



An efficient DBSCAN optimized by arithmetic optimization algorithm with opposition-based learning

Yang Yang¹ · Chen Qian¹ · Haomiao Li¹ · Yuchao Gao¹ · Jinran Wu² ·
Chan-Juan Liu³ · Shangrui Zhao⁴

Accepted: 26 May 2022 / Published online: 26 June 2022
© The Author(s) 2022

Abstract

As unsupervised learning algorithm, clustering algorithm is widely used in data processing field. Density-based spatial clustering of applications with noise algorithm (DBSCAN), as a common unsupervised learning algorithm, can achieve clusters via finding high-density areas separated by low-density areas based on cluster density. Different from other clustering methods, DBSCAN can work well for any shape clusters in the spatial database and can effectively cluster exceptional data. However, in the employment of DBSCAN, the parameters, EPS and MinPts, need to be preset for different clustering object, which greatly influences the performance of the DBSCAN. To achieve automatic optimization of parameters and improve the performance of DBSCAN, we proposed an improved DBSCAN optimized by arithmetic optimization algorithm (AOA) with opposition-based learning (OBL) named OBLAOA-DBSCAN. In details, the reverse search capability of OBL is added to AOA for obtaining proper parameters for DBSCAN, to achieve adaptive parameter optimization. In addition, our proposed OBLAOA optimizer is compared with standard AOA and several latest meta heuristic algorithms based on 8 benchmark functions from CEC2021, which validates the exploration improvement of OBL. To validate the clustering performance of the OBLAOA-DBSCAN, 5 classical clustering methods with 10 real datasets are chosen as the compare models according to the computational cost and accuracy. Based on the experimental results, we can obtain two conclusions: (1) the proposed OBLAOA-DBSCAN can provide highly accurately clusters more efficiently; and (2) the OBLAOA can significantly improve the exploration ability, which can provide better optimal parameters.

Keywords Clustering · Hyperparameter optimization · Swarm intelligence · Exploration

Chen Qian, Haomiao Li have contributed equally to this work.

Extended author information available on the last page of the article

1 Introduction

Clustering, a common unsupervised learning algorithm [1–4], groups the samples in the unlabeled dataset according to the nature of features, so that the similarity of data objects in the same cluster is the highest while that of different clusters is the lowest [5–7]. Clustering is popularly used in biology [8], medicine [9], psychology [10], statistics [11], mathematics [12] and computer science [13]. Since the early 1950s, many clustering algorithms have been proposed. In this paper, considering the novelty and effectiveness of density-based method, we will focus on density-based noise application spatial clustering algorithm (DBSCAN) and explore an adaptive method to tune the hyperparameter for DBSCAN instead of empirical setting.

1.1 Literature review

In clustering algorithms, K-means [14], as the most basic partition clustering algorithm at present, has the advantages of simple principle, strong practicability, fast convergence speed and strong model interpretation and so on. However, it is difficult to converge non-convex datasets and often stops at the local optimal solution.

Different from K-means, DBSCAN [15, 16] is another popular clustering algorithm based on density. It achieves clusters via finding high-density areas separated by low-density areas based on cluster density. Compared with other clustering algorithms based on the distance between objects, DBSCAN is suitable for finding clusters of any shape in spatial database and connecting adjacent regions with corresponding density. It can effectively deal with abnormal data, especially the clustering of spatial data [17]. Although DBSCAN has many advantages in clustering, it still has some disadvantages. For different datasets, DBSCAN needs to set the most appropriate parameters, MinPts and EPS, to achieve the best clustering effect. To some extent, the process of setting parameters limits the application of DBSCAN [18].

Over the years, to apply DBSCAN effectively, many researchers have improved DBSCAN [19] through meta-heuristic algorithm [20–23] to realize the automatic search and determination of EPS and MinPts parameters in DBSCAN. For example, Lai et al. [24] proposed a multi-segment optimization algorithm. As a special variable updating method, it has good optimization performance, can obtain good DBSCAN accuracy, and can quickly obtain appropriate EPS parameter selection. Ji'an et al. [25] proposed an adaptive DBSCAN to solve the clustering problem, taking the target solution and its motion range as noise points, in which DBSCAN ϵ The neighborhood is affected by some specific physical factors. Zhu et al. [26] applied the harmony search optimization algorithm to DBSCAN, and obtained better clustering parameters and better clustering results. Hu et al. [27] proposed a density-based clustering algorithm, KR-DBSCAN, which is based on reverse nearest neighbor and influence space. Li et al. [28] combined the improved

DBSCAN algorithm based on bat optimization and DP algorithm for clustering, and obtained good results. However, these methods still have the characteristics of low convergence accuracy, poor universality, and slow convergence speed.

Meta heuristic algorithm is a popular algorithm in recent years, such as Gray Wolf Whale (GWO), Dragonfly algorithm (DA) and Ant Lion Optimizer (ALO). It has the characteristics of high convergence accuracy and strong robustness. It can be used to solve the selection of parameters in DBSCAN. However, the common meta heuristic algorithms are easy to fall into local optimization. Therefore, we choose Arithmetic optimization algorithm (AOA) as the optimization algorithm. AOA is a new population-based metaheuristic algorithm proposed by Abualah [29], which uses four basic arithmetic operators in mathematics. AOA can not only deal with low dimensional problems [30], but also has a strong ability to solve high-dimensional problems [31]. The distribution mechanism enhances its global search ability, and the algorithm based on population [32] without optimization also helps to achieve faster convergence speed.

However, the ability of standard AOA to balance global optimization and local optimization is still insufficient, and the optimization accuracy is also insufficient. To better balance global optimization and local optimization and improve the optimization accuracy, we proposed some search strategies of improving development (local search) and exploration (global search). In addition, Opposition-based learning (OBL) [33–35] is one of the most popular strategies to enhance exploration, which can improve the population diversity of the algorithm in the search space. In the optimization problem, the strategy of checking the candidate solution and its opposite solution at the same time is adopted to speed up the convergence speed to the global optimal solution.

In general, the current clustering effect of DBSCAN is limited by the optimization results of its parameters. At present, the optimization algorithm used to solve DBSCAN parameter optimization has low convergence accuracy and is easy to fall into local optimal solution. Although the standard AOA improves the global dispersion compared with other optimization algorithms, it still has some shortcomings, such as insufficient convergence accuracy and global search ability.

1.2 The gap

To sum up, the demand for the accuracy of DBSCAN clustering algorithm is still increasing. To improve the accuracy of DBSCAN clustering algorithm, more advanced machine learning methods are needed to automatically optimize the parameters of DBSCAN clustering algorithm to improve the accuracy of clustering.

1.3 The contribution

To improve the accuracy and convergence speed of the automatic selection of DBSCAN parameters, this paper proposes a new meta-heuristic improvement strategy, OBLAOA-DBSCAN, which combines the advantages of AOA and OBL with DBSCAN to adjust dynamically the two parameters of DBSCAN. In addition,

according to the experimental results, DBSCAN improved with OBLAOA performs well in a variety of public datasets. Therefore, the contributions of this article are as follows:

- (1) An OBLAOA-DBSCAN clustering algorithm is proposed, which can realize automatic parameter search and improve the clustering accuracy and efficiency.
- (2) By adding the OBL strategy, an OBLAOA optimizer is established, which can effectively improve the exploration performance of AOA.
- (3) The proposed OBLAOA-DBSCAN algorithm can provide better clustering results than other clustering algorithms including K-means, Spectral, Optics, DPC and the combination method of DBSCAN and other meta-heuristic optimization algorithms.

1.4 The structure of the paper

The remaining contents are organized as following. Section 2 outlines some backgrounds of DBSCAN and AOA. Section 3 introduces the OBLAOA and gives the use principle and concrete operation. Section 4 illustrates the proposed OBLAOA-DBSCAN algorithm. Section 5 compares the proposed OBLAOA with the original AOA by using 12 benchmark functions. Section 6 demonstrates the superiority of the proposed algorithm with 10 datasets by comparing with some considered clustering algorithms. Section 7 concludes the paper.

2 Related work

2.1 The basic theory of DBSCAN

DBSCAN, an unsupervised learning method, is proposed by [36] handling the clustering problem efficiently based on density. DBSCAN has the capacity to identify noise points efficiently and exactly. Furthermore, it can also distinguish clusters with arbitrary shapes.

In this clustering method, two parameters, the epsilon (EPS) and MinPts, are required to be pre-set to appraise the density distribution of points. DBSCAN starts from an unvisited point randomly. Then, it counts the points fallen within the adjacent area radius of the point less than EPS.

If the number of points is more than MinPts, the current point and its nearby points from a cluster, and the starting point is marked as visited. Then, all the points in the cluster are not marked as visited that are processed in the same way recursively, to expand the cluster. Otherwise, the point is temporarily marked as a noise point. If the cluster is fully expanded, that is, all points in the cluster are marked as visited, and then the same algorithm is used to process the non-visited points. Until all objects are marked as a certain cluster or noise, the clustering process ends. The DBSCAN algorithm flow is presented in Algorithm 1.

Algorithm 1 DBSCAN

Require: dataset D , adjacent area radius EPS , minimum number of points in the neighborhood $MinPts$

Ensure: Clusters C

- 1: Set all points as *unmarked*
- 2: Do
- 3: Choose one *unmarked* points X randomly
- 4: Mark X as marked
- 5: **if** the number of points in adjacent area radius EPS of $X \geq MinPts$ **then**
- 6: Produce a new cluster C and put P into C
- 7: Let P be the set of points within EPS of X neighborhood radius
- 8: **for** P for each point x **do**
- 9: **if** x is unmarked **then**
- 10: Set X as marked
- 11: **if** the number of points in X adjacent area radius $EPS > MinPts$ **then**
- 12: put them all to X
- 13: **if** X does not belong to any cluster **then**
- 14: Put X to C
- 15: **end if**
- 16: **end if**
- 17: **end for**
- 18: **end for**
- 19: Output C
- 20: **else**
- 21: Marks X as noise points
- 22: **end if**
- 23: **repeat**
- 24: Above procedure
- 25: **until** there are no unmarked points

DBSCAN suffers from the determination of these two parameters. Previous studies have presented that these two parameters can be found by statistical and classical methods of combining different data mining ways, but these methods consume excessive time. Therefore, we introduce a meta-heuristic optimization to improve the accuracy and efficiency of finding these parameters considerably to achieve clustering faster and more precisely.

2.2 The arithmetic optimization algorithm

Arithmetic Optimization Algorithm (AOA) is a new meta-heuristic optimization algorithm [29] inspired by four major arithmetic operators (Multiplication (M), Division (D), Subtraction(S), and Addition (A)). The mathematical models of exploration and exploitation phase are detailed as follows. Note that the exploration stage and exploitation stage is conditioned by the math optimizer accelerated (MOA) function. It is calculated by

$$MOA(C_{Iter}) = \delta + C_{Iter} * \left(\frac{\gamma - \delta}{M_{Iter}} \right), \quad (1)$$

where M_{Iter} is the maximum number of iterations, and C_{Iter} represents the current iteration, which is between 1 and M_{Iter} . $MOA(C_{Iter})$ represents the value of MOA at the current iteration. γ and δ are set to 1 and 0.2 respectively. The math optimizer probability (MOP) at the current iteration is calculated by

$$MOP(C_{Iter}) = 1 - \frac{C_{Iter}^{\frac{1}{\alpha}}}{M_{Iter}^{\frac{1}{\alpha}}}, \tag{2}$$

where α is a sensitive parameter and represents the exploitation accuracy over the iterations, which is set to 0.5.

r_1, r_2, r_3 are random numbers. When $MOA < r_1$, we carry out exploration section by executing D or M . The position updating equation in the exploration stage is followed:

$$x_{i,j}(C_{Iter} + 1) = \begin{cases} x^*(C_{Iter}) \div (MOP + \epsilon) \times ((ub_j - lb_j) \times \mu + lb_j), & r_2 < 0.5 \\ x^*(C_{Iter}) \times MOP \times ((ub_j - lb_j) \times \mu + lb_j), & \text{otherwise,} \end{cases} \tag{3}$$

where $x_{i,j}(C_{Iter} + 1)$ denotes the j th dimension of the i th solution in the next iteration, and $x^*(C_{Iter})$ is the best-obtained solution in the previous iteration. ϵ is a small integer, ub_j and lb_j refer to the upper and lower bound value of j th position. μ is a control parameter, which is set to 0.5.

When $MOA \geq r_1$, we carry out exploitation section by executing S or A . In the case of $r_3 < 0.5$, S performs (first rule in Eq. 4). Otherwise, A performs the task in the position of S (second rule in Eq. 4). The position updating equation in the exploitation stage is followed:

$$x_{i,j}(C_{Iter} + 1) = \begin{cases} x^*(C_{Iter}) - MOP \times ((ub_j - lb_j) \times \mu + lb_j), & r_3 < 0.5 \\ x^*(C_{Iter}) + MOP \times ((ub_j - lb_j) \times \mu + lb_j), & \text{otherwise.} \end{cases} \tag{4}$$

2.3 The opposition-based learning

Opposition-based learning (OBL) is employed to consider candidate schemes and their inverses. Depending on which estimate, or inverse estimate is closer to the solution, the search interval can be recursively halved until the estimate or inverse estimate is close enough to the existing solution. It determines whether the original solution x is replaced by the opposite solution \bar{x} by comparing the fitness function values of them. Considering the solution $x \in [lb, ub]$, \bar{x} is calculated by the following equation:

$$\bar{x} = ub + lb - x. \tag{5}$$

This equation above can be popularized to n -dimension via:

$$\bar{x}_j = ub_j + lb_j - x_j, j = 1, 2, \dots, n. \tag{6}$$

According to the results of comparison, it ends up with storing the best of two solutions.

3 The proposed OBLAOA

OBL is committed to taking both candidate solutions and their opposite solutions into consideration, which shows greater opportunity to reach the global optimal and faster convergence acceleration than only executing S or A. It is adopted to find a solution, which is opposite to the present solution, and subsequently it determines if the opposite solution is used by comparing the fitness function values of them. For example, if $f(x^*(C_{Iter})) \leq f(\bar{x}^*(C_{Iter}))$, then $x^*(C_{Iter})$ is saved; otherwise, $\bar{x}^*(C_{Iter})$ is stored. The equation used in OBLAOA to get the opposite solution is as,

$$\bar{x}^*(C_{Iter}) = ub + lb - x^*(C_{Iter}) \quad (7)$$

where $x^*(C_{Iter})$ denotes the position of the best solution in the current iteration. $\bar{x}^*(C_{Iter})$ denotes the opposite position of the best solution in the current iteration.

The flowchart of the proposed OBLAOA is given in Fig. 1 and the pseudocode is recorded in Algorithm 2.

Algorithm 2 OBLAOA

```

1: Initialize the parameters  $\alpha$ ,  $\mu$ ,  $l$  and candidate solutions  $x$ 
2: while ( $C_{Iter} < M_{Iter}$ ) do
3:   Update the MOP value of current iteration using Eq. 2
4:   Update the MOA value of current iteration using Eq. 1
5:   for  $i = 1$  to  $N$  do
6:     for  $j = 1$  to  $n$  do
7:       Generate random number  $r_1, r_2, r_3 \in [0, 1]$ 
8:       if  $MOA < r_1$  then
9:         Exploration phase
10:        if  $r_2 < 0.5$  then
11:          Employ the division operator ( $D'' \div ''$ )
12:          Update the ith solution using the first rule in Eq. 3
13:        else
14:          Employ the multiplication operator ( $M'' \times ''$ )
15:          Update the ith solution using the second rule in Eq. 3
16:        end if
17:      else
18:        Exploitation phase
19:        if  $r_3 < 0.5$  then
20:          Employ the subtraction operator ( $S'' - ''$ )
21:          Update the ith solution using the first rule in Eq. 4
22:        else
23:          Employ the addition operator ( $A'' + ''$ )
24:          Update the ith solution using the second rule in Eq. 4
25:        end if
26:      end if
27:    end for
28:  end for
29:  Calculate the fitness function
30:  Find the opposite solution of the best solution using Eq. 7
31:  Calculate the fitness function of both the current and its opposite solution
32:  Set the better one as the  $x^*(C_{Iter} + 1)$ 
33: end while
34: return the best solution ( $x_{best}$ ).

```

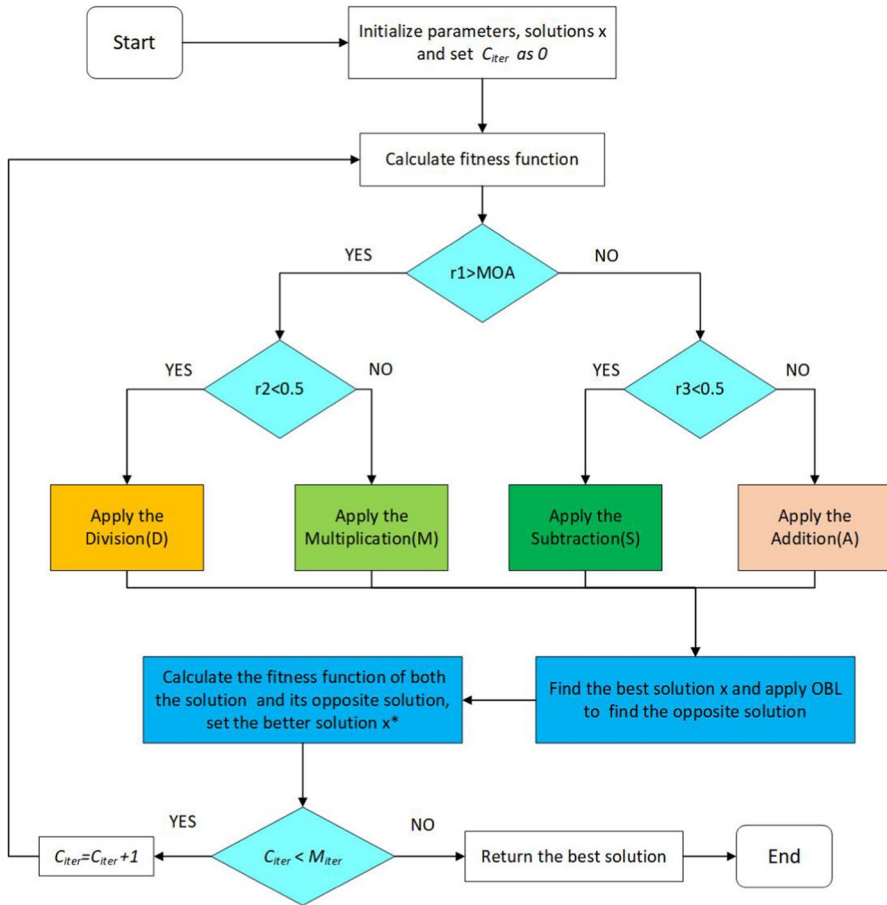


Fig. 1 The flowchart of OBLAOA

4 The improved DBSCAN with OBLAOA

In this section, we apply OBLAOA to DBSCAN to optimize two parameters of DBSCAN (EPS and MinPts). Here more advanced modification method, namely OBLAOA-DBSCAN, is proposed, which can further improve the performance of the clustering algorithm.

In details, the OBLAOA-DBSCAN can perform the optimization process of determining the parameters EPS and MinPts automatically in an extensive scope of search spaces via a meta-heuristic method. First, set the normalized range matrix of two parameters (EPS and MinPts) as the upper bounds (ub_j) and lower bounds (lb_j) of search space. Then, the OBLAOA is used to search for suitable parameters within the effective search space.

To get the best clustering results, the sum of the average Euclidean distance of each cluster, the fitness function in OBLAOA-DBSCAN is given as,

$$D(o_i, o_l) = \left(\sum_{j=1}^m (o_{ij} - o_{lj})^r \right)^{\frac{1}{r}} \quad (8)$$

where $D(o_i, o_l)$ is an Euclidean distance function that produces different metrics between object i and object l , $o_{ij}, o_{lj} (i, l = 1, \dots, n, j = 1, \dots, m)$ represents the value of the j -th attribute of object i and object l , respectively.

With the value of fitness function updates continuously, the position of best-obtained solution, which determines the value of two parameters, varies. At this time, the corresponding parameters MinPts and EPS will change. Until the fitness value no longer changes, apply the obtained parameters into DBSCAN algorithm for clustering.

When only using DBSCAN for clustering, problems, such as low accuracy of clustering results and low definition of noise points, always appear because of parameters setting manually. By introducing OBL to enhance the exploration ability of AOA, OBLAOA can provide effective parameter solutions for DBSCAN, thereby improving the clustering ability. The flowchart is shown in Fig. 2. After calculation, the time complexity of OBLAOA-DBSCAN is $O(N(1 + M \times n \log(n) + M \times n))$. Where N represents the number of candidate solutions, M is the number of iterations, and n is the dimension of solving the problem.

5 Numerical simulation

5.1 The benchmark functions

To evaluate the performance of the proposed OBLAOA optimizer, we conducted numerical simulation experiments with 8 test functions in CEC2021. The benchmark functions are presented in Table 1, and its constraint range is represented by Range in the table.

5.2 The setting of experimental parameters

The results of OBLAOA are saved and compared with five traditional methods (i.e., AOA, IAOA, DAOA, EN-GWO and WSSA) for each test case. The parameters of each algorithm are set as follows. The maximum number of iterations and population size of all algorithms are set as 500 and 20, respectively, and the number of function evaluations is 30 [37]. In addition, the initial random population set of all algorithms are the same. All CEC2021 test functions are simulated in 10 and 20 dimensions, respectively.

5.3 Analysis of the results

The results of numerical simulation are recorded in Tables 2 and 3. To verify the effectiveness of OBLAOA, we compared the results of OBLAOA with the standards AOA, IAOA, DAOA, ENGWO and WSSA. We select the corresponding average value (AVG), standard deviation (STD) and best value (BEST) as performance indicators and report them in all tables. We show better results in bold in Tables 2 and 3. In addition, Wilson’s rank test was used for all results, and all results of Wilson’s rank test (h) were 1. It can be seen from the table that OBLAOA has better performance than standard AOA and other current popular optimization algorithms (i.e., IAOA, DAOA, ENGWO and WSSA). In the test of high-dimensional meta heuristic algorithm, for all functions, the average value and optimal value of OBLAOA are better than standard AOA and current popular algorithms. In the test of low-dimensional meta-heuristic algorithm, the average and optimal values of OBLAOA are better than AOA for $F_1, F_2, F_3, F_5, F_6, F_7$ and F_8 functions. In some experiments, compared with AOA, the performance of OBLAOA is significantly improved. Taking the F_3 function of 10 dim as an example, the best index of OBLAOA is 106.76,

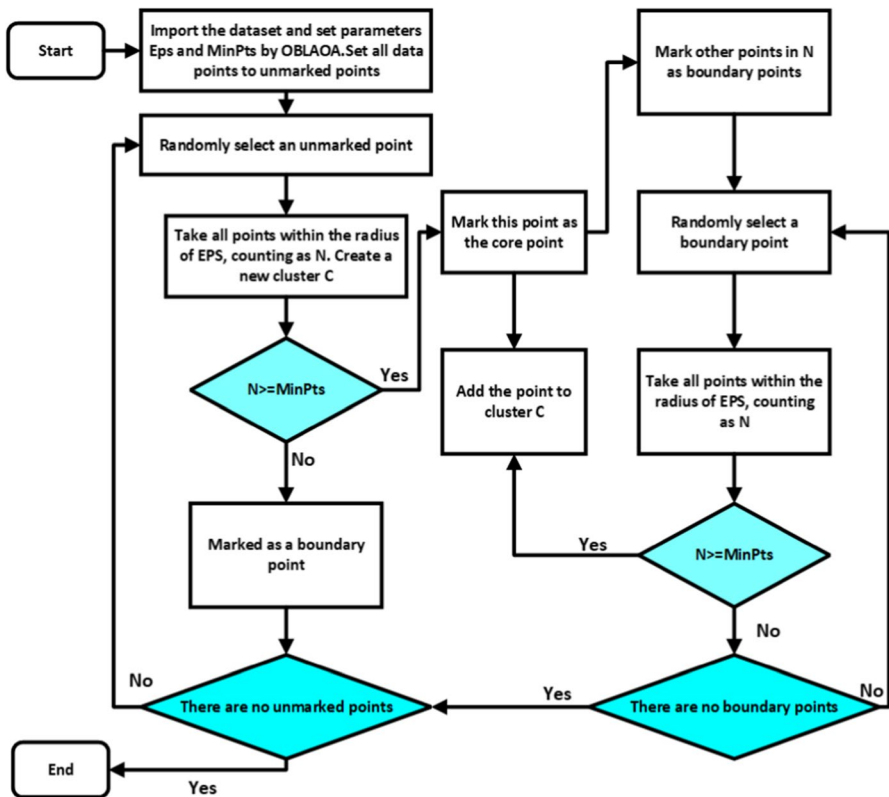


Fig. 2 The flowchart of the proposed OBLAOA-DBSCAN

which is 46.62% lower than standard AOA, 77.17% lower than DAOA, 2.67% lower than ENGWO and 80.91% lower than WSSA. As far as F_6 is concerned, the index of best is 1600, which is 44% lower than standard AOA, 21.95% lower than DAOA, 24.88% lower than ENGWO and 31.91% lower than WSSA. From the F_8 function, the index of best is $2.99e + 3$, which is 59.15% lower than standard AOA, 59.14% lower than DAOA, 54.83% lower than IAOA, 63.49% lower than ENGWO and 65.67% lower than WSSA. To sum up, our proposed OBL is better than standard AOA and other current popular algorithms in dealing with complex functions.

In order to further prove the optimization effect of OBLAOA, we selected three practical engineering problems for verification, including welded beam design [38], compression spring design [39] and design problems of I-beam [40]. The results are recorded in Table 4 and shown in Fig. 3. To verify the adequacy of the experimental results, we also carried out Wilcoxon signed rank test. The results are expressed in h , which are all 1, and recorded in Table 4. From Fig. 3, we can see that our OBLAOA has better optimization effect compared with other algorithms. Our OBLAOA has the highest convergence accuracy among all problems. Specifically, in the CSD problem, our OBLAOA converges first, and the convergence effect is greatly improved compared with ENGWO and WSSA. In general, our OBLAOA has better convergence effect in solving practical engineering problems. As can be seen from Table 4, our OBLAOA algorithm also has great advantages over standard AOA and the latest algorithm in solving practical engineering problems. Our OBLAOA has obtained the best value in all three

Table 1 The CEC2021 benchmark functions

Function	Description	Range
F_1	$f(x) = x_1^2 + 10^6 * \sum_{i=2}^D x_i^2$	[- 100,100]
F_2	$f(x) = \sum_{i=1}^D (x_i^2 - 10 * \cos(2\pi x_i))$	[- 100,100]
F_3	$f(x) = \sum_{i=1}^D (10^6)^{\frac{i-1}{D-1}} * x_i^2$	[- 100,100]
F_4	$f(x) = \left \left(\sum_{i=1}^D x_i^2 \right)^2 - \left(\sum_{i=1}^D x_i \right)^2 \right ^{1/2} + \left(0.5 * \sum_{i=1}^D x_i^2 + \sum_{i=1}^D x_i \right) / D + 0.5$	[- 100,100]
F_5	$f(x) = \sum_{i=1}^{D-1} (100 * (x_i^2 - x_{i+1})^2 + (x_i - 1)^2)$	[- 100,100]
F_6	$f(x) = \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	[- 100,100]
F_7	$f(x) = -20 * \exp(-0.2 * \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i)\right) + 20 + e$	[- 100,100]
F_8	$f(x) = \left \sum_{i=1}^D x_i^2 - D \right ^{1/4} + \left(0.5 \sum_{i=1}^D x_i^2 + \sum_{i=1}^D x_i \right) / D + 0.5$	[- 100,100]

engineering problems. Taking the WBD problem as an example, our best value is 4.25, which is 34% lower than the standard AOA algorithm, 40.3% lower than DAOA and 56.72% lower than IAOA. ENGWO and WSSA do not converge, which is quite different from OBLAOA. We can also see from Figs. 4 and 5 that OBLAOA converges earlier and faster, and the final fitness value is lower than that of other algorithms.

6 Experiment and performance evaluation

This section is summarized as follows. In Sect. 6.1, we describe the datasets in the experiment. In Sect. 6.2, we introduce the evaluation indexes that used. In Sect. 6.3, we describe the parameter setting process in detail. In Sect. 6.4, we use ten datasets to test different optimization algorithms. In Sect. 6.5, we compare the optimized OBLAOA-DBSCAN with five classical clustering algorithms.

6.1 The datasets

In this part, we use ten datasets to test the performance of our optimization algorithm OBLAOA-DBSCAN. The instance of 10 datasets is 788, 399, 373, 150, 251, 300, 198, 1980, 341 and 846. The dimensions of 10 datasets are 3, 3, 3, 5, 3, 2, 34, 3, 3 and 19. The clusters of 10 datasets are 7, 6, 2, 3, 3, 5, 2, 5, 9 and 4. Table 5 shows ten datasets as experimental data. We compared the real labels with the clustering label and use the comparison result as the evaluation index of the algorithm, therefore, we use the datasets with real labels.

6.2 The error index

In order to measure the clustering results of the improved method, we use Accuracy, Davies- Bouldin index (DBI), Silhouette index (Sil), Rand index (RI) [41, 42], Normalized Mutual Information (NMI), Homogeneity, Completeness, and V-measure [43]. Because of the datasets with the real label, we use the accuracy index to show the performance of the proposed method.

Accuracy is the ratio of correctly clustered data to total data. The correctly clustered data is obtained by comparing the cluster labels K with the actual labels C . DBI is used to measure the distance within the cluster and the distance between the clusters. The smaller DBI means the smaller distance within the cluster and the larger distance among clusters that is formulated as:

$$DBI = \frac{1}{N} \sum_{i=1}^N \left(\max_{j=1, \dots, N, j \neq i} \left(\frac{d_{ij}}{S_i + S_j} \right) \right),$$

Table 2 Results of 10-dimensional CECE2021 test functions (F_1 - F_8)

	F_1				F_2			
	Avg	Best	Std	h	Avg	Best	Std	h
AOA	2.94e+9	1.08e-3	3.10e+18	1	1.94e+3	1.77e+3	2.11e+4	1
DAOA	1.04e+10	1.04e-10	3.97e+16	1	2.02e+3	2.17e+3	2.57e+3	1
IAOA	2.95e+7	2.20e-198	3.43e+17	1	203.06	0	1.3e+5	1
ENGWO	9.90e+9	9.85e-9	2.38e+16	1	2.31e+3	2.28e+3	1.5e+3	1
WSSA	1.30e+10	1.30e-10	5.26e-9	1	2.59e+3	2.58e+3	869.1	1
OBLAOA	2.62e+7	0	3.38e-17	0	179.94	0	2.11e+4	0
	F_3				F_4			
	Avg	Best	Std	h	Avg	Best	Std	h
AOA	218.02	200.89	2.59e+3	1	4.16e+4	9.31e+5	1.52e+11	1
DAOA	559.48	468.61	28.37	1	7.74e+4	5.01e+5	2.23e+11	1
IAOA	114.78	109.62	490.39	1	3.49e+5	3.29e+5	8.46e+10	1
ENGWO	453.99	450.90	35.45	1	1.64e+5	1.35e+5	8.83e+10	1
WSSA	559.48	559.48	0	1	5.95e+6	5.92e+6	1.50e+10	1
OBLAOA	106.76	106.76	1.10e+13	0	2.37e+5	1.99e+5	9.47e+10	0
	F_5				F_6			
	Avg	Best	h	Std	Avg	Best	Std	h
AOA	2.71e+7	2.17e+3	3.40e+14	1	2.14e+3	2.86e+3	2.29e+3	1
DAOA	6.04e+7	5.01e+3	3.77e+14	1	2.18e+3	2.05e+3	4.16e+3	1
IAOA	1.53e+5	1700	1.10e+14	1	1.62e+3	1600	3.52e+3	1
ENGWO	2.11e+7	2.09e+7	5.89e+12	1	2.14e+3	2.13e+3	330.07	1
WSSA	2.85e+7	2.75e+7	2.85e+13	1	2.35e+3	2.35e+3	0.02	1
OBLAOA	106.76	1700	1.10e+13	0	1.62e+3	1600	2.35e+3	0
	F_7				F_8			
	Avg	Best	Std	h	Avg	Best	Std	h
AOA	1.24e+07	1.42e+6	3.72e+13	1	0.29	3.19e+3	3.03e+3	1
DAOA	1.67e+7	6.06e+6	4.70e+12	1	3.98e+3	3.68e+3	2.40e+4	1
IAOA	06.89e+5	5.8e+5	6.84e11	1	3.33e+3	3.33e+3	1.10e+4	1
ENGWO	7.42e+6	7.05e+06	7.08e+11	1	4.18e+3	4.13e+3	716.21	1
WSSA	1.74e+6	1.74e+7	3.08e-36	1	4.4e+3	4.4e+3	0.32	1
OBLAOA	3.23e+5	1.40e+5	8.19e+11	0	3.04	2.99e+3	9.45	0

where N is the number of clusters, d_{ij} is the average of the distance between clusters i and j . In addition, S_i and S_j are the mean distances of cluster i and cluster j .

The Silhouette value describes the similarity between different clusters. The larger this value is, the higher similarity between the target and its cluster, and the lower similarity with other clusters. The formula is as follows:

Table 3 Results of 20-dimensional CECE2021 test functions (F_1 - F_8)

	F_1				F_2			
	Avg	Best	Std	h	Avg	Best	Std	h
AOA	3.36e+10	3.2e-10	4.7e+17	1	5.7e+3	3.3e+3	4.9e+4	1
DAOA	3.7e+10	3.6e-10	7.6e+15	1	5.7e+3	5.4e+3	3.5e+4	1
IAOA	9.9e+7	2.2e-160	3e+18	1	109	0	3.2e+5	1
ENGWO	3.2e+10	3.2e-10	1.04e+17	1	5.8e+3	5.7e+3	1.26e+3	1
WSSA	3.7e+10	3.7e-10	1.3e-8	1	6.2e+3	6.2e+3	5.6e+22	1
OBLAOA	7.7e+7	2.6e-177	2.8e+18	0	108	0	1.25e+5	0
	F_3				F_4			
	Avg	Best	Std	h	Avg	Best	Std	h
AOA	1.2e+3	12e+3	3.6e+3	1	1.24e+6	1.06e+6	9.2e+11	1
DAOA	1.7e+3	1.62e+3	11	1	5.6e+6	3.6e+6	1.03e+13	1
IAOA	341	333	4e+9	1	4.5e+5	4.17e+5	4.2e+11	1
ENGWO	1.56e+3	01.56e+3	34	1	4.05e+6	4.01e+6	2.23e+11	1
WSSA	1.69e+3	1.69e+3	0	1	1.45e+7	1.45e+7	1.6e+14	1
OBLAOA	305	300	4e+3	0	4.5e+5	3.5e+5	4.5e+11	0
	F_5				F_6			
	Avg	Best	Std	h	Avg	Best	Std	h
AOA	8.1e+7	2.8e+4	4.5e+15	1	3.55e+3	3.35e+3	2.98e+3	1
DAOA	2.2e+8	3.17e+6	1.13e+13	1	3.83e+3	3.56e+3	1.14e+3	1
IAOA	4.9e+5	1700	1.13e+14	1	1.66e+3	1600	2.4e+4	1
ENGWO	1.26e+8	1.25e+8	2.78e+13	1	3.71e+3	3.7e+3	405	1
WSSA	1.84e+8	1.814e+8	1.08e+14	1	1.84e+8	1.81e+8	1.08e+14	1
OBLAOA	4.7e+5	1700	1.13e+14	0	1.64e+3	1600	1.45e+4	0
	F_7				F_8			
	Avg	Best	Std	h	Avg	Best	Std	h
AOA	6.9e+7	1.06e+7	3.49e+15	1	7.6e+3	7.32e+3	4.78e+4	1
DAOA	1.8e+8	6.12e+7	2.53e+15	1	7.65e+3	7.32e+3	4.99e+4	1
IAOA	1.53e+7	1.45e+7	8.6e+13	1	6.74e+3	6.62e+3	7.3774e+4	1
ENGWO	5.55e+7	5.46e+7	5.37e+13	1	8.19e+3	8.19e+3	1.06e+4	1
WSSA	2.12e+8	2.12e+8	1.11e+11	1	8.72e+3	8.71e+3	1.24e+03	1
OBLAOA	2.86e+6	7.6e+5	1.89e+14	0	6.6e+3	2.99e+3	5.05e+4	0

$$SIL = \frac{1}{N} \sum_{i=1}^N \left(\frac{b(i) - a(i)}{\max \{a(i), b(i)\}} \right),$$

Table 4 Results of three engineering problems

Engineering problem A: Welded Beam Design, WBD				
	Avg	Best	Std	h
AOA	101.40	2.47	2.46e+6	1
DAOA	158.78	2.3768	34.54e+6	1
IAOA	1217	2.85	1.32e+7	1
ENGWO	1922	4.38	32.85e+7	1
WSSA	335	235.41	352.12e+6	1
OBLAOA	158.78	2.37	4.54e+6	0
Engineering problem B: compression spring design, CSD				
	Avg	Best	Std	h
AOA	12.33	6.44	16.22	1
DAOA	11.75	7.12	137.92	1
IAOA	9.15	9.82	129.11	1
ENGWO	28.98	16.86	120.95	1
WSSA	33.31	26.25	380.73	1
OBLAOA	12.09	4.25	129.1167	0
Engineering problem C: design problems of I-beam, IBP				
	Avg	Best	Std	h
AOA	189.42	187.63	3.68	1
DAOA	190.63	187.28	10.41	1
IAOA	192.97	187	24.34	1
ENGWO	188.05	187.73	1.48	1
WSSA	188.73	186.73	5.46	1
OBLAOA	189.47	186.42	5.43	0

where $a(i)$ is the average distance between a cluster C_i and all other data points in the same cluster, and $b(i)$ is the average difference between a cluster C_i and other clusters.

The Rand index is a way to compare the similarity of results between two different clustering methods. The larger the value is, the clustering result that compared with the real situation is more consistent. The formula is as follows:

$$RI = \frac{x + y}{C_n^2},$$

where x represents the number of the same labels in both C and K , and y represents the number of different labels in both C and K . C_n^2 represents the number of combinations of C and K that can be made in the dataset.

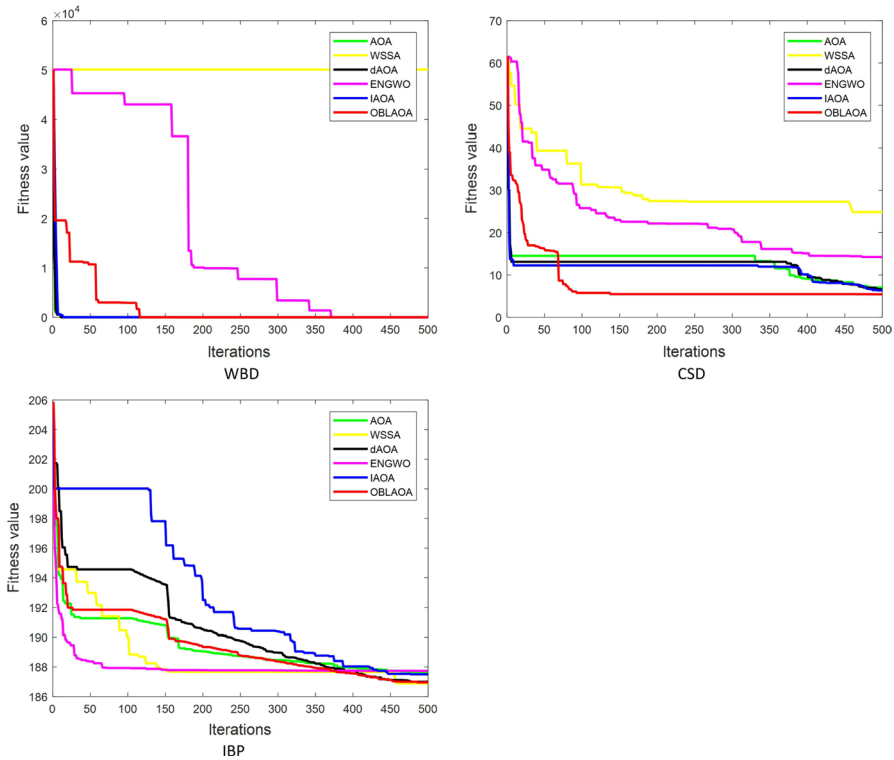


Fig. 3 Comparison of different algorithms for three engineering application problems

NMI is used to measure the coincidence degree of two datasets and refers to the correlation between two sets of results. The greater the NMI, the greater the degree of correlation between categories. The formula is as follows:

$$\begin{aligned}
 Hl &= - \sum_{i=1}^N \left(\frac{Ml}{N} * \log_2 \frac{Ml}{N} \right), Hr = - \sum_{i=1}^N \left(\frac{Mr}{N} * \log_2 \frac{Mr}{N} \right), Hlr \\
 &= - \sum_{i=1}^N \left(\frac{Ml * Mr}{N} * \log_2 \frac{Ml * Mr}{N} \right),
 \end{aligned}$$

and

$$NMI = \sqrt{\left(Hl + Hr - \frac{Hlr}{Hl} \right) * \left(Hl + Hr - \frac{Hlr}{Hr} \right)},$$

where *Ml* represents the cluster distribution of the randomly selected object from the clustering result *K*, *Mr* represents the cluster distribution of the randomly selected object from the actual labels *C*.

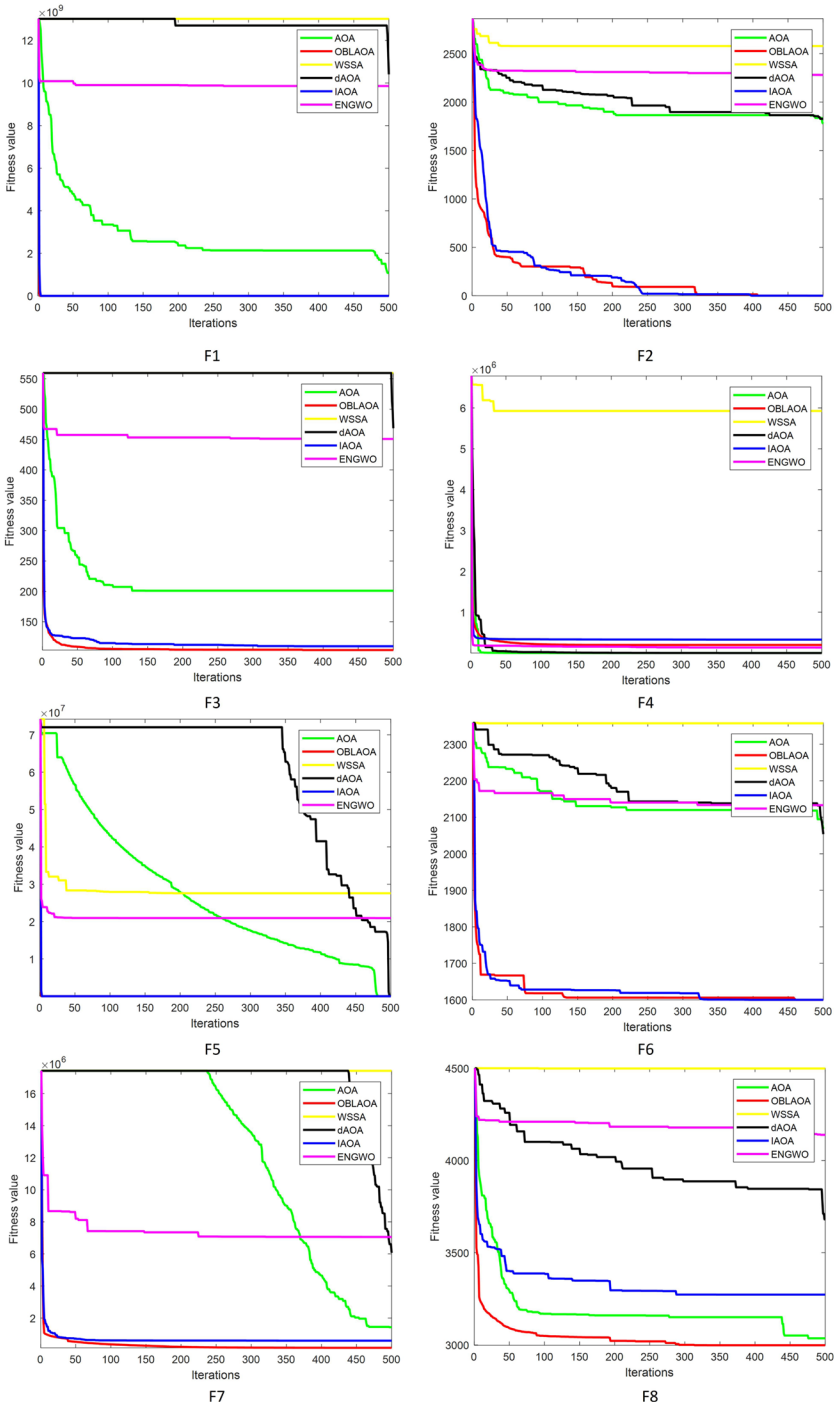


Fig. 4 10-dimensional F₁-F₈ convergence curve of CEC2021 test function

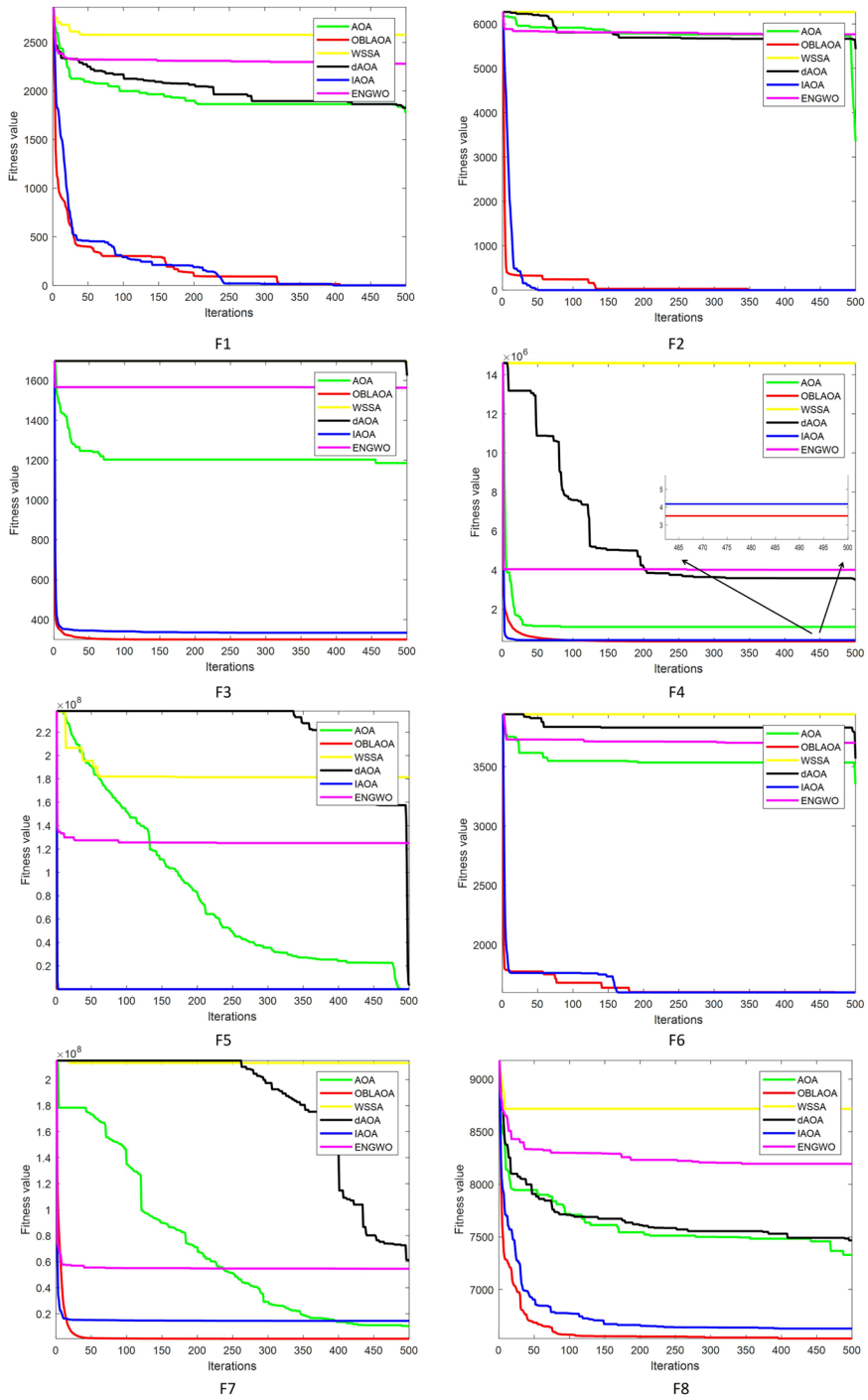


Fig. 5 20-dimensional F_1 - F_8 convergence curve of CEC2021 test function

Table 5 Datasets used in experiments

Dataset	Instance	Dimension	Cluster
Aggregation	788	3	7
Compound	399	3	6
Jain	373	3	2
Iris	150	5	3
Spiral	251	3	3
Pathbased	300	2	5
Wpbc	198	34	2
Synthesis	1980	3	5
R15	341	3	9
Vehicle1	846	19	4

Homogeneity refers to each cluster only containing one member of the same cluster. Completeness refers to all members of a cluster are in the same cluster. V-measure is average of Homogeneity and Completeness. The formula is as follows:

$$\text{homogeneity} = \frac{Hl}{Hl + Hr} - \frac{Hl \cdot Hr}{Hl}, \text{completeness} = \frac{Hl}{Hl + Hr} - \frac{Hl \cdot Hr}{Hr},$$

and

$$V - \text{measure} = \frac{2 * \text{homogeneity} * \text{completeness}}{\text{completeness} + \text{homogeneity}}.$$

The DBI index is usually less than 1, and the lower the index, the better the performance. SIL and RI index values are usually within 1, the closer they are to 1, the better the clustering performance of this method will be. The bigger Accuracy, NMI, homogeneity, completeness, and V-measure are the more real the clustering results are. Through the analysis of evaluation index, we can clearly compare the clustering performance of the new algorithm.

6.3 Experiment settings

DBSCAN [44] requires two parameters to be selected during clustering. By changing the values of EPS and MinPts parameters, we can get different clustering results. We first set up a large range of two parameters to run. We set EPS to 0-20 and MinPts to 0-40 to find an appropriate clustering results and adjust the range of parameters manually. These ranges of parameters are shown in Table 6. By comparing the results, we get a more accurate range for each dataset, which is used for the following experiments. We take EPS to one decimal and round down MinPts. We use the OBLAOA-DBSCAN algorithm to optimize these two parameters in the experiment. Firstly, we compare the optimization algorithms. The results of OBLAOA

are compared with the following algorithms: Arithmetic Optimization Algorithm (AOA), Whale Optimization Algorithm (WOA) [45], Salp Swarm Algorithm (SSA) [46], Weighted Salp Swarm Algorithms (WSSA) [47], Exponential Neighborhood Grey Wolf Optimization (ENGWO) [48], developed Arithmetic Optimization Algorithm (dAOA) [49] and improved arithmetic optimization algorithm (IAOA) [50]. Secondly, we compare our OBLAOA-DBSCAN algorithm with five classical clustering algorithms, namely K-means [51], Spectral [52], OPTICS [53], clustering by fast search and find of density peaks (DPC) [54] and the original DBSCAN.

To compare the gap conveniently and clearly among the algorithms, we set the parameters in the test as follows. The maximum number of iterations and population size of all algorithms are set to 100 and 20. In addition, we run each algorithm 20 times, and take the average result to eliminate the error in the experiment. The experimental algorithm run by MATLAB 2017b.

6.4 Experimental results of the optimization algorithm

In this part, we compared our improved optimization algorithm OBL-AOA with other seven meta-heuristic optimization algorithms. We take Euclidean distance as the fitness function and get the convergence curve of fitness function. In Tables 7, 8, 9 and 10, we show the error indexes of different algorithms and make better indexes in bold. In Fig. 6, we show the convergence curves of six datasets, and the convergence curves of other datasets are in Fig. 9 in Appendix.

The experiment shows that our OBLAOA algorithm is better than the original AOA algorithm, and it is the best among the eight optimization algorithms when we apply it into DBSCAN algorithm. We use the convergence curve and the error index to introduce them. Our optimization algorithm has better fitness function and the convergence rate, it can be seen through the convergence curve in Fig. 6. In all the datasets, the fitness function of our OBLAOA algorithm are better than the original AOA algorithm and other optimization algorithms. Fig. 6 shows that the

Table 6 The range of parameters for the investigated datasets

Dataset	EPS	MinPts
Aggregation	[1.0,3.0]	[2,25]
Compound	[1.2,2.8]	[2,15]
Jain	[2.4,3.3]	[4,16]
Iris	[0.5,1.8]	[0,30]
Spiral	[1.1,4.0]	[0,10]
Pathbased	[1.6,2.1]	[2,6]
Wpbc	[0.5,1.2]	[2,12]
Synthesis	[0.5,6.6]	[2,10]
R15	[0.5,1.2]	[0,20]
Vehicle1	[0.3,1.1]	[3,12]

convergence accuracy and rate of the OBLAOA are better than those of the AOA. In the datasets Aggregation, Jain, and Synthesis, as the function gradually converges, all algorithms converge more slowly and sometimes AOA falls into local optimal solution. However, because the OBL algorithm has strong local search capability, OBLAOA can still update the optimal solution.

Our OBLAOA algorithm performs better than the other optimization algorithms according to the results of error index in Tables 7, 8, 9 and 10. In the datasets Compound, Jain, Iris, Wpbc, Synthesis and Vehicle, we can clearly see that our OBLAOA-DBSCAN algorithm is better in accuracy. Its DBI index is smaller than the others, and its SIL, RI, NMI, homogeneity, completeness, and V-measure index are larger than the others. Their accuracy is the best of the eight algorithms, the accuracy of Compound is 0.8538, the accuracy of Jain is 0.7151, the accuracy of Iris is 1, the accuracy of Wpbc is 0.9346, the accuracy of Synthesis is 0.9998, the accuracy of Vehicle is 0.9656. Although some of the indexes are the same, our OBLAOA algorithm is better in general. In the four datasets Aggregation, Spiral, Pathbased and R15, the accuracy and the evaluation index of the different algorithms are similar. However, it can be concluded that, in general, the OBLAOA algorithm has a better effect on the analysis of clustering problems than the original AOA algorithm and other six meta-heuristic algorithms. Therefore, the OBLAOA-DBSCAN algorithm has a good influence on the clustering of datasets.

6.5 Experimental results of clustering algorithm

Specific clustering results of these datasets are recorded in Figs. 7 and 8, where we show the results by using K-means algorithm, Spectral algorithm, Optics algorithm, DPC algorithm, DBSCAN algorithm and the best clustering optimization algorithm (OBLAOA-DBSCAN). Each colour in the figure represents a cluster of data. By comparing the graphs of each cluster, we can make a basic judgment about the effect of clustering as follows. We can find that OBLAOA-DBSCAN algorithm has a better clustering result in Figs. 7 and 8. It can cluster the data into a better shape and find the actual number of clusters. The graphs of datasets without illustrations are in the in Fig. 10 in Appendix. In Tables 11 and 12, we show the error indexes of different Clustering algorithms and make better indexes in bold. The data with * in the table represents the data in articles [55] and [56].

In Fig. 7, compared with K-means, the effect of dataset Aggregation shows that our algorithm has more reliable clusters. Each cluster in the figure is clearly distinguished, while some clusters in K-means are not clearly distinguished. In Fig. 8, compared with Spectral, the effect of dataset Synthesis shows that our algorithm clusters more accurately on the left side of the graph. The cluster effect for a whole block of data is better than Spectral algorithm. From the graphs of Jain, Spiral and Pathbased in Figs. 7 and 8, the OBLAOA-DBSCAN algorithm is more accurate than K-means and Spectral for the for circular datasets.

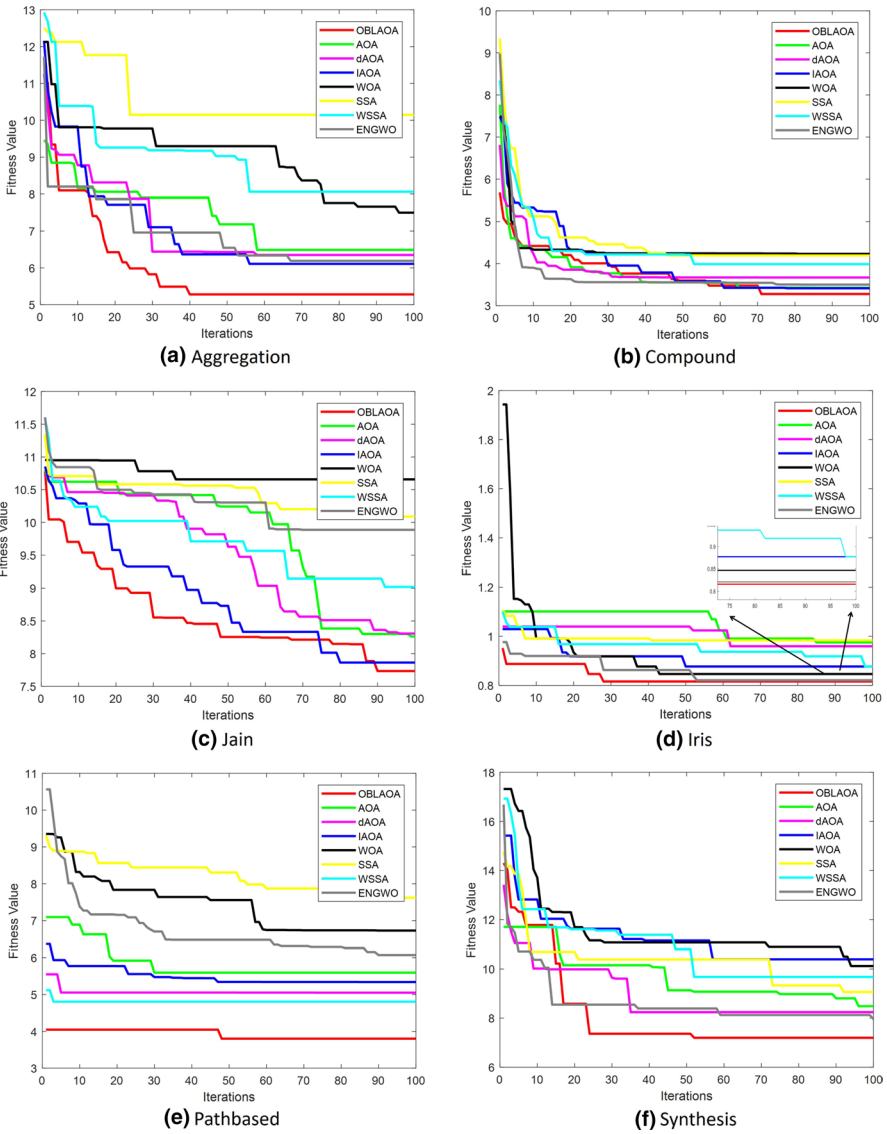


Fig. 6 Convergence curves with DBSCAN optimized by different meta-heuristic algorithms I

Table 7 The evaluation indexes of datasets in DBSCAN optimized by different meta-heuristic algorithms I

Dataset	Algorithm	Evaluation index			
		Accuracy	DBI	RI	SIL
Aggregation	WOA-DBSCAN	0.9949	0.3651	1	0.6813
	SSA-DBSCAN	0.9949	0.3651	1	0.6813
	WSSA-DBSCAN	0.9949	0.3651	1	0.6813
	ENGWO-DBSCAN	0.9949	0.3651	1	0.6813
	AOA-DBSCAN	0.9949	0.3651	1	0.6813
	dAOA-DBSCAN	0.9949	0.3651	1	0.6813
	IAOA-DBSCAN	0.9949	0.3651	1	0.6813
Compound	OBLAOA-DBSCAN	0.9949	0.3651	1	0.6813
	WOA-DBSCAN	0.7443	0.4221	0.8916	0.6488
	SSA-DBSCAN	0.7757	0.4334	0.9083	0.5890
	WSSA-DBSCAN	0.7453	0.4218	0.8922	0.6411
	ENGWO-DBSCAN	0.8321	1.0888	0.9324	0.1298
	AOA-DBSCAN	0.7966	1.0815	0.9171	0.1041
	dAOA-DBSCAN	0.8375	1.0888	0.9347	0.1341
Jain	IAOA-DBSCAN	0.7966	1.0815	0.9171	0.1041
	OBLAOA-DBSCAN	0.8538	1.0941	0.9415	0.1488
	WOA-DBSCAN	0.6728	0.4920	0.8517	0.4272
	SSA-DBSCAN	0.6518	0.4828	0.8427	0.4047
	WSSA-DBSCAN	0.6518	0.4828	0.8427	0.4047
	ENGWO-DBSCAN	0.6728	0.4920	0.8517	0.4272
	AOA-DBSCAN	0.6834	0.4999	0.8562	0.4355
Iris	dAOA-DBSCAN	0.6834	0.4999	0.8562	0.4355
	IAOA-DBSCAN	0.7151	0.5037	0.8700	0.4064
	OBLAOA-DBSCAN	0.7151	0.5037	0.8700	0.4064
	WOA-DBSCAN	0.9800	0.3773	0.9911	0.6642
	SSA-DBSCAN	0.9400	0.3760	0.9740	0.7148
	WSSA-DBSCAN	0.9600	0.3765	0.9825	0.7272
	ENGWO-DBSCAN	1	0.3654	1	0.7478
Spiral	AOA-DBSCAN	0.9400	0.3760	0.9740	0.7148
	dAOA-DBSCAN	0.9400	0.3760	0.9740	0.7148
	IAOA-DBSCAN	0.9600	0.3765	0.9825	0.7272
	OBLAOA-DBSCAN	1	0.3654	1	0.7478
	WOA-DBSCAN	1	2.2296	1	-0.1081
	SSA-DBSCAN	1	2.2296	1	-0.1081
	WSSA-DBSCAN	1	2.2296	1	-0.1081
ENGWO-DBSCAN	1	2.2296	1	-0.1081	
Spiral	AOA-DBSCAN	1	2.2296	1	-0.1081
	dAOA-DBSCAN	1	2.2296	1	-0.1081
	IAOA-DBSCAN	1	2.2296	1	-0.1081
	OBLAOA-DBSCAN	1	2.2296	1	-0.1081
	WOA-DBSCAN	1	2.2296	1	-0.1081
	SSA-DBSCAN	1	2.2296	1	-0.1081
	WSSA-DBSCAN	1	2.2296	1	-0.1081

Table 8 The evaluation indexes of datasets in DBSCAN optimized by different meta-heuristic algorithms II

Dataset	Algorithm	Evaluation index			
		Accuracy	DBI	RI	SIL
Pathbased	WOA-DBSCAN	0.8100	0.9482	0.8129	0.4349
	SSA-DBSCAN	0.8100	0.9482	0.8129	0.4349
	WSSA-DBSCAN	0.8100	0.9482	0.8129	0.4349
	ENGWO-DBSCAN	0.8100	0.9482	0.8129	0.4349
	AOA-DBSCAN	0.8233	0.9482	0.8158	0.4515
	dAOA-DBSCAN	0.8233	0.9482	0.8158	0.4515
	IAOA-DBSCAN	0.8233	0.9482	0.8158	0.4515
	OBLAOA-DBSCAN	0.8233	0.9482	0.8158	0.4515
Wpbc	WOA-DBSCAN	0.1314	0.3792	0.6699	0.0751
	SSA-DBSCAN	0.2595	0.4859	0.7075	0.1425
	WSSA-DBSCAN	0.2595	0.4859	0.7075	0.1425
	ENGWO-DBSCAN	0.8913	0.7829	0.9505	0.3282
	AOA-DBSCAN	0.8270	0.7552	0.9221	0.2937
	dAOA-DBSCAN	0.8270	0.7552	0.9221	0.2937
	IAOA-DBSCAN	0.8913	0.7829	0.9505	0.3282
	OBLAOA-DBSCAN	0.9346	0.8023	0.9700	0.3447
Synthesis	WOA-DBSCAN	0.9884	0.1926	0.9956	0.8542
	SSA-DBSCAN	0.9860	0.2152	0.9930	0.8606
	WSSA-DBSCAN	0.9884	0.1926	0.9956	0.8542
	ENGWO-DBSCAN	0.9970	0.1791	0.9985	0.8534
	AOA-DBSCAN	0.9970	0.1791	0.9985	0.8534
	dAOA-DBSCAN	0.9970	0.1791	0.9985	0.8534
	IAOA-DBSCAN	0.9860	0.2152	0.9930	0.8606
	OBLAOA-DBSCAN	0.9998	0.1787	0.9999	0.8517
R15	WOA-DBSCAN	1	0.3044	1	0.8966
	SSA-DBSCAN	0.9971	0.3047	0.9986	0.8946
	WSSA-DBSCAN	1	0.3044	1	0.8966
	ENGWO-DBSCAN	1	0.3044	1	0.8966
	AOA-DBSCAN	1	0.3044	1	0.8966
	dAOA-DBSCAN	1	0.3044	1	0.8966
	IAOA-DBSCAN	1	0.3044	1	0.8966
	OBLAOA-DBSCAN	1	0.3044	1	0.8966
Vehicle	WOA-DBSCAN	0.4823	1.5198	0.9605	0.1637
	SSA-DBSCAN	0.9561	1.5527	0.9835	0.1772
	WSSA-DBSCAN	0.9561	1.5527	0.9835	0.1772
	ENGWO-DBSCAN	0.9624	1.5534	0.9859	0.1790
	AOA-DBSCAN	0.9561	1.5527	0.9835	0.1772
	dAOA-DBSCAN	0.9248	1.5301	0.9718	0.1701
	IAOA-DBSCAN	0.9624	1.5534	0.9859	0.1790
	OBLAOA-DBSCAN	0.9656	1.5541	0.9871	0.1808

In Fig. 8, we can see from the graphs of datasets Pathbased and R15 that our algorithm clusters more accurately than Optics when dealing with dense data. When dealing with discrete data points, the Optics algorithm marks them as noise points. Our algorithm is more accurate when dealing with these points. We can draw this conclusion from the picture of dataset Synthesis. Through the datasets Aggregation and Jain in Fig. 7, it can be obtained that DPC algorithm marks the boundary points as noise points when dealing with data. Therefore, we can find that OBLAOA-DBSCAN algorithm has a better clustering effect than Optics and DPC on circular datasets by comparing cluster graphs. In addition, OBLAOA-DBSCAN correctly identifies sets of data points in areas of lower local density, and edge data points. In contrast, the original DBSCAN failed to accurately cluster these points.

We can find the Accuracy, RI, Sil, NMI, homogeneity, completeness, and V-measure index of OBLAOA-DBSCAN algorithm are significantly higher than those of K-means, Spectral, Optics, DPC and DBSCAN algorithm, the DBI index of the OBLAOA-DBSCAN algorithm is lower than that of K-means and Spectral algorithm from Tables 11 and 12. Therefore, the accuracy of improved OBLAOA-DBSCAN algorithm is better than the original DBSCAN in the dataset clustering.

Compared with the indexes of other articles in Table 11, our algorithm has better NMI indexes than K-means and original DBSCAN algorithms. On dataset Compound, OBLAOA-DBSCAN's NMI index is 48.74% higher than K-means's and 1.04% higher than DBSCAN's. On dataset Iris, OBLAOA-DBSCAN's NMI index is 13.25% higher than K-means' and 56.25% higher than DBSCAN's. Compared with the indexes of other articles in Table 12, our algorithm has better NMI indexes than K-means and DPC algorithms. On dataset Aggregation, OBLAOA-DBSCAN's NMI index is 17.52% higher than K-means's and 2.08% higher than DPC's. On dataset Jain, OBLAOA-DBSCAN's NMI index is 72.16% higher than K-means's and 2.74% higher than DPC's. On dataset Pathbased, OBLAOA-DBSCAN's NMI index is 28.99% higher than K-means's and 16.22% higher than DPC's. On dataset R15, OBLAOA-DBSCAN's NMI index is 0.58% higher than K-means's and 0.67% higher than DPC's.

In Table 11, we can find the index DBI and RI of dataset Spiral and Pathbased are not best, but their accuracy is better than those compared with real labels. According to figure, we can draw a conclusion that for circular datasets like Figs. 7 and 8, our DBSCAN algorithm can determine the shape of clustering more accurately and get better results. In Table 11, we can see that the SIL index has a negative set of values on circular dataset Spiral, but the clustering shapes are more consistent with the real labels. Through the above comparative analysis, we can find that OBLAOA-DBSCAN algorithm not only optimizes better than other optimization algorithms, but also performs better in clustering analysis compared with some classical clustering algorithms. In general, we can conclude that OBLAOA-DBSCAN algorithm has a very good effect on the clustering of datasets.

Table 9 The evaluation indexes of datasets in DBSCAN optimized by different meta-heuristic algorithms III

Dataset	Algorithm	Evaluation index			
		NMI	Homogeneity	Completeness	Vmeasure
Aggregation	WOA-DBSCAN	1	1	1	1
	SSA-DBSCAN	1	1	1	1
	WSSA-DBSCAN	1	1	1	1
	ENGWO-DBSCAN	1	1	1	1
	AOA-DBSCAN	1	1	1	1
	dAOA-DBSCAN	1	1	1	1
	IAOA-DBSCAN	1	1	1	1
	OBLAOA-DBSCAN	1	1	1	1
Compound	WOA-DBSCAN	0.8073	0.9531	0.6838	0.7963
	SSA-DBSCAN	0.8021	0.8928	0.7207	0.7976
	WSSA-DBSCAN	0.8061	0.9478	0.6856	0.7956
	ENGWO-DBSCAN	0.8736	0.9434	0.8089	0.8710
	AOA-DBSCAN	0.8364	0.9118	0.7672	0.8333
	dAOA-DBSCAN	0.8796	0.9488	0.8154	0.8770
	IAOA-DBSCAN	0.8364	0.9118	0.7672	0.8333
	OBLAOA-DBSCAN	0.9049	0.9729	0.8417	0.9026
Jain	WOA-DBSCAN	0.5987	0.6584	0.5409	0.5939
	SSA-DBSCAN	0.5781	0.6435	0.5194	0.5748
	WSSA-DBSCAN	0.5781	0.6435	0.5194	0.5748
	ENGWO-DBSCAN	0.5987	0.6584	0.5409	0.5939
	AOA-DBSCAN	0.6062	0.6660	0.5518	0.6035
	dAOA-DBSCAN	0.6062	0.6660	0.5518	0.6035
	IAOA-DBSCAN	0.6353	0.6894	0.5855	0.6332
	OBLAOA-DBSCAN	0.6353	0.6894	0.5855	0.6332
Iris	WOA-DBSCAN	0.9702	0.9703	0.9701	0.9702
	SSA-DBSCAN	0.9306	0.9311	0.9300	0.9306
	WSSA-DBSCAN	0.9488	0.9490	0.9486	0.9488
	ENGWO-DBSCAN	1	1	1	1
	AOA-DBSCAN	0.9306	0.9311	0.9300	0.9306
	dAOA-DBSCAN	0.9306	0.9311	0.9300	0.9306
	IAOA-DBSCAN	0.9488	0.9490	0.9486	0.9488
	OBLAOA-DBSCAN	1	1	1	1
Spiral	WOA-DBSCAN	1	1	1	1
	SSA-DBSCAN	1	1	1	1
	WSSA-DBSCAN	1	1	1	1
	ENGWO-DBSCAN	1	1	1	1
	AOA-DBSCAN	1	1	1	1
	dAOA-DBSCAN	1	1	1	1
	IAOA-DBSCAN	1	1	1	1
	OBLAOA-DBSCAN	1	1	1	1

Table 10 The evaluation indexes of datasets in DBSCAN optimized by different meta-heuristic algorithms IV

Dataset	Algorithm	Evaluation index			
		NMI	Homogeneity	Completeness	SIL
Pathbased	WOA-DBSCAN	0.6907	0.7114	0.6706	0.6904
	SSA-DBSCAN	0.6907	0.7114	0.6706	0.6904
	WSSA-DBSCAN	0.6907	0.7114	0.6706	0.6904
	ENGWO-DBSCAN	0.6907	0.7114	0.6706	0.6904
	AOA-DBSCAN	0.7012	0.7226	0.6804	0.7009
	dAOA-DBSCAN	0.7012	0.7226	0.6804	0.7009
	IAOA-DBSCAN	0.7012	0.7226	0.6804	0.7009
	OBLAOA-DBSCAN	0.7012	0.7226	0.6804	0.7009
Wpbc	WOA-DBSCAN	0.1655	0.3324	0.0824	0.1320
	SSA-DBSCAN	0.2649	0.4102	0.1711	0.2415
	WSSA-DBSCAN	0.2649	0.4102	0.1711	0.2415
	ENGWO-DBSCAN	0.8199	0.8443	0.7961	0.8195
	AOA-DBSCAN	0.7438	0.7817	0.7078	0.7429
	dAOA-DBSCAN	0.7438	0.7817	0.7078	0.7429
	IAOA-DBSCAN	0.8199	0.8443	0.7961	0.8195
	OBLAOA-DBSCAN	0.8786	0.8936	0.8637	0.8784
Synthesis	WOA-DBSCAN	0.9674	0.9820	0.9530	0.9673
	SSA-DBSCAN	0.9505	0.9809	0.9386	0.9593
	WSSA-DBSCAN	0.9674	0.9820	0.9530	0.9673
	ENGWO-DBSCAN	0.9836	0.9914	0.9758	0.9835
	AOA-DBSCAN	0.9836	0.9914	0.9758	0.9835
	dAOA-DBSCAN	0.9836	0.9914	0.9758	0.9835
	IAOA-DBSCAN	0.9505	0.9809	0.9386	0.9593
	OBLAOA-DBSCAN	0.9934	0.9980	0.9888	0.9934
R15	WOA-DBSCAN	1	1	1	1
	SSA-DBSCAN	0.9937	0.9937	0.9937	0.9937
	WSSA-DBSCAN	1	1	1	1
	ENGWO-DBSCAN	1	1	1	1
	AOA-DBSCAN	1	1	1	1
	dAOA-DBSCAN	1	1	1	1
	IAOA-DBSCAN	1	1	1	1
	OBLAOA-DBSCAN	1	1	1	1
Vehicle	WOA-DBSCAN	0.8938	0.8953	0.8924	0.8938
	SSA-DBSCAN	0.9436	0.9439	0.9432	0.9436
	WSSA-DBSCAN	0.9436	0.9439	0.9432	0.9436
	ENGWO-DBSCAN	0.9500	0.9503	0.9497	0.9500
	AOA-DBSCAN	0.9436	9439	0.9432	0.9436
	dAOA-DBSCAN	0.9161	0.9169	0.9154	0.9161
	IAOA-DBSCAN	0.9500	0.9503	0.9497	0.9500
	OBLAOA-DBSCAN	0.9535	0.9538	0.9533	0.9585

Table 11 The evaluation indexes of datasets in different clustering algorithms I

Dataset	Algorithm	Evaluation index			
		Accuracy	DBI	RI	SIL
Aggregation	K-means	0.9226	0.8668	0.5323	0.6729
	Spectral	0.9727	0.9870	0.3853	0.6808
	Optics	0.5533	0.8301	0.9274	0.4362
	DPC	0.6586	0.3738	0.9055	0.3023
	DBSCAN	0.9898	0.3657	0.9993	0.5877
	OBLAOA-DBSCAN	0.9949	0.3651	1	0.6813
Compound	K-means	0.5740*	0.6115	0.8234	0.5329
	Spectral	0.5789	0.5692	0.8462	0.606
	Optics	0.4286	0.9662	0.9302	0.0496
	DPC	0.2130	0.4052	0.8444	0.3998
	DBSCAN	0.8450*	1.0888	0.9347	0.1341
	OBLAOA-DBSCAN	0.8538	1.0941	0.9415	0.1488
Jain	K-means	0.7748	0.3923	0.6501	0.6722
	Spectral	0.7105	0.3879	0.5875	0.6466
	Optics	0.1930	0.5667	0.6876	0.3049
	DPC	0.6793	2.1268	0.5624	0.0688
	DBSCAN	0.6834	0.4999	0.8562	0.4355
	OBLAOA-DBSCAN	0.7151	0.5037	0.8700	0.4064
Iris	K-means	0.8830*	0.3960	0.8997	0.7242
	Spectral	0.9933	0.3824	0.9910	0.7540
	Optics	0.6600	0.6453	0.7719	0.6943
	DPC	0.9067	0.3902	0.8923	0.7023
	DBSCAN	0.6400*	0.3654	1	0.7478
	OBLAOA-DBSCAN	1	0.3654	1	0.7478
Spiral	K-means	0.3720	0.6149	0.5156	0.5492
	Spectral	0.2815	0.5707	0.5339	0.5415
	Optics	0.9760	2.1960	0.9888	-0.0206
	DPC	1	2.2296	1	-0.1081
	DBSCAN	1	2.2296	1	-0.1081
	OBLAOA-DBSCAN	1	2.2296	1	-0.1081
Pathbased	K-means	0.7500	0.4437	0.7319	0.7482
	Spectral	0.7500	0.4437	0.7319	0.7482
	Optics	0.6167	1.3392	0.7499	0.2123
	DPC	0.6467	0.8539	0.6769	0.4039
	DBSCAN	0.8100	0.9482	0.8129	0.4349
	OBLAOA-DBSCAN	0.8233	0.9482	0.8158	0.4515

Table 11 (continued)

Dataset	Algorithm	Evaluation index			
		Accuracy	DBI	RI	SIL
Wpbc	K-means	0.6010	1	0.5180	0.2872
	Spectral	0.6041	1.4287	0.5201	0.0796
	Optics	0.7020	1.6713	0.5795	0.3222
	DPC	0.6670	0.2529	0.6308	0.5213
	DBSCAN	0.8270	0.7552	0.9221	0.2937
	OBLAOA-DBSCAN	0.9346	0.8023	0.9700	0.3447
Synthesis	K-means	0.6074	0.5452	0.7573	0.6496
	Spectral	0.8025	0.2735	0.9018	0.7964
	Optics	0.9788	0.3227	0.9989	0.8517
	DPC	0.9829	0.3680	0.9965	0.7789
	DBSCAN	0.9884	0.1926	0.9956	0.8542
	OBLAOA-DBSCAN	0.9998	0.1787	0.9999	0.8517
R15	K-means	0.8106	0.4353	0.9598	0.6844
	Spectral	0.8409	0.3953	0.9659	0.7473
	Optics	0.1202	1.0480	0.2310	0.5570
	DPC	0.9941	0.3500	0.9973	0.8398
	DBSCAN	1	0.3044	1	0.8966
	OBLAOA-DBSCAN	1	0.3044	1	0.8966
Vehicle	K-means	0.2920	1.4350	0.6530	0.3075
	Spectral	0.3014	0.7864	0.6903	0.3264
	Optics	0.2577	0.9972	0.2625	0.2327
	DPC	0.3262	0.8936	0.5118	0.0943
	DBSCAN	0.9309	1.5369	0.9741	0.1717
	OBLAOA-DBSCAN	0.9656	1.7856	0.9871	0.1808

Table 12 The evaluation indexes of datasets in different clustering algorithms II

Dataset	Algorithms	Evaluation Index			
		NMI	Homogeneity	Completeness	Vmeasure
Aggregation	K-means	0.8509*	0.7911	0.8722	0.8297
	Spectral	0.9927	0.9912	0.9942	0.9927
	Optics	0.8833	0.9766	0.7990	0.8789
	DPC	0.9769*	0.8221	0.7950	0.8083
	DBSCAN	0.9960	0.9954	0.9966	0.9960
	OBLAOA-DBSCAN	1	1	1	1
Compound	K-means	0.6604	0.6351	0.6867	0.6599
	Spectral	0.7236	0.6890	0.7600	0.7228
	Optics	0.8317	0.9104	0.7598	0.8283
	DPC	0.7597	0.7639	0.7559	0.7597
	DBSCAN	0.8796	0.9488	0.8154	0.8770
	OBLAOA-DBSCAN	0.9049	0.9729	0.8417	0.9026
Jain	K-means	0.3690*	0.3375	0.4070	0.3690
	Spectral	0.3072	0.2804	0.3367	0.3060
	Optics	0.2216	0.3209	0.1530	0.2072
	DPC	0.6183*	0.1048	0.1264	0.1146
	DBSCAN	0.6062	0.6660	0.5518	0.6035
	OBLAOA-DBSCAN	0.6353	0.6894	0.5855	0.6332
Iris	K-means	7.7766*	0.7650	0.7515	0.7582
	Spectral	0.9702	0.9703	0.9701	0.9702
	Optics	0.7220	0.9063	0.5752	0.7037
	DPC	0.8350*	0.7960	0.8156	0.8057
	DBSCAN	1	1.0000	1	1
	OBLAOA-DBSCAN	1	1.0000	1	1
Spiral	K-means	0.0636	0.0619	0.0654	0.0636
	Spectral	0.0586	0.0579	0.0594	0.0586
	Optics	0.9582	0.9597	0.9567	0.9582
	DPC	1	1	1	1
	DBSCAN	1	1	1	1
	OBLAOA-DBSCAN	1	1	1	1
Pathbased	K-means	0.5463*	0.5846	0.5140	0.5470
	Spectral	0.5482	0.5846	0.5140	0.5470
	Optics	0.6398	0.7799	0.5248	0.6274
	DPC	0.5390*	0.4845	0.3597	0.4129
	DBSCAN	0.6907	0.7114	0.6706	0.6904
	OBLAOA-DBSCAN	0.7012	0.7226	0.6804	0.7009

Table 12 (continued)

Dataset	Algorithms	Evaluation Index			
		NMI	Homogeneity	Completeness	Vmeasure
Wpbc	K-means	0.0270	0.0241	0.0302	0.0268
	Spectral	0.4389	0.5421	0.3553	0.4293
	Optics	0.0089	0.0124	0.0064	0.0084
	DPC	0.0104	0.0432	0.0025	0.0047
	DBSCAN	0.7438	0.7817	0.7078	0.7429
	OBLAOA-DBSCAN	0.8786	0.8936	0.8637	0.8784
Synthesis	K-means	0.6313	0.8546	0.4734	0.6309
	Spectral	0.8209	0.7201	0.9346	0.8140
	Optics	0.9826	0.9801	0.9852	0.9826
	DPC	0.9726	0.9619	0.9835	0.9726
	DBSCAN	0.9674	0.9820	0.9530	0.9673
	OBLAOA-DBSCAN	0.9934	0.9980	0.9888	0.9934
R15	K-means	0.9942*	0.8839	0.9182	0.9007
	Spectral	0.9441	0.9225	0.9630	0.9439
	Optics	0.2991	0.1244	0.7190	0.2122
	DPC	0.9933*	0.9874	0.9874	0.9874
	DBSCAN	1	1	1	1
	OBLAOA-DBSCAN	1	1	1	1
Vehicle	K-means	0.0999	0.0997	1	0.0999
	Spectral	0.3927	0.3489	0.4420	0.3900
	Optics	0.0351	0.0078	0.1585	0.0148
	DPC	0.1106	0.0770	0.1587	0.1037
	DBSCAN	0.9208	0.9215	0.9202	0.9208
	OBLAOA-DBSCAN	0.9535	0.9538	0.9533	0.9585

7 Conclusion

In this paper, we have proposed a new clustering algorithm named OBLAOA-DBSCAN. In this algorithm, we introduce OBL into AOA algorithm and develop an OBLAOA optimizer to improve the global search ability and convergence accuracy of standard AOA algorithm. Then, we use the improved OBLAOA algorithm to adjust the EPS and MinPts parameters of DBSCAN in order to improve the clustering effect of DBSCAN and propose a hybrid clustering algorithm (OBLAOA-DBSCAN). In our numerical simulation, we have demonstrated that the improved OBLAOA is more effective than the original AOA and other current

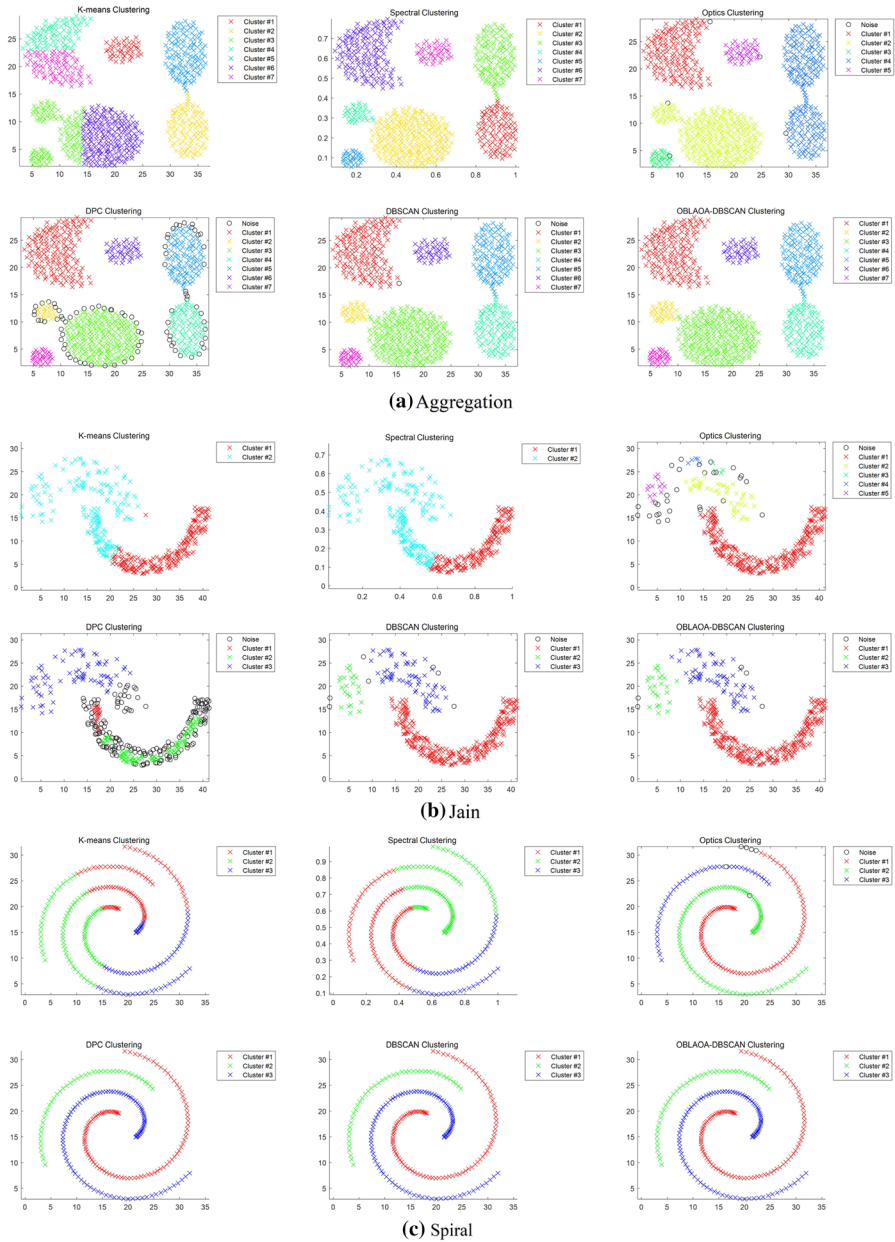


Fig. 7 The results with different clustering algorithms I

popular algorithms. In addition, we also have validated the effectiveness of our proposed OBLAOA-DBSCAN algorithm by many clustering projects and found that the proposed clustering algorithm can achieve an accurate and reliable clustering results with less computational costs.

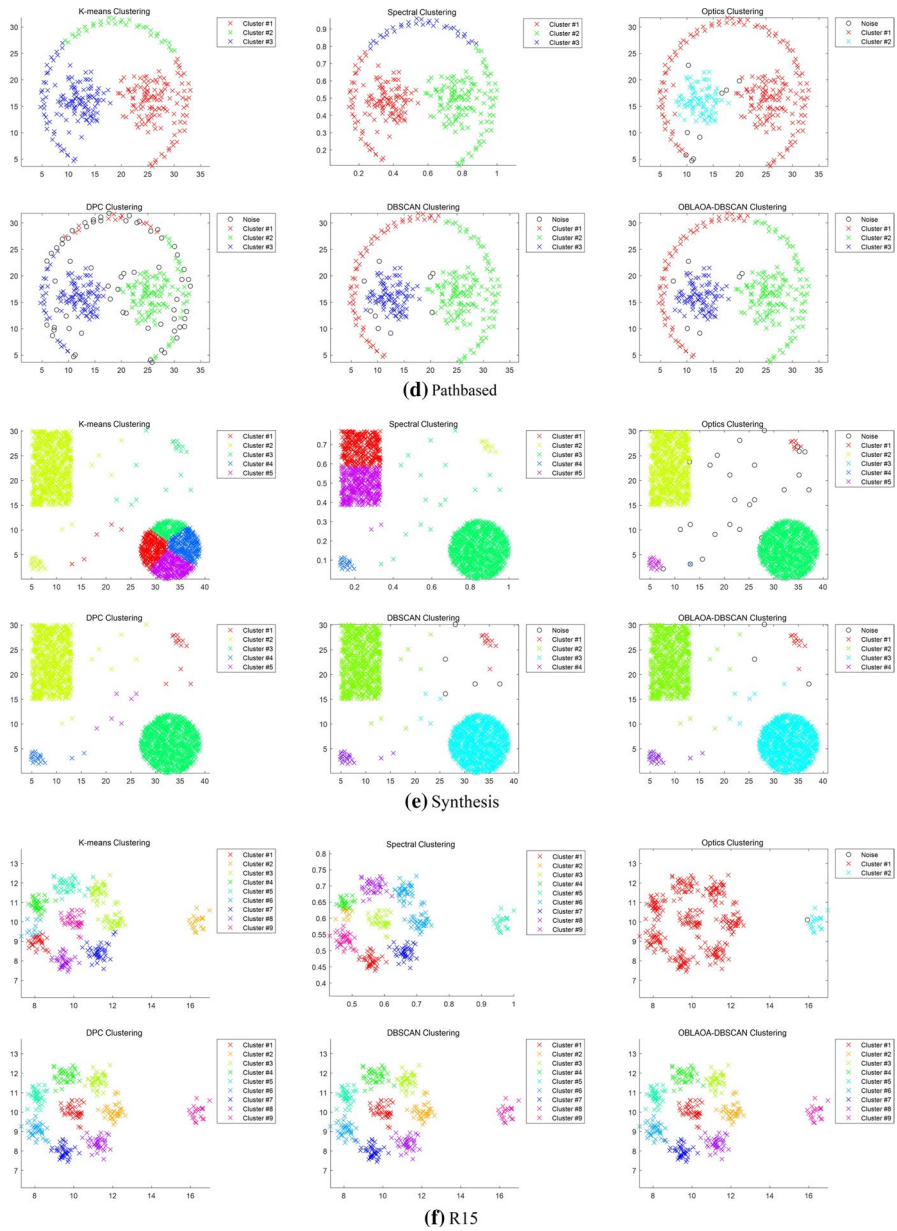


Fig. 8 The results with different clustering algorithms II

Although OBLAOA-DBSCAN can achieve significant improvement, there are still some insufficient, such as the selection of the best parameters of the optimization algorithm, the global search ability and clustering effect of the optimization algorithm need to be further improved. In the future, we will apply

OBLAOA-DBSCAN to clustering problems on more datasets. In addition, OBLAOA can also be applied to other application problems like clustering model, such as image classification and recognition, speech signal classification, electrical information classification and so on, which needs further research by other researchers.

Appendix A The additional results for our experiments

See Figs. 9 and 10.

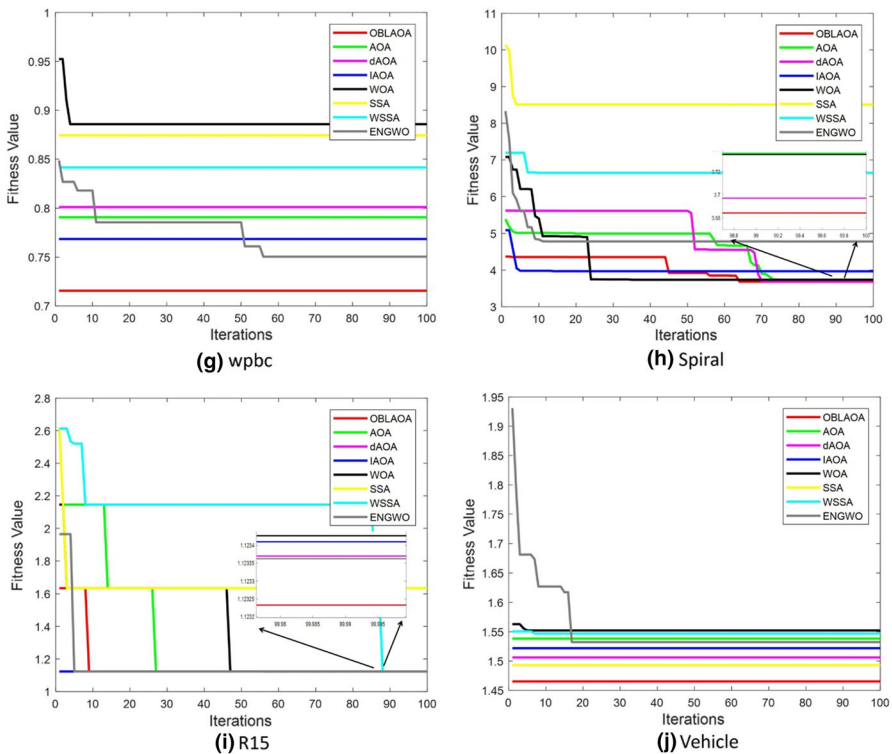


Fig. 9 Convergence curves with DBSCAN optimized by different meta-heuristic algorithms II

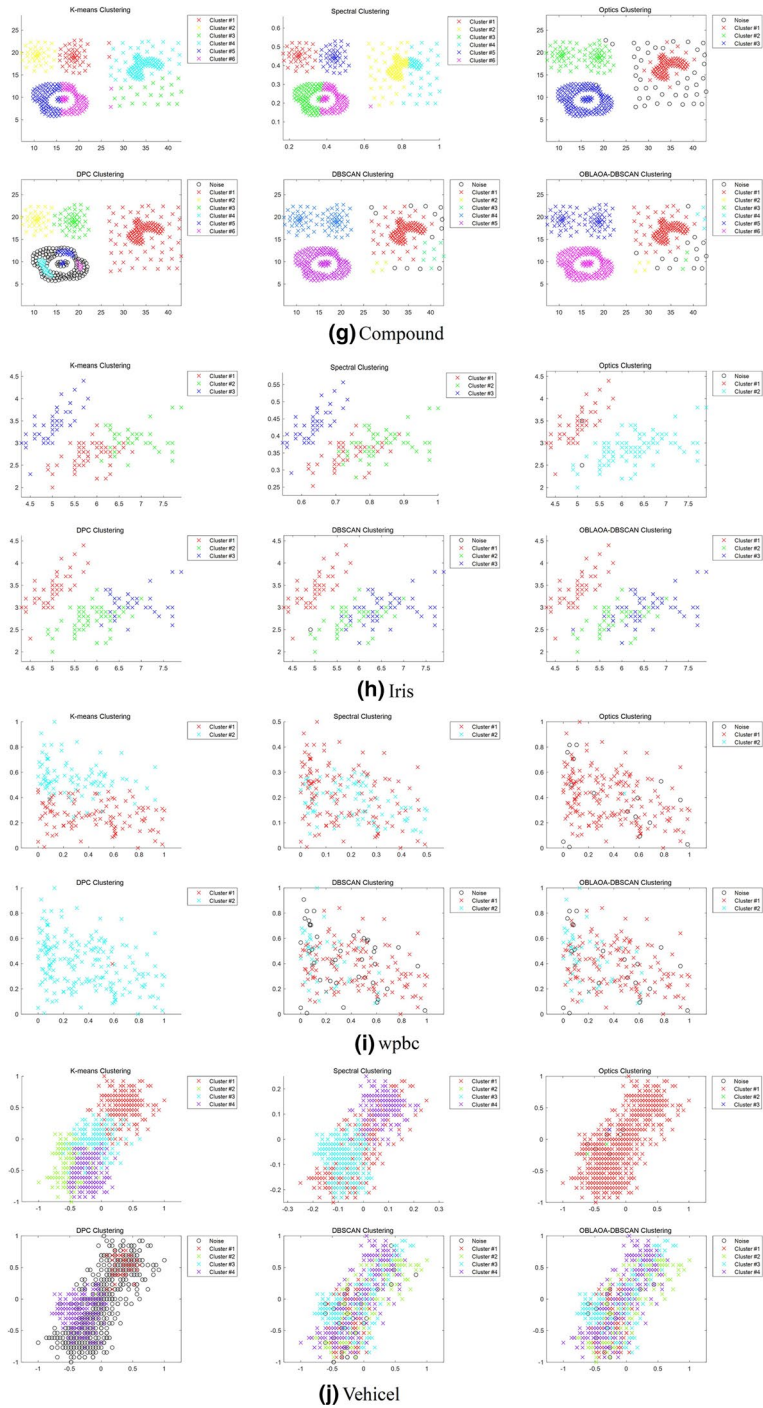


Fig. 10 The results with different clustering algorithms III

Acknowledgements The authors would like to thank the six excellent reviewers for their constructive comments and suggestions, which have led to a much-improved paper. Also, the authors would like to acknowledge Ms. Xin Jiang and Ms. Xia Lin for their preparation for the original manuscript. This work is supported in part by the National Natural Science Foundation of China under Grant 61873130 and Grant 61833011, in part by the Natural Science Foundation of Jiangsu Province under Grant BK20191377, in part by the 1311 Talent Project of Nanjing University of Posts and Telecommunications, in part by Natural Science Foundation of Nanjing University of Posts and Telecommunications under Grant NY220194 and under Grant NY221082 by the Australian Research Council project DP160104292 and the National Natural Science Foundation of China under Grant 62001337.

Authors's contribution YY: Project administration; QC: Software, Visualization, Formal analysis, Writing-review and editing; HL: Software, Visualization, Formal analysis, Writing-original draft; YG: Writing-review and editing; JW: Investigation, Formal analysis, Writing-review and editing; C-JL: Writing-review and editing; SZ: Writing-review and editing.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions.

Data availability Data are available from the second and third author.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

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

References

1. Yuvaraj N, Suresh Ghana Dhas C (2020) High-performance link-based cluster ensemble approach for categorical data clustering. *J Supercomput* 76(6):4556–4579
2. Hussein S, Kandel P, Bolan CW, Wallace MB, Bagci U (2019) Lung and pancreatic tumor characterization in the deep learning era: novel supervised and unsupervised learning approaches. *IEEE Trans Med Imag* 38(8):1777–1787
3. Wu J, Wang YG, Burrage K, Tian YC, Lawson B, Ding Z (2020) An improved firefly algorithm for global continuous optimization problems. *Expert Syst Appl* 149:113340
4. Chen H, Li W, Yang X (2020) A whale optimization algorithm with chaos mechanism based on quasi-opposition for global optimization problems. *Expert Syst Appl* 158:113612
5. Edwin Dhas P, Sankara Gomathi B (2020) A novel clustering algorithm by clubbing GHFCM and GWO for microarray gene data. *J Supercomput* 76(8):5679–5693
6. Wang C, Koh JM, Yu T, Xie NG, Cheong KH (2020) Material and shape optimization of bi-directional functionally graded plates by GIGA and an improved multi-objective particle swarm optimization algorithm. *Computer Methods Appl Mech Eng* 366:113017
7. Nanda SJ, Panda G (2014) A survey on nature inspired metaheuristic algorithms for partitional clustering. *Swarm Evol Comput* 16:1–18

8. Hu L, Zhang J, Pan X, Yan H, You ZH (2021) HiSCF: leveraging higher-order structures for clustering analysis in biological networks. *Bioinformatics*. 37(4):542–550
9. Chen YJ, Chen MZ, Zhang HW, Wu GS, Guo SR (2021) Effect of Guo Qing Yi Tang combined with Western medicine cluster therapy on acute pancreatitis. *Am J Emergency Med* 50:66–70
10. Rochat L, Bianchi-Demicheli F, Aboujaoude E, Khazaal Y (2019) The psychology of swiping: a cluster analysis of the mobile dating app Tinder. *J Behav Addict* 8(4):804–813
11. Kim S, Jung I (2017) Optimizing the maximum reported cluster size in the spatial scan statistic for ordinal data. *PLoS One* 12(7):182234
12. Celebi ME (2014) *Partitional clustering algorithms*. Springer, New York
13. Medová J, Bakusová J (2019) Application of hierarchical cluster analysis in educational research: Distinguishing between transmissive and constructivist oriented mathematics teachers. *Statistika: Stat Econ J* 99:142–150
14. Gong W, Pang L, Wang J, Xia M, Zhang Y (2021) A social-aware K means clustering algorithm for D2D multicast communication under SDN architecture. *AEU-Int J Electron Commun* 132:153610
15. Raj S, Improved Ghosh D, Optimal DBSCAN, for Embedded Applications Using High-Resolution Automotive Radar. In, (2020) 21st International Radar Symposium (IRS). *IEEE* 2020:343–346
16. Mardani K, Maghooli K (2021) Enhancing retinal blood vessel segmentation in medical images using combined segmentation modes extracted by DBSCAN and morphological reconstruction. *Biomed Signal Process Control* 69:102837
17. Fouedjio F (2020) Clustering of multivariate geostatistical data. *Wiley Interdiscipl Rev: Comput Stat* 12(5):1510
18. Wang L, Wang H, Han X, Zhou W (2021) A novel adaptive density-based spatial clustering of application with noise based on bird swarm optimization algorithm. *Computer Commun* 174:205–214
19. Wang C, Ji M, Wang J, Wen W, Li T, Sun Y (2019) An improved DBSCAN method for LiDAR data segmentation with automatic Eps estimation. *Sensors* 19(1):172
20. Jian Z, Zhu G (2021) Affine invariance of meta-heuristic algorithms. *Inf Sci* 576:37–53
21. Agarwal P, Mehta S, Abraham A (2021) A meta-heuristic density-based subspace clustering algorithm for high-dimensional data. *Soft Comput* 25:10237–10256
22. Zhang H, Nguyen H, Bui XN, Pradhan B, Mai NL, Vu DA (2021) Proposing two novel hybrid intelligence models for forecasting copper price based on extreme learning machine and meta-heuristic algorithms. *Resour Policy* 73:102195
23. Singh H, Singh B, Kaur M (2021) An improved elephant herding optimization for global optimization problems. *Eng Computers* 55:1–33
24. Lai W, Zhou M, Hu F, Bian K, Song Q (2019) A new DBSCAN parameters determination method based on improved MVO. *IEEE Access* 7:104085–104095
25. Jian S, Li D, Yu Y (2021) Research on Taxi Operation Characteristics by Improved DBSCAN Density Clustering Algorithm and K-means Clustering Algorithm. In: *Journal of Physics: Conference Series*. vol. 1952. IOP Publishing; p. 042103
26. Zhu Q, Tang X, Elahi A (2021) Application of the novel harmony search optimization algorithm for DBSCAN clustering. *Expert Syst Appl* 178:115054
27. Hu L, Liu H, Zhang J, Liu A (2021) KR-DBSCAN: a density-based clustering algorithm based on reverse nearest neighbor and influence space. *Expert Syst Appl* 186:115763
28. Li M, Bi X, Wang L, Han X (2021) A method of two-stage clustering learning based on improved DBSCAN and density peak algorithm. *Computer Commun* 167:75–84
29. Abualigah L, Diabat A, Mirjalili S, Abd Elaziz M, Gandomi AH (2021) The arithmetic optimization algorithm. *Computer Methods Appl Mech Eng* 376:113609
30. Brust JJ, Marcia RF, Petra CG (2019) Large-scale quasi-Newton trust-region methods with low-dimensional linear equality constraints. *Comput Optim Appl* 74(3):669–701
31. Bouhlef MA, Martins JR (2019) Gradient-enhanced kriging for high-dimensional problems. *Eng Computers* 35(1):157–173
32. Fu G, Wang C, Zhang D, Zhao J, Wang H (2019) A multiobjective particle swarm optimization algorithm based on multipopulation coevolution for weapon-target assignment. *Math Probl Eng* 2019:1424590
33. Elgamal ZM, Yasin NM, Sabri AQM, Sihwail R, Tubishat M, Jarrah H (2021) Improved equilibrium optimization algorithm using elite opposition-based learning and new local search strategy for feature selection in medical datasets. *Computation* 9(6):68

34. Lei D, You T, Cai L (2021) Parameter identification of roll motion equation of ship in regular wave using opposition based learning gaussian bare bone imperialist competition algorithm. *IEEJ Trans Electr Electron Eng* 16(8):1086–1092
35. Nekooei-Joghndani A, Safi-Esfahani F (2021) Dynamic scheduling of independent tasks in cloud computing applying a new hybrid metaheuristic algorithm including Gabor filter, opposition-based learning, multi-verse optimizer, and multi-tracker optimization algorithms. *J Supercomput* 78:1182–1243
36. Ester M, Kriegel H, Sander J, Xu X, Idrissov A, Nascimento M et al (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. In: *KDD*. vol. 2. ACM Press. p. 49–60
37. Abualigah L, Ewees AA, Al-qaness MAA, Elaziz MA, Yousri D, Ibrahim RA et al (2022) Boosting arithmetic optimization algorithm by sine cosine algorithm and levy flight distribution for solving engineering optimization problems. *Neural Comput Appl* 34(11):8823–8852
38. Kamil AT, Saleh HM, Abd-Alla IH (2021) A multi-swarm structure for particle swarm optimization: Solving the welded beam design problem. In: *Journal of Physics: Conference Series*. vol. 1804. IOP Publishing. p. 01201
39. Gupta S (2021) Enhanced harmony search algorithm with non-linear control parameters for global optimization and engineering design problems. *Eng Computers* 87:1–24
40. Kumar N, Mahato SK, Bhunia AK (2021) Design of an efficient hybridized CS-PSO algorithm and its applications for solving constrained and bound constrained structural engineering design problems. *Results Control Optim* 5:100064
41. Rad MH, Abdolrazzagah-Nezhad M (2020) A new hybridization of DBSCAN and fuzzy earthworm optimization algorithm for data cube clustering. *Soft Comput* 24(20):15529–15549
42. Gholizadeh N, Saadatfar H, Hanafi N (2021) K-DBSCAN: an improved DBSCAN algorithm for big data. *J Supercomput* 77(6):6214–6235
43. Zhang H, Guo H, Wang X, Ji Y, Wu QJ (2020) Clothescounter: a framework for star-oriented clothes mining from videos. *Neurocomputing* 377:38–48
44. Bryant A, Cios K (2017) RNN-DBSCAN: a density-based clustering algorithm using reverse nearest neighbor density estimates. *IEEE Trans Knowl Data Eng* 30(6):1109–1121
45. Jiang J, Feng T, Liu C (2021) An improved nonlinear grey bernoulli model based on the whale optimization algorithm and its application. *Math Probl Eng* 2021:6691724
46. Abd El-sattar S, Kamel S, Ebeed M, Jurado F (2021) An improved version of salp swarm algorithm for solving optimal power flow problem. *Soft Comput* 25(5):4027–4052
47. Chouhan N, Bhatt UR, Upadhyay R (2021) Weighted salp swarm and salp swarm algorithms in fiWi access network: a new paradigm for ONU placement. *Opt Fiber Technol* 63:102505
48. Mohakud R, Dash R (2022) Skin cancer image segmentation utilizing a novel EN-GWO based hyper-parameter optimized FCEDN. *J King Saud Univ-Computer Inf Sci* 45:1–16
49. Xu YP, Tan JW, Zhu DJ, Ouyang P, Taheri B (2021) Model identification of the proton exchange membrane fuel cells by extreme learning machine and a developed version of arithmetic optimization algorithm. *Energy Rep* 7:2332–2342
50. Kaveh A, Hamedani KB (2022) Improved arithmetic optimization algorithm and its application to discrete structural optimization. In: *Structures*. vol. 35. Elsevier; p. 748–764
51. Karczmarek P, Kiersztyn A, Pedrycz W, Al E (2020) K-Means-based isolation forest. *Knowl-Based Syst* 195:105659
52. Allab K, Labiod L, Nadif M (2016) Power simultaneous spectral data embedding and clustering. In: *Proceedings of the 2016 SIAM International Conference on Data Mining*. SIAM; p. 270–278
53. Kim JH, Choi JH, Yoo KH, Nasridinov A (2019) AA-DBSCAN: an approximate adaptive DBSCAN for finding clusters with varying densities. *J Supercomput* 75(1):142–169
54. Rodriguez A, Laio A (2014) Clustering by fast search and find of density peaks. *Science* 344(6191):1492–1496
55. Guo W, Xu P, Dai F, Hou Z (2022) Harris hawks optimization algorithm based on elite fractional mutation for data clustering. *Appl Intell* 89:1–27
56. Zhang Y, Ding S, Wang L, Wang Y, Ding L (2021) Chameleon algorithm based on mutual k-nearest neighbors. *Appl Intell* 51(4):2031–2044

Authors and Affiliations

Yang Yang¹  · Chen Qian¹ · Haomiao Li¹ · Yuchao Gao¹ · Jinran Wu²  ·
Chan-Juan Liu³  · Shangrui Zhao⁴ 

✉ Jinran Wu
j73.wu@qut.edu.au

✉ Chan-Juan Liu
liuchanjuan@shcc.edu.cn

Yang Yang
yyang@njupt.edu.cn

Chen Qian
b18040931@njupt.edu.cn

Haomiao Li
b19041019@njupt.edu.cn

Yuchao Gao
b18040933@njupt.edu.cn

Shangrui Zhao
zhaosr@whut.edu.cn

¹ College of Automation and College of Artificial Intelligence, Nanjing University of Posts and Telecommunications, Nanjing 210023, Jiangsu, People's Republic of China

² School of Mathematical Sciences, Queensland University of Technology, Brisbane 4001, Queensland, Australia

³ School of Business Administration and Customs Affairs, Shanghai Customs College, Shanghai, Shanghai 201204, People's Republic of China

⁴ School of Science, Wuhan University of Technology, Wuhan 430070, Hubei, People's Republic of China