IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 4, 2019, accepted July 17, 2019, date of publication July 26, 2019, date of current version August 13, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2931334

A New DBSCAN Parameters Determination Method Based on Improved MVO

WENHAO LAI, MENGRAN ZHOU, FENG HU, KAI BIAN, AND QI SONG
School of Electrical and Information Engineering, Anhui University of Science and Technology, Huainan 232000, China

Corresponding author: Mengran Zhou (mrzhou8521@163.com)

This work was supported in part by the National 135 Program ''National Key Research Program'' under Grant 2018YFF0301000, and in part by the National Safe Production Critical Incident to Key Technologies Science and Technology Project of China under Grant anhui-0001-2016AQ.

ABSTRACT Density-based spatial clustering of applications with noise (DBSCAN) is a typical kind of algorithm based on density clustering in unsupervised learning. It can cluster data of arbitrary shape and also identify noise samples in the dataset. However, an unavoidable defect of the DBSCAN algorithm exists since the clustering performance is quite sensitive to the parameter settings of *MinPts* and *Eps*, and there is no theory to guide the setting of its parameters. Therefore, a new method is proposed to optimize the DBSCAN parameters in this paper. Multi-verse optimizer algorithm, a special variable updating method with excellent optimization performance, is selected and improved for optimizing the parameters of DBSCAN, which not only can quickly find out the highest clustering accuracy of DBSCAN, but also find the interval of *Eps* corresponding to the highest accuracy. In order to search the range of *Eps* more quickly and efficiently, we design a new mechanism for the variable update of MVO. The experimental results show that the improved MVO is used to optimize DBSCAN, which not only can quickly find out its highest clustering accuracy but also can search the parameters of *MinPts* and *Eps* corresponding to the highest clustering accuracy efficiently.

INDEX TERMS Improved MVO, DBSCAN, parameter optimization, unsupervised learning.

I. INTRODUCTION

Cluster analysis is one of the most effective methods commonly used in data mining, which aims to find potential and valuable information in the dataset. DBSCAN [1] is a typical representative based on the density clustering algorithm which can cluster clusters of any shape and identify noise samples in the data. Besides, DBSCAN has a significant advantage in that it does not require category information of cluster data when clustering. These advantages make it an increasingly popular clustering algorithm. For example, Kellner Dominik *et al*. selected DBSCAN to cluster extended objects in high-resolution radar data [2], Shen Jianbing *et al*. used DBSCAN to segment super pixels in real time [3], and Pavlis Michalis *et al*. employed DBSCAN to recognize UK local retail agglomerations [4].

Although DBSCAN has significant advantages in clustering, it also has the same defects as other clustering algorithms, that is, the clustering performance depends on the parameter settings. In different datasets, the optimal clustering results of DBSCAN will have different values of the parameters *MinPts* and *Eps*. Even in the same dataset, *MinPts* takes different values, and the optimal value of *Eps* is quite different. In addition, there are no theoretical guidance parameters for setting *MinPts* and *Eps*, which leads to the selection of reasonable DBSCAN parameters completely depends on personal experience and a large number of experimental trials. Since the clustering result of DBSCAN is sensitive to parameters, the parameters *MinPts* and *Eps* must be set reasonably in the application, which limits the extensive use of DBSCAN to some extent. For this problem, many scholars have done some research on the direction of the parameters setting of DBSCAN. In references [5] and [6], the parameter *Eps* is automatically determined by using the k-dist list. In reference [7], the authors propose a hierarchical adaptive alternating optimization method to find the optimal parameter combination of DBSCAN. In reference [8], a normalized density list is generated by evaluating the local density of the dataset by using the Affinity Propagation algorithm, and then the density list is combined to determine the parameters of

The associate editor coordinating the review of this manuscript and approving it for publication was Marko Beko.

the DBSCAN. In reference [9], the histogram equalization is applied to the pairwise similarity matrix of input data, and then the optimal parameter combination of DBSCAN is determined by dominant sets (DSets). In reference [10], the binary differential evolution algorithm is used to optimize the optimal combination parameters *MinPts* and *Eps*. These methods have promoted the development and application of DBSCAN to some extent, but in this paper, a new optimize DBSCAN parameters method based on the meta-heuristic optimization algorithm is proposed, which is not for finding a value of the parameter *Eps* in the optimal clustering but finding the interval of *Eps*. This new method allows DBSCAN to select a more reasonable value of *Eps* from the optimal range when clustering.

Meta-heuristic algorithms are a class of optimization algorithms that are inspired by physical phenomena or biological behaviors in nature, such as MFO [11], SOS [12], GA [13], PSO [14], FA [15], BA [16], GWO [17], *etc*. They are widely used in machine learning [18]–[21], data mining [22], [23], engineering design [24]–[26], industrial control [27], [28], and power systems [29] due to superior optimization performance. The swarm intelligence optimization algorithm is a typical representative in the meta-heuristic algorithm, and these algorithms are simple, flexible, easy to implement but not easy to fall into local optimum. MVO is a young and more advanced SI optimization algorithm proposed by Seyedali *et al.* in 2015 [30]. It inherits all the advantages from the SI optimization algorithm while also having better optimization capabilities, higher search efficiency, and fewer parameters to adjust. Once MVO was proposed, it successfully solved classic engineering problems such as welded beam design, gear train design, pressure vessel design and cantilever beam design [30]. Therefore, the MVO algorithm is also particularly concerned by researchers. In reference [31], MVO and some famous meta-heuristic algorithms including GA, PSO, BA, FA are used for parameter optimization of SVM respectively. After comparison and analysis, the parameters of MVO optimized allow the SVM to have a higher recognition rate. In reference [32], the authors implemented MVO to train multi-layer perceptrons, and the results show that the MVO-trained feedforward multi-layer perceptron in multiple datasets has better performance in avoiding local optimum and convergence speed. In [33], the authors used MVO to optimize the DGM (1, 1) parameters and predict the annual peak of the electrical load. In the comparison of multiple prediction models, MVO-DGM (1, 1) shows better convergence speed and higher prediction accuracy.

Considering the characteristics of variable update in MVO algorithm and its excellent optimization performance, we select and improve it to optimize the parameters of DBSCAN, which not only can quickly find the highest clustering accuracy of DBSCAN but also can search for the interval of *Eps* corresponding to the highest accuracy rate. No free lunch (NFL) theorem has described that there is no algorithm could solve all optimization problems [34], and MVO is not an exception. To make MVO more quickly and efficiently search for the maximum interval of *Eps* corresponding to the highest accuracy rate, we have further improved the MVO algorithm, that is, design a new mechanism for MVO's universe variable update. Obtaining the optimal value interval of *Eps* can allow DBSCAN to select more reasonable parameters from the interval, thus better guaranteeing its clustering performance.

II. MVO OPTIMIZATION AND DBSCAN CLUSTERING THEROIES

A. MVO OPTIMIZATION THEROY

Mirjalili S *et al.* proposed a new heuristic optimization algorithm called MVO, which was inspired by the theory of multiverse in physics. The MVO algorithm treats the inflation rate of the universe as a function of fitness, building mathematical models through white holes, black holes, and wormholes. In the optimization model, the higher the inflation rate, the higher the probability of occurrence of white holes and the lower probability of black holes. The direction of information transmission between different universes is from white holes to black holes, that is, from the universe with a higher inflation rate transmits information to the universe with a lower inflation rate. Therefore, the transmission of information from the universe with a high inflation rate to the universe with a low inflation rate is in the process of optimization, which in turn increases the average inflation rate of all universes. The wormholes appear randomly in any universe to maintain the diversity of the universe during the iteration.

Assume that

$$U = \begin{bmatrix} u_1^1 & u_1^2 & \cdots & u_1^{n_2} \\ u_2^1 & u_2^2 & \cdots & u_2^{n_2} \\ \vdots & \vdots & \cdots & \vdots \\ u_{n_1}^1 & u_{n_1}^2 & \cdots & u_{n_1}^{n_2} \end{bmatrix}$$

where n_1 is the number of parallel universes and n_2 is the number of parameters(variables) of each universe.

The white/black holes are established by the roulette wheel mechanism, and their selection rules are as shown in (1). In addition, the universe is sorted according to the normal expansion rate in each iteration before the information is transmitted.

$$u_i^j = \begin{cases} u_k^j & r_1 < N_i\left(U_i\right) \\ u_i^j & r_1 \geq N_i\left(U_i\right) \end{cases} \tag{1}$$

where u_i^j is the jth variable of the ith universe, u_k^j is the jth variable of the kth universe selected by the roulette mechanism, $N_i(U_i)$ is the normalized inflation rate of the i-th universe, and r_1 is a random variable with a value between 0 and 1.

After the selection of the white/black holes, information exchange will take place in different universes. However, before the variables of the universe are updated, two parameters must be calculated, they are the wormhole existence probability (*WEP*) and the travel distance (*TDR*). The evaluation rules for *WEP* and *TDR* are shown in (2).

$$\begin{cases} WEP = \alpha + m * (\beta - \alpha)/M \\ TDR = 1 - (m/M)^{1/p} \end{cases} \tag{2}$$

where M is the maximum number of iterations, m is the current iteration, α and β are the minimum and maximum values of *WEP* respectively, p is a constant that closely affects the accuracy of development.

The linear increase of *WEP* in iterations can lead to an increase in probability of wormholes emergence. Reducing the value of *TDR* in the iteration allows for a more accurate local search around the optimal universe. In this paper, the values of α and β are 0.2 and 1 respectively, the value of p is 6.

Then update the variables of the universe.

$$u = \begin{cases} U^j + TDR \times ((ub_j - lb_j) \times r_4 + lb_j), & r_3 < 0.5 \\ U^j - TDR \times ((ub_j - lb_j) \times r_4 + lb_j), & r_3 \geq 0.5 \end{cases} \tag{3}$$

$$u_i^j = \begin{cases} u, & r_2 < WEP \\ u_i^j, & r_2 \geq WEP \end{cases} \tag{4}$$

where ub_j and lb_j are the upper and lower boundaries of the jth variable of the universe, respectively, r_2, r_3, and r_4 are random variables with values between [0, 1], U^j is the jth variable of the optimal universe.

A notable characteristic of MVO in the process of updating universe variables is the use of three random variables, r_2, r_3, and r_4, and the contribution of this feature is that it effectively avoids falling into local optimums. However, in author's view, this characteristic has another contribution, which provides a possibility that MVO can search for the interval of optimized parameters, which is one of the reasons why MVO is used to optimize the parameters of DBSCAN.

The global optimal inflation rate is *binf*, the optimal inflation rate for the mth iteration is inf_m, and the universe corresponding to inf_m is U_m. The flowchart of the MVO algorithm is shown in Fig.1.

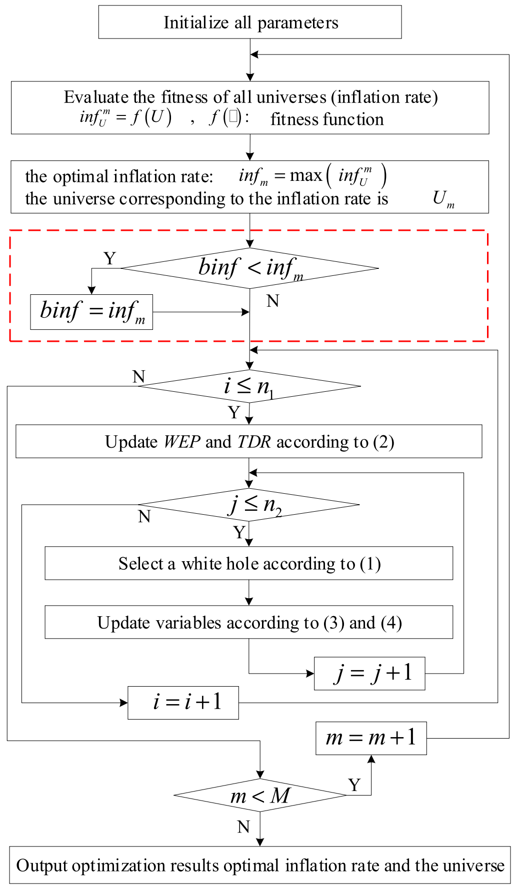


FIGURE 1. The flowchart of the MVO algorithm.

B. DBSCAN CLUSTERING THEROY

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is an unsupervised clustering algorithm proposed by Martin *et al*. Different from partitioning and hierarchical clustering, DBSCAN is a considerably representative density-based clustering algorithm that defines clusters as the largest set of points connected by density. Therefore, the design idea of the DBSCAN algorithm is to find the set of maximum density connected samples in the dataset according to the density reachability relationship, and the samples in this set are considered to be the same class.

Let the dataset is $X = (x_1, x_2, \cdots x_n)$. The set of samples whose distance from $x (x \in X)$ is not greater than *Eps* is $N_{Eps}(x) = \{x' \in C | dis(x', x) \leq Eps\}$, called *Eps-neighborhood*. Several core concepts about DBSCAN are as follows:

Definition 1 (Core Object): If the number of samples in $N_{Eps}(x)$ is not less than *MinPts*, then x is the core object.

Definition 2 (Directly Density-Reachable): If $x' \in N_{Eps}(x)$ and x is the core object, then$x'$is said to be directly density-reachable from x.

Definition 3 (Density-Reachable): There exists a chain $P_1, \cdots, P_i, \cdots, P_t$. If $P_1 = x'$, $P_t = x$, and P_{i+1} is directly density-reachable from P_i, then x' is said to be density-reachable from x.

Definition 4 (Density-Connected): A point x is density-connected to point x' if there exists core object c such that both x and x' directly density-reachable from c.

Definition 5 (Noise): A sample that is neither a core object nor within the set of the $N_{Eps}(x)$ is called noise.

Let the set of core objects is ψ, the unvisited sample set is X', the number of clusters searched is Q, and the clustering result is C. The clustering steps of the DBSCAN algorithm are as follows:

Step 1: Assign values to the parameters *MinPts* and *Eps*, initialize all variables.

Step 2: Find out all the core objects of the dataset X.

Let i $= 1, 2, 3, \cdots$,evaluate the distance between all samples, and find the $N_{Eps}(x_i)$ of each sample x_i. If $|N_{Eps}(x_i)| >$ *MinPts*, then add x_i to the set of core objects ($\psi = \psi \cup x_i$).

Step 3: If the core object set is empty, go to step 7, otherwise, go to step 4.

Step 4: Randomly select an object x_c in the core object set. The current cluster core queue $\psi' = \{x_c\}$, the category label

$q(q \leftarrow q+1)$, and the unvisited sample set $X'(X' \leftarrow X' - x_c)$ are then updated.

Step 5: For a core object x_c' of current cluster core queue ψ', find the subset of $N_{Eps}(x_c')$, update current cluster sample set $C_q = C_q \cup (N_{Eps}(x_c') \cap X')$, unvisited sample set and current cluster core queue.

Step 6: If the current core object queue ψ' is empty, the search for current cluster's sample is finished, update the cluster set and the core object set, return to step 3.

Step 7: Output clustering results $C = \{C_1, C_2, C_3, \cdots C_Q\}$, calculate clustering accuracy.

III. OPTIMIZED DBSCAN-BASED ON IMPROVED MVO

A. REASONS FOR IMPROVING THE MVO ALGORITHM

The essence of DBSCAN algorithm clustering is to use the density to divide the samples of the dataset into different clusters. The basis of clustering is the distance between each sample, which makes DBSCAN accurately cluster them for a given sample set. The corresponding *Eps* is an interval. A clearer explanation of this is shown in Fig. 2.

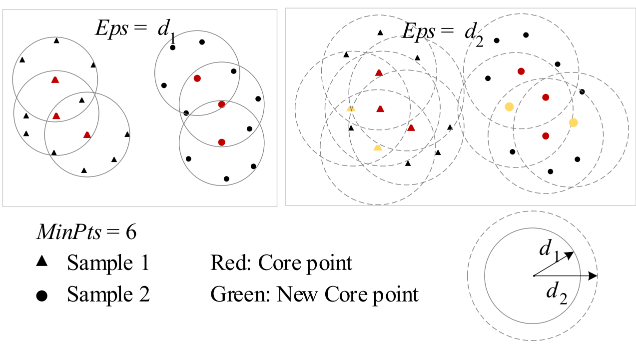


FIGURE 2. Example of DBSCAN clustering results.

In Fig.2, the value of *MinPts* is 6. Whether the value of *Eps* is d_1 or d_2, DBSCAN algorithm can accurately distinguish cluster 1 and cluster 2. Furthermore, when $Eps \in [d_1 d_2]$, DBSCAN can accurately identify cluster 1 and cluster 2, and the difference is that various values of *Eps* may lead to a varying number of core objects.

From Fig. 1, it can be observed that the traditional MVO is used to optimize DBSCAN, the return value is the universe corresponding to the optimal inflation rate and the optimal inflation rate, and the value of *Eps* corresponding to the highest clustering accuracy of DBSCAN is an interval. Therefore, we improve MVO, denoted IMVO1, and used it to search for the highest clustering accuracy of DBSCAN and the corresponding interval of parameter *Eps*.

B. THE IMPROVES OF MVO

There are two parameters that DBSCAN needs to optimize for clustering, namely *MinPts* and *Eps*. In order to simplify the process of optimizing DBSCAN parameters, our solution is to firstly determine the parameter *MinPts* corresponding to the highest clustering accuracy and then optimize the parameter *Eps*. Therefore, in this paper, the only parameter that DBSCAN algorithm needs to optimize is *Eps*, that is, there is only one variable in each universe. Let the universe set is $U = \{u_1, u_2, u_3, \cdots u_n\}$, the upper and lower boundaries of the universe variables corresponding to the global optimal inflation rate are u_{max} and u_{min}, respectively. The global optimal inflation rate is $binf$, the optimal inflation rate for the mth iteration is inf_m, and the universe corresponding to inf_m is U_m.The flowchart of the improved part of the MVO algorithm is shown in Fig. 3.

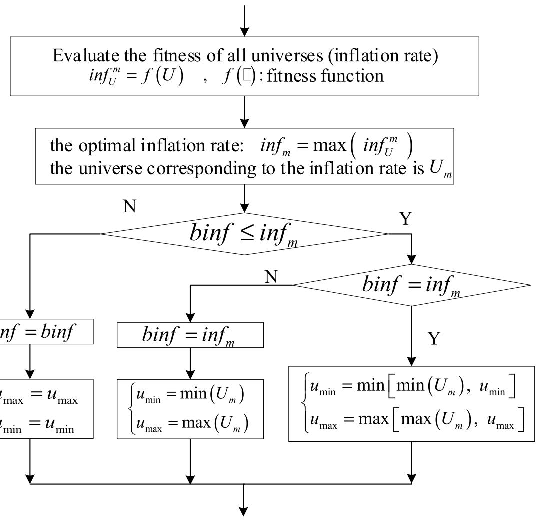


FIGURE 3. The improved part of the MVO algorithm.

In order to find out the maximum range of *Eps* corresponding to the maximum clustering accuracy of DBSCAN more efficiently, we further improve the optimization algorithm based on IMVO1, denoted as IMVO2. In this section, we improve the (3) that is related to the universe variable update mechanism, as given in (5), as shown at the bottom of the next page.

where M is the maximum number of iterations, and m is the current iteration. Our universe variable update mechanism is more concerned with the values around the maximum and minimum values of the *Eps* that have been searched. Since only the *Eps* found at the highest clustering accuracy is meaningful, when $m > M/2$ starts using the new variable update mechanism.

Compared to (3), the new variable update method has two major advantages. One is to search only near the maximum and minimum *Eps* that has been searched; the other is that the variable update does not use the universe boundary information [*lb*,*ub*]. These features make IMVO2 faster and more adaptable to different datasets when performing outstanding searches on *Eps*. The pseudocode of IMVO2-DBSCAN is as shown on the right.

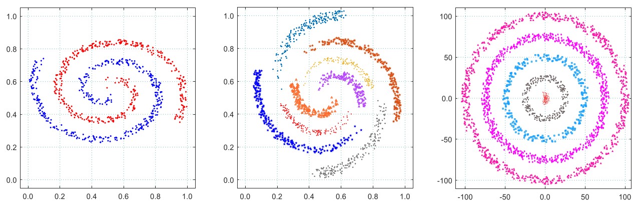
In pseudocode, the fitness of IMVO2 is DBSCAN clustering accuracy. DBSCAN is an unsupervised learning algorithm whose output is the cluster number of each sample. Sometimes the number of clusters output by DBSCAN is much larger than the number of classes of input samples. We cannot use it to calculate clustering accuracy directly.

When evaluating clustering accuracy, the selection of clusters mainly follows three principles. First, select at most one of the clusters with the same input label. Second, not all samples with the same cluster number but different input labels are selected. Third, there should be as many samples as possible in the cluster.

IV. EXPERIMENTS AND RESULTS

A. EXPERIMENTS SETUP

In our experiments, three artificial datasets were selected, denoted D1 which contains 2 categories of 800 samples, D2 which contains 5 categories of 1500 samples, and D3 which contains 8 categories of 2400 samples. In order to make the verification of the performance of the improved MVO algorithm search *Eps* more reasonable, the inter-class distances and shapes of the three artificial datasets selected are different. The shape of the cluster contained in the artificial datasets is shown in Fig.4.



(a) (b) (c)

FIGURE 4. Artificial data. a, b, and c are datasets D1, D2, and D3, respectively.

One of the advantages of MVO is that there are fewer parameters to be set, and the author Seyedali *et al*. of MVO algorithm also gives the most reasonable settings for key parameters, so the focus of this paper is on the improved MVO optimization DBSCAN parameters. To verify the performance of the MVO, we also used the GA, PSO, MFO, and SOS to optimize the DBSCAN parameters. In the experiment, the parameter settings of GA, PSO, MFO, SOS, MVO, and IMVO are shown in Table 1.

In our experiments, each optimization model searches the DBSCAN parameters multiple times independently and selects the optimal results. All experiments in this paper are run on Intel (R) Core (TM) i7-9700K CPU @ 3.60GHz with 16GB memory.

B. THE OPTIMAL SELECTION OF MINPTS

The premise of accurate clustering of DBSCAN algorithm requires reasonable setting of parameters *MinPts* and *Eps*.

TABLE 1. Initial parameters of GA, PSO, MFO, SOS, MVO, and IMVO.

Parameter	Algorithm	Value
Crossover ratio	GA	0.6
Mutation ratio	GA	0.1
Max fit eval	SOS	1.0
Organisms size	SOS	20
Population size	GA, PSO, MFO	20
Min *WEP*	MVO, IMVO	0.2
Max *WEP*	MVO, IMVO	1.0
Universes	MVO, IMVO	20
Iterations	GA, PSO, MVO, MFO, SOS	200
	IMVO	400
[lb, ub] of D1	GA, PSO, MVO, IMVO, MFO, SOS	[0.001, 1.0]
[lb, ub] of D2	GA, PSO, MVO, IMVO, MFO, SOS	[0.001, 1.0]
[lb, ub] of D3	GA, PSO, MVO, IMVO, MFO, SOS	[0.001, 100]

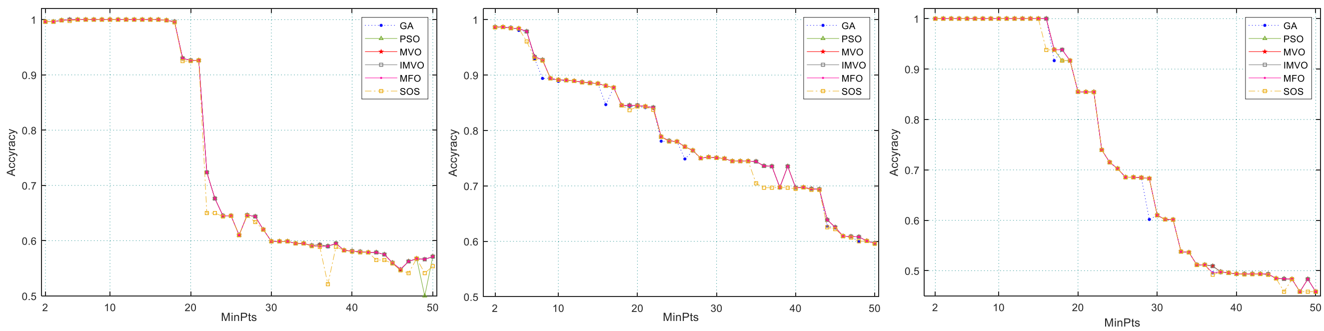
Before using the improved MVO to optimize the DBSCAN parameter *Eps*, the parameter *MinPts* must be set reasonably. GA, PSO, MVO, and IMVO are used to find the highest clustering accuracy of DBSCAN when *MinPts* takes different values, and the results are shown in Fig.5.

In the experiment, the optimization speed of SOS is the fastest but easily fall into local optimum, relatively. Although MFO's optimized performance is as good as MVO, the variable update mechanism of MVO allows us to improve it to optimize the interval of Eps. Therefore, we select and improve MVO to optimize the parameters of DBSCAN. As can be seen from Figure 5, we modify the variable update mechanism of MVO to optimize the interval of *Eps* is not deteriorating the excellent optimization performance of MVO. In addition, we can also know that the value of *MinPts* has a great influence on the maximum clustering accuracy of DBSCAN. Taking dataset D1 as an example, a reasonable set of *MinPts* can accurately cluster all samples, but if *MinPts* is setting unreasonably, the clustering accuracy of DBSCAN will be dramatically reduced to below 0.6. In general, the larger the value of *MinPts* is, the worse the DBSCAN accurate clustering dataset, but, in different datasets, the impact of *MinPts* on DBSCAN clustering will be slightly different. After comparison and analysis, in the next section of the experiment, the value of *MinPts* was selected to be 5.

C. OPTIMIZATION OF PARAMETER EPS OF DBSCAN

The improved MVO is used to search DBSCAN for optimal clustering accuracy and possible values of parameter *Eps*.

$$u = \begin{cases} \begin{cases} U + TDR \times ((ub - lb) \times r_4 + lb), & r_3 < 0.5 \\ U - TDR \times ((ub - lb) \times r_4 + lb), & r_3 \geq 0.5 \end{cases}, & m \leq \dfrac{M}{2} \\ \begin{cases} u_{\max} + 10 \times (r_4 - 0.4) \times TDR \div m, & r_3 < 0.5 \\ u_{\min} - 10 \times (r_4 - 0.4) \times TDR \div m, & r_3 \geq 0.5 \end{cases}, & m > \dfrac{M}{2} \end{cases} \tag{5}$$



(a) (b) (c)

FIGURE 5. Optimal clustering accuracy of GA, PSO, MFO, SOS, MVO, and IMVO search when *MinPts* takes different values. a, b, and c are datasets D1, D2, and D3, respectively.

Algorithm IMVO2-DBSCAN

Input: *MinPts, Iterations, Boundaries, Universes.*
Initialization all parameters
for *each iteration indexed by m in M*
  Update WEP and TDR according to (2).
  for *each universe indexed by g in* u_1
    Evaluate the fitness: $inf_g^m \leftarrow DBSCAN(X, u_g, MinPts)$
    # DBSCAN: please refer to steps 1 *to* 6. $u_g = Eps$
  end for
  Find the optimal inflation rate: $inf_m \leftarrow max\ (inf^m)$
  # where $inf^m = [inf_1^m, \cdots, inf_g^m, \cdots, inf_{n1}^m]$
  Find the universe U_m *corresponding to the* inf_m
  if $inf_m > binf$
    $u_{max} \leftarrow max\ (U_m)$; $u_{min} \leftarrow min\ (U_m)$; $binf \leftarrow inf_m$
  else if $inf_m == binf$
    $u_{max} \leftarrow max[u_{max}, max(U_m)]$; $u_{min} \leftarrow max[u_{min}, max(U_m)]$
  else
    $u_{max} \leftarrow u_{max}$; $u_{min} \leftarrow u_{min}$; $binf \leftarrow binf$;
  end if
  for *each universe indexed by g in* n_1
    $r_1 \leftarrow random(0,1)$; $r_3 \leftarrow random(0,1)$; $r_4 \leftarrow random(0,1)$;
    Select the universe according to (1).
    if $m < M/2$

$$u = \begin{cases} U^j + TDR \times ((ub - lb) \times r_4 + lb), & r_3 < 0.5 \\ U^j - TDR \times ((ub - lb) \times r_4 + lb), & r_3 \geq 0.5 \end{cases}$$

    else

$$u = \begin{cases} u_{max} + 10 \times TDR \times (r_4 - 0.4) \div m, & r_3 < 0.5 \\ u_{max} - 10 \times TDR \times (r_4 - 0.4) \div m, & r_3 \geq 0.5 \end{cases}$$

    end if
    $r_2 \leftarrow random(0,1)$
    Update the position of the universe u_g *according to (4)*
    end for
end for
Output: *binf: global optimal inflation rate,*
  $[u_{min}, u_{max}]$: *binfcorresponding boundary of the universe*

When $MinPts = 5$, the optimization curves of MVO and IMVO1 are shown in Fig. 6.

As can be seen from Fig. 6, for the dataset D1 and D2, when the DBSCAN clustering accuracy is the largest, the value and the interval of the *Eps* are small extremely, and for dataset D3, when the DBSCAN clustering accuracy is optimal, the value range of *Eps* is relatively large. In a small interval, the original MVO can quickly search for a suitable value of *Eps*, which allows DBSCAN to cluster with the highest accuracy. However, the MVO algorithm can only find an optimal *Eps* value, and IMVO1 is still working after searching for the maximum clustering accuracy of DBSCAN, searching for the *Eps* interval corresponding to the accuracy.

It also can be seen from Fig.6 that the search speed of IMVO1 is fast when the interval of the *Eps* is large, but in the dataset D1 and D2, the range of the *Eps* are small, and the search speed of the IMVO1 is relatively slow. To search the interval of *Eps* as efficiently as possible, we further improved the IMVO1 algorithm, indicated IMVO2, to perform a more refined search around the maximum and minimum *Eps* which are obtained so far. When $MinPts = 5$, the optimized curves of IMVO2 are shown in Fig. 7.

We designed IMVO2 to search for *Eps* more finely after half of the maximum number of iterations. In this paper, the maximum iteration of the improved MVO is set to 400, which means that IMVO2 searches for the maximum and minimum values of *Eps* more finely from the 201st iteration.

As can be seen from Figures 7a and 7b, when the iteration is greater than 200, the distance between the curves 'Max *Eps*' and 'Min *Eps*' is grown rapidly. Especially in Fig. 7b, when there is no fine search after searching for the maximum clustering accuracy of DBSCAN, even after more than 100 iterations, the distance between 'Max *Eps*' and 'Min *Eps*' is still relatively small, and the new search mechanism begins to work, the distance between 'Max *Eps*' and 'Min *Eps*' increases rapidly. In summary, our improved IMVO2 can not only quickly search for the maximum clustering accuracy of DBSCAN, but also search for the *Eps* range corresponding to the maximum clustering accuracy quickly and efficiently.

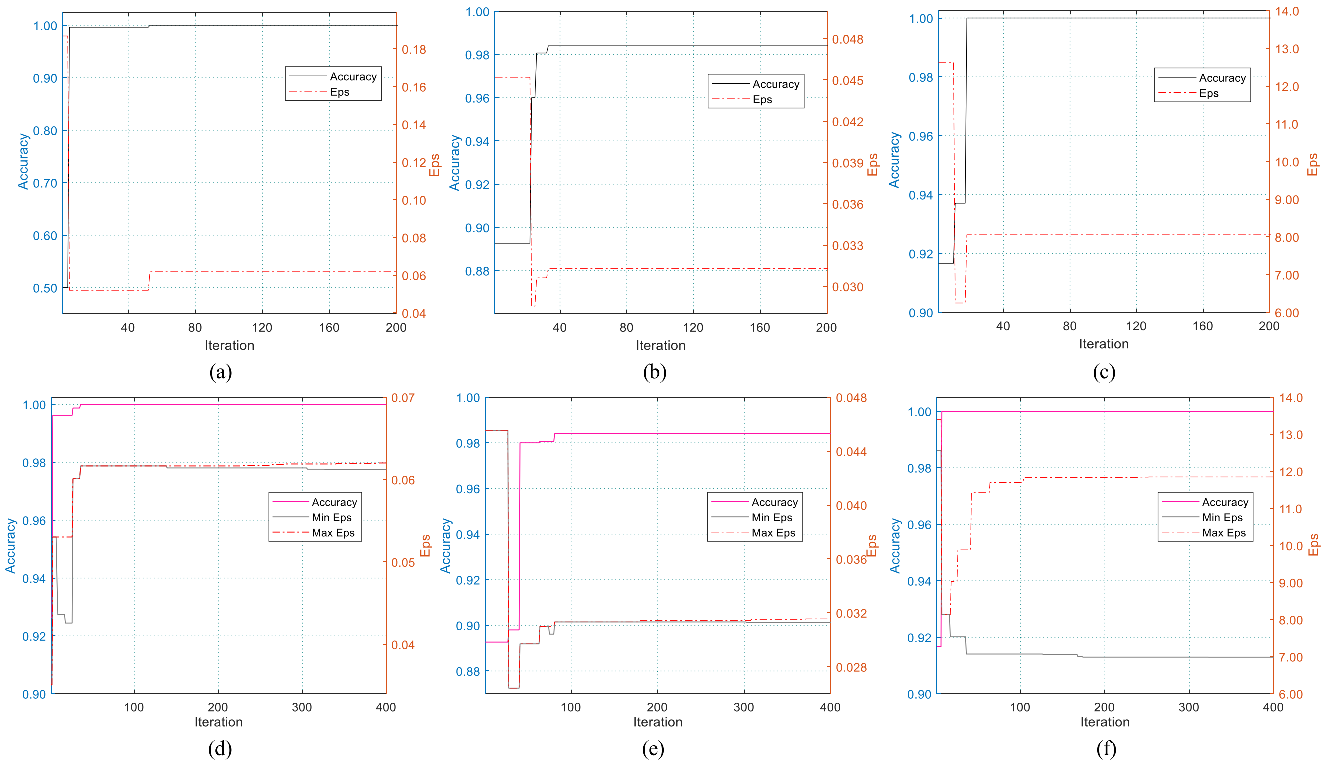


FIGURE 6. MVO/IMVO1 Optimization results when *MinPts* = 5. a, b, c d, e, and f are the optimization results of MVO and IMVO1 on D1, D2, and D3, respectively. Eps_{max} and Eps_{min} are the maximum and minimum values o *Eps* searched by IMVO1 at the current clustering accuracy.

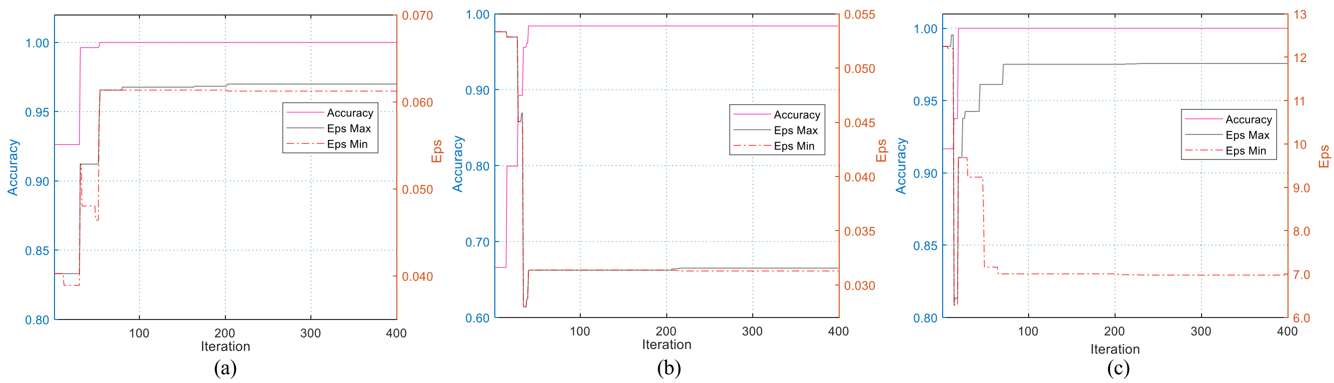


FIGURE 7. The optimized curves of IMVO2. a, b, and c are the optimization results on D1, D2, and D3, respectively. Eps_{max} and Eps_{min} are the maximum and minimum values of *Eps* searched by IMVO2 at the current clustering accuracy.

In order to make the comparison more apparent, we use IMVO1 and IMVO2 to search for the value interval of *Eps* which corresponds to the maximum clustering accuracy of DBSCAN when the of MinPts are different (regardless of whether the value of *MinPts* is the best), and the results are shown in Tables 2 and 3.

In Table 3, the range of *Eps* of IMVO2 search is larger than the range of *Eps* searched by IMVO1, especially in dataset D3, this is because IMVO1 updates the universe variable mechanism to $u = U \pm TDR \times (ub - lb) \times r_4 + lb$ and the search accuracy is affected by the boundary $[lb,ub]$. IMVO2 does not use universe variable boundary information in the finer search for the *Eps* interval, and only searches around Eps_{min} and Eps_{max}.

The maximum and minimum values of *Eps* for IMVO2 search are greater than and smaller than the maximum and minimum values of *Eps* for IMVO1 search, which means that the interval of *Eps* for IMVO2 search is larger than the range for *Eps* searched by IMVO1, that is, the *Eps* of the IMVO2 search is more excellent.

D. THE VERIFICATION OF IMVO2 OPTIMIZATION RESULTS

Compared with IMVO1, IMVO2 has a stronger ability to search for parameters *Eps*. To verify the *Eps* validity of

TABLE 2. Optimal clustering accuracy.

MinPts	D1		D2		D3	
	IMVO1	IMVO2	IMVO1	IMVO2	IMVO1	IMVO2
5	1.00000	1.00000	0.98400	0.98400	1.00000	1.00000
10	1.00000	1.00000	0.89133	0.89133	1.00000	1.00000
15	1.00000	1.00000	0.88467	0.88467	1.00000	1.00000
20	0.92625	0.92625	0.84533	0.84533	0.85500	0.85500
25	0.64500	0.64500	0.78000	0.78000	0.70250	0.70250
30	0.59875	0.59875	0.75067	0.75067	0.61000	0.61000
35	0.59125	0.59125	0.74400	0.74400	0.51167	0.51167
40	0.58125	0.58125	0.69667	0.69667	0.49375	0.49375
45	0.56000	0.56000	0.62533	0.62533	0.48458	0.48458
50	0.57125	0.57125	0.59667	0.59667	0.45833	0.45833

the IMVO2 search, we randomly select 500 points in each interval [Eps_{min}, Eps_{max}] as the value of the parameter *Eps*, and then use it for DBSCAN clustering, and the selection mechanism of *Eps* is shown in (6).

$$Eps = (Eps_{\max} - Eps_{\min}) \times rand() + Eps_{\min} \tag{6}$$

where $rand()$ is a random number with a value between 0 and 1. We also select points outside the [Eps_{min}, Eps_{max}] interval as *Eps* values (greater than Eps_{max} and less than Eps_{min}, respectively) for clustering, and the results are shown in Tables 4, 5, and 6.

It can be seen from Tables 4, 5, and 6 that all values of *Eps* are within the interval [Eps_{min}, Eps_{max}], and the clustering results of DBSCAN are the same. However, when the value of *Eps* is outside the interval, it will cause the maximum clustering accuracy of DBSCAN to decrease. This indicates that the interval of *Eps* corresponding to the highest clustering accuracy of DBSCAN for IMVO2 search is effective and optimal. In addition, the value of *Eps* larger than Eps_{max} has a greater influence on the clustering accuracy of DBSCAN. Sometimes even if the value of *Eps* is slightly larger than Eps_{max}, the maximum clustering accuracy of DBSCAN will be drastically reduced, since increment of *Eps* also increases the possibility that different classes are treated as the same class.

E. APPLICATION OF IMVO2-DBSCAN IN UCI DATASETS

One advantage of DBSCAN based on density clustering is that it can recognize noise samples. We improved MVO to optimize DBSCAN parameters, which gave DBSCAN another unique advantage in data mining. When the dataset

TABLE 3. IMVO1/IMVO2 search *Eps* results correspond to the optimal clustering accuracy.

MinPts	D1([Eps_{min}, Eps_{max}])		D2 ([Eps_{min}, Eps_{max}])		D3 ([Eps_{min}, Eps_{max}])	
	IMVO1	IMVO2	IMVO1	IMVO2	IMVO1	IMVO2
5	[0.061238, 0.062032]	[0.061230, 0.062036]	[0.031292, 0.031544]	[0.031288, 0.031553]	[6.98331, 11.85552]	[6.97701, 11.85556]
10	[0.061257, 0.076508]	[0.061230, 0.076527]	[0.052377, 0.053133]	[0.052366, 0.053148]	[11.01362, 11.85365]	[11.01357, 11.85556]
15	[0.071541, 0.076526]	[0.071537, 0.076527]	[0.053556, 0.053807]	[0.053551, 0.053821]	[11.76684, 11.85322]	[11.76673, 11.85556]
20	[0.077645, 0.078179]	[0.077636, 0.078188]	[0.053375, 0.053899]	[0.053366, 0.053905]	[13.10074, 13.47624]	[13.09998, 13.47683]
25	[0.076008, 0.078122]	[0.075980, 0.078123]	[0.053621, 0.053897]	[0.053621, 0.053905]	[11.80360, 11.85544]	[11.80336, 11.85556]
30	[0.096439, 0.096819]	[0.096438, 0.096821]	[0.068361, 0.068874]	[0.068355, 0.068884]	[13.81743, 13.82565]	[13.81694, 13.82648]
35	[0.102123, 0.102804]	[0.102122, 0.102813]	[0.071967, 0.073043]	[0.071962, 0.073046]	[13.83536, 13.97579]	[13.83511, 13.97647]
40	[0.108508, 0.109095]	[0.108507, 0.109097]	[0.077271, 0.077803]	[0.077263, 0.077805]	[15.17684, 15.52365]	[15.17679, 15.52461]
45	[0.114048, 0.114048]	[0.114046, 0.114058]	[0.081048, 0.081115]	[0.081042, 0.081123]	[16.31019, 16.45805]	[16.30906, 16.46158]
50	[0.114517, 0.114546]	[0.114450, 0.114547]	[0.081043, 0.081116]	[0.081037, 0.081123]	[17.59248, 20.07741]	[17.58753, 20.08188]

TABLE 4. Clustering performance verification of *Eps* searched by IMVO2 in dataset D1.

MinPts	Outside the interval		Within the interval			Outside the interval	
	$Eps = Eps_{min} - 0.000001$	*Accuracy*	$Eps =$[Eps_{min}, Eps_{max}]	$Accuracy_{min}$	$Accuracy_{max}$	$Eps = Eps_{max} + 0.000001$	*Accuracy*
5	0.061229	0.99875	[0.061230, 0.062036]	1.00000	1.00000	0.062037	0.50000
10	0.061229	0.99875	[0.061230, 0.076527]	1.00000	1.00000	0.076528	0.50000
15	0.071536	0.99875	[0.071537, 0.076527]	1.00000	1.00000	0.076528	0.50000
20	0.077635	0.92500	[0.077636, 0.078188]	0.92625	0.92625	0.078189	0.56500
25	0.075979	0.64375	[0.075980, 0.078123]	0.64500	0.64500	0.078124	0.64125
30	0.096437	0.59750	[0.096438, 0.096821]	0.59875	0.59875	0.096822	0.50000
35	0.102121	0.52625	[0.102122, 0.102813]	0.59125	0.59125	0.102814	0.59000
40	0.108506	0.57750	[0.108507, 0.109097]	0.58125	0.58125	0.109098	0.58000
45	0.114045	0.55875	[0.114046, 0.114058]	0.56000	0.56000	0.114059	0.49000
50	0.114449	0.57000	[0.114450, 0.114547]	0.57125	0.57125	0.114548	0.50750

TABLE 5. Clustering performance verification of *Eps* searched by IMVO2 in dataset D2.

MinPts	Outside the interval		Within the interval			Outside the interval	
	$Eps = Eps_{min} - 0.000001$	*Accuracy*	$Eps = [Eps_{min}, Eps_{max}]$	$Accuracy_{min}$	$Accuracy_{max}$	$Eps = Eps_{max} + 0.000001$	*Accuracy*
5	0.031287	0.98067	[0.031288, 0.031553]	0.98400	0.98400	0.031554	0.88533
10	0.052365	0.89067	[0.052366, 0.053148]	0.89133	0.89133	0.053149	0.75800
15	0.053550	0.88400	[0.053551, 0.053821]	0.88467	0.88467	0.053822	0.75200
20	0.053365	0.84333	[0.053366, 0.053905]	0.84533	0.84533	0.053906	0.75200
25	0.053620	0.77867	[0.053621, 0.053905]	0.78000	0.78000	0.053906	0.68867
30	0.068354	0.74933	[0.068355, 0.068884]	0.75067	0.75067	0.068885	0.62133
35	0.071961	0.70467	[0.071962, 0.073046]	0.74400	0.74400	0.073047	0.61933
40	0.077262	0.69467	[0.077263, 0.077805]	0.69667	0.69667	0.077806	0.61133
45	0.081041	0.62400	[0.081042, 0.081123]	0.62533	0.62533	0.081124	0.53533
50	0.081036	0.59533	[0.081037, 0.081123]	0.59667	0.59667	0.081124	0.50467

TABLE 6. Clustering performance verification of *Eps* searched by IMVO2 in dataset D3.

MinPts	Outside the interval		Within the interval			Outside the interval	
	$Eps = Eps_{min} - 0.000001$	*Accuracy*	$Eps = [Eps_{min}, Eps_{max}]$	$Accuracy_{min}$	$Accuracy_{max}$	$Eps = Eps_{max} + 0.000001$	*Accuracy*
5	6.97700	0.93792	[6.97701, 11.85556]	1.00000	1.00000	11.85557	0.91667
10	11.01356	0.93833	[11.01357, 11.85556]	1.00000	1.00000	11.85557	0.91667
15	11.76672	0.93917	[11.76673, 11.85556]	1.00000	1.00000	11.85557	0.91667
20	13.09997	0.85458	[13.09998, 13.47683]	0.85500	0.85500	13.47684	0.52167
25	11.80335	0.70208	[11.80336, 11.85556]	0.70250	0.70250	11.85557	0.61917
30	13.81693	0.60958	[13.81694, 13.82648]	0.61000	0.61000	13.82649	0.60875
35	13.83510	0.51125	[13.83511, 13.97647]	0.51167	0.51167	13.97648	0.49458
40	15.17678	0.49250	[15.17679, 15.52461]	0.49375	0.49375	15.52462	0.49000
45	16.30905	0.38625	[16.30906, 16.46158]	0.48458	0.48458	16.46159	0.48417
50	17.58752	0.41625	[17.58753, 20.08188]	0.45833	0.45833	20.08189	0.33333

TABLE 7. Optimization results of IMVO2 in public datasets.

Datasets	Classes	*MinPts*	$[Eps_{min}, Eps_{max}]$	*Accuracy*
Seeds subset	Kama, Canadian	18	[0.25177, 0.25322]	80.00%
Seeds	**Rosa**, Kama, Canadian	18	[0.25264, 0.25322]	70.48%
Iris subset	versicolor virginica	3	[0.40001, 0.41231]	74.00%
Iris	**setosa**, versicolor, virginica	3	[0.40001, 0.41231]	80.67%
Segment subset	2, 3, …,7	3	[0.16195, 0.16296]	54.14%
Segment	**1**, 2, 3, …,7	3	[0.16195, 0.16296]	52.20%

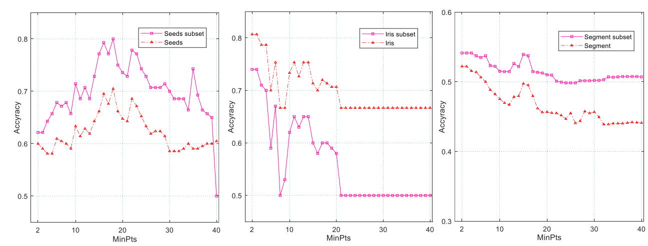


FIGURE 8. The optimal clustering accuracy of DBSCAN searched by IMVO2 when *MinPts* take different values. a, b, and c are the optimization results in Seeds, Iris, and Segment, respectively.

to be clustered has unknown label samples and their density is similar, the parameters of DBSCAN can be optimized to mine unknown label samples by using known label samples.

In this section, we test the performance of IMVO2-DBSCAN in public datasets and demonstrate this advantage.

Three UCI Machine Learning datasets were selected, named Iris which contains 3 classes with 4 features, Seeds which contains 3 classes with 7 features, and Segment which contains 7 classes with 19 features, respectively. The subset of the dataset is used for parameters optimization of DBSCAN, and the selection of the subset is shown in Table 7. The clustering accuracy of DBSCAN with different values of *MinPts* is shown in Fig. 8.

In Figure 8, the trend of the optimal accuracy of DBSCAN clustering Seeds, Iris, and Segment are consistent with the tendency of clustering accuracy in their subsets, when *MinPts* takes different values. Also, the optimal parameter *MinPts* is the same, which are 18, 3, and 3, respectively. The optimal *MinPts* is selected and then optimize the parameter *Eps*. The results are shown in Table 7.

In Table 7, the interval of the *Eps* corresponding to the optimal clustering accuracy of all the samples of the data set overlaps with the range of the *Eps* searched by IMVO2 on its subset, which means that we can find a value in the *Eps* interval searched in the subset for clustering all samples of the dataset. The results show that we can optimize DBSCAN parameters by known label samples to mine unknown label samples, which is impossible for supervised learning algorithms.

We improve MVO to optimize the parameters of DBSCAN, not only can find the optimal clustering accuracy of DBSCAN but also find the range of *Eps* corresponding to the accuracy. This allows us to select a more appropriate value from the obtained *Eps* interval for DBSCAN clustering. In addition, the output of our improved MVO is the interval of the optimal variable, which can guide the solution of other optimization problems that the optimal parameters are continuous intervals.

V. CONCLUSION

Accurate clustering of DBSCAN requires the reasonable setting of parameters *MinPts* and *Eps*. For different datasets, the optimal *MinPts* and *Eps* corresponding to the highest clustering accuracy of DBSCAN are different. Even if the value of *MinPts* is set differently in the same dataset, the difference in optimal *Eps* will be very large, and sometimes the reasonable interval of *Eps* is quite small. This creates great difficulties for manually setting parameters. For example, in dataset D1, when $MinPts = 45$, the value range of *Eps* corresponding to the highest clustering accuracy of DBSCAN is [0.114046, 0.114058], which is a challenging to search manually. We use MVO for DBSCAN parameter optimization and improve it, denoted IMVO1, to find the interval of *Eps* when DBSCAN is clustering accurately. To optimize the parameter *Eps* corresponding to the maximum clustering accuracy of DBSCAN faster and more efficiently, we further improve IMVO1, denoted IMVO2. The experimental results show that the IMVO2 we designed not only can optimize the parameters of DBSCAN quickly and effectively but also the search interval can guide setting the parameters more reasonably.

REFERENCES

[1] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. Knowl. Disco. Data Mining*, Portland, OR, USA, Aug. 1996, pp. 226–231.

[2] D. Kellner, J. Klappstein, and K. Dietmayer, "Grid-based DBSCAN for clustering extended objects in radar data," in *Proc. IEEE Intell. Vehicles Symp.*, Alcala de Henares, Spain, Jun. 2012, pp. 365–370.

[3] J. Shen, X. Hao, Z. Liang, Y. Liu, W. Wang, and L. Shao, "Real-time superpixel segmentation by DBSCAN clustering algorithm," *IEEE Trans. Image Process.*, vol. 25, no. 12, pp. 5933–5942, Dec. 2016.

[4] M. Pavlis, L. Dolega, and A. Singleton, "A Modified DBSCAN clustering method to estimate retail center extent," *Geograph. Anal.*, vol. 50, no. 2, pp. 141–161, Sep. 2017.

[5] F. O. Ozkok and M. Celik, "A new approach to determine Eps parameter of DBSCAN algorithm," *Int. J. Intell. Syst. Appl. Eng.*, vol. 5, no. 4, pp. 247–251, 2017.

[6] W.-T. Wang, Y.-L. Wu, C.-Y. Tang, and M.-K. Hor, "Adaptive density-based spatial clustering of applications with noise (DBSCAN) according to data," in *Proc. Int. Conf. Mach. Learn. Cybern. (ICMLC)*, Guangzhou, China, Jul. 2015, pp. 445–451.

[7] A. Dockhorn, C. Braune, and R. Kruse, "An alternating optimization approach based on hierarchical adaptations of DBSCAN," in *Proc. IEEE Symp. Ser. Comput. Intell.*, Cape Town, South Africa, Dec. 2015, pp. 749–755.

[8] X. Chen, W. Liu, H. Qiu, and J. Lai, "APSCAN: A parameter free algorithm for clustering," *Pattern Recognit. Lett.*, vol. 32, no. 7, pp. 973–986, May 2011.

[9] J. Hou, H. Gao, and X. Li, "DSets-DBSCAN: A parameter-free clustering algorithm," *IEEE Trans. Image Process.*, vol. 25, no. 7, pp. 3182–3193, Jul. 2016.

[10] A. Karami and R. Johansson, "Choosing DBSCAN parameters automatically using differential evolution," *Int. J. Comput. Appl.*, vol. 91, no. 7, pp. 1–11, Apr. 2014.

[11] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *Knowl. Based Syst.*, vol. 89, pp. 228–249, Nov. 2015.

[12] M.-Y. Cheng and D. Prayogo, "Symbiotic Organisms Search: A new meta-heuristic optimization algorithm," *Comput. Struct.*, vol. 139, pp. 98–112, Jul. 2014.

[13] M. Srinivas and L. M. Patnaik, "Genetic algorithms: A survey," *Computer*, vol. 27, no. 6, pp. 17–26, Jun. 1994.

[14] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. 6th Int. Symp. Micro Mach. Hum. Sci.*, Nagoya, Japan, Oct. 1995, pp. 39–43.

[15] X. S. Yang, "Firefly algorithm, stochastic test functions and design optimization," *Int. J. Bio Inspired Comput.*, vol. 2, no. 2, pp. 78–84, Mar. 2010.

[16] X. S. Yang, "A new metaheuristic bat-inspired algorithm," in *Nature Inspired Cooperative Strategies for Optimization*. Berlin, Germany: Springer, 2010, pp. 65–74. [Online]. Available: https://link.springer.com/chapter/10.1007%2F978-3-642-12538-6_6

[17] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.

[18] A. V. Phan, M. L. Nguyen, and L. T. Bui, "Feature weighting and SVM parameters optimization based on genetic algorithms for classification problems," *J. Appl. Intell.*, vol. 46, no. 2, pp. 455–469, Mar. 2017.

[19] M. Chen and M. Liu, "Classification of flour types based on PSO-BP neural network," in *Proc. Chin. Control Decis. Conf. (CCDC)*, Shenyang, China, Jun. 2018, pp. 2591–2595.

[20] H. T. Kahraman, "A novel and powerful hybrid classifier method: Development and testing of heuristic k-nn algorithm with fuzzy distance metric," *Data Knowl. Eng.*, vol. 103, pp. 44–59, May 2016.

[21] C. Yilmaz, H. T. Kahraman, and S. Söyler, "Passive mine detection and classification method based on hybrid model," *IEEE Access*, vol. 6, pp. 47870–47888, 2018.

[22] P. Hu, Y. Wang, H. Wang, R. Zhao, C. Yuan, Y. Zheng, Q. Lu, Y. Li, I. Masood, "ALO-DM: A smart approach based on ant lion optimizer with differential mutation operator in big data analytics," in *Pro. Int. Conf. Database Syst. Adv. Appl.*, Gold Coast, QLD, Australia, May 2018, pp. 64–73.

[23] O. Buchtala, M. Klimek, and B. Sick, "Evolutionary optimization of radial basis function classifiers for data mining applications," *IEEE Trans. Syst., Man, Cybern. B. Cybern.*, vol. 35, no. 5, pp. 928–947, Oct. 2005.

[24] Rao, R. Venkata, "Design optimization of a spur gear train using TLBO and ETLBO algorithms," in *Teaching Learning Based Optimization Algorithm*. Cham, Switzerland: Springer, 2016, pp. 91–101.

[25] M. Rabbani, S. Mohammadi, and M. Mobini, "Optimum design of a CCHP system based on Economical, energy and environmental considerations using GA and PSO," *Int. J. Ind. Eng. Comput.*, vol. 9, no. 1, pp. 99–122, Apr. 2018.

[26] R. Bayindir, I. Colak, S. Sagiroglu, and H. T. Kahraman, "Application of adaptive artificial neural network method to model the excitation currents of synchronous motors," in *Proc. 11th Int. Conf. Mach. Learn. Appl.*, Boca Raton, FL, USA, Dec. 2012, pp. 498–502.

[27] M. K. Merugumalla and N. P. Kumar, "Optimized PID controller for BLDC motor using nature-inspired algorithms," *Int. J. Appl. Eng. Res.*, vol. 12, no. 1, pp. 415–422, Jan. 2017.

[28] Y, Arya and N. Kumar, "BFOA-scaled fractional order fuzzy PID controller applied to AGC of multi-area multi-source electric power generating systems," *Swarm Evol. Comput.*, vol. 32, pp. 202–218, Feb. 2017.

[29] M. K. Dosoglu, U. Guvenc, S. Duman, Y. Sonmez, and H. T. Kahraman, "Symbiotic organisms search optimization algorithm for economic/emission dispatch problem in power systems," *Neural Comput. Appl.*, vol. 29, no. 3, pp. 721–737, Feb. 2018.

[30] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, "Multi-verse optimizer: A nature-inspired algorithm for global optimization," *Neural Comput. Appl.*, vol. 27, no. 2, pp. 495–513, Feb. 2015.
[31] H. Faris, M. A. Hassonah, A. M. Al-Zoubi, S. Mirjalili, and I. Aljarah, "A multi-verse optimizer approach for feature selection and optimizing SVM parameters based on a robust system architecture," *Neural Comput. Appl.*, vol. 30, no. 8, pp. 2355–2369, Oct. 2018.
[32] H. Faris, I. Aljarah, and S. Mirjalili, "Training feedforward neural networks using multi-verse optimizer for binary classification problems," *Appl. Intell.*, vol. 45, no. 2, pp. 322–332, Sep. 2016.
[33] H. Zhao, X. Han, and S. Guo, "DGM (1, 1) model optimized by MVO (multi-verse optimizer) for annual peak load forecasting," *Neural Comput. Appl.*, vol. 30, no. 6, pp. 1811–1825, Sep. 2018.
[34] D. H. Wolper and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997.

WENHAO LAI is currently pursuing the Ph.D. degree with the Anhui University of Science and Technology. His research interests include pattern recognition, machine learning, and photoelectric information processing.

MENGRAN ZHOU received the Ph.D. degree in optics from the Anhui Institute of Optics and Fine Mechanics (AIOFM), Chinese Academy of Sciences, in 2006. He is currently a Professor with the Anhui University of Science and Technology. His research interests include photoelectric information processing and coal mine safety monitoring and control.

FENG HU is currently pursuing the Ph.D. degree with the Anhui University of Science and Technology. His research interests include machine learning, photoelectric information processing, and coal mine safety monitoring.

KAI BIAN is currently pursuing the master's degree with the Anhui University of Science and Technology. His research interests include machine learning, photoelectric information processing, spectra analysis, and medical data processing.

QI SONG is currently pursuing the master's degree with the Anhui University of Science and Technology. His research interests include picture processing, photoelectric information processing, and artificial intelligence.