Clustering Algorithm of Density Difference Optimized by Mixed Teaching and Learning

Hailong Chen¹ · Miaomiao Ge¹ · Yutong Xue¹

Received: 9 February 2020 / Accepted: 27 April 2020 / Published online: 16 May 2020
© The Author(s) 2020

Abstract

Density peak clustering (DPC) algorithm is to find clustering centers by calculating the local density and distance of data points based on the distance between data points and the cutoff distance (d_c) set manually. Generally, the attribute calculation between data points is simply obtained by Euclidean distance. However, when the density distribution of data points in data sets is uneven, there are high-density and low-density points, and the d_c value is set artificially and randomly, this will seriously affect the clustering results of DPC algorithm. For this reason, a clustering algorithm which combines teaching and learning optimization algorithm and density gap is proposed (NSTLBO-DGDPC). First, in order to consider the influence of data point attributes and neighborhoods, the density difference distance is introduced to replace the Euclidean distance of the original algorithm. Secondly, because manual selection of clustering centers may produce incorrect clustering results, the standard deviation of high-density distance is used to determine the clustering centers of clustering algorithm. Finally, using the teaching and learning optimization algorithm (TLBO) to find the optimal value, in order to avoid the algorithm falling into local optimum. When the population density reaches a certain threshold, the niche selection strategy is introduced to discharge the similarity value, and then the nonlinear decreasing strategy is used to update the students in the teaching stage and the learning stage to obtain the optimal d_c solution. In this paper, the accuracy and convergence of the improved TLBO algorithm (NSTLBO) are verified by ten benchmark functions. Simulation experiments show that the NSTLBO algorithm has better performance. Clustering algorithm integrating teaching and learning optimization algorithm and density gap proposed in this paper are validated by using eight synthetic data sets and eight real data sets. The simulation results show that the algorithm has better clustering quality and effect.

Keywords Clustering · Density gap · Niche · Nonlinear decreasing strategy · TLBO

Introduction

With the advent of the era of big data, it is becoming more and more important to obtain valuable and potential knowledge and information from massive data. Cluster analysis is a multivariate statistical method that is divided into groups. According to the degree of similarity between each abstract object, it is divided into several groups, and similar objects are combined into one set. Clustering [1] is a process in which each data point in a data set is aggregated to several centers of the same feature. That is, the process of dividing a

data set into classes consisting of similar objects. Therefore, cluster analysis is often used as a preprocessing for other data mining operations [2]. Because cluster analysis techniques can mine useful and implicit undiscovered information and knowledge from large amounts of raw data, it has been widely used in many applications, including image pattern recognition, Web search, biology and security, and data analysis, text mining and other fields. Generally, clustering algorithm based on partition, clustering algorithm based on hierarchy, clustering algorithm based on density, clustering algorithm based on grid, followed by neural grid clustering algorithm, clustering algorithm based on statistics and fuzzy clustering algorithm [3–9]. *K*-means algorithm is one of the most widely used clustering algorithms based on partition [10]. The Euclidean distance method is used to calculate the minimum of the sum of the squares of the distances between the data points and the cluster center points as the objective

✉ Hailong Chen
hrbustchl@163.com

¹ Department of Computer Science and Technology, Harbin University of Science and Technology, Harbin 150080, China

function of the algorithm. The K -means algorithm has a fast convergence speed, a simple algorithm and can effectively process large data sets [11], however, it is necessary to manually determine the number of clusters and randomly select the initial cluster centers. The DBSCAN algorithm [12] is the most classic density-based clustering algorithm. The DBSCAN algorithm requires two parameters: the density threshold parameter and the distance parameter. The disadvantage is that the parameter affects the cluster quality and cannot process the high-dimensional data well. Secondly, if the sample set has uneven density and large difference between clustering spacing, it will lead to poor clustering quality. Another density clustering algorithm is a new density peak clustering algorithm DPC (density peak clustering) proposed by Rodriguez and Laio [13]. Although the algorithm is simple and efficient, there is no need to calculate the objective function iteratively to determine the cluster center. However, the cutoff distance and cluster center of the algorithm must be manually set or randomly set according to experience. The choice of cutoff distance determines the algorithm. The calculation of local density and relative distance, which in turn, affects the clustering results. Therefore, finding the optimal value of the cutoff distance is a challenge. In order to improve DPC algorithm, some scholars have put forward their own ideas in the past 3 years. There are many ways to set truncation distance manually in these studies. For example, Xu [14] proposed the k -nearest neighbor algorithm based on Gauss kernel to calculate the local density and use the difference vector to automatically select the cluster center. Du [15] proposed to combine FJP algorithm and use fuzzy neighborhood relations to define fuzzy joint points to calculate local density. Li [16] proposed a new comparative model to replace the calculation of the original relative distance in the clustering of comparative density peaks. Bai [17] proposed an accelerated algorithm based on concept approximation with less distance calculation and then proposed an approximate algorithm based on sample clustering to obtain the approximate clustering results of the original algorithm quickly in order to take into account the scalability of the algorithm. Swarm intelligence algorithm has great advantages in solving global optimum and is insensitive to parameters, so swarm intelligence algorithm is used to optimize the intercept and clustering center. Swarm intelligence algorithms include ant colony optimization algorithm [18], artificial bee colony algorithm [19], firefly algorithm [20], particle swarm optimization algorithm [21], teaching and learning optimization algorithm [22], etc. Among them, the TLOB algorithm is an optimization tool inspired by the knowledge transfer mechanism of teachers and students in the classroom. Compared with other optimization algorithms, the most prominent feature of TLBO algorithm is that special control parameters that do not need to be adjusted before execution. The PSO algorithm needs

to adjust speed and inertia weights, and the ABC algorithm requires three individuals. GA algorithm needs to adjust crossover rate and mutation rate. However, the TLBO algorithm has the disadvantages of being easily trapped in a local optimum and slow convergence. In order to increase the diversity of new populations, Zou [23] proposed a teaching and learning optimization algorithm (DRLTLBO) for differential and exclusion learning. Ender [24] proposed to combine extreme machine learning with TLBO to solve data classification problems. Kumar [25] used a variety of chaotic mechanisms to solve the problem of TLBO algorithm lacking trade-off between premature convergence and local search and global search. Bourahla [26] used the random local search mechanism of the bat algorithm to the TLBO algorithm and proved its effectiveness. Niu [27] proposed an improved teaching-based optimization algorithm (MTLBO) based on the actual "teaching learning" situation to speed up the convergence of the algorithm. Zhang [28] integrated the number spiral strategy and the triangle mutation rule into the TLBO algorithm in order to improve the shortcomings of the TLBO convergence rate and the tendency to fall into the local optimal solution. At present, many learners apply group intelligence algorithms combined with clustering algorithms to data mining. Anari proposed a new clustering algorithm based on learning automata and ant colony intelligence and solved the problem of optimum placement of data items in grid by LA method [29]. Majdi used genetic algorithm and particle swarm optimization to optimize the pre-determined FCM clustering model, which improves the accuracy of model estimation [30]. Zang [31] proposed a new clustering method based on potential entropy of data field to extract the optimal threshold value and uses Gauss method to automatically determine the clustering center. In this paper, TLBO has the advantages of fast convergence and fast search speed to optimize the cutoff distance in DPC algorithm. On this basis, a clustering algorithm combining TLBO algorithm and density gap is proposed (NSTLBO-DGDPC). Because Euclidean distance calculation does not consider the influence of data point attributes and neighborhood, a density gap distance instead of Euclidean distance. Then, the standard deviation of high-density distance is used to determine the clustering center of the clustering algorithm, which can avoid the influence of manual selection of clustering center on clustering results. Finally, in order to avoid the TLBO algorithm falling into local optimum, niche selection strategy is introduced to exclude similar values when the population density reaches a certain threshold, and then the nonlinear decline strategy is used to update the individual students in the teaching and learning stages to find the optimal d_c value.

The main contributions of this paper are summarized as follows:

- (1) In this paper, we proposed to use the weighted density difference distance instead of the Euclidean distance to calculate the local density of the algorithm.
- (2) In order to be able to correctly select the clustering center, the final clustering center is selected by using the standard deviation.
- (3) In order to solve the DPC algorithm manually set the truncation distance, use the TLBO algorithm to find the optimal d_c value.
- (4) TLBO algorithm is easy to fall into local optimal. Therefore, in this paper, we propose to introduce a niche selection strategy to eliminate similar values when the population density reaches a certain threshold and then use a nonlinear decreasing strategy to update the individual students in the teaching and learning stages and finally obtain the optimal d_c solution.
- (5) NSTLBO algorithm improves the accuracy and convergence of the algorithm. The simulation function of the benchmark function proves that the NSTLBO algorithm has better performance.
- (6) Using the synthetic data set and the real data set to verify the proposed fusion teaching and learning optimization algorithm and the clustering algorithm of density difference distance, it is shown that the algorithm can calculate the number of real data sets without manually setting the d_c value. The ACC, AMI, and ARI values of the proposed algorithm have improved, indicating that the algorithm has better clustering quality.

The organization of this paper is as follows: in the second section, the DPC algorithm and the TLBO algorithm are briefly introduced. In third section, a clustering algorithm that combines teaching and learning optimization algorithms and density difference distances is introduced. In subsection, the density difference distance and standard deviation optimization selection clustering center of the clustering algorithm are introduced. The improved TLBO algorithm (NSTLBO) is introduced in subsection. The algorithm steps are described in subsection. The simulation experiments and analysis of the algorithm in this paper are discussed in fourth section. Finally, the fifth section covers conclusion and future work.

Preliminaries

Density Peak Clustering Algorithm DPC

The algorithm is based on two simple and intuitive assumptions. Cluster centers are surrounded by neighborhoods with lower local density, while the distance between cluster centers and any points with higher local density is relatively large. According to these two hypotheses, the algorithm first finds

the potential density peak, labels it as the cluster center and then assigns other points to the cluster. For each data point x_i , DPC clustering algorithm must calculate the distance between local density ρ_i and high-density point δ_i . The local density of data points is defined as Eq. (1):

$$\rho_i = \sum_j \chi(d(x_i, x_j) - d_c), \quad \chi(x) = \begin{cases} 1, & x < 0 \\ 0, & x \geq 0 \end{cases} \quad (1)$$

d_c is cutoff distance. $d(x_i, x_j)$ is the Euclidean distance between data point x_i and data point x_j . The δ_i of data point x_i is determined by Eq. (2).

$$\delta_i = \begin{cases} \min(d(x_i, x_j)), & \rho_j > \rho_i \\ \max(d(x_i, x_j)), & x_j \in X, \text{ otherwise} \end{cases} \quad (2)$$

Teaching and Learning Optimization Algorithms

TLBO algorithm is proposed inspired by the teaching process. It has two important stages: teaching stage and learning stage. The algorithm uses these two stages to influence the population to find the optimal solution. The results of each student in the class are mapped to the solution space. The results of each student are the fitness value of the solution. The best students are regarded as teachers, and then other students are taught to improve their learning performance. The space dimension of the solution is d , the total number of students (population) is n , x_i is the i th student, $f(x_i)$ is the academic achievement (fitness value), and $x_{teacher}$ is the teacher. $mean = \frac{1}{n} \sum_{i=1}^n X_i$ is expressed as the average of students.

In the teaching stage: among all the students, the students with the best scores are selected as the teachers of the next round of learning process. The teachers improve the performance of the whole class by teaching the students' knowledge to make the students' grades close to their own learning level. Any student in the class after the teacher stage, the difference between the teacher and student averages in the t th iteration is shown in Eq. (3):

$$Difference(t) = r_i \times (x_{teacher}(t) - TF \times mean(t)) \quad (3)$$

$$TF = round[1 + rand(0, 1)], \quad r_i = rand(0, 1) \quad (4)$$

where TF takes the value 1 or 2, and t is the number of iterations. The resulting description of the teacher knowledge level update is as shown in Eq. (5):

$$x_{i+1}(t) = x_i(t) + Difference(t) \quad (5)$$

$$\text{If } f(x_{i+1}(t)) > f(x_i(t)), \text{ then } x_i(t) = x_{i+1}(t), \text{ otherwise } x_i(t) = x_i(t) \quad (6)$$

In the learning phase: any student in the class randomly searches for other students x_j in the class according to their

own understanding of the learning content and enhances their knowledge understanding ability. If the student's x_j score is better than the student's x_i score, learn from the student x_j and improve his or her knowledge. In the learning phase, the student update description formula is obtained by communicating with different students as shown in Eqs. (7, 8):

$$\text{If } f(x_i(t)) < f(x_j(t)), \text{ then } x_{i+1}(t) = x_i(t) + r_i(x_j(t) - x_i(t)) \quad (7)$$

$$\text{If } f(x_i(t)) > f(x_j(t)), \text{ then } x_j(t) = x_j(t) + r_i(x_i(t) - x_j(t)) \quad (8)$$

Algorithm 1: TLBO algorithm

Step 1: Initialize the student population n , dimension d ;
Step 2: Calculate the student average *mean*, $x_{teacher}$;
Step 3: Update the student individual by using Eq. (5), and calculate the updated student individual score;
Step 4: If the individual score of the student is improved after the update, the student individual replaces the student individual before the update with Eq. (6);
Step 5: Randomly select one student individual to compare with another student individual and use Eqs. (7) and (8) to update the student individual and calculate the student individual score;
Step 6: If the individual achievement of the student is improved after the update, the updated student individual replaces the student individual before the update;
Step 7: If the algorithm satisfies the iterative condition, input the optimal student individual;

Niche Selection Strategy

Niche is originated from biology. That is to say, the same kind of creatures forms a small living environment and separate different kinds of individuals [32]. This idea is applied to the neighborhood of computing science, which is to divide the data with the same data characteristics into one kind and separate the data with different characteristics. DeJong [33] proposed a niche method based on the exclusion mechanism in 1975, that is, in a limited space, various biological individuals can survive and compete for survival resources. Applying this idea to intelligent algorithms can prevent the algorithm from premature convergence and improve population diversity.

Algorithm 2: Selection strategy based on crowding mechanism niche algorithm

Step 1: Set the total number of groups, the crowding factor CF (value 2 or 3);

Step 2: randomly select $1/CF$ individuals of the total population as the crowded individuals;

Step 3: Execute Eq. (9) to calculate the Hamming distance between the new individual and the crowded individual:

$$L = \|x_i - x_j\| = \sum_{k=1}^N (x_{k,i} - x_{k,j})^2 \quad (9)$$

Step 4: Based on the similarity of Hamming distance, the individuals similar to the crowded individuals are excluded to form a new current group;

Clustering Algorithm Integrating Teaching and Learning Optimization Algorithm and Density Gap

Standard Deviation Optimization Selection Cluster Center

The DPC algorithm uses the local density and relative distance of the data set to determine the cluster center. It can be seen that the similarity measure between objects is the key of clustering, and the calculation of its properties is calculated by simple Euclidean distance. When the data set density is different, the clustering result will be greatly affected. Assumption $d(x_1, x_2) = d(x_2, x_3) = d(x_3, x_4)$ Since the DPC algorithm uses the Euclidean distance to measure the similarity, the similarity between x_2 and x_3 is equal to the similarity between x_3 and x_4 . Generally, the similarity of similar data points is large, and the similarity of data points of different classes is small. Secondly, if x_1 and x_2 belong to high-density, x_3 and x_4 belong to low density. According to the literature [34], the density of similar or co-manifold data points is similar, the similarity is large, and the density of different types of data points is higher than that of similar data points. The density differs greatly and the similarity is small. Therefore, such clustering is incorrect. In [35], the similarity metric should be influenced by the environment and neighbors. Therefore, the reciprocal of the weighted distance of the K nearest neighbors of any data point can be considered as the neighborhood density, according to the neighborhood density difference between the data points, adjust the distance between the two points. This takes into account both the manifold neighborhood of the data points and less noise.

In order to describe the method we propose, suppose the data set $D = \{x_1, x_2, \dots, x_n\} \in R^{n \times m}$ is to be determined.

Definition 1 The distance between any two data points is:

$$d(x_i, x_j) = \sqrt{(x_i - x_j)^T (x_i - x_j)}$$

In order to consider the neighborhood information of the data point, the Euclidean distance of the k th neighbor x_k of

any data point x_i is used, but the data point x_i is susceptible to noise, and therefore, the Weighted distance of the k th neighbor x_k of the data point x_i is used.

Definition 2 The weighted distance of K data points in the k -nearest neighbor of any data point x_i is:

$$\varphi_{x_i} = \sum_{j=1}^K \omega(x_i, x_j) d(x_i, x_j), \text{ in } \omega(x_i, x_j) = 1 - \ln \left(\frac{d(x_i, x_j)}{\sum_{k=1}^K d(x_i, x_k)} \right)$$

Definition 3 Density of arbitrary data points

$\theta_{x_i} = \frac{1}{\phi_{x_i}}$. When the density of the neighborhood of the data point x_i is high and the distribution is dense, ϕ_{x_i} is small. When the density of the neighborhood where the data point x_i is located is low and the distribution is sparse, ϕ_{x_i} is large.

Definition 4 Density difference distance

$d'(x_i, x_j) = d(x_i, x_j) \left(1 + \frac{|\theta_{x_i} - \theta_{x_j}|}{\theta_{\max}} \right)$, in $\theta_{\max} = \{ |\theta_{x_i} - \theta_{x_j}|; i, j = 1, \dots, n \}$ is the maximum value of the density difference, The ratio of $|\theta_{x_i} - \theta_{x_j}|$ and $\theta_{\max}$ adjusts the distance. If the difference in density is larger, the similarity is smaller, resulting in a density difference distance. The local density ρ_i of the data point x_i becomes a formula:

$$\rho_i = \sum_j \chi(d'(x_i, x_j) - d_c) \quad (10)$$

In order to solve the problem that the DPC algorithm is difficult to obtain an accurate clustering center by manually selecting the clustering center by using the decision graph, this paper combines the method proposed into select the exact cluster number without manual intervention to optimize the algorithm, which is based on the following Eqs. (11) and (12):

$$EC_i = (\delta_i) \geq 2\sigma(\delta_i) \quad (11)$$

EC_i is the expectation of the cluster center, and $\sigma(\delta_i)$ is the standard deviation of all high-density distances calculated according to Eq. (12).

$$DC_i = EC_i \geq \mu(\rho_i) \quad (12)$$

DC_i is the cluster center after separation, and $\mu(\rho_i)$ is the mean of ρ_i . Because the DPC algorithm may generate a large δ_i value and a low density cluster center, consider using Eq. (12) to separate this cluster center from the cluster class center of EC_i . In this way, DC_i has a higher density and a higher density of higher density than adjacent data points. After determining the cluster center, the remaining points are assigned to the points with the smallest distance between all the high local density points.

Improvement of Teaching and Learning Optimization Algorithm

Nonlinear Decreasing Strategy Learning

Because the TLBO algorithm focuses on the learning mode of global exploration in the early stage, the local development ability of the algorithm is weakened in the search process. Although under the influence of teaching factors, with the increasing number of iterations of the algorithm, it may also fall into the local optimum situation. In order to enable the algorithm to learn adaptively in both teaching and to increase the search range and prevent local optimal deadlock, this paper proposes to introduce the weight of the nonlinear decrement strategy in the teacher stage and the student stage, as shown in Eq. (13).

$$w(t) = \frac{1}{2} \left[(w_{\max} - w_{\min}) \left(1 - \frac{t}{t_{\max}} \right)^2 + (w_{\max} - w_{\min}) \left(\frac{t_{\max} - t}{t_{\max}} \right) \right] + w_{\min} \quad (13)$$

where $w_{\max}$ is the maximum weight and $w_{\min}$ is the minimum weight.

In this way, the balance algorithm uses the larger weight to explore more regions in the early stage and uses the smaller weights to better perform local development in the later stage. The improved individual description formula of the student stage is as shown in (14):

$$x_{i+1}(t) = w(t) \times x_i(t) + \text{Difference}(t) \quad (14)$$

In the learning phase, the student individual x_i and the different student individual x_j exchange learning and get the knowledge update description formula as shown in (15, 16):

$$x_{i+1}(t) = w(t) \times x_i(t) + r_i(x_i(t) - x_j(t)), f(x_i(t)) < f(x_j(t)) \quad (15)$$

$$x_j(t) = w(t) \times x_j(t) + r_i(x_i(t) - x_j(t)), f(x_i(t)) > f(x_j(t)) \quad (16)$$

By introducing dynamic $w(t)$, the algorithm dynamically updates the knowledge level of teachers teaching individual students, increases the diversity of the population and the search range of the algorithm, thereby reducing the situation that the algorithm falls into local optimum and speeds up the convergence.

Opportunity for Niche Selection Strategy

The teaching and learning optimization algorithm may fall into local optimum in the later stage of iteration, which

leads to the problem of population diversity decline. Therefore, the niche strategy is used to improve the shortcomings of teaching and learning optimization algorithms that are easy to fall into local optimum. In the process of increasing the number of iterations, if the algorithm appears that a large number of individual adaptive function values are close together and then come together, a niche selection strategy is introduced at this time. In order to make the teaching and learning optimization algorithm and the niche selection strategy better combined, a variable is needed to measure the degree of aggregation between individuals. In this paper, the discrete coefficient in statistics is used to describe the aggregation of the population Eq. (17):

$$g(t) = \frac{1}{f_{avg}(t)} \times \sqrt{\frac{1}{n} \sum_{i=1}^n (f(x_i(t)) - (f_{avg}(t)))^2} \quad (17)$$

where $f(x_i(t))$ represents the fitness function value of the current i th student individual; $f_{avg}(t)$ is the current student population fitness average, and t is the number of iterations. Discrete coefficients can be used to reflect the degree of dispersion between individuals within a population. In the process of teaching and learning, the learning knowledge level of students is likely to approximate the same value, and the fitness function values may gradually become similar, resulting in the $g(t)$ value changing. Small, the degree of dispersion is getting lower and lower, which indicates that the population is getting more and more concentrated at this time. Set a threshold q , which is usually between 0 and 0.3 depending on the actual situation. When $g(t) < q$, it indicates that the population aggregation is high, and it is necessary to introduce a niche selection strategy, that is, randomly select 30% of the total population to form the crowded members and then compare the similarity between the newly generated population and the crowded members after adding $w(t)$, crowd out the students with high similarity, so as to avoid falling into the local optimum situation and maintain the diversity of the whole population.

Algorithm Implementation

Algorithm 3: NSTLBO algorithm

Step 1–Step 6: This process is consistent with the original TLBO algorithm, see “[Teaching and Learning Optimization Algorithms](#)” section;

Step 7: Determine the degree of population aggregation. If $g(t)$ is less than q , execute Algorithm 2, “[Niche Selection Strategy](#)” section. Continue to step 4;

Step 8: update the student individual by using Eq. (14), and calculate the updated student individual score;

Step 9: If the individual achievement of the student is improved after the update, the student individual replaces the student individual before the update by using Eq. (6);

Step 10: randomly select one student individual to compare with another student individual, and use Eqs. (15) and (16) to update the student individual and calculate the student individual score;

Step 11: If the individual score of the student is improved after the update, the updated student individual replaces the student individual before the update;

Step 12: $t = t + 1$; satisfy the iterative condition and input the optimal student individual;

Clustering Algorithm Integrating Teaching and Learning Optimization Algorithm and Density Gap

The contour index can reflect the compactness and separation of data points. Therefore, this paper uses the contour index as the fitness function of the swarm intelligence algorithm, in order to find the best d_c value. The contour index of the sample o is as shown in Eq. (18).

$$S(o) = \frac{[b(o) - a(o)]}{\max\{a(o), b(o)\}} \quad (18)$$

First, divide the data set of n sample points in the data set into k clusters, $c_i (i = 1, 2, \dots, k)$, $a(o)$ is the average difference or distance between the sample point o in cluster c and all other sample stores in their class. $d(o, c_i)$ is the distance between the sample point o and another class c_i of all samples, $b(o) = \min(d(o, c_i))$ Where $i = 1, 2, \dots, k$ and $i \neq j$. The $S(o)$ value reflects the quality of the clustering results. If the value of $S(o)$ is larger, the clustering quality of the clustering algorithm is better.

The algorithm idea of this paper is to first set the cutoff distance d_c to make the data object average local density as data according to the literature [11]. Then, randomly select a d_c value in this interval $[d_{low}, d_{high}]$, calculate the density difference distance $d'(x_i, x_j)$ by using the Definitions 1–4, calculate the local density ρ_i and the relative distance δ_i according to Eqs. (2) and (10), and then according to Eqs. (11) and (12) Select cluster center, the clustering result is evaluated by the fitness function, that is, the contour index $S(o)$, the clustering is good or bad. Repeat the different d_c values in the interval until the optimal d_c value is selected. The whole process is to use the NSTLBO algorithm as the medium to select the optimal d_c solution, and the optimal d_c value is substituted into the improved density peak clustering algorithm, and finally the clustering result is obtained. The algorithm steps are as follows:

Algorithm 4: NSTLBO-DGDPC algorithm

Step 1: Set the d_c interval [dlow, dhigh], initialize the population size of the NSTLBO algorithm 10, the number of iterations 10, and the threshold q is 0.2;

Step 2: randomly select a d_c value, and calculate the density difference distance $d'(x_i, x_j)$ by using the Definitions 1–4;

Step 3: Calculate the local density ρ_i and the relative distance δ_i according to f Eq. (10) and Eq. (2);

Step 4: Select a cluster center according to Eq. (11) and Eq. (12);

Step 5: According to the fitness Eq. (18), determine the fitness function value corresponding to each student;

Step 6: Run the NSTLBO algorithm to find the current optimal student individual and calculate the individual fitness function value of the student;

Step 7: Determine the current population aggregation degree, and if the aggregation degree is less than the threshold, use the niche selection strategy to form a new group;

Step 8: Perform the teaching phase: use the non-decreasing learning strategy to update the student individual and calculate the updated student individual fitness function value;

Step 9: If the updated student individual score is improved, the student individual updated by Eq. (14) replaces the student individual before the update;

Step 10: Execution stage: randomly select two different individual students to compare the academic scores, and use Eqs. (15) and (16) to update the individual students and calculate the individual student scores;

Step 11: If the algorithm reaches the maximum number of iterations, the output is a good value, otherwise it returns to step 2; if the algorithm does not reach the maximum number of iterations, the algorithm is terminated, and the optimal value is output, and the process returns to step 5;

Simulation Experiment and Validation

According to the above solution, the flowchart of the algorithm is designed as shown in Fig. 1.

NSTLBO Algorithm Experimental Proof Validation

Firstly, the optimization performance of NSTLBO is verified. Secondly, the clustering performance of the proposed algorithm is verified. All the experiments mentioned in this paper are compiled and debugged under Windows 1064 bit operating system through MATLAB R2014a. The hardware environment is AMD A10 (2.30 GHz) with 8 GB of memory.

In order to verify the optimization performance of NSTLBO algorithm, this paper uses PSO algorithm, ABC algorithm, TLBO algorithm, and CTLBO algorithm to test and compare with NSTLBO algorithm. In this paper, ten standard test functions are selected from CEC, and the

Fig. 1 The flowchart of algorithm

performance of the algorithm is verified. Among them, f_1 – f_3 , f_7 , f_8 is a continuous unimodal function, f_4 – f_6 , f_9 , f_{10} is a continuous multimodal function. For single peak function, there is only one global optimal value, which is used to test the development ability of the algorithm. Multi-peak function has many local optimal solutions, which is used to study the search ability of the algorithm. In order to verify the performance of the algorithm, it is necessary to calculate the average value and standard deviation of the test function. The smaller the average, the better the algorithm. The lower standard deviation indicates that the algorithm is more stable, and its expression and related parameters are shown in Table 1.

Assuming that the population size is set to 50, the maximum number of iterations is set to 500, and the dimensions are 50 and 100, respectively. Inertial weight $w = 0.7289$, $c_1 = c_2 = 1.5968$ for PSO algorithm. The $w_{\max}$ of NSTLBO algorithm is 0.9, $w_{\min}$ is 0.1, q is 0.3. In the experiment, each algorithm runs 50 times for each test function. The optimal value, the optimal average value, and the optimal variance obtained by the algorithm are shown in Table 2 (50 dimensions) and Table 3 (100 dimensions), bold fonts are the best results of algorithms in experiments. In order to further visually compare the optimization performance of the algorithm, the average fitness evolution curves and statistical box diagrams of f_1 – f_{10} are drawn as shown in Figs. 2, 3, 4, and 5.

As can be seen from the data in Table 2, on the test function, f_1 , although f_1 is a function that is relatively easy to optimize. However, the average and standard deviation of ABC and PSO are relatively poor. The NSTLBO algorithm proposed in this paper achieves the global optimal value both in the mean and standard deviation. On the test function f_2 , f_2 is a unimodal function that is difficult to optimize. The optimal value of the NSTLBO algorithm is lower than the optimal value of the PSO. The average value is slightly lower than the CTLBO algorithm, and the most obvious is the NSTLBO algorithm. The standard deviation is the highest compared to other algorithms. On test functions f_3 and f_4 , where f_4 is a multimodal function, the NSTLBO algorithm maintains a good performance on f_1 , and both the mean and standard deviation reach a global optimal value of zero. On the test function f_5 , the average and standard deviation of the solution of the NSTLBO algorithm are higher than those of the ABC and PSO algorithms, which are slightly higher than the TLBO and CTLBO algorithms, so the NSTLBO algorithm performs generally on the test function. On the test function f_6 , from the average and standard deviation of the solution, it is superior to the other four algorithms. On the test function f_7 , the average value of the NSTLBO algorithm is smaller than the average value of the ABC algorithm, larger than the average values of the TLBO and CTLBO

Table 1 Details of test function from CEC

Function name	Function	Search space	$f_{\min}$
Sphere	$f_1 = \sum_{i=1}^D x_i^2$	[100, 100]	0
Rosenbrock	$f_2 = \sum_{i=1}^{D-1} \left(100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right)$	[−30, 30]	0
Schwefel2.22	$f_3 = \sum_{i=1}^D x_i + \prod_{i=1}^D x_i $	[100, 100]	0
Rastrigrin	$f_4 = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10)$	[5.12, 5.12]	0
Ackley	$f_5 = 20 - 20 \exp \left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} - \exp \left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i) \right) \right) + e$	[−32, 32]	0
Girewank	$f_6 = 1 + \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos \left(\frac{x_i}{\sqrt{i}} \right)$	[−600, 600]	0
Schwefel2.26	$f_7 = \sum_{i=1}^D -x_i \sin \left(\sqrt{ x_i } \right)$	[−500, 500]	0
Quaritic	$f_8 = \sum_{i=1}^D ix_i^2 + \text{rand}(0, 1)$	[−1.28, 1.28]	0
Step	$f_9 = \sum_{i=1}^D (x_i + 0.5)^2$	[−100, 100]	0
Zakharov	$f_{10} = \sum_{i=0}^D x_i^2 + \left(0.5 \left(\sum_{i=0}^D ix_i \right) \right)^2 + \left(0.5 \left(\sum_{i=0}^D ix_i \right) \right)^4$	[−10, 10]	0

Table 2 Comparisons of algorithm test results (50 Dim)

Test function	Function	Optimal value	Average value	Standard deviation
f_1	ABC	2.9803e-05	7.3298e-04	8.2928e-04
	PSO	1.3580e+00	1.6259e+01	1.5921e+00
	TLBO	1.9674e-90	7.3162e-90	4.1036e-89
	CTLBO	2.1627e-156	5.0469e-157	2.0770e-156
	NSTLBO	0e+00	0e+00	0e+00
f_2	ABC	1.0045e+02	1.4046e+02	3.4782e+01
	PSO	1.6022e+00	1.9556e+02	2.8586e+02
	TLBO	4.8919e+01	4.8902e+01	0.0364e+01
	CTLBO	4.8909e+01	4.8131e+01	0.0360e+01
	NSTLBO	4.8930e+01	4.8930e+01	0.0259e+00
f_3	ABC	3.1498e+01	3.4046e+01	2.6637e+00
	PSO	1.4807e+02	1.6634e+02	4.6136e+01
	TLBO	0e+00	0e+00	0e+00
	CTLBO	0e+00	0e+00	0e+00
	NSTLBO	0e+00	0e+00	0e+00
f_4	ABC	3.1995e-04	3.2200e-04	2.0894e-06
	PSO	3.0591e-02	3.8952e-02	4.1841e-02
	TLBO	0e+00	0e+00	0e+00
	CTLBO	0e+00	0e+00	0e+00
	NSTLBO	0e+00	0e+00	0e+00
f_5	ABC	0.0396e+00	0.1786e+00	0.1459e+00
	PSO	1.7793e+00	1.8472e+00	2.6726e-01
	TLBO	2.6645e-15	0.8002e+00	3.9589e+00
	CTLBO	2.6645e-15	2.5935e-15	5.0243e-16
	NSTLBO	1.2407e-15	2.5935e-15	5.0243e-16
f_6	ABC	-1.4801e+02	-1.4785e+02	0.1358e+00
	PSO	7.8877e+00	8.1844e+00	1.4755e+00
	TLBO	2.9145e-44	3.2037e-44	4.6041e-44
	CTLBO	1.2589e-78	5.9949e-78	1.5802e-77
	NSTLBO	1.4330e-128	1.4330e-128	2.3949e-128
f_7	ABC	0.0434e+00	0.0805e+00	0.3354e+00
	PSO	-5.8679e+02	-5.0630e+02	9.4352e+01
	TLBO	-6.05121e+02	-6.5721e+02	3.7343e+02
	CTLBO	-5.0230e+02	-5.0610e+02	2.9392e+02
	NSTLBO	-3.2136e+02	-3.2302e+02	2.3845e+02
f_8	ABC	0.5655e+00	0.7931e+00	0.2578e+00
	PSO	1.8565e+05	6.6908e+05	3.5422e+06
	TLBO	1.6122e+01	1.6802e+01	0.3414e+00
	CTLBO	1.6531e+01	1.6856e+01	0.4980e+00
	NSTLBO	1.6439e+01	1.6469e+01	0.3848e+00
f_9	ABC	2.0896e-04	9.3152e-04	6.9156e-04
	PSO	3.4050e+00	3.9331e+00	2.2876e+00
	TLBO	1.0323e+01	1.0323e+01	0.6206e+00
	CTLBO	9.9560e+01	9.9560e+01	0.9791e+00
	NSTLBO	9.8838e+01	9.8838e+01	0.9877e+00
f_{10}	ABC	1.8887e+01	2.0389e+01	2.3288e+01
	PSO	3.7056e+02	7.2686e+03	9.9594e+04
	TLBO	9.5346e-156	9.5544e-156	2.7530e-155
	CTLBO	1.6021e-194	1.6035e-194	0
	NSTLBO	2.1001e-257	2.1841e-257	0

Bold fonts are the best results of algorithms in experiments.

Table 3 Comparisons of algorithm test results (100 Dim)

Test function	Algorithm	Optimal value	Average value	Standard deviation
f_1	ABC	0.2886e+00	2.8668e+00	3.6105e+00
	PSO	2.4826e+00	2.9235e+00	3.1930e+00
	TLBO	1.5276e-156	1.5276e-157	9.1666e-156
	CTLBO	6.5803e-157	6.5803e-157	2.1749e-156
	NSTLBO	1.4840e-256	1.4840e-256	0e+00
f_2	ABC	1.9595e+02	1.9760e+02	1.4982e+00
	PSO	4.1344e+02	5.2056e+02	1.0055e+03
	TLBO	9.8920e+01	9.8920e+01	0.0438e+00
	CTLBO	9.8912e+01	9.8912e+01	0.0386e+00
	NSTLBO	9.8933e+01	9.8933e+01	0.0248e+00
f_3	ABC	1.9740e+02	2.1191e+02	1.2671e+01
	PSO	4.7554e+02	4.9742e+02	6.9089e+00
	TLBO	0e+00	0e+00	0e+00
	CTLBO	0e+00	0e+00	0e+00
	NSTLBO	0e+00	0e+00	0e+00
f_4	ABC	0.0744e+00	0.0856e+00	0.0106e+00
	PSO	7.1938e-02	8.2147e-02	4.3124e-02
	TLBO	0e+00	0e+00	0e+00
	CTLBO	0e+00	0e+00	0e+00
	NSTLBO	0e+00	0e+00	0e+00
f_5	ABC	2.2802e+00	2.3276e+00	0.0414e+00
	PSO	1.7145e+00	1.7876e+00	2.5059e-01
	TLBO	1.2020e-12	1.0204e-12	7.1973e-12
	CTLBO	0.4060e+00	0.4060e+00	2.8707e+00
	NSTLBO	1.5227e-15	1.5227e-15	1.6741e-15
f_6	ABC	-2.9265e+02	-2.8341e+02	8.7171e+00
	PSO	1.5301e+01	1.6102e+01	3.3719e+00
	TLBO	6.5143e-78	6.5143e-78	1.0341e-77
	CTLBO	2.7788e-78	2.7788e-78	2.7657e-78
	NSTLBO	1.1943e-128	1.1943e-128	2.2544e-128
f_7	ABC	2.6938e+00	3.87765e+00	1.20554e+00
	PSO	-2.6684e+02	-2.5234e+02	3.2958e+01
	TLBO	-9.4933e+02	-9.4932e+02	4.6518e+02
	CTLBO	-6.5670e+02	-6.5670e+02	3.9042e+02
	NSTLBO	-5.0117e+02	-5.0916e+02	2.6641e+02
f_8	ABC	1.7877e+00	2.0484e+00	0.3251e+00
	PSO	4.3057e+07	6.6316e+07	1.6656e+08
	TLBO	3.8051e+01	3.8051e+01	0.8414e+00
	CTLBO	3.84118e+01	3.84118e+01	0.6458e+00
	NSTLBO	3.7851e+01	3.7851e+01	0.5879e+00
f_9	ABC	0.2659e+00	0.5269e+00	0.4098e+00
	PSO	1.7777e+01	1.8467e+01	3.8249e+01
	TLBO	2.2769e+01	2.2769e+01	0.9242e+00
	CTLBO	2.26200e+01	2.26200e+01	0.7945e+00
	NSTLBO	2.2655e+01	2.2655e+01	0.7945e+00
f_{10}	ABC	4.2166e+01	4.5331e+01	3.9359e+00
	PSO	2.7300e+05	6.3276e+05	4.7108e+06
	TLBO	3.5531e-155	3.5531e-155	9.4794e-155
	CTLBO	1.8683e-194	1.8683e-194	0
	NSTLBO	2.6713e-255	2.6713e-255	0

Bold fonts are the best results of algorithms in experiments.

Fig. 2 Evolution curve of average fitness of algorithms (50 Dim)

algorithms, and the standard deviation is smaller than the values of the PSO, TLBO, and CTLBO algorithms. On the test function f_8 , the average and standard deviation of the NSTLBO algorithm are similar to the average and standard deviation of the PSO, TLBO, and CTLBO algorithms. On the test function f_9 , the performance of other algorithms is

better than that of NSTLBO algorithm. On the test function f_{10} , the average and standard deviation of the NSTLBO algorithm are significantly higher than those of other algorithms, indicating that the performance of the NSTLBO algorithm is better.

Fig. 3 Statistical box diagram of the average fitness of the algorithms (50 Dim)

As can be seen from the data in Table 3, the standard deviation of the NSTLBO algorithm on f_1 is 0 when the dimension of the function is increased to 100 dimensions. However, the overall solution ability on the test functions f_1 and f_2 is reduced, and it does not perform well in the 50-dimensional function. On the test function f_2 , compared with the other four algorithms, only the optimal value and

standard deviation of the solution of the CTLBO algorithm are excellent. For the NSTLBO algorithm, the other three algorithms have no NSTLBO algorithm with high precision. On the test functions f_3 and f_4 , the values of the mean and standard deviation of the solutions of the TLBO, CTLBO, and NSTLBO algorithms compared to the ABC and PSO algorithms are both zero. In the test functions f_5 and f_6 , the

Fig. 4 Average fitness of the algorithms (100 Dim)

mean and standard deviation of the solution and the function in the 50-dimensional state, the value does not have much amplitude, indicating that the NSTLBO algorithm has no change in value when the function dimension is 50-dimensional or 100-dimensional, and the performance It is normal on functions f_5 and f_6 . On the test function f_7 , the average

value of the NSTLBO algorithm is smaller than the average value of Table 2, and the standard deviation is larger than the standard deviation of Table 2, indicating that the stability of the function decreases when the dimension is high. On the test function f_8 , the average value of the NSTLBO algorithm is larger than the average value of Table 2. The

Fig. 5 Statistical box diagram of the average fitness of the algorithms (100 Dim)

standard deviation is smaller than the standard deviation of PSO, TLBO, and CTLBO, and larger than the standard deviation of ABC. On the test function f_9 , the average and standard deviation of the NSTLBO algorithm are similar to the values of PSO, TLBO, and CTLBO, which are smaller than the average and standard deviation of Table 2. On the

test function f_{10} , the average and standard deviation of the NSTLBO algorithm are significantly higher than the values of other algorithms, which are similar to the values of the CTLBO algorithm in Table 2, and smaller than the values of the ABC, PSO, and TLBO algorithms, indicating that the performance of the NSTLBO algorithm is better than other

algorithms. After the dimension becomes larger, in addition to the CTLBO algorithm, the stability of the NSTLBO algorithm becomes higher.

From the convergence curves on (a) (c), (d), (e), (f), and (j) in Fig. 2, it can be seen that the NSTLBO algorithm tests the functions $f_1, f_3, f_4, f_5, f_6, f_{10}$'s convergence curve declines fastest and can reach the optimal value quickly with fewer iterations. It can be seen from (a), (c), (d), (f), (g), (j) in Fig. 2 that the convergence curves under the test functions f_1, f_3, f_4, f_6, f_7 , and f_{10} are relatively smooth. It shows that the NSTLBO algorithm performs best on the test functions f_1, f_3, f_4 , and f_6 and performs generally on the test functions f_5, f_6 , and f_{10} . From the statistical box diagrams (a), (b), (e), (f), (g), (h), (i), and (j) in Fig. 3, it can be seen that the NSTLBO algorithm is testing the function f_1 . The sliding intervals on $f_2, f_5, f_6, f_7, f_8, f_9$, and f_{10} are larger. Except for f_8 and f_9 , the minimum values are smaller than those of other algorithms. Only f_3 and f_4 have good stability.

From the convergence curve in Fig. 4, it can be seen that when the function is in 100 dimensions, except for f_7, f_8 , and f_9 , the convergence curves of the NSTLBO algorithm are all under the other algorithms. It can be seen in Fig. 4b that when the number of iterations is far, when it is far less than 100, the curve has already become a straight line. In Fig. 4b, e, it is observed that when the number of iterations approaches 100, the curve becomes a straight line. Therefore, under the two test functions f_2 and f_5 , the NSTLBO algorithm, the convergence speed and solution accuracy of the two methods are basically unchanged. From the statistical box diagram in Fig. 5, the function dimension is increased to 100 dimensions, and the NSTLBO algorithm still has good stability on f_3 and f_4 . Except f_7, f_8 , and f_9 , it can be seen that the fitness value is the smallest compared to other functions. The fitness value of the NSTLBO algorithm is the smallest.

Analysis of the Experimental Proof of Algorithm NSTLBO-DGDPC

In order to verify the clustering performance of the proposed algorithm, DBSCAN algorithm and DPC algorithm are used to test and compare. In the experiment, the parameters $Eps=0.05$, $Minps=25$, and the parameters d_c or k use the values of Ref. [36]. In this paper, sixteen standard data sets including eight synthetic data sets and eight real data sets are selected as shown in Table 4. This paper chooses three commonly used clustering evaluation indicators: clustering accuracy (ACC), adjusting mutual information (AMI), and adjusting random index (ARI) [37] and carries out simulation experiments on three clustering algorithms. The experimental results of synthetic data sets are shown in Table 5. The real data set Table 6 shows the clustering effect of the composite data set as shown in Fig. 6.

Table 4 Data set

Data set	Name	Numbers	Dimensions	Classes
Synthetic data sets	D31	3100	2	31
	Flame	240	2	2
	Aggregation	788	2	7
	Spiral	312	2	3
	Pathbased	10992	16	10
	R15	600	2	15
	Jain	373	2	2
	Compand	399	2	6
	Iris	150	4	3
	Wine	178	13	3
Real data sets	Wdbc	569	30	2
	Parkinsons	195	23	2
	Seeds	210	7	3
	Glass	214	9	7
	Pima	768	8	2
	Segmentation	2310	19	7

Table 5 Comparison of composite data sets ACC, ARI, and AMI

Data set	Algorithm	ACC	ARI	AMI
D31	DBSCAN	0.7126	0.5674	0.7736
	DPC	0.9623	0.9252	0.9431
	NSTLBO-DGDPC	0.9667	0.9353	0.9551
Aggregation	DBSCAN	1	1	1
	DPC	1	1	1
	NSTLBO-DGDPC	1	1	1
Spiral	DBSCAN	1	1	1
	DPC	1	1	1
	NSTLBO-DGDPC	1	1	1
Flame	DBSCAN	0.7986	0.7133	0.6884
	DPC	1	1	1
	NSTLBO-DGDPC	1	1	1
Pathbased	DBSCAN	0.6600	0.6274	0.5461
	DPC	0.7231	0.6155	0.5637
	NSTLBO-DGDPC	0.7233	0.7265	0.6138
R15	DBSCAN	0.6650	0.5239	0.7539
	DPC	0.7983	0.8144	0.8847
	NSTLBO-DGDPC	0.9933	0.9856	0.9884
Jain	DBSCAN	0.4048	0.2386	0.3256
	DPC	0.6246	0.0441	0.2166
	NSTLBO-DGDPC	0.6353	0.4885	0.5265
Compased	DBSCAN	0.6340	0.4644	0.6300
	DPC	0.6491	0.5713	0.7110
	NSTLBO-DGDPC	0.8396	0.8244	0.8392

From the three clustering evaluation indices in Table 5, compared with DPC algorithm, the ACC value of the proposed algorithm is increased by 0.4% on D31, which can be

Table 6 Comparison of real data sets ACC, ARI, and AMI

Data set	Algorithm	ACC	ARI	AMI
Iris	DBSCAN	0.8866	0.7136	0.7410
	DPC	0.8806	0.6811	0.7380
	NSTLBO-DGDPC	0.9667	0.9037	0.8831
Wine	DBSCAN	0.8390	0.5492	0.5949
	DPC	0.8450	0.5613	0.7299
	NSTLBO-DGDPC	0.9121	0.7414	0.7480
Wdbc	DBSCAN	0.7397	0.1351	0.0920
	DPC	0.8062	0.3558	0.3575
	NSTLBO-DGDPC	0.8131	0.3771	0.3311
Parkinsons	DBSCAN	0.6474	0.2026	0.1455
	DPC	0.5854	0.0146	0.1451
	NSTLBO-DGDPC	0.8265	0.2686	0.2171
Seeds	DBSCAN	0.6428	0.4889	0.4323
	DPC	0.7323	0.5983	0.5475
	NSTLBO-DGDPC	0.8676	0.6195	0.5742
Glass	DBSCAN	0.4112	0.1129	0.2393
	DPC	0.4579	0.1258	0.2405
	NSTLBO-DGDPC	0.4859	0.2061	0.2530
Pima	DBSCAN	0.5532	0.0387	0.0071
	DPC	0.6314	0.0153	0.0021
	NSTLBO-DGDPC	0.6379	0.0143	0.2530
Segmention	DBSCAN	0.5571	0.3667	0.5130
	DPC	0.5809	0.3368	0.5141
	NSTLBO-DGDPC	0.6904	0.4687	0.5713

said that the accuracy is similar to DPC algorithm. Its ARI and AMI values are slightly higher than DPC algorithm. Compared with DBSCAN algorithm, the ACC and AMI values are increased by 26% and 19%, and its ARI values are significantly higher than DPC algorithm. This shows that this algorithm can find clustering centers on complex data sets and divide the data well. On aggregation and spiral synthetic data sets, the clustering evaluation indexes of the three algorithms are equal. On flame synthetic data sets, the values of ACC and ARI are equal to those of DPC algorithm. The values of ACC, ARI, and AMI are significantly higher than those of DBSCAN algorithm. On the Pathbased, R15, and Jain synthetic data sets, the ACC, ARI, and AMI values are significantly higher than the other two algorithms. It shows that compared with DPC algorithm, DBSCAN algorithm can process data sets and achieve better clustering results by combining the improved swarm intelligence algorithm with clustering algorithm.

From the three clustering evaluation indexes in Table 6, the ACC, ARI, and AMI values of the proposed algorithm are higher than those of DPC and DBSCAN algorithms on iris and wine real data. On the real data set of wdbc, the ACC value and ARI value are higher than the other two algorithms, but the AMI value is lower than the DPC algorithm.

On the Parkinsons real data set, the three clustering indices are significantly higher than DPC and DBSCAN algorithms. On the real data sets of Parkinsons, Seeds, Pima, Glass, and Segmention, the three clustering indexes are significantly higher than the DPC algorithm and DBSCAN algorithm. On the Glass real data set, the ACC and ARI values of each algorithm are smaller than the ACC and ARI values of the Seeds, Pima, and Segmention real data sets. It also shows that the proposed algorithm can achieve better clustering results in both synthetic and real data sets.

Figure 6 is the clustering effect diagram of DPC algorithm and this algorithm on eight kinds of composite data sets with different complexity. Figure 5a on the D31, aggregation, Jain, and Compand data sets, compared with the DPC algorithm in Fig. 6b, this algorithm can divide the data into true classes. The clustering effect of the two algorithms is similar on the spiral, flame, Pathbased, and R15 data sets.

Conclusion and Future Work

The cutoff distance d_c determines the local density and high-density distance of the DPC algorithm, so the setting of the d_c value is very important. In order to solve the DPC algorithm manually set the cutoff distance and select the clustering center manually, the DPC algorithm distance calculation does not consider the impact of data attributes and neighborhoods on the clustering results. Based on the DPC and TLBO algorithms, in this paper we propose a clustering algorithm that combines teaching and learning optimization algorithms and density difference distances. The d_c value range is set to the population search interval, and the NSTLBO algorithm is used to find the optimal d_c value. Second, considering the influence of data attributes and its neighborhood factors, the weighted density difference distance is established and replaced by the Euclidean distance, and finally the weighted distance is used to calculate the local density value of the algorithm. Because the DPC algorithm selects the points with larger high-density distance and larger local density when determining the clustering center, the clustering center is selected by using the standard deviation of the distance and the mean density to separate these data points, thereby improving the correct clustering of the DPC algorithm. The results of simulation experiments show that the algorithm in this paper can calculate the number of classes of the real data set without manually setting the d_c value and has good clustering quality and effect.

For our future work, we will focus on an adaptive parameter selection method to avoid manually setting the parameters of the density peak clustering algorithm. In addition, with the increase in data sets, it will cause a lot of overhead to calculate the distance between data points. We will also

(a) The clustering effect of NSTLBO-DGDPC algorithm on synthetic data sets

(b) The clustering effect of DPC algorithm on synthetic data sets

Fig. 6 Clustering effect of two algorithms on synthetic data sets

study how to improve the computing speed of density peak clustering algorithm in dealing with large data sets, whether using high-performance hardware or using MapReduce to process big data in parallel is worth further study.

Acknowledgements This work was financially supported by National Natural Science Foundation of China (NSFC) (Grant No. 61772160). Special Foundation of Scientific and Technological Innovation for Young Scientists of Harbin, China (Grant No. 2017RAQXJ045).

Compliance with Ethical Standards

Conflict of interest The authors declared that there was no conflict of interest regarding the publication of this paper.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Bousbaci A, Kamel N. Efficient data distribution and results merging for parallel data clustering in mapreduce environment. *Appl Intell*. 2018;48(8):2408–28.
- Qian WN, Zhou AY. Analyzing popular clustering algorithms from different viewpoints. *J Soft*. 2002;13(8):1382–94.
- Xu JH, Liu H. Web users clustering analysis based on k-means algorithm. In: *Information Networking and Automation (ICINA)*. IEEE; 2010. p. v26–29.
- Guha S, Rastogi R, Shim K. ROCK: a robust clustering algorithm for categorical attributes. In: *Proceedings of the IEEE Conference on Data Engineering*. 1999.
- Ester M, Kriegel HP, Xu X. A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In: *Proceedings of international conference on knowledge discovery and data mining*. AAAI Press; 1996. p. 226–31.
- Kohonen T. The self organizing maps. *Proc IEEE*. 1990;78(9):1464–80.
- Wang W, Yang J, Muntz R. STING: a statistical information grid approach to spatial data mining. In: *Proceedings of 23rd VLDB Conference*; 1997. p. 186–95.
- Fraley C, Raftery AE. Model-based clustering, discriminant analysis and density estimation. *J Am Stat Assoc*. 2002;97(458):611–31.
- Sun HJ, Wang SR, Jiang QS. FCM-based model selection algorithms for determining the number of clusters. *Pattern Recognit*. 2004;37:2027–37.
- Anil KJ. Data clustering: 50 years beyond K-Means. *Pattern Recognit Lett*. 2010;31(8):651–66.
- Sudipto G, Rajeev R, Kyuseok S. Cure: an efficient clustering algorithm for large databases. *Inf. Syst*. 1998;26(1):35–58.
- Ester M, Kriegel HP, Sander J, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In: *Proceedings of the second international conference on knowledge discovery and data mining*, vol. 96. AAAI Press; 1996. p. 226–31.
- Rodriguez A, Laio A. Clustering by fast search and find of density peaks. *Science*. 2014;344(6191):1492–6.
- Xu ML, Li YH, Li RX, et al. An extended adaptive density peaks clustering for overlapping community detection in social networks. *Neurocomputing*. 2019;337:287–302.
- Du MJ, Ding SF, Xue Y. A robust density peaks clustering algorithm using fuzzy neighborhood. *Int J Mach Learn Cybern*. 2018;9(7):1131–40.
- Li ZJ, Tang YC. Comparative density peaks clustering. *Expert Syst Appl*. 2018;95:236–47.
- Bai L, Cheng XQ, Liang JY, et al. Fast graph clustering with a new description model for community detection. *Inf Sci*. 2017;71:375–86.
- Reynolds CW. Flocks, herds, and schools: a distributed behavioral model. *ACM*. 1987;21(4):25–34.
- Gao WF, Liu SY, Huang LL. Enhancing artificial bee colony algorithm using more information-based search equations. *Inf Sci*. 2014;270:112–33.
- Mauder T, Sandera C, Stetina J, et al. Optimization of the quality of continuously cast steel slabs using the firefly algorithm. *Mater Technol*. 2011;45(4):347–50.
- Kennedy J, Eberhart R. Particle swarm optimization. In: *Proceedings of ICNN95-international Conference on Neural Networks*. 1995. p. 1942–8.
- Rao RV, Savsani VJ, Rai DP. Teaching–learning based optimization: a novel method for constrained mechanical design optimization problems. *Comput Aided Des*. 2011;43(3):303–15.
- Zou F, Chen D, Lu R, et al. Teaching–learning-based optimization with differential and repulsion learning for global optimization and nonlinear modeling. *Soft Comput*. 2017;1:1–29.
- Sevinç E, Dökeroğlu T. A novel hybrid teaching-learning-based optimization algorithm for the classification of data by using extreme learning machines. *Turk J Electr Eng Comput Sci*. 2019;27:1523–33.
- Kumar Y, Singh PK. A chaotic teaching learning based optimization algorithm for clustering problems. *Appl Intell*. 2018;49:1036–62.
- Kheireddine B, Zoubida B, Tarik H. Improved version of teaching learning-based optimization algorithm using random local search: TLBO-RLS. *COMPEL Int J Comput Math Electr Electron Eng*. 2019;38:1048–60.
- Niu P, Ma Y, Yan S. A modified teaching–learning-based optimization algorithm for numerical function optimization. *Int J Mach Learn Cybern*. 2018;10:1357–71.
- Zhang Z, Huang H, Huang C, et al. TLBO, logarithmic helix and triangular mutation global optimization. *Neurol Sci*. 2018;31:4435–50.
- Kumar Y, Kuma P. A chaotic teaching learning based optimization algorithm for clustering problems. *Appl Intell*. 2019;49(3):1036–62.
- Zou F, Chen DB, Xu QZ. A survey of teaching–learning-based optimization. *Neurocomputing*. 2018;335:366–83.
- Zang WK, Ren LY, Zhang WQ, et al. Automatic density peaks clustering using DNA genetic algorithm optimized data field and gaussian process. *Neurocomputing*. 2017;31(8):366–83.
- DeJong KA. An analysis of the behavior of class of genetic adaptive systems. University of Michigan; 1975.
- Wan M, Yin SQ, Ta T, et al. Optimized fuzzy clustering by fast search and find of density peaks. In: *Proceedings of 3rd IEEE International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*; IEEE. 2018. p. 83–7.
- Krumhansl C. Concerning the applicability of geometric models to similarity data: the interrelationship between similarity and spatial density. *Memory Cognit*. 1978;85(5):445–63.
- Li D, Luo K, Sun Z. Fuzzy clustering of new niche fireflies. *Comput Eng Sci*. 2017;39(05):1005–10.
- Xie J, Gao H, Xie W, et al. Grant, Robust clustering by detecting density peaks and assigning points based on fuzzy weighted K-nearest neighbors. *Inf Sci*. 2018;354:19–40.
- Nguyen XV, Epps J, Bailey J. Information theoretic measures for clusterings comparison: is a correction for chance necessary. In: *Proceedings of the 26th Annual Machine Learning*; 2009. p. 1073–80.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.