + F_2(x_{r2} - x_{r3}) \quad (7)$$

where two random individuals x_{r2}, x_{r3} and current individual x_i are taken from P, x_{best} is randomly taken from external archive Archive, with $x_{r2} \neq x_{r3} \neq x_i \neq x_{best}$. Mutation factors F_1, F_2 are random numbers uniformly generated in $[0.8, 0.9]$; - (c) Merge the generated offspring populations to obtain offspring population Q, i.e., $Q = NPOP1 \cup NPOP2$.

d) For each individual in offspring population Q, perform operations similar to step b) (a)(b), accumulating objective evaluations as $nmb\_obj =$

$nmb\_obj + |Q|$ (where $|.|$ denotes set cardinality). Identify the non-dominated solution set of Q and merge it with Archive.

e) Update and prune parent population P and external archive Archive: Set $P = P \cup Q$. If $|P| > N$, update P using the $I_{\varepsilon+}$ indicator-based approach. Determine non-dominated solutions from Archive and assign them to Archive. If $|Archive| > nArchive$, update Archive using the Lp-norm distance-based diversity maintenance strategy. Supplementary details: - (a) The $I_{\varepsilon+}$ indicator describes the minimum distance required for one solution to dominate another in objective space [16], given by:

$$I_{\varepsilon+}(x_1, x_2) = \min_{\varepsilon}(f_i(x_1) - \varepsilon \leq f_i(x_2), 1 \leq i \leq m) \quad (8)$$

where m is the number of objectives. Individual fitness values are assigned based on this indicator. Equation (9) gives the fitness calculation for individual x_1 [17]:

$$F(x_1) = \sum_{x_2 \in P \setminus \{x_1\}} -e^{-I_{\varepsilon+}(x_2, x_1)/0.05} \quad (9)$$

During population update, individuals with smaller fitness values are sequentially removed until the specified population size is reached; - (b) The Lp-norm is defined as [18]:

$$L_p(x, y) = \left[\sum_{i=1}^{d}(x_i - y_i)^p\right]^{1/p} \quad (10)$$

where d is the decision vector dimension, and $L_p(x, y)$ represents the Lp-norm distance between vectors $x = (x_1, \cdots, x_d)$ and $y = (y_1, \cdots, y_d)$ in d-dimensional decision space. When Archive exceeds capacity, first add individuals with maximum/minimum objective values in each objective to an empty external archive $Archive'$. Then repeatedly select from Archive the individual with the maximum shortest Lp-norm distance to existing individuals in $Archive'$ and add it to $Archive'$ until $Archive'$ reaches capacity $nArchive$. Finally, set $Archive = Archive'$.

f) Termination condition: If objective evaluation count nmb_obj reaches the problem-specific nmb_obj_max, the algorithm stops and outputs the current external archive Archive as an approximate Pareto-optimal solution set, selecting the solution with highest accuracy as the optimal solution. Otherwise, return to Step 3.

2.4 Algorithm Time Complexity Analysis

Let m be the number of objectives and N be the population size. The time complexity of MODEAC-CD primarily includes: (a) center screening with maximum complexity $O((K_{max}N)^2) = O(nN^2)$; (b) non-dominated solution identification

in offspring population with complexity $O(N \log^{m-2} N)$ [17]; (c) indicator-based population update with complexity $O(N^2)$ [17]; (d) Archive update via dominance comparisons with complexity $O(N \log^{m-2} N)$ [17] and Lp-norm distance calculations for diversity maintenance with complexity $O(mN^2)$ [17]. Generally, the number of data points n exceeds the number of objectives m, so the worst-case total complexity is $\max\{O(N \log^{m-2} N), O(nN^2)\}$.

3.1 Experimental Data Description

To validate MODEAC-CD' s clustering effectiveness, two types of experimental datasets were selected. The first group consists of four UCI (University of California, Irvine) datasets, commonly used as public benchmark test sets in data mining [19]. The second group comprises four artificial datasets with spherical data characteristics [10]. Detailed dataset properties are shown in .

3.2 Comparison Algorithms

Literature [10] designed two differential evolution-based automatic clustering algorithms, PSIMACDE and DEAFC_DO, and compared them with three classic automatic clustering algorithms (GADE [20], VGAPS [21], ACDE [4]) on a set of datasets. Results showed PSIMACDE and DEAFC_DO outperformed the classic algorithms in both cluster number and accuracy for most datasets. In 2014, Rodriguez et al. proposed the density peak clustering algorithm (RLCLu) in *Science* [7], which states that a data point must satisfy two conditions to be a cluster center: being surrounded by lower-density neighbors and having relatively large distances to higher-density points. RLCLu identifies cluster centers through decision graphs, but its main difference from MODEAC-CD is that RLCLu cannot automatically identify centers, lacks clustering criterion functions, and requires manual center selection via decision graphs, introducing uncertainty.

3.3 Parameter Settings

For each clustering problem, the first three algorithms were independently run 20 times for statistical results. To ensure fair comparison, all three algorithms used the same stopping criterion: maximum objective evaluations $nmb\_obj\_max = 30020$. MODEAC-CD parameters were tuned based on experimental results, while PSIMACDE and DEAFC_DO parameters followed literature [10]. Specific values are shown in . The crossover probability CR for the first three algorithms was randomly generated in $[CR_{min}, CR_{max}]$. For RLCLu, decision graph results remain identical across runs without parameter changes, allowing

operators to manually select centers with relatively high local density and large distances to higher-density points. However, manual selection is error-prone for special datasets. RLCLu only requires defining the percentage of nearest neighbors, here set to $percent = 2$.

3.4 Experimental Results and Analysis

Three metrics were used for comparison: number of clusters, clustering accuracy [10], and adjusted rand index (ARI) [25], along with graphical analysis. The number of clusters should match the true number, higher accuracy is better, and ARI measures the probability of data pairs co-occurring in the same class across two partitions, with larger values indicating better clustering.

1) Graphical Analysis

[Figure 2: see original paper] and [Figure 3: see original paper] show the best clustering results of the four algorithms after 20 independent runs on datasets AD_5_2 and square4, with x and y axes representing 2D coordinates. For AD_5_2, MODEAC-CD, PSIMACDE, and RLCLu correctly identified 5 clusters matching the true number, while DEAFC_DO found only 4 clusters. For square4, MODEAC-CD, PSIMACDE, and RLCLu correctly identified 4 clusters, with MODEAC-CD producing more uniform and reasonable data distribution due to its class-center density strategy. DEAFC_DO identified only 2 clusters because it is a single-objective algorithm, while the other differential evolution-based algorithms are multi-objective, yielding superior overall partitions by simultaneously optimizing within-class and between-class distances. RLCLu also performed well by considering both local density and inter-center distances, effectively filtering noise/outlier points surrounding each class core.

2) Cluster Number and ARI Analysis

presents the mean and variance of cluster numbers and ARI across 20 runs on both dataset types. MODEAC-CD achieved the best cluster numbers (closest to true values with smallest variance) in 7 of 8 datasets, demonstrating excellent stability. PSIMACDE performed well on artificial datasets, while DEAFC_DO showed good results on artificial datasets Square1 and Long1, and UCI datasets Diabetes and Liver. RLCLu performed well on datasets Iris, Long1, and AD_5_2.

ARI results show MODEAC-CD achieved the best performance on three UCI datasets and the best overall performance on artificial datasets long1 and AD_5_2. PSIMACDE performed best on Iris and Square4, DEAFC_DO on Square1, and RLCLu showed relatively good performance on artificial datasets. Notably, all four algorithms achieved lower ARI values on UCI datasets, confirming that increased feature dimensionality raises clustering difficulty.

3) Accuracy Analysis

shows the mean and variance of clustering accuracy across 20 runs. MODEAC-CD achieved the highest accuracy on six datasets, PSIMACDE on Iris, DEAFC_DO on Square1, and RLCLu showed relatively good accuracy on Square1 and Long1. All algorithms achieved 100% accuracy on dataset Square1, indicating correct clustering.

4) Runtime Analysis

compares average runtime. MODEAC-CD requires relatively longer runtime due to its computationally intensive center screening strategy. However, since automatic clustering is an offline optimization problem with low real-time requirements, MODEAC-CD' s runtime remains acceptable.

---

4 Conclusion

To achieve accurate clustering without prior knowledge of cluster numbers, this paper employs real-valued, fixed-length individual encoding for automatic clustering. To effectively avoid erroneous clustering caused by algorithmic randomness, the Multi-Objective Differential Evolution Automatic Clustering algorithm based on Class-Center Density (MODEAC-CD) is proposed. The class-center density strategy screens out randomly selected centers that deviate from the dataset or are overly concentrated before clustering, preventing their participation. To rapidly obtain optimal centers, an improved clustering criterion function dynamically penalizes cluster numbers. Comparative experiments with two state-of-the-art automatic clustering algorithms and the classic RLCLu algorithm on two dataset types demonstrate that MODEAC-CD achieves superior clustering results, with cluster numbers matching or closer to true values, higher ARI performance, and greater accuracy, validating the feasibility and effectiveness of the proposed strategies.

However, MODEAC-CD's clustering performance on certain complex-structured datasets remains suboptimal. For example, the glass dataset contains clusters with vastly different sizes and relatively tight data distributions, posing clustering challenges. Future research will focus on selecting more effective clustering mechanisms to address such datasets.

References

[4] Swagatam D, Ajith A, Amit K. Automatic clustering using an improved differential evolution algorithm [J]. IEEE Trans on Systems, 2008, 38(1): 218-237.

[5] Saha I, Maulik U, Bandyopaghyay S. A new differential evolution based fuzzy clustering for automatic cluster evolution [C]// Proc of IEEE International Advance Computing Conference. 2009: 706-711.

[6] Maulik U, Saha I. Automatic fuzzy clustering using modified differential evolution for image classification [J]. IEEE Trans on Geoscience and Remote Sensing, 2010, 48(9): 3503-3510.

[7] Rodriguez A, Laio A. Clustering by fast search and find of density peaks [J]. Science, 2014, 344(6191): 1492-1496.

[8] 李涛, 葛洪伟, 苏树智. 自动确定聚类中心的密度峰聚类 [J]. 计算机科学与探索, 2016, 11(10): 1614-1622. (Li Tao, Ge Hongwei, Su Shuzhi. Automatic determination of density peak clustering in cluster center [J]. Computer Science and Exploration, 2016, 11(10): 1614-1622.)

[9] Ye Xuanzuo, Li Dinghao, He Xiongxiong. An algorithm for automatic recognition of cluster centers based on local density clustering [C]// Proc of the 29th Chinese Control and Decision Conference. 2017: 1347-1351.

[10] 武小龙. 基于改进的差分进化自动聚类算法研究 [D]. 西安: 西安电子科技大学, 2013. (Wu Xiaolong. Research on improved automatic clustering algorithm based on differential evolution [D]. Xi' an: Xidian University, 2013.)

[11] 李建. 聚类融合研究及其应用 [D]. 哈尔滨: 哈尔滨工程大学, 2014. (Li Jian. Research and application of cluster fusion [D]. Harbin: Harbin Engineering University, 2014.)

[12] 张素洁, 赵怀慈. 最优聚类个数和初始聚类中心点选取算法研究 [J]. 计算机应用研究, 2017, 34(6): 1617-1620. (Zhang Sujie, Zhao Huaici. Research on optimal clustering number and initial clustering center selection algorithm [J]. Application Research of Computers, 2017, 34(6): 1617-1620.)

[13] 黎凡, 王新, 和晓萍. 一种基于局部密度的 K-means 算法 [J]. 云南民族大学学报: 自然科学版, 2014, 23(6): 439-442. (Li Fan, Wang Xin, He Xiaoping. A K-means algorithm based on local density [J]. Journal of Yunnan Minzu University: Natural Science Edition, 2014, 23(6): 439-442.)

[14] Ding Jie, Noshad M, Tarokh V. Learning the number of autoregressive mixtures in time series using the gap statistics [C]// Proc of the 15th IEEE International Conference on Data Mining. 2015: 1441-1446.

[15] Nafchi H Z, Shahkolaei A. Mean deviation similarity index: efficient and reliable full-reference image quality evaluator [J]. IEEE Access. 2016(4): 5579-5590.

[16] Zitzler E, Kunzl S. Indicator-based selection in multi-objective search, in parallel problem solving from nature [C]// Proc of International Conference on Parallel Problem Solving from Nature. 2004: 832–842.

[17] Wang Handing, Jiao Licheng, Yao Xin. Two_Arch2: an improved two-archive algorithm for many-objective optimization [J]. IEEE Trans on Evolu-

tionary Computation, 2015, 19(4): 524-541.

[18] Aggarwal C C, Hinneburg A, Keim D A. On the surprising behavior of distance metrics in high dimensional space [C]// Proc of International Conference on Database Theory. 2001: 420-434.

[19] Zhang Li, Zhang Chengjin, Xu Qingyang, et al. Weighted-KNN and its application on UCI [C]// Proc of IEEE International Conference on Information and Automation. 2015: 1748-1750.

[20] Kundu D, Suresh K, Ghosh S. Automatic clustering using a synergy of genetic algorithm and multi-objective differential evolution [C]// Proc of International Conference on Hybrid Artificial Intelligence Systems. 2009: 336-343.

[21] Bandyopadhyay S, Saha S. A point symmetry based clustering technique for automatic evolution of clusters [J]. IEEE Trans on Knowledge and Data Engineering, 2008, 20(11): 1–17.

[22] Gao Bo, Wang Jun. Multi-objective fuzzy clustering for synthetic aperture radar imagery [J]. IEEE Geoscience and Remote Sensing Letters, 2015, 12(11): 2341-2345.

[23] 龚文引. 差分演化算法的改进及其在聚类分析中的应用研究 [D]. 武汉: 中国地质大学, 2010. (Gong Wenyin. Improvement of differential evolution algorithm and its application in clustering analysis [D]. Wuhan: China University of Geosciences, 2010.)

[24] Ashok P, Kadhar G M. Detecting outliers on uci repository datasets by adaptive rough fuzzy clustering method [C]// Proc of Online International Conference on Green Engineering and Technologies. 2016: 1-6.

[25] Park S, Choi H, Lee B. hc-OTU: a fast and accurate method for clustering operational taxonomic units based on homopolymer compaction [J]. IEEE/ACM Trans on Computational Biology and Bioinformatics, 2018, 15(2): 441-451.

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