IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 25 July 2023, accepted 12 August 2023, date of publication 22 August 2023,
date of current version 30 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3307412

RESEARCH ARTICLE

WOA-DBSCAN: Application of Whale Optimization Algorithm in DBSCAN Parameter Adaption

XINLIANG ZHANG AND SHIBO ZHOU
College of Navigation, Jimei University, Xiamen 361021, China

Corresponding author: Shibo Zhou (zhoushibo@jmu.edu.cn)

This work was supported in part by the Natural Science Foundation of Fujian Province under Grant 2020J01658, in part by the Open Project Fund of National Local Joint Engineering Research Center for Ship Assisted Navigation Technology under Grant HHXY2020002, and in part by the Doctoral Start-Up Fund of Jimei University under Grant ZQ2019012.

ABSTRACT Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is a classic density-based clustering method that can identify clusters of arbitrary shapes in noisy datasets. However, DBSCAN requires two input parameters: the neighborhood distance value (Eps) and the minimum number of sample points in its neighborhood (MinPts), to perform clustering on a dataset. The quality of clustering is highly sensitive to these two parameters. This paper introduces a parameter-adaptive DBSCAN clustering algorithm based on the Whale Optimization Algorithm (WOA-DBSCAN) to tackle this issue. The algorithm determines the parameter range based on the dataset distribution and utilizes the silhouette coefficient as the objective function. It iteratively selects the two input parameters of DBSCAN within the parameter range using the WOA. This approach ultimately achieves adaptive clustering of DBSCAN. Experimental results on five typical artificial datasets and six real UCI datasets demonstrate the effectiveness of the proposed WOA-DBSCAN algorithm. Compared with DBSCAN and its related optimization algorithms, WOA-DBSCAN shows significant improvements. The F-values of WOA-DBSCAN increased by 9.8%, 13.2%, and 2%, respectively, in two-dimensional artificial datasets. Additionally, the accuracy values on low to medium dimensional real datasets increased by 22.3%, 10%, and 23.3%. Hence, WOA-DBSCAN can maintain the clustering ability of DBSCAN while achieving adaptive parameter clustering.

INDEX TERMS DBSCAN algorithm, parameter adaptive, whale optimization algorithm, data mining.

I. INTRODUCTION

Cluster analysis is the process of dividing a set of objects into categories and making the objects in each category possess similarities to each other but differ from the objects in the other categories. In short, it is the process of grouping similar data points into the same category or cluster based on a similarity measure between the data. In addition, since clustering is an essential technique in unsupervised machine learning that does not require a training dataset, its most prominent advantage is that it can classify different datasets into different categories directly based on their distributional characteristics when faced with many unknown datasets. It saves the manual labeling of data sets and has been used in applications in data processing and database management [1].

The commonly used methods in cluster analysis include K-mean, hierarchical, and density clustering. Among the density clustering methods, the DBSCAN (Density-Based Spatial Clustering of Applications with Noise) algorithm [2] is one of the most popular algorithms, which defines clustering as the largest group of points connected by density. This makes the DBSCAN algorithm highly adaptable to most datasets, identifying distributions of arbitrary shape in the dataset and separating outliers that do not belong to any cluster. Therefore, DBSCAN is widely used in power system analysis [3], traffic flow prediction [4], image segmentation and other fields [5]. Although the DBSCAN algorithm has been widely used, it still has some drawbacks: two input

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

parameters are required, the neighborhood distance value of the samples (Eps) and the minimum number of sample points in the neighborhood (MinPts). Clustering results are susceptible to these two parameters [6], and good clustering results often depend on the user's domain knowledge, which can affect the usefulness of DBSCAN.

Meta-heuristics are optimization algorithms that solve complex problems by simulating specific behavioral mechanisms in natural and social systems. Standard meta-heuristic algorithms include Genetic Algorithms, Particle Swarm Optimization, Simulated Annealing, Ant Colony Optimization. These algorithms are widely used in scheduling optimization, industrial control and power systems due to their ability to improve the quality of the current solution through multiple iterations. The population intelligence optimization algorithm is a typical representative of meta-heuristic algorithms, which is simple, flexible, easy to implement, and can effectively avoid falling into the local optimal solution. WOA (Whale Optimization Algorithm) is a new optimization algorithm proposed by Mirjalili and Lewis [7], inspired by the humpback whale's hunting behavior and adopts the spiral and approximation to search for the optimal solution. The WOA algorithm has high optimization accuracy and is able to quickly converge to the optimal solution at a relatively low cost, achieving good results in solving optimization problems [8]. In addition, due to the characteristics of a whale optimization algorithm with a small number of parameters and global solid search ability, some researchers and scholars apply it in the field of clustering. Nasiri et al. [9] proposed a method to complete clustering by a whale optimization algorithm by initializing the clustering center, dividing the objects into different groups, and then the sum of the distances of the objects in the groups as the fitness function, and finally achieved the clustering of the data, and compared to the other heuristic methods, the WOA algorithm performed best. Singh et al. [10] improved on this by proposing an Enhanced Whale Optimization Algorithm (E-WOA) for clustering, which improves the search space through the positional updating of the water wave search algorithm, and employs the forbidden and neighborhood search strategies to enhance the algorithm's global search capability, compared to other heuristics, the E-WOA algorithm has more superior performance and feasibility than other heuristics. Although the meta-heuristic algorithm can achieve data clustering autonomously, it requires a significant amount of time for iteration. It is unable to identify the noise in the dataset, which makes the meta-heuristic algorithm more demanding on the dataset compared to the most popular algorithms based on density clustering.

In summary, we propose a WOA-based parameter adaptive DBSCAN algorithm (WOA-DBSCAN) based on the parameter sensitivity of the DBSCAN algorithm as well as the solid global searching ability of the WOA algorithm, which exploits the fast convergence and strong optimization ability of the WOA algorithm [24] to iteratively optimize the parameters of the DBSCAN adaptively. The silhouette coefficients are used as the fitness function of the WOA-DBSCAN algorithm, and the optimal number of clusters ensures the clustering quality. Compared to the DBSCAN algorithm, this algorithm excels in efficiently acquiring parameters for clustering. It also demonstrates outstanding performance on real datasets in medium to low dimensions; this suggests its capability to aid in tasks like database management, image segmentation, and anomaly detection in real-world scenarios. Additionally, the algorithm boasts simplicity in structure, low implementation complexity, and high clustering precision.

In order to provide a comprehensive account of the algorithm, this paper focuses on demonstrating the fundamentals of the algorithm, illustrating the working principle of WOA-DBSCAN, performing experimental analyses, and providing conclusions.

The contributions of this paper are as follows:

(1) A new composite algorithm is proposed to improve the efficiency of the DBSCAN algorithm by combining the whale optimization algorithm.

(2) The proposed algorithm is verified by testing on five artificial and six real datasets, which show excellent performance on low and medium-dimensional real datasets.

(3) The difficulty of parameter selection in the DBSCAN algorithm is better addressed.

(4) Alleviates the problem of decreasing flexibility faced by existing adaptive DBSCAN parameter algorithms, thus maintaining the clustering quality.

The rest of the paper is organized as follows. Section II reports related work in parameter optimization for DBSCAN algorithms, distinguishing between K-nearest neighbor, mathematical and meta-heuristic approaches and focusing on the strengths and weaknesses of the current research in all three approaches. Section III introduces the basic concepts and operational procedures of the DBSCAN and WOA algorithms and details the implementation of the WOA-DBSCAN algorithm. In Section IV, we present the experimental study conducted to evaluate the modeling capabilities of WOA-DBSCAN on synthetic and natural datasets, compare it with the results obtained by several popular clustering algorithms, and discuss the results. Finally, Section V concludes and provides directions for the following research phase.

II. RELATED WORK

Three main approaches have been proposed in the literature for estimating the parameters of the DBSCAN algorithm: The k-nearest neighbor algorithm, the mathematical algorithm, and the metaheuristic algorithm. In the K-nearest neighbor algorithm, a list of parameters is generated from the K-nearest neighbor distribution in the dataset. Then the parameters in the list are evaluated individually to obtain optimal clustering. In mathematical algorithms, the parameters used for clustering are estimated utilizing matrices or probabilities. Finally, there is a meta-heuristic algorithm, which searches for the optimal solution utilizing spatial search by evaluating

the internal metrics of the clustering effect as an objective function. The applications and challenges of the DBSCAN algorithm have been presented in [3], [4], [5], and [6], and in this section, we discuss the research of the three approaches.

A. K NEAREST NEIGHBOR ALGORITHM

K nearest neighbor algorithm is based on the distribution matrix of each object in the dataset according to the order from small to large to obtain K average distance values, and then through the indicators to evaluate the clustering effect of the K distance values, finally, the optimal solution is obtained among the K distance values. Sunita and Parag [11] employed KNN to develop an adaptive method for determining a sample point's neighborhood distance (Eps) in a variable-density dataset. By analyzing the density distribution of each attribute of the sample points, they derived the overall density distribution characteristics, enabling clustering datasets with uneven density distributions. This method performs exceptionally well on high-dimensional datasets. Cassisi et al. [12] introduced the IS-DBSCAN algorithm based on spatial hierarchy, utilizing the selection of reverse nearest neighbors for parameter optimization. It provides users with guidance for input parameters when employing the DBSCAN algorithm. The algorithm only requires one input parameter, K, which reduces the number of input parameters but significantly increases the time complexity. Lv et al. [13] further optimized the IS-DBSCAN algorithm and proposed the ISB-DBSCAN algorithm. This algorithm redefines the neighbor relationship of sample points and introduces the concept of kernel density reachability. It also introduces a new data index structure to speed up the algorithm's runtime and reduce the DBSCAN's dependence on parameters. Bryant and Cios [14] proposed the RNN-DBSCAN algorithm, estimating the observed density of clusters by traversing the K nearest neighbor graph and considering the inverse nearest neighbor number of the samples. Li et al. [15] introduced the KANN-DBSCAN algorithm, an adaptive DBSCAN clustering algorithm based on K-mean nearest neighbors. The algorithm utilizes the K-mean nearest neighbor method to generate candidate datasets and determines the number of clusters under different K values. The optimal Eps parameter is identified when the generated clusters are consistent for three consecutive times, and the corresponding MinPts are obtained using the mathematical expectation method. Li et al. [16] proposed a Partition KMNN-DBSCAN Algorithm, which constitutes a K-median nearest neighbor set as a list of Eps by calculating the K-nearest neighbor distance matrix of the input dataset and then finding the median of the K-nearest neighbor distances of all elemental points. The median method and the given list of Eps parameter values are then used to generate a list of Minpts parameter values. For the current Eps parameter list, the number of elemental points contained in the Eps neighborhood of all elemental points under different Eps is obtained sequentially. The median of the number of element points in the Eps neighborhood of all element points is used as the Minpts value corresponding to the current Eps value. The Minpts values corresponding to all Eps values are obtained to form a list of Minpts parameters, and the Minpts values correspond to the Eps values. Different K corresponds to different Eps and Minpts parameter values. The optimal parameters are then judged based on the stability of the K values. However, the method has two drawbacks: the quality of the parameter list is not controllable, and MinPts is generated from the Eps parameters by a bijective function, which reduces the flexibility of the parameter input. Li et al. [17] optimized DBSCAN by transforming Eps and MinPts into the input nearest neighbor parameter K, leveraging the basic properties of the nearest neighbor graph. The feasibility of the algorithm was verified using an artificial dataset. Li et al. [18] proposed the GNN-DBSCAN algorithm, combining grid division and K-nearest neighbors. Core sample points are selected through grid division, and data sets are clustered based on the dynamic radius of K-nearest neighbors. The nearest neighbor parameter K requires user input and is adaptively changed dynamically to avoid using multiple input parameters. Chen et al. [19] employed the K-nearest neighbor method to determine the distribution characteristics of the dataset. They generated a list of Eps and MinPts parameters and then determined the optimal parameters based on inter-cluster and intra-cluster density, yielding good clustering results. While the K-nearest neighbor algorithm accurately finds the Eps parameter in the DBSCAN algorithm, finding the MinPts parameter is only possible mathematically or by transforming it into other parameters to achieve the best result. This indirect approach to finding the parameter selection can impact the flexibility of parameter selection.

B. MATHEMATICAL ALGORITHMS

The methods of reducing the influence of parameters in mathematical algorithms are divided into three main types. The first is to reduce the input parameters to a single, reducing the difficulty of determining the parameters due to permutations and combinations; the second is to form the optimal set of solutions; and the third is to fit the parameters according to the results of the parameters and the clustering. Jeong et al. [20] utilized a quad-tree to define the density layer. They proposed the AA-DBSCAN algorithm, which automatically determines the Eps parameter but does not adopt the MinPts parameter. Wu et al. [21] proposed a linear DBSCAN algorithm called ISH-DBSCAN, which maps original sample points to hash buckets using multiple hash functions. This ensures that close original sample points remain similar after mapping. Initial parameter optimization was achieved, but the effectiveness of determining the parameters was average. Hou et al. [22] introduced a parameter-free clustering algorithm based on dividing the dataset. They applied histogram equalization to the similarity matrix and selected parameters based on the results to form dominant sets (D-sets). The parameters for the final input

of the DBSCAN algorithm are automatically determined by the dominance sets (D-sets). Wang et al. [23] proposed the MDBSCAN algorithm, which combines the idea of dataset division with the adjacency table in statistics. This algorithm generates two different Eps parameters and gradually determines the values of Eps and MinPts using the adjacency list. However, the optimization process is more complex and can involve multiple parameters. Wang and Lin [24] introduced an improved adaptive parameter DBSCAN algorithm. They determine the value range of Eps through kernel density estimation and then calculate MinPts using the mathematical expectation method. The maximum value of the silhouette is selected to determine the corresponding Eps and MinPts. Lu et al. [25] proposed an adaptive grey clustering algorithm (SAG-DBSCAN) based on a grey relational matrix to obtain a local density metric, which is capable of dividing the dataset into dense and discrete subsets and then clustering the densely populated subsets using the DBSCAN algorithm, where the input parameter MinPts is determined by the number of densely populated subsets, and the Eps is the set. Eps is the maximum nearest neighbor distance of the dense subset. After completing the clustering of the dense subsets, the data in the discrete subsets are then divided according to the clustering with the dense subset class clusters. The parameter optimization research based on mathematical statistics also involves studying the distance matrix of the dataset. However, similar to the DBSCAN algorithm optimized by K-nearest neighbors, there is still an issue of non-adaptability in the MinPts parameter.

C. META-HEURISTIC ALGORITHMS

Meta-heuristic algorithms are feature models refined by simulating the recognition of relevant behaviors and functions in biological, physical, social, and other fields. This class of algorithms achieves the search for optimal solutions by setting the internal clustering evaluation index as the objective function, relying on the unique search mechanism of the algorithm as well as its powerful search capability. By simulating the behaviors of different mechanisms in reality, this kind of algorithm can exactly fit with the density space clustering algorithms such as DBSCAN and DPC [9]. As a result, recent research on DBSCAN parameter optimization has been dominated by meta-heuristic algorithms. Hua et al. [26] proposed the PACA-DBSCAN algorithm, which utilizes the ant colony algorithm to optimize DBSCAN. It aims to reduce the sensitivity of DBSCAN parameters concerning the density of sample points. Juan and Julián [27] combined genetic algorithms with DBSCAN. Their method initially groups data information using elemental analysis and then iteratively optimizes the solution parameters using genetic algorithms. Promising results have been achieved in clustering various datasets. However, genetic algorithms still face challenges such as coding difficulties, slow convergence, and long iteration cycles in optimizing numbers. Mina and Majid [28] introduced a fuzzy earthworm algorithm-based optimization for DBSCAN. Their approach employs a fuzzy logic controller to adjust and optimize the parameters of the DBSCAN dynamically, showing good clustering results, particularly on 3D datasets. However, the dynamic changes of parameters in the adaptive clustering optimization process increase the algorithm's optimization difficulty. Adibifard et al. [29] proposed an improved adaptive DBSCAN, GA-DBSCAN-KMEANS, based on genetic algorithms and the K-means algorithm. Genetic algorithms are used to set subpopulation optimization parameters, and the K-means algorithm matches the best-quality subpopulation to generate new individuals, resulting in higher accuracy results. However, this algorithm involves a large number of iterations and lacks efficiency. Cao et al. [30] suggested using the particle swarm optimization algorithm to solve the optimal DBSCAN parameters. They utilize the DBI index as the fitness function and employ the particle swarm algorithm for iterative optimization, achieving an adaptive DBSCAN algorithm. Zhu et al. [31] proposed the HS-DBSCAN algorithm based on the parameters of the harmonic-optimized DBSCAN algorithm. They enhance the algorithm's robustness by predicting appropriate clustering parameters through a novel harmonic search algorithm, leading to improved clustering outcomes. However, the algorithm exhibits slow convergence. Zhou et al. [32] proposed an adaptive density spatial clustering method (CSA-DBSCAN) incorporating chameleon swarm algorithms, which optimizes the value of Eps to an exact value of 0.01, and the input parameters are used as the location of the chameleon swarms for optimization. Then the noise points are assigned to different class clusters by K Nearest Neighbor Algorithm (KNN). The algorithm can find accurate clustering results quickly and segment color images efficiently. However, the algorithm has two problems: the search boundary is not determined, and the number of iterations is too many. Yang et al. [33] used a novel meta-heuristic algorithm-arithmetic optimization method. They combined it with oppositional learning to implement an algorithm for DBSCAN parameter optimization (OBLAOA-DBSCAN). It features fast convergence speed and high accuracy through a mathematical optimizer that selects different optimization strategies at initialization and gradual convergence.

Compared with the CSA-DBSCAN algorithm, the OBLAOA-DBSCAN algorithm determines the parameter ranges through the normalized distance matrix and better improves the efficiency of the spatial search through the feature of opposites learning and double examination at each iteration.

Although intelligent heuristics can simultaneously optimize the parameters in the DBSCAN algorithm, issues such as excessive iterations and suboptimal optimization are commonly observed.

III. METHODOLOGY

A. DBSCAN ALGORITHM THEORY

The DBSCAN algorithm was proposed by Ester et al. in 1996 [2] as a density-based clustering algorithm. The main

Algorithm 1 DBSCAN Algorithm

Input: Sample set $D = \{x, x_2, \ldots, {}^{x}_{m}\}$;
Neighborhood parameters (Eps, MinPts);
1: Initializing a collection of core objects;
2: **for** j = 1, 2, . . . , m;
3: Determine the ε -neighborhood of sample x_j N_{Eps} (x_j);
4: **if** $|N_{Eps}(x_j)| \geq MinPts$;
5: Add sample x_j to the set of core objects: $\Omega = \Omega \bigcup \{x_j\}$;
6: **end if**;
7: **end for**;
8: Initialize the number of clusters: k=0;
9: Initialize the set of unvisited samples: $\Gamma = D$;
10: **while** $\Omega \neq \emptyset$;
11: Record the current set of unvisited samples: $\Gamma_{old} = \Gamma$;
12: Random selection of a core object $o \in \Omega$, Initializing the queue $Q =< o >$;
13: $\Gamma = \Gamma \backslash \{o\}$;
14: **while** $Q \neq \emptyset$;
15: Fetch the first sample in the Q queue q;
16: **if** $|N_{Eps}(q)| \geq MinPts$;
17: $\triangle = N_{Eps}(q) \bigcap \Gamma$;
18: Add the samples in $\triangle$ to the queue Q;
19: $\Gamma = \Gamma \backslash \triangle$;
20: **end if;**
21: **end while;**
22: $k = k + 1$, Number of clusters generated $C_k = \Gamma_{old} \backslash \Gamma$;
23: $\Omega = \Omega \backslash C_k$;
24: **end while;**
Output: Cluster division $C = \{C, C_2, \ldots C_k\}$.

idea is to define clusters as the maximal sets of density-connected points and partition the regions with sufficient density into clusters. The algorithm starts by randomly selecting an unvisited point and counts the number of points within the adjacent area radius of the point, which is less than Eps. If the number of points is greater than or equal to MinPts, the current point and its nearby points form a cluster, and the starting point is marked as visited. Then, all the points in the cluster are recursively processed in the same way to expand the cluster. If the number of neighboring points is less than MinPts, the point is temporarily marked as a noise point. This algorithm continuously processes unvisited points until all data points are assigned to a cluster or marked as noise. If the cluster is fully expanded, all points are marked as visited, and then the same algorithm is used to process non-visited points. The clustering process ends when all objects are marked as a particular cluster or noise. The pseudo-code for the DBSCAN is shown in algorithm 1.

B. WHALE OPTIMIZATION ALGORITHM THEORY (WOA)

Professor MIRJALILI created a specialized whale optimization algorithm. The algorithm is known for its simplicity and fast convergence, which combines three different behaviors: prey enveloping, hunting, and searching, and requires only the input of the population number S and the number of iterations T. The enveloping process is simulated by equations (1)-(4).

$$\chi_{\mathrm{k}}^{\mathrm{j+1}} = \chi_{\mathrm{k}}^{*} - \mathrm{A} \cdot \mathrm{D_k} \tag{1}$$

$$D_k = \left| C \cdot X_k^* - X_k^j \right| \tag{2}$$

$$A = 2a \cdot r_1 - a \tag{3}$$

$$C = 2 \cdot r_2 \tag{4}$$

In Equation 1, X_k^{j+1} represents the k-th component of the sample space coordinate X^{j+1}, X_k^* is the current optimal solution position. In Equation 4, X_k^j is the current whale position In (3) and (4), A and C are vector coefficients, a linearly decreases from 2 to 0 with the increase of the number of iterations, r_1 and r_2 are random numbers between 0 and 1.

1) BUBBLE ATTACK PHASE

Since whales spit out bubbles as they spiral swim, the entire bubble net attack is divided into two parts: shrink-wrap and update the spiral position. In the shrink-wrap method, the coefficient vectors are varied to simulate the behavior of humpback whales. In the spiral position updating method, the spiral motion of the whale is found based on the spiral equation expressed in (5). A humpback whale shrink-wrap or

spiral motion can be calculated using (6).

$$X_k^{j+1} = D_k \cdot e^{bl} \cdot \cos(2\pi l) + X_k^* \tag{5}$$

$$X_k^{j+1} = \begin{cases} X_k^* - A \cdot D_k, & p < 0.5 \\ X_k^* + D_k \cdot e^{bl} \cdot \cos(2\pi l), & p \geq 0.5 \end{cases} \tag{6}$$

In Equation 5, b is a constant defining the shape of the logarithmic spiral, l is a random number in $[-1, 1]$, and D_k is a random variable defined by (1). In Equation 6, the variable p is a random number in $[0, 1]$, A is a coefficient vector represented by (3).

2) PREY SEARCH PHASE

In order to avoid the solution at this stage being locally optimal, WOA uses search predation to expand the search area, the basic idea is that in the mathematical model of predator-prey behavior with constricted enclosures, the range of values of A is restricted to $[-1, 1]$, but when $|A| \geq 1$, the whale individuals choose one whale individual at random from the current whale population to approach. Search predation causes the current whale individual to deviate from the target prey, and enhance the global search ability of the whale population. The mathematical model of search predation behavior is shown in (7).

$$X_k^{j+1} = X_{rand}(t) - A \cdot D_k \tag{7}$$

In Equation 7, $X_{rand}(t)$ is the random selection of whale individuals from the current population. In the optimization process of specific problems, individual whales use different position update methods to continuously approach the optimal solution.

3) STEPS OF THE WHALE OPTIMIZATION ALGORITHM

The WOA algorithm first initializes a random set of solutions, and in each iteration, the search agent updates the position of the initial solution based on the randomly selected search agent or the optimal solution obtained so far. The parameter a in (3) is linearly decreased from 2 to 0 with the number of iterations, so as to gradually approach the optimal solution from exploration. When $|A| > 1$, A random search agent is selected, and when $|A| < 1$, the optimal solution is selected to update the search agent position. According to the randomly varying p value in (6), the whale can switch between spiral and circular movements. Finally, the run is terminated by meeting the termination criteria. The specific steps of the WOA algorithm are as Algorithm 2.

C. WOA-DBSCAN ALGORITHM

1) BASIC IDEA

The DBSCAN algorithm is a well-known density-based clustering algorithm capable of identifying clusters of arbitrary shapes in noisy datasets while effectively handling outliers. However, the clustering performance of the DBSCAN algorithm is significantly affected by two input parameters, and enhancing its performance often requires manual parameter adjustment through numerous experiments.

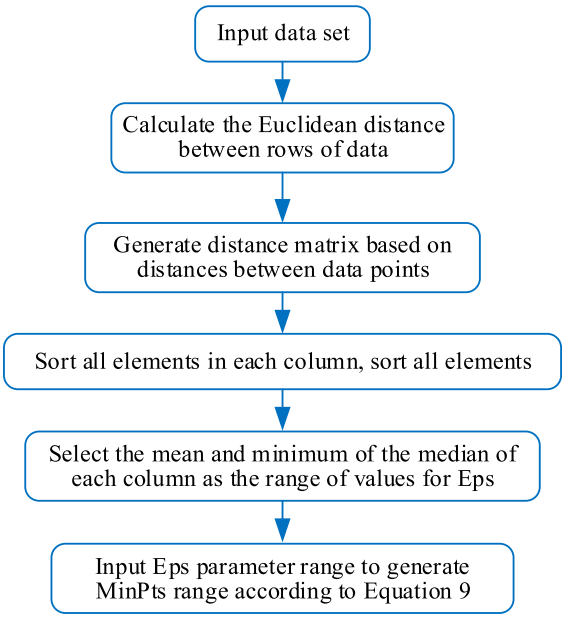

FIGURE 1. Flow chart for parameter range determination.

Furthermore, the DBSCAN algorithm lacks robustness, as changes in the target dataset necessitate the readjustment of the Eps and MinPts parameters, limiting its applicability. The Whale Optimization Algorithm (WOA) is a novel heuristic algorithm that demonstrates faster convergence when addressing multi-objective optimization problems compared to other algorithms. Moreover, the WOA algorithm produces more acceptable optimal solutions through parameter range exploration and optimal cluster selection. To address these issues, this paper proposes the WOA-DBSCAN algorithm. By employing the WOA algorithm, it swiftly identifies the global optimal solution. The algorithm automatically determines the number of clusters in the dataset using density peaks and incorporates the silhouette coefficient as the fitness function. It iteratively searches for the optimal value of the silhouette coefficient and ultimately provides the optimal solution to optimize the input parameters Eps and MinPts of DBSCAN.

2) DETERMINING PARAMETER RANGES AND OPTIMAL NUMBER OF CLUSTERS

a: ADAPTIVE CALCULATION OF PARAMETER RANGE

The range of Eps parameters is determined by the distribution characteristics of sample points in the data set, the parameter range adaptive calculation flow is shown in Fig. 1 and the main steps are as follows:

Step 1 Calculate the distance matrix between samples in the data set according to (8);

Step 2 Integration of the elements of the distance matrix into a single column by superposition;

Step 3 Sort the entire column elements, and select the median as the maximum Eps value, combined with the minimum value, it is the value range of Eps;

Algorithm 2 Whale Optimization Algorithm

Input: the whale population X_i ($i = 1, 2, \ldots, n$), Number of iterations T;
Fitness calculation rules F ();
1: z = find (max (F (X_i)));
2: X* = X_z;
3: t = 1;
4: **while** (t < T);
5: **for** each search agent;
6: Generate A, C and a based on position, randomly generate parameters 1 and p;
7: **if1** (p < 0.5);
8: **if2** (|A| < 1);
9: $X_k^{j+1} = X_k^* - A \cdot D_k$;
10: **else**;
11: $X_k^{j+1} = X_{noul}(t) - A \cdot D_k$;
12: **end if2;**
13: **else;**
14: $X_k^{j+1} = D_k \cdot e^{bl} \cdot \cos(2\pi l) + X_k^*$;
15: **end if1;**
16: **end for;**
17: Update when optimal solution exists X*;
18: t = t + 1;
19: **end while;**
Output: X*.

Step 4 According to the mathematical expectation method of (9), the average number of sample points is calculated within the maximum Eps-neighbor, obtained the range of MinPts in the entire data set.

$$D_{ij} = \left(x_i - x_j\right)\left(x_i - x_j\right)^T \tag{8}$$

In Equation 8, D_{ij} represents the value in the distance matrix. Unlike formulas (1) and (2), x_i and x_j represents the sample point.

$$MinPts = E(Eps) = \frac{1}{n}\sum_{i=1}^{n} P_i \tag{9}$$

In Equation 9, P_i represents the number of sample points included in each Eps of sample point i. The integrated process is shown in Fig. 1.

In this paper, the S2 dataset [34] is utilized as a case study, where Fig. 2 illustrates the distribution of sample points. The dataset comprises 2000 sample points categorized into 5 groups. Following the aforementioned steps, the calculated median Eps for the dataset is 18.46. Hence, the Eps parameter range for the S2 dataset is [0, 18.46], and the value range for MinPts is [0, 431].

b: DETERMINING THE OPTIMAL NUMBER OF CLUSTERS

In this paper, the decision diagram of the density peak clustering algorithm is used to adaptively calculate the number of clusters in the data set. This algorithm is based on two basic assumptions: (1) the local density of a cluster center (density peak point) is greater than that of its neighbors around it; (2) the distances between different cluster centers are relatively large. In order to find the cluster centers that satisfy these two conditions at the same time, which algorithm calculates the local density of each sample point in the data set ρ and its distance δ to the sample point whose local density is larger than that of the sample point, construct a $\rho - \delta$ visual-decision diagram, selecting the sample points with larger ρ and δ as the center of each cluster in the dataset. To automatically determine the cluster center of the dataset, $\gamma_i = \rho_i \times \delta_i$ is used as the weight of the cluster center, γ_i is arranged in descending order, and the slope of the two-point line segment is used to represent the downward trend of the weight of the cluster center. Using the analysis proposed by Liu et al. [35] to compare the distance relationship between suspected center points to determine the number of clusters. Taking the data set S2 as an example, the values of the first 50 points of the weight γ_i

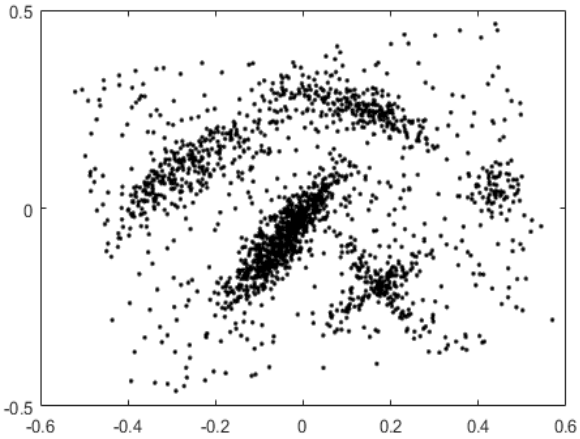

FIGURE 2. Two-dimensional display of the S2 dataset.

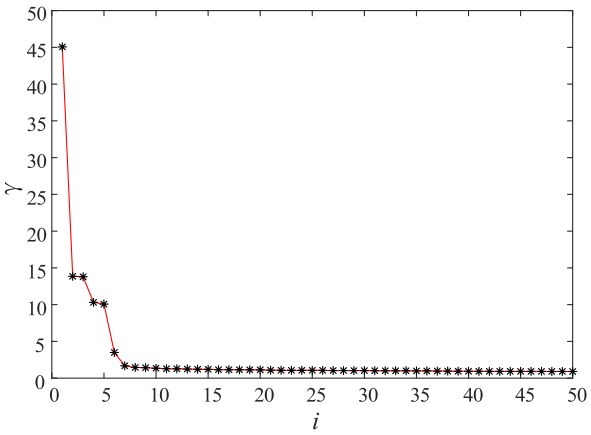

FIGURE 3. Optimal number of clusters decision diagram.

of the cluster center are shown in Fig. 3. It can be seen from the figure that the slope of the fifth point and the sixth point is obviously greater than that of the sixth point and the seventh point. So the number of cluster centers in dataset S2 is 5.

3) FITNESS FUNCTION SELECTION

The parameter optimization process of the WOA-DBSCAN algorithm involves the selection of a fitness function, which directly impacts the accuracy of the optimization results. This study chooses the silhouette coefficient as the fitness function for the WOA algorithm to optimize the parameters. Compared to other clusters, the silhouette coefficient measures the similarity between a sample point and its cluster. Among various evaluation indicators like the Davies - Bouldin Index (DBI) and Calinski - Harabasz score (CH), the silhouette coefficient is widely used and capable of assessing the clustering quality effectively [36]. A more significant silhouette coefficient indicates stronger intra-cluster relationships and greater inter-cluster distance, aligning with density-based clustering algorithms such as DBSCAN, DPC, and MDCA principles. Using Euclidean distance enables a more precise representation of dissimilarities between clusters. While the silhouette coefficient may provide lower evaluations for concave-shaped cluster structures, we have enhanced the reliability of the optimal solution by determining parameter ranges and selecting the best cluster number. The calculation of the silhouette coefficient is presented in (12).

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{10}$$

In Equation 10, $a(i)$ represents the average distance of the i-th object to other objects in the cluster where it belongs, and $b(i)$ represents the average distance of the i-th object to the objects in other clusters except the cluster where i is located. where $s(i) \in [-1, 1]$, and the closer $s(i)$d is to 1, the higher the clustering quality.

4) ITERATIVE PROCESS OF PARAMETER OPTIMIZATION

The input parameters of DBSCAN algorithm, Eps and MinPts, were optimized by simulating the enveloping

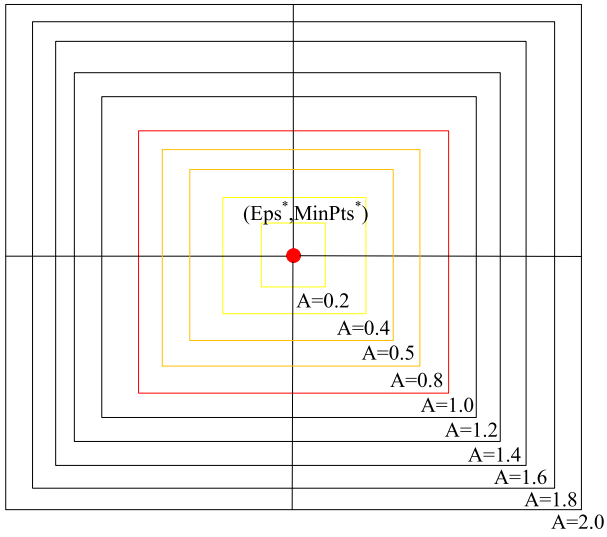

FIGURE 4. Coefficient vector distribution plot.

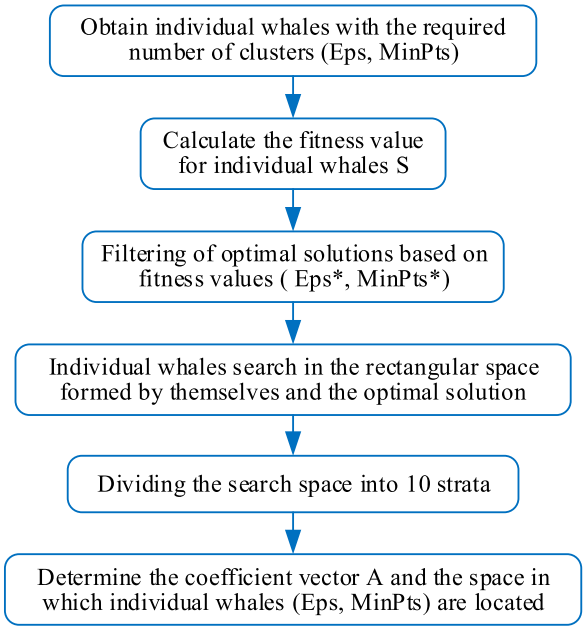

FIGURE 5. Iterative process flow chart.

predation behavior of whales. The input parameter of the DBSCAN algorithm is taken as the coordinate of the individual whale (Eps, $MinPts$), and the optimal solution coordinate of each iteration is (Eps^*, $MinPts^*$), where the value range of (Eps, $MinPts$) is determined according to the method in Section A of Methodology. Simulate the behavior of whale individuals (Eps, $MinPts$) to surround and prey to the optimal solution (Eps^*, $MinPts^*$). The coefficient vector division structure is shown in Fig. 4, and the iterative process flow chart is shown in Fig. 5.

The simulated whale algorithm involves bubble attack and random search behavior which is divided into two parts: shrinking encircling and spiral hunting, as shown in Fig. 6. The current solution (Eps, $MinPts$) and the optimal solution (Eps^*, $MinPts^*$) form the search space, and based on the

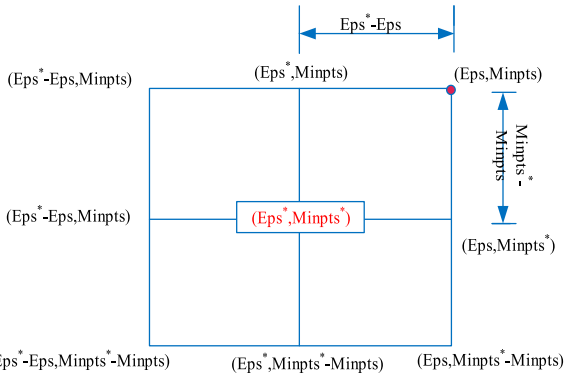

FIGURE 6. Shrink bounding update schematic.

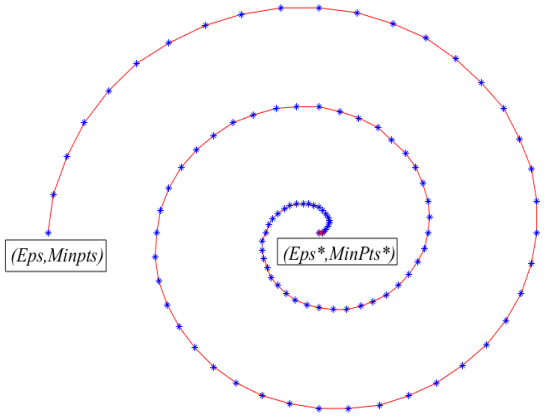

FIGURE 7. Schematic diagram of spiral position update.

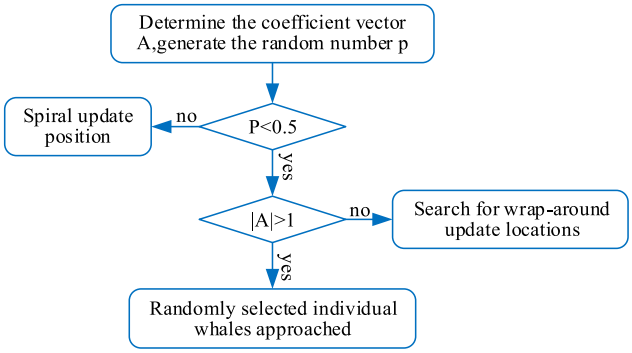

FIGURE 8. Flow chart of the update of the shrinkage boundary.

change in the iteration count, the current solution randomly updates to any position in the next interval. Fig. 7 shows the spiral hunting behavior of the whale, where the current solution (*Eps*, *MinPts*) approaches the optimal solution (*Eps**, *MinPts**) along a spiral path. The hunting behavior in the WOA-DBSCAN algorithm involves randomly selecting a whale individual to update its position using the shrinking encircling method, as illustrated in Fig. 6.

This method determines the search prey for the selected whale individual. The flow chart is shown in Fig. 8 and steps of bubble attack and searching for prey are as follows:

Step 1. Determine whether the coefficient vector A of the searched individual is [−1, 1].

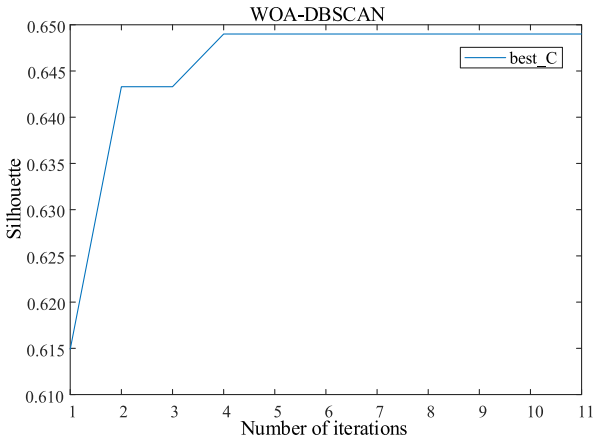

FIGURE 9. The optimal fitness function iteration value change graph of the S2 dataset.

Step 2. Use the individual whose coefficient vector A is in [−1, 1] to implement the bubble attack behavior to the optimal solution (*Eps**, *MinPts**), 50% of the individuals use the spiral The trajectory updates the position, and other individuals use the method of contraction encircling to update the position parameters.

Step 3: The individuals whose coefficient vector implements the behavior of searching for prey, that is, randomly select other whale individuals, and then, update the position by contraction encircling.

Step 4 Determine whether the number of iterations is reached or whether all whale individuals converge to the optimal solution. If the convergence times have not been reached or the optimal solution has not been reached, return to step 1 of the surrounding prey behavior. otherwise, the optimal solution is output.

Taking dataset S2 as an example, the WOA-DBSCAN algorithm optimizes the input parameters Eps and MinPts of the DBSCAN algorithm. The algorithm utilizes 500 randomly generated whale individuals and performs 10 iterations. The iterative changes of the fitness function value are depicted in Fig. 9. By the third iteration, the fitness value of the optimal solution stabilizes at 0.648. At this point, the optimal solution corresponds to a parameter Eps of 6.0002 and MinPts of 37. Therefore, the WOA-DBSCAN algorithm effectively achieves adaptive clustering in the DBSCAN algorithm based on the dataset's data distribution characteristics.

5) ALGORITHM IMPLEMENTATION STEPS

In the WOA-DBSCAN algorithm, we only need to enter the sample set D, the number of whales S, and the number of iterations T. The pseudo-code for the WOA-DBSCAN algorithm is shown in Algorithm 3, where the section about the whale optimization algorithm is in lines 10–18. The running procedure is shown in Fig. 10.

6) ALGORITHM PERFORMANCE ANALYSIS

For a dataset with *n* number of sample points, the space complexity of the DBSCAN algorithm mainly comes from the

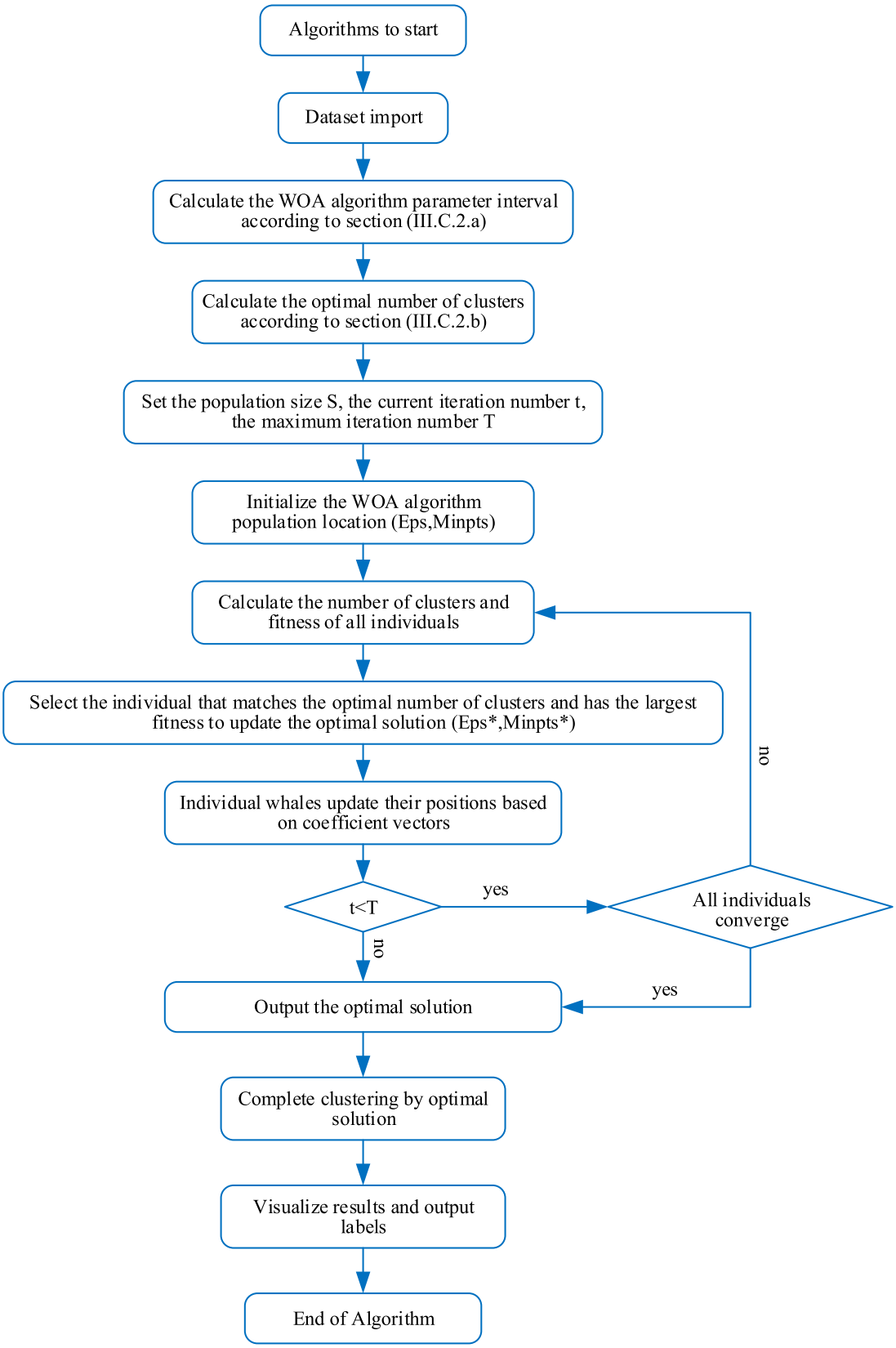

FIGURE 10. Core algorithm flow chart.

Algorithm 3 WOA-DBSCAN Algorithm

Input: Sample set $D = \{x_1, x_2, \ldots, x_m\}$,Population Size S, Number of iterations T;
1: Calculate the distance matrix for dataset D $D^* = \{\xi_1, \xi_2, \ldots, \xi_m\}$;
2: Calculate the median of each column in D* $K = \{k_1, k_2, \ldots, k_m\}$;
3: Get the range of Eps values $Eps = [\min(K), \max(K)]$;
4: Get the range of values for MinPts $MinPts = [E(\min(K)), E(\max(K))]$;
5: Calculate the local density ρ and the sample point distance δ, Determine the number of clusters based on the distance relationship (*best_n*);
6: Generate (*Eps, MinPts*) list of individual whalesa parameters based on Eps and MinPts;
7: t = 1;
8: **While** (t < T);
9: Obtain the profile coefficient value s(i) of an individual whale with the number of clusters n;
10: The solution that matches the number of clusters and has the largest Sil houette is the optimal solution;
11: Determining the magnitude of the coefficient A by the position of (*Eps, MinPts*) and (*Eps, MinPts**);
12: **if** $|A| \leq 1$;
13: 50% of individual whales $X_k^{j+1} = X_k^* - A \cdot D_k$, others $X_k^{j+1} = X_{\text{rame}}(t) - A \cdot D_k$;
14: **else**;
15: $X_k^{j+1} = X_{\text{radi}}(t) - A \cdot D_k$;
16: **end if**;
17: t = t + 1;
18: **end while**;
19: Initializing a collection of core objects;
20 **for** j = 1, 2, . . . , m;
21: Determine the ε-neighborhood of sample x_j : $N_{\text{Fop}}\left(x_j\right)$;
22: **if** $\left|N_{Ess}\left(x_j\right)\right| \geq MinPts^*$;
23: Add sample x_j to the set of core objects: $\Omega = \Omega \bigcup \left\{x_j\right\}$;
24: **end if**;
25: **end for**;
26: Initialize the number of clusters: k = 0;
27: Initialize the set of unvisited samples: $\Gamma = D$;
28: **while** $\Omega \neq \varnothing$;
29: Record the current set of unvisited samples: $\Gamma_{\text{old}} = \Gamma$;
30: Random selection of a core object $o \in \Omega$, Initializing the queue $Q =< 0 >$;
31: $\Gamma = \Gamma \backslash \{o\}$;
32: **while** $Q \neq \varnothing$;
33: Fetch the first sample in the queue q;
34: **if** $N_{Eps^*}(q) \geq MinPts^*$;
35: $\Delta = N_{Eps^*}(q) \cap \Gamma$;
36: Add the samples in Δ to the queue Q;
37: $\Gamma = \Gamma \backslash \Delta$;
38: **end if**;
39: **end while**;
40: $k = k + 1$, Number of clusters generated $C_k = \Gamma_{\text{old}} \backslash \Gamma$;
41: $\Omega = \Omega \backslash C_k$;
42: end while;
Output: Cluster division $C = \{C_1, C_2, \ldots, C_k\}$.

cluster labels and the identification of the sample point categories (core points, boundary points, and noise points), so the space complexity of the DBSCAN algorithm is $O(n)$. Compared with the DBSCAN algorithm, the WOA-DBSCAN algorithm adds the optimization and iterative process of the whale algorithm, and the part that mainly increases the space complexity is the fitness function. When the number of samples generated by the whale algorithm is m, the generated space complexity is $O(m)$. Therefore, the space complexity of the WOA-DBSCAN algorithm is $O(n + m)$.

The time complexity of the WOA-DBSCAN algorithm is mainly spent in the DBSCAN algorithm and during the optimization iterative process of the whale algorithm. When the number of sample points in the data set is n, the time

complexity of the DBSCAN algorithm mainly comes from the time required to find the points in the Eps-neighbor of each sample point and the determination of the type of each sample point according to the points in the Eps-neighbor., the worst-case time complexity of the DBSCAN algorithm is $O(n^2)$. The time complexity of the whale optimization algorithm mainly comes from the process of optimization, iteration and each fitness function calculation of individuals in the population. Under the assumption that the population size is S and the number of iterations is T, the difference between Eps and MinPts for each optimization is calculated. The parameters are all two-dimensional matrices, so the time complexity of the whale algorithm in the optimization phase is $O(2ST)$. In the iterative process of the whale algorithm, the time complexity of the total number of iterations is $O(n^2T)$. The time complexity of the fitness function optimization mainly comes from the calculation of the silhouette coefficient. The total silhouette coefficient of the clustering effect evaluation needs to first calculate the silhouette coefficient of a single vector, and then average the silhouette coefficients of all sample points. The class results in the total silhouette coefficient, so the time complexity of the total silhouette coefficient is $O(n^2)$. To sum up, the upper limit of the total time complexity of the WOA-DBSCAN algorithm is $O(n^2(2+T)+2ST)$, which is consistent with the DBSCAN algorithm in magnitude.

IV. EXPERIMENT

A. EXPERIMENTAL ENVIRONMENT AND COMPARISON ALGORITHMS

The WOA-DBSCAN algorithm is implemented in MATLAB, using a Windows 10 operating system with a 64-bit architecture. The hardware environment consists of an Intel Core I5-7200 processor, 4GB of RAM, and a 128GB hard disk.

B. RELATIONSHIPS BETWEEN THE ALGORITHMS USED IN THE EXPERIMENTS

RNN-DBSCAN [17] is an improved version of the DBSCAN clustering algorithm based on reverse K-nearest neighbors. This algorithm reduces the two parameters, Eps and MinPts, to the expected quantity of reverse nearest neighbors, denoted as K, and performs clustering by controlling the input of K. KANN-DBSCAN [15] is similar to the RNN-DBSCAN algorithm in that it also estimates parameter values using nearest neighbor algorithms. However, KANN-DBSCAN determines the Eps parameter based on the average nearest neighbor distance and then establishes a relationship with MinPts through mathematical expectations. It provides a parameter list containing the parameters required to achieve excellent clustering, which are selected based on their performance. AF-DBSCAN [37] clusters data by mathematically fitting the clustering results, and eventually identifies the optimal parameters based on the fitting results obtained from multiple experiments. DBSCAN [2] is the most classic density-based clustering algorithm and one of the most widely used algorithms currently. In this experiment, the DBSCAN algorithm achieved the optimal results after multiple parameter adjustments based on existing research. WOA-DBSCAN algorithm draws inspiration from K-nearest neighbors and density peak clustering algorithms. It aims to accelerate the iteration of the Whale Optimization Algorithm and ensure the clustering quality of the DBSCAN algorithm. The algorithm utilizes silhouette coefficient to search for the optimal solution.

TABLE 1. Introduction to the data set.

Dataset	Observations	Classes	Dimensions
Aggregation[38]	788	7	2
Compound[38]	399	6	2
R15[38]	600	15	2
Spiral[38]	312	3	2
P2glob[39]	2000	4	2
Iris[40]	150	3	4
Wine[40]	178	3	13
Sym[40]	350	3	2
Seeds[40]	210	3	7
Zoo[40]	101	7	16
Landsat[40]	2000	6	36

C. EXPERIMENTAL DATASET

In this paper, five well-known artificial data sets and six real data sets are used for testing, and the performance of the algorithm is comprehensively evaluated. The details of the dataset and its summary information are shown in Table 1. All data sets are under the same conditions; run the WOA-DBSCAN algorithm and compare the obtained results. The Aggregation dataset represents a cluster-connected dataset, Compound consists of clustered datasets with uneven cluster density, R15 represents clustered datasets with uniform but unconnected density, Spiral represents a bar-like dataset with uniform density, and P2Glob contains two different shapes with uniform density. Among the six real datasets, the Sym dataset was obtained from the Waikato Environment for Knowledge Analysis (WEKA) data mining software. In contrast, the remaining datasets were sourced from the UCI machine learning repository [40], which is used to test the performance of the algorithm under accurate data. Detailed descriptions of the datasets and their clustering effects can be found in Sections B and D of Part IV.

D. CLUSTERING EVALUATION INDICATORS

To assess the feasibility of the WOA-DBSCAN algorithm and compare its clustering quality with other algorithms, this paper introduces the F- value as a clustering-specific index [41]. The F value ($F-Score$) is a comprehensive measure used to evaluate clustering results. Precision, which represents the accuracy of clustering, is defined as the ratio of correctly identified data to the total number of identified data.

Recall rate, on the other hand, represents the ratio of correctly identified data to the actual total number of data. The F value is calculated using (11).

$$F = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{11}$$

Furthermore, we have introduced additional clustering metrics to comprehensively evaluate the clustering results of different algorithms. These metrics include Accuracy (ACC) [42], Adjusted Mutual Information (AMI) [43], and Adjusted Rand Index (ARI) [44]. The accuracy rate, ACC, measures the ratio of correctly clustered records to the total number of records, and its calculation formula is provided in (12).

$$ACC = \frac{n_{correct}}{n_{tatal}} \tag{12}$$

Among them, $n_{correct}$ represents the number of correct clusters, n_{tatal} represents the number of all clusters, and the value range of ACC is [0, 1] The closer it is to 1, the better the clustering effect.

Mutual information is usually used to measure the degree of agreement between two data distributions. The actual category labels are used to evaluate the clustering quality. The value range of AMI is $[-1, 1]$. The closer it is to 1, the better the clustering effect. The definition of the index is shown in the (13).

$$AMI(U, V) = \frac{MI(U, V) - E\{MI(U, V)\}}{F(H(U), H(V)) - E\{MI(U, V)\}} \tag{13}$$

In Equation 13, $E\{MI(U, V)\}$ is the expectation of $MI(U, V)$, and the calculation method is shown in (14).

$$\begin{aligned} &E\{MI(U, V)\} \\ &= \sum_{i=1}^{R}\sum_{j=1}^{C}\sum_{k=(a_i+b_j-N)^+}^{\min(a_i,b_j)} \frac{k}{N}\log\left(\frac{N \times k}{a_i \times b_j}\right) \\ &\quad \cdot \frac{a_i!b_j!(N-a_i)!(N-b_j)!}{N!k!(a_i-k)!(b_j-k)!(N-a_i-b_j+k)!} \end{aligned} \tag{14}$$

In Equation 14, $(a_i + b_j - N)$ is $\max(1,\ a_i + b_j - N)$, a_i and b_j are the sum of the i-th row and j-th column of $MI(U, V)$, respectively, see (15) and (16).

$$a_i = \sum_{j=1}^{C} m_{ij} \tag{15}$$

$$b_j = \sum_{i=1}^{R} m_{ij} \tag{16}$$

Adjusted Rand Index to evaluate the pros and cons of the clustering algorithm by comparing the results of the clustering algorithm with the real classification situation. The value range of ARI is $[-1, 1]$. The definition of ARI index is shown in (17).

$$ARI = \frac{RI - E|RI|}{\max(RI) - E|RI|} \tag{17}$$

E. EXPERIMENTS WITH ARTIFICIAL DATA SETS AND DISCUSSION

Fig. 11 presents the visualization of clustering results obtained by WOA-DBSCAN and other algorithms on various artificial datasets, with the DBSCAN algorithm's parameters optimized for multiple inputs. To provide a more comprehensive evaluation of the clustering quality of different comparison algorithms, Table 2 displays the clustering index F values and lists the parameter values used for each algorithm in the experiment. The parameters for WOA-DBSCAN, RNN-DBSCAN, KANN-DBSCAN, and AF-DBSCAN are all calculated through self-adaptive calculations. The parameters for DBSCAN are the ones that yield the best clustering performance. In Table 2, the bold and emphasized values indicate superior experimental results.

The Aggregation dataset represents a cluster-connected dataset with uniform density. The RNN-DBSCAN, WOA-DBSCAN, KANN-DBSCAN, and DBSCAN algorithms can accurately cluster this dataset. However, the AF-DBSCAN algorithm needs help to identify accurate clusters. Evaluation results based on F-Score, ACC, ARI, and AMI indicate that the RNN-DBSCAN algorithm achieves the best clustering results, followed closely by the WOA-DBSCAN algorithm, with a slight difference between them. The DBSCAN algorithm performs relatively well after adjusting its parameters.

The Compound dataset represents uneven density, different cluster shapes, and inclusive clusters. The WOA-DBSCAN, KANN-DBSCAN, and DBSCAN algorithms can accurately identify the cluster types in this dataset. Evaluation results based on F-Score, ACC, ARI, and AMI show that the WOA-DBSCAN algorithm achieves higher accuracy, followed by the DBSCAN algorithm with multiple parameter adjustments and the KNN-DBSCAN algorithm. The evaluation results for all three algorithms are above 0.8.

The R15 and Spiral datasets represent a single shape and uniform density. The WOA-DBSCAN, RNN-DBSCAN, KANN-DBSCAN, and DBSCAN algorithms can accurately identify clusters with precise clustering. The KANN-DBSCAN algorithm and the regular DBSCAN algorithm sometimes misclassify a few points as noise, while the AF-DBSCAN algorithm merges clusters in the central part that should not be merged. Evaluation results based on F-Score, ACC, ARI, and AMI show that all five algorithms perform well, with the WOA-DBSCAN algorithm demonstrating the best overall performance.

The P2glob dataset represents a dataset with two different shapes. The RNN-DBSCAN algorithm, affected by the reduced input parameter dimension, has some impact on the clustering effect when the cluster shapes are significantly different. The KANN-DBSCAN algorithm, which performs well on other datasets, exhibits poor performance on the P2glob dataset. Analysis shows that the Eps and MinPts parameters generated by the KANN-DBSCAN algorithm have a gradient ascent feature and are highly correlated. As a

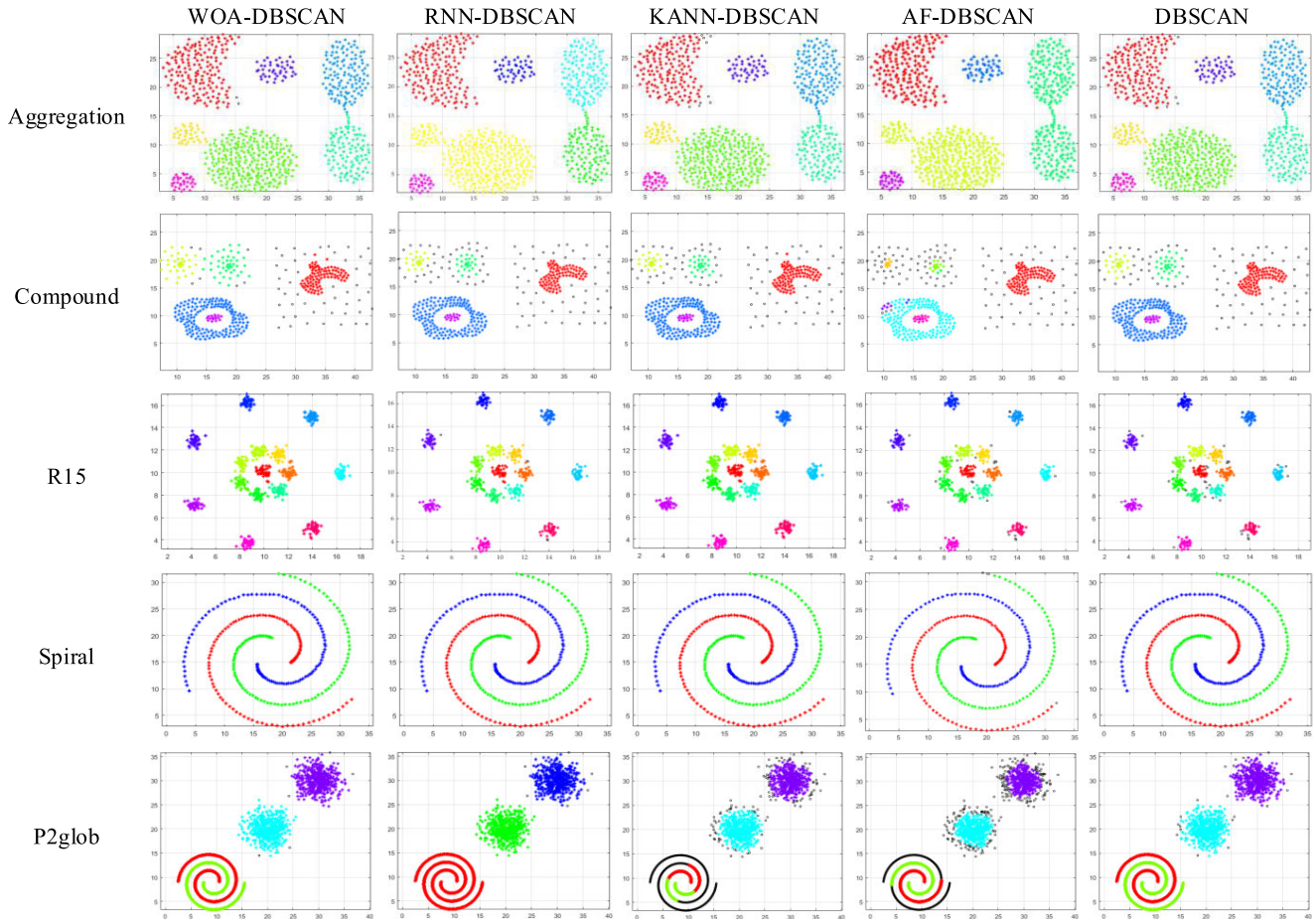

FIGURE 11. Visualization of clustering results of artificial datasets by different algorithms.

result, when the Eps parameter is reasonable, the MinPts parameter becomes unreasonable, leading to a lack of correct solutions. Evaluation results based on F-Score, ACC, ARI, and AMI show that the WOA-DBSCAN algorithm performs the best on this dataset, followed by the regular DBSCAN algorithm.

In summary, the WOA-DBSCAN algorithm, compared to other optimized DBSCAN parameter algorithms, achieves better clustering results on various types of two-dimensional artificial datasets. Compared to the DBSCAN algorithm, the adaptive parameters improve accuracy (ACC) while maintaining the flexibility of the DBSCAN algorithm.

F. UCI DATASET VALIDATION AND DISCUSSION

Table 3 presents the clustering results of the WOA-DBSCAN, along with the comparison algorithms, RNN-DBSCAN, KANN-DBSCAN, AF-DBSCAN, and DBSCAN, applied to six UCI datasets. The bold and emphasized values in the table represent superior experimental outcomes.

The Iris dataset, consisting of 150 iris flower samples with four attributes, demonstrates good performance by both the WOA-DBSCAN and AF-DBSCAN algorithms. The WOA-DBSCAN algorithm achieves the best clustering result with a comprehensive accuracy of 0.9, followed by the AF-DBSCAN algorithm with a comprehensive accuracy of 0.742. The RNN-DBSCAN, KANN-DBSCAN, and DBSCAN show similar performance on this dataset.

For the Wine dataset, which contains 178 wine samples with 13 features and three clusters, the RNN-DBSCAN algorithm achieves the best clustering result, followed by the parameter-adjusted DBSCAN algorithm. The accuracy of the clustering results for all algorithms is above 0.6.

The Sym dataset, composed of 350 samples with two features and three categories, demonstrates good performance by the WOA-DBSCAN, KANN-DBSCAN, and AF-DBSCAN algorithms. The WOA-DBSCAN algorithm achieves the highest clustering effect with a value of 0.912, while the RNN-DBSCAN, KANN-DBSCAN, and AF-DBSCAN algorithms perform similarly with clustering effects around 0.72.

The Seeds dataset consists of three types of wheat seeds described by seven geometric parameters. The experimental results in Table 3 show that the WOA-DBSCAN, RNN-DBSCAN, AF-DBSCAN, and DBSCAN algorithms achieve similar clustering effects, with the RNN-DBSCAN algorithm performing slightly better.

TABLE 2. Clustering parameters and F-Score of different comparison algorithms.

Data set	Algorithm	Cluster number	*F-Score*	*ACC*	*AMI*	*ARI*
Aggregation	WOA-DBSCAN	7	0.992	0.979	0.971	0.984
	RNN-DBSCAN	7	**0.999**	**0.999**	**0.999**	**0.999**
	KANN-DBSCAN	7	0.985	0.985	0.976	0.979
	AF-DBSCAN	5	0.827	0.815	0.888	0.808
	DBSCAN	7	0.987	0.965	0.979	0.987
Compound	WOA-DBSCAN	6	**0.975**	**0.954**	**0.936**	**0.967**
	RNN-DBSCAN	6	0.933	0.894	0.884	0.892
	KANN-DBSCAN	6	0.902	0.825	0.871	0.874
	AF-DBSCAN	6	0.845	0.728	0.745	0.731
	DBSCAN	5	0.937	0.873	0.834	0.852
R15	WOA-DBSCAN	15	**0.992**	**0.991**	**0.988**	0.982
	RNN-DBSCAN	15	0.990	0.990	0.937	**0.984**
	KANN-DBSCAN	15	0.990	0.989	0.983	0.978
	AF-DBSCAN	14	0.898	0.882	0.931	0.884
	DBSCAN	15	0.945	0.946	0.934	0.921
Spiral	WOA-DBSCAN	3	**0.999**	**0.999**	**0.999**	**0.999**
	RNN-DBSCAN	3	**0.999**	**0.999**	**0.999**	**0.999**
	KANN-DBSCAN	3	**0.999**	0.993	0.985	0.990
	AF-DBSCAN	3	0.990	0.967	0.926	0.951
	DBSCAN	3	**0.999**	**0.999**	**0.999**	**0.999**
P2glob	WOA-DBSCAN	4	**0.997**	**0.996**	**0.992**	**0.995**
	RNN-DBSCAN	3	0.698	0.694	0.652	0.687
	KANN-DBSCAN	4	0.589	0.588	0.591	0.514
	AF-DBSCAN	4	0.7315	0.731	0.683	0.574
	DBSCAN	4	0.9905	0.990	0.980	0.987

The Zoo dataset, representing animals from a zoo, contains samples from six animal classes described by 17 parameters. According to the experimental data in Table 3, the WOA-DBSCAN algorithm performs the best, followed by the RNN-DBSCAN algorithm. In contrast, the KANN-DBSCAN, AF-DBSCAN, and DBSCAN algorithms show similar clustering effects.

In conclusion, the WOA-DBSCAN algorithm outperforms other adaptive DBSCAN algorithms by efficiently clustering multidimensional real datasets without input parameters. Determining the parameter ranges and the number of class clusters helps the whale optimization algorithm converge more quickly. The constraint of increasing the number of class clusters also ensures that the optimal solution found by WOA-DBSCAN matches the actual characteristics of the dataset. This is one of the reasons why the algorithm performs well on low and medium-dimensional real datasets. However, using Euclidean distance for sample distance calculation is quickly limited by the dimension, which leads to a decrease in clustering stability when the dataset dimension is high. The RNN-DBSCAN algorithm is highly stable, and the clustering effect is better after parameter tuning. The DBSCAN algorithm lacks practical parameter tuning based on the clustering effect of the actual dataset. The KANN-DBSCAN algorithm shows abnormal convergence in cluster number generation for high-dimensional real datasets. The AF- DBSCAN algorithm performs poorly on two-dimensional artificial datasets with suboptimal clustering effects.

G. SUMMARY OF EXPERIMENTAL RESULTS AND PARAMETER SENSITIVITY ANALYSIS

1) SUMMARY OF EXPERIMENTAL RESULTS

For each object in the datasets, the classification result is deterministic, allowing us to evaluate the clustering performance using ACC, AMI, and ARI metrics. WOA-DBSCAN algorithm, which we proposed, combines the K-Medians Nearest Neighbor algorithm [16] to determine the value ranges of Eps and MinPts. It also incorporates the concept of density peak clustering to identify the optimal number of clusters. This guarantees the speed of convergence of the algorithm and the quality of the optimal solution, resulting in excellent clustering performance on both 2D artificial

TABLE 3. Comparison of ACC, AMI, and ARI indicators of each algorithm.

Data set	Clustering Algorithm	ACC	AMI	ARI
Iris	WOA-DBSCAN	**0.940**	**0.840**	**0.889**
	RNN-DBSCAN	0.706	0.593	0.664
	KANN-DBSCAN	0.613	0.402	0.388
	AF-DBSCAN	0.866	0.673	0.687
	DBSCAN	0.647	0.535	0.515
Wine	WOA-DBSCAN	0.635	0.360	0.239
	RNN-DBSCAN	**0.742**	**0.496**	0.359
	KANN-DBSCAN	0.534	0.156	0.152
	AF-DBSCAN	0.609	0.301	0.396
	DBSCAN	0.674	0.344	**0.414**
Sym	WOA-DBSCAN	**0.9285**	**0.884**	**0.923**
	RNN-DBSCAN	0.725	0.785	0.742
	KANN-DBSCAN	0.808	0.665	0.682
	AF-DBSCAN	0.802	0.654	0.72
	DBSCAN	0.754	0.427	0.534
Seeds	WOA-DBSCAN	**0.567**	0.424	0.323
	RNN-DBSCAN	0.536	**0.511**	**0.534**
	KANN-DBSCAN	0.171	0.251	0.207
	AF-DBSCAN	0.523	0.428	0.344
	DBSCAN	0.519	0.322	0.235
Zoo	WOA-DBSCAN	0.829	**0.804**	**0.709**
	RNN-DBSCAN	**0.842**	0.683	0.594
	KANN-DBSCAN	0.600	0.699	0.604
	AF-DBSCAN	0.537	0.572	0.500
	DBSCAN	0.605	0.743	0.653
Landsat	WOA-DBSCAN	0.463	0.432	0.512
	RNN-DBSCAN	**0.733**	**0.546**	0.342
	KANN-DBSCAN	0.435	0.387	0.406
	AF-DBSCAN	0.374	0.342	**0.517**
	DBSCAN	0.593	0.387	0.436

datasets and some real-world datasets. RNN-DBSCAN performs well on 2D artificial datasets, with all evaluation metrics exceeding 0.9. However, as this algorithm simulates the Eps and MinPts parameters through the reverse nearest neighbor count, it limits the clustering capability of the DBSCAN algorithm. Therefore, in some UCI real-world datasets, the DBSCAN algorithm with multiple parameter adjustments achieves even better clustering results. KANN-DBSCAN shows good performance on most artificial datasets but is not suitable for datasets with significantly different shapes.

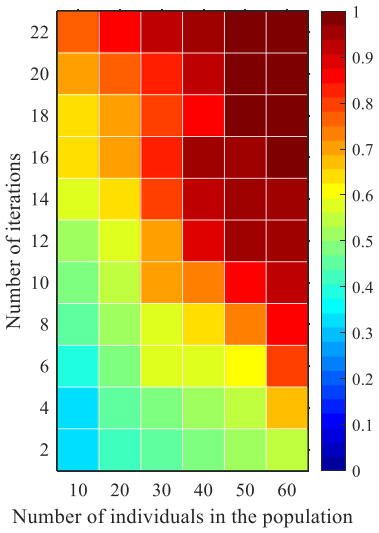

FIGURE 12. ARI of WOA-DBSCAN on aggregation dataset with different parameters.

Additionally, during experiments with high-dimensional real-world data, KANN-DBSCAN encounters the issue of parameter K failing to converge. AF-DBSCAN determines the most appropriate parameters by fitting the impact of parameter changes on the DBSCAN algorithm. However, we found that this method has relatively high requirements for datasets, especially being sensitive to datasets with uneven densities. DBSCAN is the result of multiple parameter adjustments based on existing research. During the experiments, the Eps parameter's step size for DBSCAN was set to 0.01, and the MinPts parameter's step size was set to 1. Grid-based parameter tuning was performed for each dataset. However, due to the step size, the ARI on some datasets could not reach the optimal value, which might lead to a potential underestimation of DBSCAN algorithm's clustering capability in our experimental results.

2) PARAMETER SENSITIVITY ANALYSIS

Parameter sensitivity analysis is a method to assess the extent to which algorithm outputs are affected by changes in parameters. For the WOA-DBSCAN algorithm, the objective of parameter sensitivity analysis is to understand how adjusting parameter values impacts clustering results. We conducted this analysis using the representative Aggregation dataset, with a primary focus on two parameters: the population size (S) and the number of iterations (T). The population size is varied from 10 to 60 with a step size of 10, while the number of iterations ranges from 1 to 22 with a step size of 2. Figure 12 displays the average value of the Adjusted Rand Index (ARI) across multiple experiments.

As shown in Figure 12, as the values of S and T increase, the ARI metric approaches 1, and this process is linear before reaching the optimal clustering. Therefore, when S and T are large enough, WOA-DBSCAN can achieve parameter-adaptive clustering.

V. CONCLUSION

Clustering is a widely used data mining technique that helps reveal data distributions and interesting patterns [45]. DBSCAN, a well-known density-based clustering algorithm, excels in clustering analysis and can handle datasets with noise and arbitrary shapes. However, its clustering quality heavily depends on input parameters. This paper introduces WOA-DBSCAN, a WOA-optimized DBSCAN algorithm, to address this concern. WOA-DBSCAN automatically determines the number of clusters in a dataset by utilizing density peaks, employs the silhouette coefficient as the fitness function, and seeks optimal values for DBSCAN's input parameters, Eps and MinPts. By doing so, WOA-DBSCAN effectively resolves the sensitivity issue of input parameters in the DBSCAN algorithm and achieves parameter adaptability for DBSCAN. Experimental results on artificial and natural UCI datasets demonstrate that WOA-DBSCAN outperforms traditional DBSCAN and its various enhanced algorithms. It also exhibits impressive performance on medium and low-dimensional real datasets.

Furthermore, the example dataset shows that WOA-DBSCAN produces good clustering results even in noisy data. Therefore, the algorithm can be widely used in anomaly detection, such as abnormal behavior recognition, state evaluation, pattern recognition, etc. However, WOA-DBSCAN needs further research. We plan to expand this project in the future to include two key areas. First, we will further evaluate its scalability to ensure the effective operation of the project in different scenarios. Second, we will experiment with more and more complex data sets to adapt to the needs of different scenarios.

REFERENCES

[1] X. Huang, T. Ma, C. Liu, and S. Liu, "GriT-DBSCAN: A spatial clustering algorithm for very large databases," *Pattern Recognit.*, vol. 142, Oct. 2023, Art. no. 109658.

[2] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. KDD*, Jan. 1996, pp. 226–231.

[3] P.-K. Kao, B. J. VanSaders, S. C. Glotzer, and M. J. Solomon, "Accelerated annealing of colloidal crystal monolayers by means of cyclically applied electric fields," *Sci. Rep.*, vol. 11, no. 1, May 2021, Art. no. 11042.

[4] Z. Huang, S. Gao, C. Cai, H. Zheng, Z. Pan, and W. Li, "A rapid density method for taxi passengers hot spot recognition and visualization based on DBSCAN+," *Sci. Rep.*, vol. 11, no. 1, May 2021, Art. no. 9420.

[5] R. M. Sterbentz, K. L. Haley, and J. O. Island, "Universal image segmentation for optical identification of 2D materials," *Sci. Rep.*, vol. 11, no. 1, Mar. 2021, Art. no. 5808.

[6] Z. Xia and S. Chong, "WiFi-based indoor passive fall detection for medical Internet of Things," *Comput. Electr. Eng.*, vol. 109, Aug. 2023, Art. no. 108763, doi: 10.1016/j.compeleceng.2023.108763.

[7] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, May 2016, doi: 10.1016/j.advengsoft.2016.01.008.

[8] M. V. Anaraki, S. Farzin, S.-F. Mousavi, and H. Karami, "Uncertainty analysis of climate change impacts on flood frequency by using hybrid machine learning methods," *Water Resour. Manage.*, vol. 35, no. 1, pp. 199–223, Jan. 2021, doi: 10.1007/s11269-020-02719-w.

[9] J. Nasiri and F. M. Khiyabani, "A whale optimization algorithm (WOA) approach for clustering," *Cogent Math. Statist.*, vol. 5, no. 1, Jan. 2018, Art. no. 1483565.

[10] H. Singh et al., "An enhanced whale optimization algorithm for clustering," *Multimedia Tools Appl.*, vol. 82, no. 3, pp. 4599–4618, Jan. 2023.

[11] S. Jahirabadkar and P. Kulkarni, "Algorithm to determine ε-distance parameter in density based clustering," *Expert Syst. Appl.*, vol. 41, no. 6, pp. 2939–2946, May 2014.
[12] C. Cassisi, A. Ferro, R. Giugno, G. Pigola, and A. Pulvirenti, "Enhancing density-based clustering: Parameter reduction and outlier detection," *Inf. Syst.*, vol. 38, no. 3, pp. 317–330, May 2013, doi: 10.1016/j.is.2012.09.001.
[13] Y. Lv et al., "An efficient and scalable density-based clustering algorithm for datasets with complex structures," *Neurocomputing*, vol. 171, pp. 9–22, Jan. 2016.
[14] A. Bryant and K. Cios, "RNN-DBSCAN: A density-based clustering algorithm using reverse nearest neighbor density estimates," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 6, pp. 1109–1121, Jun. 2018, doi: 10.1109/TKDE.2017.2787640.
[15] W. J. Li, S. Q. Yan, and Y. Jiang, "Research on adaptive determination of DBSCAN algorithm parameters," *Comput. Eng. Appl.*, vol. 55, pp. 1–7, May 2019.
[16] Y. Li, Z. Yang, S. Jiao, and Y. Li, "Partition KMNN-DBSCAN algorithm and its application in extraction of rail damage data," *Math. Problems Eng.*, vol. 2022, pp. 1–10, Jul. 2022, doi: 10.1155/2022/4699573.
[17] H. Li, X. Liu, T. Li, and R. Gan, "A novel density-based clustering algorithm using nearest neighbor graph," *Pattern Recognit.*, vol. 102, Jun. 2020, Art. no. 107206, doi: 10.1016/j.patcog.2020.107206.
[18] L. Yihong, W. Yunpeng, L. Tao, L. Xiaolong, and S. Han, "GNN-DBSCAN: A new density-based algorithm using grid and the nearest neighbor," *J. Intell. Fuzzy Syst.*, vol. 41, no. 6, pp. 7589–7601, Dec. 2021.
[19] S. H. Chen, M. l. Yi, and Y. X. Zhang, "Wafer graph preprocessing based on optimized DBSCAN clustering algorithm," *J. Control Decis.*, vol. 36, no. 11, pp. 2713–2721, Nov. 2021.
[20] J.-H. Kim, J.-H. Choi, K.-H. Yoo, and A. Nasridinov, "AA-DBSCAN: An approximate adaptive DBSCAN for finding clusters with varying densities," *J. Supercomput.*, vol. 75, no. 1, pp. 142–169, Jan. 2019.
[21] Y.-P. Wu, J.-J. Guo, and X.-J. Zhang, "A linear DBSCAN algorithm based on LSH," in *Proc. Int. Conf. Mach. Learn. Cybern.*, Hong Kong, Aug. 2007, pp. 2608–2614, doi: 10.1109/icmlc.2007.4370588.
[22] 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.
[23] S. Wang, Y. Liu, and B. Shen, "MDBSCAN: Multi-level density based spatial clustering of applications with noise," in *Proc. 11th Int. Knowl. Manage. Org. Conf. Changing Face Knowl. Manage. Impacting Soc.*, Hagen, Germany, Jul. 2016, pp. 1–5, doi: 10.1145/2925995.2926040.
[24] W. Guang and L. Guoyu, "Improved adaptive parameter DBSCAN clustering algorithm," *Comput. Eng. Appl.*, vol. 56, no. 14, pp. 45–51, 2020.
[25] S. Z. Lu, "A self-adaptive grey DBSCAN clustering method," *J. Grey Syst.*, vol. 34, Dec. 2022, Art. no. 475984.
[26] H. Jiang, J. Li, S. Yi, X. Wang, and X. Hu, "A new hybrid method based on partitioning-based DBSCAN and ant clustering," *Expert Syst. Appl.*, vol. 38, no. 8, pp. 9373–9381, Aug. 2011.
[27] J. C. Perafán-López and J. Sierra-Pérez, "An unsupervised pattern recognition methodology based on factor analysis and a genetic-DBSCAN algorithm to infer operational conditions from strain measurements in structural applications," *Chin. J. Aeronaut.*, vol. 34, no. 2, pp. 165–181, Feb. 2021.
[28] M. H. Rad and M. Abdolrazzagh-Nezhad, "A new hybridization of DBSCAN and fuzzy earthworm optimization algorithm for data cube clustering," *Soft Comput.*, vol. 24, no. 20, pp. 15529–15549, Apr. 2020.
[29] M. Adibifard, A. Sheidaie, and M. Sharifi, "An intelligent heuristic-clustering algorithm to determine the most probable reservoir model from pressure–time series in underground reservoirs," *Soft Comput.*, vol. 24, no. 20, pp. 15773–15794, Apr. 2020.
[30] P. Y. Cao, C. Z. Yang, and L. M. Shi, "Unknown radar signal processing method based on PSO-DBSCAN and SCGAN," *J. Syst. Eng. Electron.*, vol. 44, no. 34, pp. 1158–1165, Apr. 2022.
[31] Q. T. Zhu, M. Xiang, and A. Elahi, "Application of the novel harmony search optimization algorithm for DBSCAN clustering," *Expert Syst. With Appl.*, vol. 178, Sep. 2021, Art. no. 115054, doi: 10.1016/j.eswa.2021.115054.
[32] W. Zhou, L. Wang, X. Han, Y. Wang, Y. Zhang, and Z. Jia, "Adaptive density spatial clustering method fusing chameleon swarm algorithm," *Entropy*, vol. 25, no. 5, p. 782, May 2023.
[33] Y. Yang, "An efficient DBSCAN optimized by arithmetic optimization algorithm with opposition-based learning," *J. Supercomput.*, vol. 78, no. 18, pp. 19566–19604, Jun. 2022, doi: 10.1007/s11227-022-04634-w.
[34] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, Jun. 2014, doi: 10.1126/science.1242072.
[35] J. X. Liu, X. F. Dong, and C. Xu, "Terminal area track clustering and anomaly identification based on density peaks," *J. Transp. Eng.*, vol. 21, no. 5, pp. 214–226, Oct. 2021.
[36] N. Matsumoto, Y. Hamakawa, K. Tatsumura, and K. Kudo, "Distance-based clustering using QUBO formulations," *Sci. Rep.*, vol. 12, no. 1, Feb. 2022, Art. no. 2669, doi: 10.1038/s41598-022-06559-z.
[37] Z. P. Zhou, Z. F. Wang, S. W. Zhu, and Z. W. Sun, "An improved adaptive fast AF-DBSCAN clustering algorithm," *CAAI Trans. Intell. Syst.*, vol. 11, no. 1, pp. 93–98, Jan. 2016.
[38] University of Eastern Finland. *Clustering Datasets: Shape Sets*. [Online]. Available: https://cs.joensuu.fi/sipu/datasets/
[39] R. Laxhammar, "Artificial intelligence for situation assessment," M.S. thesis, Dept. Comput. Sci. Commun., KTH, Sweden, Europe, 2007.
[40] M. Lichman. (2013). *UCI Machine Learning Repository*. [Online]. Available: http://archive.ics.uci.edu/m
[41] Liran Zvibel. *Weka Release Built-in Datasets*. [Online]. Available: https://waikato.github.io/weka-wiki/datasets/
[42] M. Manaa, A. Obaid, and M. Dosh, "Unsupervised approach for email spam filtering using data mining," *EAI Endorsed Trans. Energy Web*, vol. 8, Jul. 2018, Art. no. 168962, doi: 10.4108/eai.9-3-2021.168962.
[43] S. Huang, J. Luo, K. Pu, and M. Wu, "Diagnosis system of microscopic hyperspectral image of hepatobiliary tumors based on convolutional neural network," *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–13, Mar. 2022, doi: 10.1155/2022/3794844.
[44] X. Ping, F. Yang, H. Zhang, C. Xing, W. Zhang, and Y. Wang, "Evaluation of hybrid forecasting methods for organic Rankine cycle: Unsupervised learning-based outlier removal and partial mutual information-based feature selection," *Appl. Energy*, vol. 311, Apr. 2022, Art. no. 118682, doi: 10.1016/j.apenergy.2022.118682.
[45] D. Huang, C.-D. Wang, H. Peng, J. Lai, and C.-K. Kwoh, "Enhanced ensemble clustering via fast propagation of cluster-wise similarities," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 51, no. 1, pp. 508–520, Jan. 2021, doi: 10.1109/TSMC.2018.2876202.

XINLIANG ZHANG was born in 1997. He received the bachelor's degree in logistics management from the Fujian University of Technology, Fujian, China, in 2020. He is currently pursuing the master's degree in transportation engineering with Jimei University. He is a dedicated master's student with a strong passion for research in the field of transportation engineering. His current research interests encompass a wide range of areas including data mining, deep learning, and artificial intelligence. He is eager to deepen his understanding of these fields and apply his findings to practical applications, particularly in the domains of transportation and data mining.

SHIBO ZHOU received the B.S. degree from the Navigation College, Jimei University, in 2003, the M.S. degree from Shanghai Maritime University, in 2005, and the Ph.D. degree from Beijing Jiaotong university, in 2018. He is currently a Professor with Jimei University. In the fields of maritime safety and computer science, he has actively participated and published over 50 journal articles, including more than ten articles that have been indexed by the Web of Science system. His main research interests include data mining, system analysis, and integration. His extensive research contributions, broad research interests, and eagerness to apply his findings to real-world applications make him a valuable asset to the academic and professional community.