IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 12, 2021, accepted March 18, 2021, date of publication March 22, 2021, date of current version March 30, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3067729

Battlefield Target Grouping by a Hybridization of an Improved Whale Optimization Algorithm and Affinity Propagation

YUXIAN DUAN[1,2], **CHANGYUN LIU**[1], **AND SONG LI**[1]
[1]Air and Missile Defense College, Air Force Engineering University, Xi'an 710051, China
[2]Graduate College, Air Force Engineering University, Xi'an 710051, China

Corresponding author: Song Li (elizabeth.leary@students.clatsopcc.edu)

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant 61703426.

ABSTRACT Target grouping, which is essentially a data clustering problem, is a research hotspot in the field of battlefield situation assessment. To address an unknown battlefield environment, affinity propagation based on an improved whale optimization algorithm (APBWOA) is proposed from the perspective of clustering. First, we propose a whale optimization algorithm based on a chaotic map and nonlinear inertia weight improvement called CPIW-WOA, which uses an improved circle map to generate initial populations and introduces nonlinear inertia weights to improve its convergence efficiency. The test results on nine benchmark functions show that the CPIW-WOA algorithm has superior performance to existing methods. Second, based on the fact that it fully considers the weights of the attributes in a given sample, the weighted Mahalanobis distance is adopted to replace the Euclidean distance. In addition, the silhouette index is introduced to determine the optimal number of clusters. By iteratively updating through CPIW-WOA to search for the optimal settings, the limitation of manually entering specified parameters can be overcome. Test results on real-world datasets show that the new method is more accurate and effective than other methods; therefore, it can provide effective solutions with respect to battlefield target grouping.

INDEX TERMS Clustering algorithm, affinity propagation, whale optimization algorithm, target grouping, situation assessment.

I. INTRODUCTION

Target grouping, which requires taking a combination of spatial geometric factors, temporal factors, and functional dependencies into account, is an important issue in the field of battlefield situation assessment. On the basis of key battlefield elements, including the operational environment, human geography, force distribution of enemies, and operational strength, the state information of the target needs to be extracted and mined. In that case, a clear view of the battlefield can be provided for situational understanding. From the perspective of game theory, battlefield target grouping is a process of forward reasoning and gradual simplification of the amount of information. In fact, target grouping is essentially a data clustering problem. By clustering and analyzing the targets, the scattered units are grouped into clusters, thereby effectively reducing the complexity of the battlefield view. At the same time, the relationships at each level can be shown clearly for commanders, and this facilitates reasonable and scientific decisions.

In fields such as data mining and machine learning, cluster analysis occupies an important position. The purpose of clustering is to divide datasets into clusters of classes that contain different information granularities by computing the similarities between points. Therefore, objects in the same class are more similar to each other, while objects in different classes are less similar to each other. In recent years, clustering has been widely used in traffic scheduling [1], health care [2], asset allocation [3], climate analysis [4], energy consumption [5], and other fields. There are many traditional clustering methods, but each has its limitations. For example, k-means requires the manual specification of initial clustering centers and the number of clusters, and it is more sensitive to noise and outliers than other approaches [6]. Fuzzy c-means (FCM) clustering suffers from a high dependence on the initial values, and it is prone to falling into

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

local minima [7]. The density-based DBSCAN algorithm has weaknesses such as high time complexity and is not applicable to large datasets [8].

In 2007, Frey and Dueck proposed a division-based clustering algorithm called affinity propagation (AP) [9]. Based on factor graphs and the sum-product algorithm, it treats all data points as potential clustering centers. Next, it continuously passes messages along the edges of the network (a[i,k] vs. r[i,k]) through an update loop. In this way, aggregation can be achieved, and categories are identified. Compared with other algorithms, such as k-means, affinity propagation does not need to predefine the number of subpopulations and the initial clustering centers. Experiments have shown that affinity propagation finds better clustering centers than those output by the K-means algorithm [10]. Furthermore, affinity propagation does not require the similarity matrix to be symmetric, so it can solve problems such as those in non-Euclidean spaces and large-scale matrix computations [11]. In addition, affinity propagation recursively passes messages in the form of factor graphs until the sum of similarities between each data point and the clustering center is minimized, making the sum of squared errors relatively low [12].

Despite the outstanding advantages of affinity propagation, some problems have been exposed in its applications and research process. First, message transmission between nodes is achieved by a similarity measure. Therefore, the setting of the similarity measure has a great impact on the performance of the algorithm [13]. However, Euclidean distance is limited by the dimensionality of the data. It is only applicable in cases where the sample attributes are independent of each other, and it performs poorly in cases of complex potential cluster distributions, such as situations with high-dimensional sparsity and nonlinearity. Thus, the applicability of the algorithm is limited [14]. Next, the preference needs to be input in advance, and the accuracy of manual adjustment directly affects the final clustering effect. Under this circumstance, since there is no linear relationship between the preference and the number of clusters corresponding to each other, it costs much time to find the optimal clustering result. Third, it easily falls into local optima and has difficulty guaranteeing convergence to the global optimum.

To address the shortcomings of affinity propagation, many scholars have proposed optimization solutions. These studies mainly focused on three aspects. The first was to adjust and improve the similarity measure so that the algorithm could adapt to data distributions in different spatiotemporal dimensional spaces. For example, Kun *et al.* [15] designed a comprehensive similarity measure that combined topological similarity with feature similarity to describe the feature information among nodes. [16] introduced a global kernel function with high generalization capability to define the similarity measure of AP clustering and enhance the learning capability of the global kernel. Taheri and Bouyer [17] proposed an adaptive similarity matrix improvement method to reflect the aggregation probabilities of data points using the centroids of the samples in the clusters, and this resulted in an accuracy improvement. Second, the data size was reduced and analyzed with respect to the time complexity problem. Two types of parallel architectures were proposed in a study by [18] to accelerate affinity propagation. Considering its large memory size and great computing capacity, the distributed system was chosen to minimize the global communication cost. Jiang *et al.* [19] proposed an adaptive AP algorithm, which continuously adjusts the threshold value according to the association results. The aims were to break through the spatial limitation problem and effectively reduce the clustering time. Li *et al.* [20] presented an adjustable preference affinity propagation (APAP). The operational efficiency was optimized by adjusting the preference during the iteration process. Third, affinity propagation has been extended for applications in various fields. For instance, fault diagnosis and treatment [21], face recognition [22], construction engineering [23] and radio communication [24] were all examined.

Clustering is an optimization problem. Furthermore, finding the best clusters is an NP problem [25]. In recent years, as one of the research hotspots in the field of optimization, metaheuristic algorithms have provided a novel research idea for clustering analysis. Compared with heuristic algorithms used for local search, metaheuristic strategies do not need to draw on the information of the target gradient function and the advantages of specific conditions, making them more applicable and general [26]. Due to their simplicity and ease of implementation, metaheuristic algorithms are usually able to efficiently explore the search space for approximate solutions. Therefore, it can be used to overcome the sensitivity of AP to the initial parameters and its difficulty in finding the best clusters [27]. On this basis, the clustering validity of the original algorithms can be optimized. Some scholars have conducted research and experiments. [28] applied an improved fruit fly optimization algorithm to optimally select the preference for affinity propagation. A novel adaptive affinity propagation based on improved cuckoo search (CAAP) was introduced in the literature [29] to adjust the preference and the damping factor. To enhance the ability of AP to respond to dynamic environments, Liu *et al.* [30] combined the particle swarm algorithm with the original algorithm. The robustness and stability of the combined algorithm were also improved. Hussain and Iqbal [31] clustered data by using a genetic algorithmic framework. The similarity function was defined as the differences between clusters, and the labels were adjusted by greedily optimizing the objective function. Combining quantum computing and ant colony optimization algorithms, [32] introduced quantum coding and quantum revolving gates to increase the population diversity, making the algorithm powerful in terms of global search. Wang *et al.* [33] conducted a study on semi-supervised affinity propagation (SAP). The idea of fireworks explosion was introduced to balance the global and local searching abilities of the algorithm. All the above algorithms have made certain improvements in avoiding local optima and improving the global search ability of AP. However, there was

still some room for enhancement with regard to the search efficiency of the algorithm.

Inspired by the hunting behavior of humpback whales, Mirjalili and Lewis [34] proposed the whale optimization algorithm (WOA) in 2016. The prey (the optimal solution) is hunted by simulating the humpback whale bubble-net strategy of circular contraction, random hunting, and updating the spiral position. The algorithm obtains the local optimal solution through its shrinking envelope mechanism and spiral update mechanism, and it acquires the global optimal solution through the random search mechanism. The hunting behavior is shown in Figure 1. Due to its fewer adjustment parameters (A and C) than other algorithms, ease of implementation, and applicability for solving various types of optimization problems, it has received wide attention [35]. Nevertheless, in the absence of any prior knowledge about the global optimal solution, WOA uses a stochastic method to initialize the population of individuals. In this case, useful information from the solution space is not guaranteed to be extracted effectively, and this may result in an uneven distribution of the initial population. Consequently, this affects the solution efficiency of the algorithm to some extent. In addition, like other meta-heuristic algorithms, the classic WOA also has shortcomings such as slow convergence and easy falling into local optima [36].

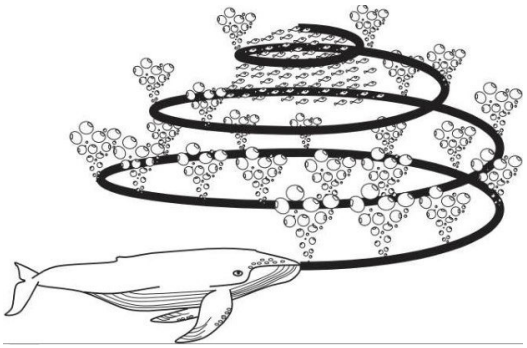


FIGURE 1. The hunting behavior of humpback whales [34].

Since being proposed, researchers have designed many variants and improvement schemes to solve the drawbacks of the WOA, such as its slow convergence and its tendency to fall into local optimality. The research has focused on three main areas: adding noise disturbances, improving the internal structure, and hybrid strategies. Noise disturbances have become a popular technique with the main purpose of preventing algorithms from falling into local optima and better balancing exploration and exploitation [37]. Among these methods, the most representative methods are Levy flight (LF) and the chaos strategy. For example, Liu *et al.* [38] proposed a hybrid WOA enhanced with Levy flight and differential evolution (WOA-LFDE), which enhanced the global search and iterative convergence and improved the diversity of the population. Luo *et al.* [39] introduced a chaos strategy to generate high-quality populations for initialization and improve the convergence efficiency. The second type of method enhances performance by improving the internal structure of the WOA. Fan *et al.* [40] presented a new adaptive inertia weight, which improved the convergence efficiency and the solution accuracy. Guo *et al.* [41] improved the position update strategy by calculating the adaptive neighborhood radius. Consequently, the search agents can be learned in their respective neighborhoods, replacing the original WOA in the random learning and enhancing the ability of the algorithm to jump out of local optima. Turgut *et al.* [42] introduced the concept of topology to maintain exploration and exploitation. On this basis, the exploitation of the fertile areas can be maintained through interaction between topological neighbors. The third type of method combines the WOA with other strategies or methods. Heidari *et al.* [43] developed a variant of WOA, which mixed WOA with the β-hill climbing (BHC) algorithm. The local search engine of the BHC algorithm was further exploited during the development phase. The purpose was to prevent the algorithm from falling into local optima and enhance its local search capabilities. Chakraborty *et al.* [44] presented an enhanced WOA method (WOAmM), which combined the WOA with symbiotic organisms search (SOS). The location update strategy of using randomly selected agents instead of the global best was introduced to improve the search accuracy and convergence performance. Ewees *et al.* [45] proposed a hybrid multiobjective optimization method based on differential evolution (DE) and the WOA. By exploring the search space extensively, further optimization was sought in the most salient regions. Reddy *et al.* [46] presented a hybrid social whale optimization algorithm. By combining the WOA with social group optimization (SGO), the runtime was shortened, and a balance between exploration and exploitation was achieved.

To address the above problems, this paper proposes affinity propagation based on an improved whale optimization algorithm (APBWOA). First, the weighted Mahalanobis distance is used instead of the Euclidean distance to improve the similarity measure in the original affinity propagation algorithm. Considering the importance of attributes, the entropy weighting method is introduced to the traditional Mahalanobis distance. The purpose of doing this is to overcome the defects of the Euclidean distance and adaptively adjust the geometric distribution of the given dataset. After that, a new algorithm based on the whale optimization algorithm is defined for automatically obtaining the optimal preference for affinity propagation. The new algorithm overcomes the problem of falling into local optima easily. In general, our work focuses on the following three aspects.

(1) We propose a whale optimization algorithm based on a chaotic map and nonlinear inertia weight improvement (CPIW-WOA). The improved circle map is used to optimize the initialization generation strategy, thereby enhancing the traversal of the initial population. In addition, the nonlinear inertia weight is introduced to improve the convergence efficiency of the algorithm. The effectiveness of the algorithm is verified on nine benchmark functions.

(2) Furthermore, the CPIW-WOA algorithm is used to perform a spatial search for the preference to address the limitation that the parameters of affinity propagation need to be input in advance. Additionally, the weighted Mahalanobis distance is defined to optimize the similarity measure. For overcoming the defect of the Mahalanobis distance, the entropy weight method is introduced to consider the importance of each attribute of the given sample. In addition, the silhouette index is used for validity evaluation. On this basis, the process of APBWOA is given.

(3) Finally, we explore the performance of APBWOA on an artificial dataset and the UCI dataset. The F-measure, NMI index and RI are applied to evaluate the clustering results compared to those generated by AP, APAP, SAP and CAAP. The experimental results show that the proposed algorithm has higher accuracy these methods. To better illustrate the superiority of APBOP in target grouping, a simulation scenario is designed for testing purposes. The simulation results show the correctness and effectiveness of the method.

II. BASIC THEORY

A. THE MODEL FOR TARGET GROUPING

Based on multisource information fusion, target clustering is a correlation and clustering analysis of the location, heading, movement status, task execution and other attributes of the target. In the field of battlefield target clustering, it is a key research component for excluding irrelevant and redundant information from the large amount of information known about the target. Next, an effective battlefield situation map can be formed to deepen the understanding and judgment of the battlefield situation. From the bottom up, the divisions of spatial, functional, interaction and enemy identification clusters are involved.

Assume a set of battlefield targets is denoted by K. $K = \{T_1, T_2 \ldots\ldots T_n\}$ consists of n objects. The attribute for each target is represented as $T_i (i = 1, 2 \ldots\ldots n)$. $T_i = \{T_{i1}, T_{i2} \ldots\ldots T_{im}\}$. T_{im} is the m_{th} characteristic information of the i_{th} target, including its batch number, speed, location, type, and identification friend or foe (IFF) system. A spatial group is a cluster obtained based on the overall structure and target location in a certain time and space range. By dividing targets into clusters, the distribution characteristics of the targets can be reflected. Therefore, members of the same spatial cluster have similar attributes, locations and behaviors. In detail, the set K is divided into n clusters, where the elements of any two clusters are not identical to each other. The result of spatial grouping is shown as follows:

$$S = S_1 \cup S_2 \cup S_3 \ldots\ldots \cup S_n \ \forall p, q \in [1, n], S_p \cap S_q \neq \emptyset, p \neq q \tag{1}$$

B. AFFINITY PROPAGATION

First, it is necessary to define the similarity between the given data points through the negative Euclidean distance measure. Assume a dataset $x_i = \{x_1, x_2 \ldots\ldots x_n\}$ consists of n objects.

$$S(i, j) = -\left\| x_i^2 - x_j^2 \right\| \quad i, j \in [1, n] \tag{2}$$

The similarity represents the degree to which a node serves as a cluster center. The greater the similarity is, the more likely it is to be selected as a clustering center. Unlike other algorithms, such as k-means, affinity propagation does not need to initialize the number of clusters and clustering centers. In contrast, it requires specifying the preference in advance, and it is usually set to a constant.

The algorithm conveys information by calculating the responsibility and availability of each node. The responsibility represents the attractiveness of node j as a cluster center to node i. The degree of availability represents the possibility of node i selecting node j as a cluster center. Both are calculated as follows:

$$r(i, j) = \begin{cases} S(i, j) - \max\limits_{j \neq k} \{a(i, k) + S(i, k)\} & (i \neq j) \\ S(i, j) - \max\limits_{j \neq k} \{S(i, k)\} & (i = j) \end{cases} \tag{3}$$

$$a(i, j) = \begin{cases} \min\{0, r(j, j) + \sum\limits_{k \neq i, j} \max[0, r(k, j)]\} & (i \neq j) \\ \sum\limits_{k \neq j} \max[0, r(k, j)]\} & (i = j) \end{cases} \tag{4}$$

In the iterative process, a damping factor λ, which acts on the responsibility and availability values, is introduced to avoid oscillations. The smaller the value is, the stronger the global search capability of affinity propagation and the faster the iteration speed. The iterative process is shown in Equations 5 and 6:

$$r_t(i, j) = (1 - \lambda) r_t(i, j) + \lambda r_{t-1}(i, j) \tag{5}$$

$$a_t(i, j) = (1 - \lambda) a_t(i, j) + \lambda a_{t-1}(i, j) \tag{6}$$

Finally, the algorithm selects the largest node as the clustering center through continuous iteration, as shown in Equation 7.

$$c_i = arg \max_j \{r(i, j) + a(i, j)\} \tag{7}$$

C. WHALE OPTIMIZATION ALGORITHM

In this section, we focus on the model and workflow of the WOA. The algorithm consists of three parts: an initialization phase, a local search phase and a global search phase.

1) INITIALIZATION PHASE

Assume that there are N humpback whales in the D-dimensional search space. The position of the k_{th} whale in the space is k_{th}. $X_k = \{X_k^1, X_k^2 \ldots\ldots X_k^D\} (k = 1, 2 \ldots\ldots N)$. Therefore, the definition of the initialized population is as shown in Equation 8.

$$X_k^m(0) = l^m + rand(0, 1)(u^m - l^m) \quad (1 \leq m \leq D) \tag{8}$$

where u^m and l^m represent the upper and lower bounds of the m_{th} dimension, respectively. The initial population $P(0) = \{X_1(0), X_2(0) \ldots\ldots X_k(0)\}$.

2) LOCAL SEARCH PHASE

The local search phase consists of two parts: the contracting envelope mechanism and the spiral renewal mechanism. First, the humpback whale population encircles the target when the prey is locked. Since the global optimal solution is not known a priori, the current optimal position of the humpback whale population is set as the position of the target prey in the WOA algorithm. After the n_{th} iteration, the position of the target solution is as follows:

$$X_n^d = \{X_n^1, X_n^2 \ldots\ldots X_n^D\} \quad (n = 1, 2 \ldots\ldots t_{\max}) \tag{9}$$

where $t_{\max}$ denotes the maximum number of iterations. Other individuals adjust their own positions according to the position of the target prey. The update equation for this process is shown in Equation 10.

$$X(t+1) = X'(t) - A * d(t) \tag{10}$$

In the above equation, t represents the number of iterations. The position of the optimal solution for the current population is X'. X is the current position of the individual. The covariates A and $d(t)$ control the step size, which is determined by Equations 11 and 12.

$$A = 2 \cdot a \cdot r - a \tag{11}$$

$$d(t) = |C \cdot X'(t) - X(t)| \tag{12}$$

During the iterative process, r is a random number between 0 and 1, while C is a random number between 0 and 2. a decreases linearly from 2 to 0, and this can be expressed as:

$$a = 2 - 2 \cdot \frac{t}{t_{\max}} \tag{13}$$

By adjusting parameters A and C, the humpback whales can be controlled to search for the current prey. At the same time, the range of the envelope can be continuously reduced by decreasing the value of a. When $|A| \leq 1$, the humpback whales in the group always move within the envelope to complete the envelope surrounding the prey and search for the local optimal solution. Second, humpback whales spiral upward to hunt while decreasing the circle. The equation for this spiral update is as follows:

$$X(t+1) = D' \cdot e^{bl} \cdot \cos(2\pi l) + X'(t) \tag{14}$$

$$D' = \left|X'(t) - X(t)\right| \tag{15}$$

In this process, D' denotes the distance between the current individual and the optimal individual. b is a constant that defines the logarithmic spiral shape. l is a random number between -1 and 1. To simulate the humpback whale predation behavior, the WOA algorithm uses a coin toss to select the chosen humpback whale behavior. In other words, the individuals have the same chance of contracting the envelope and updating the spiral, both of which are 0.5, as shown in Equation 16.

$$X(t+1) = \begin{cases} X'(t) - A * d(t) & (p < 0.5) \\ D' \cdot e^{bl} \cdot \cos(2\pi l) + X'(t)] & (p > 0.5) \end{cases} \tag{16}$$

3) GLOBAL SEARCH PHASE

When $|A| > 1$, the humpback whale leaves the current enclosure and conducts a random search outside the enclosure. Unlike in the enclosure search stage, the position of the best individual is no longer used as a reference at this time; instead, a random selection is made in the group. At this time, $d(t)$ represents the absolute value of the distance between the current individual and the randomly selected individual. The equations are shown as follows:

$$X(t+1) = X''(t) - A * d(t) \tag{17}$$

$$d(t) = |C \cdot X''(t) - X(t)| \tag{18}$$

III. THE PROPOSED ALGORITHM

It can be found that in the traditional AP algorithm, there are two main shortcomings that limit its clustering performance. First, Euclidean distance is used to define the node similarity, and this measure is easily affected by dimensionality. Furthermore, the weighting relationships of sample attributes are ignored. Thus, the final result may be biased compared with the actual situation. Second, the preference and the damping factor in affinity propagation need to be set manually in advance. The sizes of the parameters, which need to be constantly and dynamically adjusted according to the actual situation, directly affect the number of clusters generated. To deal with these problems, APBWOA is proposed. First, the entropy weight method is introduced to improve the Mahalanobis distance and use it as the similarity measure. Furthermore, a whale optimization algorithm based on the chaotic circle map and nonlinear inertia weight improvement (CPIW-WOA) is defined to globally seek the preference in affinity propagation. Based on that, the problem of manually entering parameters in affinity propagation can be solved effectively. Finally, the silhouette index is used as the fitness function to compare the clustering results obtained through the use of different parameters.

A. WEIGHTED MAHALANOBIS DISTANCE

In the process of clustering analysis, the definition of the similarity measure or distance directly affects the final clustering results [47]. In previous studies, similarity calculations have generally been performed in two ways. The first is to perform feature projection on all objects, as this approach visually reflects the association relationships through images. The second is to calculate the distances, which reflect the similarity relationships between objects by measuring the differences between them [48]. In the traditional AP algorithm, the similarity between two points is measured by calculating the Euclidean distance. If the distance between two points is lower, the similarity is greater. Although this method is simple and easy to calculate, it only takes the characteristics of the samples themselves into account and is easily influenced by their magnitudes. In addition, the correlations between different samples are ignored. Under this circumstance, the relative importance degree of each sample attribute is not taken seriously.

Compared with the Euclidean distance, the Mahalanobis distance [49] is independent of the magnitude and fully accounts for the degrees of association between attributes by normalizing the variance. Moreover, the Mahalanobis distance can adjust the data distribution to make the relationships between features more realistic so that the data clustering process is more adaptive and different cluster shapes can be identified.

For a single data point, the expression for the Mahalanobis distance is shown in Equation 19.

$$D_M x) = \sqrt{(x-\mu)^T S^{-1}(x-\mu)} \tag{19}$$

where $x = (x_1, x_2 \ldots\ldots x_n)^T$ denotes the vector of sample data and $\mu = (\mu_1, \mu_2 \ldots\ldots \mu_n)^T$ represents the means of the samples. S is the covariance matrix. For two different data samples, the Mahalanobis distance is shown in Equation 20:

$$D_M x) = \sqrt{(x-y)^T S^{-1}(x-y)} \tag{20}$$

However, the covariance matrix usually represents the overall sample covariance matrix in the Mahalanobis distance. Therefore, it cannot distinguish the relative importance values of the sample categories. For this reason, the entropy weighting method is considered for introduction into the Mahalanobis distance. The entropy weighting method is an objective assignment method that judges the dispersion of an attribute based on its entropy value. The lower the weight is, the greater the information entropy of the attribute, indicating that the degree of dispersion of the attribute is smaller and that its role in the evaluation process is smaller. Based on this, the entropy weight method can reflect the importance of the attribute. Assume there are m attributes in the n_{th} sample. d_{ij} denotes the j_{th} attribute value of the i_{th} sample. The weight is calculated as:

$$p_{ij} = \frac{d_{ij}}{\sum_{i=1}^{n} d_{ij}} \tag{21}$$

The entropy of the j_{th} attribute is shown as follows:

$$E_j = -\frac{1}{\ln n} \sum_{i=1}^{n} p_{ij} \ln p_{ij} \tag{22}$$

Then, the weight of the j_{th} attribute is expressed by Equation 23:

$$\omega_j = \frac{1 - E_j}{\sum_{j=1}^{m} (1 - E_j)} \tag{23}$$

The weight $\omega_j (j = 1, 2 \ldots\ldots n)$ should satisfy the following equation.

$$\sum_{j=1}^{n} \omega_j = 1 \tag{24}$$

Thus, the weighted Mahalanobis distance is expressed as:

$$d(x, y) = \sqrt{(x-y)^T \beta^T S^{-1} \beta (x-y)} \tag{25}$$

where β denotes the weight matrix, $\beta = diag(\sqrt{\omega_1}, \sqrt{\omega_2} \ldots \ldots \sqrt{\omega_n})$. ω_j is the weight of each attribute obtained based on the entropy weighting method. The covariance matrix is linearly transformed, and the weighted martingale distance can be defined as:

$$\begin{aligned} d(x, y) &= \sqrt{(x-y)^T \beta^T S^{-1} \beta (x-y)} \\ &= \sqrt{(x-y)^T \beta^T S^{-\frac{1}{2}} S^{-\frac{1}{2}} \beta (x-y)} \\ &= \sqrt{\beta^T S^{-\frac{1}{2}} (x-y)^T} \cdot \sqrt{\beta S^{-\frac{1}{2}} (x-y)} \\ &= \sqrt{\beta^T U \Lambda^{-\frac{1}{2}} U^T (x-y)^T} \cdot \sqrt{\beta U \Lambda^{-\frac{1}{2}} U^T (x-y)} \end{aligned} \tag{26}$$

where $\lambda_1, \lambda_2 \ldots\ldots \lambda_n$ represents the eigenvalues of the sample covariance matrix. $\Lambda = diag(\lambda_1, \lambda_2 \ldots\ldots \lambda_n)$. U is a matrix composed of unitary eigenvectors, $U = (U_1, U_2 \ldots\ldots U_n)$.

B. THE WHALE OPTIMIZATION ALGORITHM BASED ON A CHAOTIC MAP AND NONLINEAR INERTIA WEIGHT IMPROVEMENT

Compared with other heuristic algorithms, the whale optimization algorithm is characterized by fewer adjustment parameters and greater simplicity, but there are two aspects that affect the improvement in the convergence speed of the algorithm. First, the process for generating the initialized population in this algorithm completely relies on random probability and lacks traversal, so it cannot effectively guarantee a uniform distribution for the population in the search space. Consequently, the convergence efficiency and accuracy of the algorithm are affected to a certain extent. Second, the algorithmic process relies on the convergence factor alone. Hence, it is difficult to achieve a balance between local search capabilities and global search capabilities. Therefore, CPIW-WOA is proposed to overcome these defects. First, the initialized population is generated optimally using an improved circle map from chaos theory to improve the traversal and non-repeatability of the search space. Second, a nonlinear inertia weight is introduced to enhance the search accuracy and balance the local search ability and global exploitation ability of the algorithm.

1) INITIALIZATION STRATEGY

In metaheuristic algorithms, the distribution of the initialized population largely affects the convergence rate and accuracy of the algorithm [50]. In the WOA, the population is initialized by means of a random probability distribution. Although simple and easy to implement, it lacks traversal and may result in an excessive concentration or loss of values, thereby reducing the convergence speed [51]. Compared to random search, which relies mainly on probability distributions, chaos maps allow for searching the space with a higher probability and convergence rate, as well as with higher ergodicity and sensitivity [52]. Research has found that compared with other maps, circle maps have superior exploration

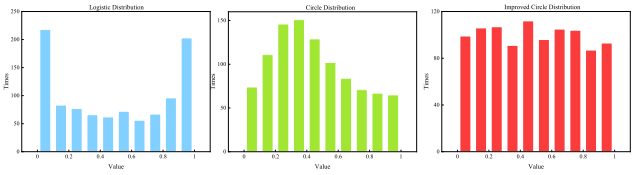
ability and can significantly improve the convergence rate of the algorithm [53]. Therefore, the improved circle map is introduced to improve the initialization method in this paper. In the solution space, the circle map can be expressed as:

$$z_{n+1} = z_n + b - (\frac{a}{2\pi})\sin(2\pi z_n) \bmod (1), z_n \in (0, 1) \quad (27)$$

To further enhance the ergodicity and distribution ranges of chaotic sequences, we have made some improvements. The equation of the improved circle map is shown in Equation 28.

$$z_{n+1} = 2z_n + b - (\frac{a}{4\pi})\sin(2\pi z_n) \bmod (1), z_n \in (0, 1) \quad (28)$$

where $a = 0.5$ and $b = 0.2$. With the same initial independent variables, the logistic map, circle map and the improved circle map are executed 1000 times. The results are shown in Figure 2.



(a) (b) (c)

FIGURE 2. Comparison between the obtained distributions. (a) Logistic map. (b) circle map. (c) The improved circle map.

As seen in Figure 2, the traversal of the improved circle map is wider, more nonrepeatable, and more uniformly distributed in the space of [0,1] than those of the logistic map and circle map. Therefore, the improved circle map is used to initialize the population in this paper. The initialized population is generated using chaotic variables and mapped to (0,1) in the following way.

$$X_k^m(0) = l^m + z^m(t)(u^m - l^m) \quad (1 \le m \le D) \quad (29)$$

2) NONLINEAR INERTIA WEIGHTS

In the WOA, when the local optimal solution is obtained, all other individuals in the space converge to it. Consequently, the chance of finding other solutions is reduced, while the algorithm may converge to the local optimal solution prematurely. Dealing with this problem, Shi *et al.* introduced inertia weights into the particle swarm algorithm. By controlling the sizes of the inertia weights, a balance between the global search ability and local search ability of the algorithm can be achieved [54]. To improve the convergence accuracy and convergence rate, a nonlinear inertia weight is introduced in this paper as follows:

$$\omega = 0.01^{e^{(1-\frac{t_{max}+t}{t_{max}-t})}} \quad (30)$$

where $t \in [0, t_{max})$. In the early iterations of the global search, the inertia weight is large, the rate of nonlinear change is fast, and the global search ability is strong. The purpose of this is to avoid falling into a local optimum, as doing so affects the convergence accuracy of the algorithm. In the later iterations, the nonlinear change rate is smaller and slower, so the optimal solution is sought in a certain region. Furthermore, the development ability is strong, and the convergence rate is accelerated. By introducing nonlinear inertia weights, effective control over the positions of the particles can be achieved, thus improving the optimization-seeking accuracy of the algorithm. The improved position update formula is shown in Equations 31 to 33.

$$X(t+1) = \omega * X'(t) - A * d(t) \quad (p < 0.5 \ \ and \ \ |A| \le 1) \quad (31)$$

$$X(t+1) = \omega * X''(t) - A * d(t) \quad (p < 0.5 \ \ and \ \ |A| > 1) \quad (32)$$

$$X(t+1) = D' \cdot e^{bl} \cdot \cos(2\pi l) + \omega * X'(t) \quad (p > 0.5) \quad (33)$$

TABLE 1. Details of nine benchmark functions.

Function	Dimension	Range	f_{min}
$f_1(x) = \sum_{i=1}^{n} x_i^2$	30	[-100,100]	0
$f_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	[-10,10]	0
$f_3(x) = \sum_{i=1}^{n} ix_i^4 + random[0, 1)$	30	[-1.28,1.28]	0
$f_4(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	30	[-5.12,5.12]	0
$f_5(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}}) + 1$	30	[-600,600]	0
$f_6(x) = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) - \exp(\frac{1}{n}\sum_{i=1}^{n} \cos(2\pi x_i)) + 20 + e$	30	[-32,32]	0
$f_7(x) = \sum_{i=1}^{11} [a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}]^2$	4	[-5,5]	0.00030
$f_8(x) = -\sum_{i=1}^{4} c_i \exp(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2)$	6	[0,1]	-3.32
$f_9(x) = -\sum_{i=1}^{7} [(X - a_i)(X - a_i)^T + C_i]^{-1}$	4	[0,10]	-10.4028

3) VALIDATION CERTIFICATION

In this section, we verify the effectiveness of CPIW-WOA. To show its superiority in terms of performance, the whale algorithm (WOA) [34], grey wolf algorithm (GWO) [55] and whale optimization algorithm using OBL (OBWOA) [56] are selected. In addition, nine classic benchmark functions are used for testing, all of which are all from the literature [34]. The definitions, dimension settings, upper and lower bounds and reference function minima of these functions are listed in Table 1. For the unimodal functions ($f_1(x)$ to $f_3(x)$), there is only one extreme point in the given search space, and it can be used to detect the convergence rate and the accuracy of the algorithm. $f_4(x)$ to $f_6(x)$ are multimodal functions with multiple local extreme points in the given search space. Therefore, they can be used to evaluate the global exploitation ability. Additionally, the ability of each algorithm to jump out of the local optimum can also be evaluated. Finally, $f_7(x)$ to $f_9(x)$ are composite functions, which are used to study the ability to balance exploration and exploitation. The simulation experiment is implemented on an Intel(R) Core (TM) i5-9300H CPU @ 2.40 GHz with 16 GB of memory and the Windows 10 operating system, and it is programmed in MATLAB R2019b.

TABLE 2. Parameter settings of different algorithms.

Algorithm	Parameter	Range
Whale optimization algorithm (WOA)	a	(2,0)
	a_2	(2,1)
	b	1
Grey wolf optimizer (GWO)	a	(2,0)
	$\vec{A}$	[-a,a]
	$\vec{C}$	[0,2]
Whale optimization algorithm using opposition-based learning (OBWOA)	a	(2,0)
	a_2	(2,1)
	b	1
Whale optimization algorithm based on a chaotic map and nonlinear inertia weight improvement (CPIW-WOA)	a	(2,0)
	a_2	(2,1)
	b	1

TABLE 3. Comparison of the results obtained by different algorithms.

F		WOA	GWO	OBWOA	CPIW-WOA
$f_1(x)$	**Ave**	1.30E-151	4.33E-59	9.01E-91	0.00E+00
	Std	6.62E-151	1.06E-58	4.84E-90	0.00E+00
$f_2(x)$	**Ave**	7.45E-105	7.35E-35	7.97E-94	8.99E-284
	Std	3.00E-104	7.18E-35	3.35E-93	0.00E+00
$f_3(x)$	**Ave**	1.69E-03	2.76E-03	4.57E-03	4.34E-05
	Std	1.99E-03	1.49E-03	2.58E-03	3.63E-05
$f_4(x)$	**Ave**	0.00E+00	4.30E-01	3.79E-15	0.00E+00
	Std	0.00E+00	1.42E+00	2.08E-14	0.00E+00
$f_5(x)$	**Ave**	1.81E-02	2.19E-03	1.04E-01	0.00E+00
	Std	3.71E-02	5.96E-03	2.73E-01	0.00E+00
$f_6(x)$	**Ave**	3.85E-15	1.65E-14	5.15E-15	8.88E-16
	Std	2.48E-15	3.56E-15	3.15E-15	0.00E+00
$f_7(x)$	**Ave**	5.92E-04	4.38E-03	1.79E-03	3.27E-04
	Std	4.12E-04	8.13E-03	3.59E-03	5.51E-05
$f_8(x)$	**Ave**	-3.22E+00	-3.27E+00	-3.07E+00	-3.29E+00
	Std	1.70E-01	6.42E-02	1.83E-01	5.34E-02
$f_9(x)$	**Ave**	-8.61E+00	-1.04E+01	-5.28E+00	-1.04E+01
	Std	3.08E+00	3.27E-04	2.62E+00	2.33E-08

To make the experiment fairer, the parameters of WOA, GWO, OBWOA and CPIW-WOA were set in compliance with the original literature, which are shown in Table 2. The parameters are kept consistent except for those listed in the table. The population size *SearchAgents_no* of WOA, GWO, and CPIW-WOA is set to 30, while the parameter is set to 15 in OBWOA. The maximum number of iterations *Max_iter* is 1000. The other parameters of the algorithms are set identically. All algorithms are run 30 times, and each time, the generated initialized populations are different. The dimensions of the benchmark functions $f_1(x)$ to $f_6(x)$ are all 30. Table 3 shows the means (Ave) and standard deviations (Std) of the algorithms.

From the results shown in Table 3, it can be found that CPIW-WOA achieves the theoretical optimal values for $f_1(x)$, $f_4(x)$ and $f_5(x)$. Compared with other algorithms, CPIW-WOA achieves the closest theoretical optima for other functions. The accuracy of the obtained solution is greatly improved on $f_2(x)$ and $f_6(x)$, and the results are stable and unchanged. This demonstrates that CPIW-WOA has good stability. Additionally, the WOA obtains the theoretical optimum for the function $f_4(x)$ and ranks second in terms of performance with $f_1(x)$, $f_2(x)$, and $f_7(x)$. GWO has better stability in solving functions $f_8(x)$ and $f_9(x)$. Overall, CPIW-WOA has the best performance, and OBWOA has the worst performance.

The above results demonstrate the superiority of the CPIW-WOA algorithm. In comparison with other algorithms, CPIW-WOA has a competitive advantage with regard to the mean and standard deviation performances. Among the four unimodal functions, CPIW-WOA obtains the theoretical optima in two of them, indicating that the algorithm has excellent accuracy in terms of finding the optima. For the multimodal functions, CPIW-WOA finds one of the global optimal solutions. CPIW-WOA also outperforms the other algorithms for the remaining three test functions. Consequently, this indicates that the proposed algorithm has superior capabilities for jumping out of local optima. Additionally, the global search capability is improved. Furthermore, CPIW-WOA has the best robustness and the most stable results when solving the composite functions, indicating that the algorithm can achieve a balance between exploration and exploitation.

To further test the convergence performances of the selected algorithms, Figure 3 shows the comparison results obtained by different algorithms on nine benchmark functions. It is obvious that CPIW-WOA has a faster convergence rate than those of the other algorithms. Since the introduction of nonlinear inertia weights significantly improves the convergence trend of the original algorithm, CPIW-WOA can be judged to have better convergence efficiency than the compared approaches.

C. FITNESS FUNCTION

Clustering validity testing refers to the evaluation of clustering results. The aim is to determine the most suitable partition for a particular dataset. To perform validity tests on clustering results, an adaptation function is required. In this paper, we select the silhouette index to evaluate the clustering results. The silhouette reflects the intraclass compactness and interclass separability of the clustering structure, and it has been used to evaluate many clustering algorithms [57]. In addition, the optimal number of clusters can be estimated. The definition of the silhouette index is shown as follows:

$$sil(i) = \frac{[b(i) - a(i)]}{\max\{a(i), b(i)\}} \tag{34}$$

Let a dataset be divided into N clusters $C_i (i = 1, 2 \ldots\ldots N)$. $a(t)$ is the average dissimilarity or distance between sample point t and other points in C_i. $d(t, C_j)$ is the average dissimilarity or distance from sample point t in cluster C_i to all samples in another cluster C_j. $b(t) = \min\{d(t, C_j)\}$. After that, the average of all the sample silhouette indices is calculated as the final result, as shown in Equation 35.

$$Sil = \overline{sil(i)} = \frac{1}{N}\sum_{i=1}^{N} sil(i) \tag{35}$$

Based on the quality of clustering, the silhouette index of each sample varies between [-1,1]. In general, the larger the silhouette value is, the better the clustering quality is.

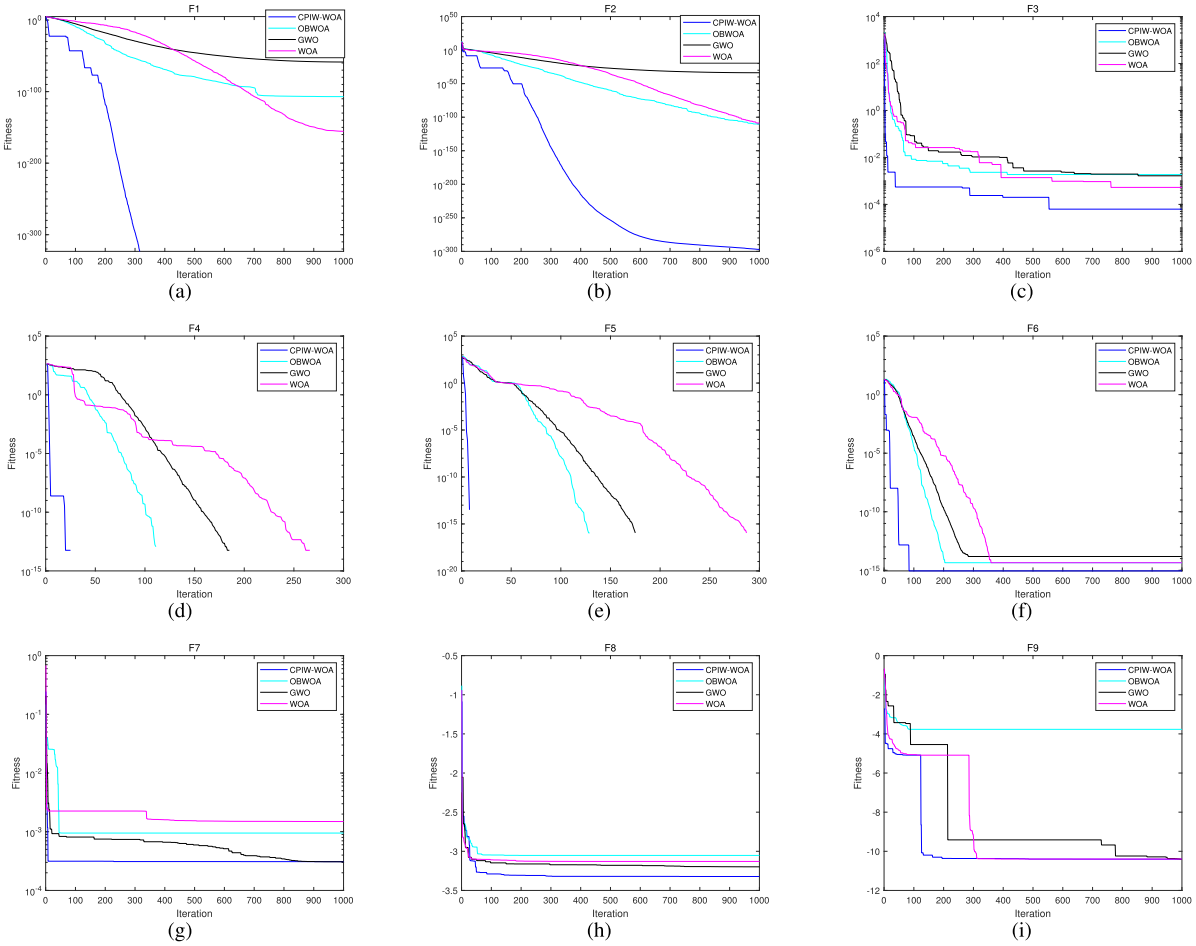


FIGURE 3. Convergence test results with different benchmark functions.

D. WORKFLOW

In APBWOA, we first improve the similarity metric by using the weighted Mahalanobis distance. Second, aiming to conduct a comprehensive and effective search for the preference, CPIW-WOA is proposed to adjust the preference adaptively. By introducing an improved circle mapping and a nonlinear inertia weight, the convergence efficiency and the search quality of the algorithm are enhanced. The workflow of APBWOA is given in Figure 4. In addition, the pseudocode of the standard APBWOA is shown in Algorithm 1.

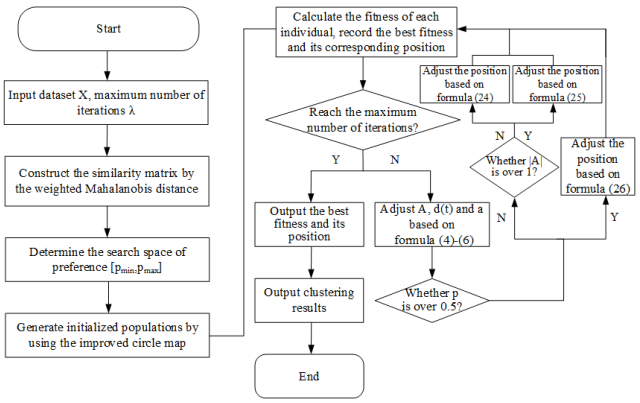


FIGURE 4. The workflow of APBWOA.

IV. EXPERIMENTS AND ANALYSIS

A. EXPERIMENTS ON STANDARD DATASETS

Furthermore, to verify the feasibility and effectiveness of APBWOA, artificial datasets and UCI datasets are selected for simulation experiments. The real-world datasets are from the University of California, Irvine (UCI) machine learning repository [58], including Iris, Wine, Seeds, and ionosphere. The title, numbers, attributes and clusters of the datasets are shown in Table 4. In addition, affinity propagation (AP) [9], adjustable preference affinity propagation (APAP) [20], semi-supervised affinity propagation (SAP) [33] and a novel adaptive affinity propagation based on improved cuckoo search (CAAP) [29] are selected for comparison with APBWOA. The simulation environment for experimentation is an Intel(R) Core (TM) i5-9300H CPU @ 2.40 GHz with 16 GB of RAM and the Windows 10 operating system. The simulation software is the MATLAB R2019b experimental platform.

Algorithm 1 The Pseudo Code of APBWOA

Input:

Dataset $C = \{C_1, C_2, \ldots\ldots C_i\}(i = 1, 2\ldots\ldots N)$, the maximum number of iterations $t_{\max}$, the damping factor λ

Output:

The result of clustering $C_s = \{C_1, C_2, \ldots\ldots C_K\}$

Initialization: normalize the input data to eliminate the influences of different dimensions
Construct the matrix of similarities $[S_{ij}]_{n\times n}$ between all pairs of data points by the weighted Mahalanobis distance
Determine the sample space of the preference $[p_{\min}, p_{\max}]$
while t=1:$t_{\max}$ **do**
  Generate initialized populations by Equation 29
  Update a based on Equation 13
  Update A and $d(t)$ based on Equations 11 and 12
  if $p < 0.5$ **then**
    if $|A| \leq 1$ **then**
      Update the current position by Equation 31
    else if $|A| > 1$ **then**
      Update the current position by Equation 32
    end if
  else if $p \geq 0.5$ **then**
    Update the current position by Equation 33
  end if
  Calculate and update the fitness and corresponding position if there is a better solution
  t=t+1
end while
Regard the best position as the preference
Execute the procedure of affinity propagation
return C_s

TABLE 4. The basic information about the selected datasets.

Title	Number	Attribute	Cluster
Two-moon	500	2	2
Aggregation	400	2	4
Iris	150	3	4
Wine	178	3	13
Seeds	210	3	7
ionosphere	351	34	2

1) EXPERIMENT SETTINGS

To shorten the running time and reduce the number of oscillations, it is necessary to specify the search space of the preference. A previous study [58] has shown that when $p = \frac{p_m}{2}$, the number of clusters $N \geq \sqrt{n}$, where p_m represents the median of the similarity between all samples and n represents the sizes of the samples. Therefore, the search space of the preference is defined as $[p_{\min}, \frac{p_m}{2}]$. In addition, the damping factor λ is set to 0.5. The maximum number of iterations *Max_iteration* is 500. The individual size *SearchAgents_no* is 30. For affinity propagation, the preference is set to p_m. The maximum number of iterations is 1000, and the number of successive convergences is 100.

2) EVALUATION INDICATORS

To further evaluate the efficiency of the selected algorithms, the F-measure (F), normalized mutual information (NMI) index and Rand index (RI) are used as the evaluation criteria.

The F-measure is an external evaluation indicator expressed in terms of both precision and recall, and it reflects the recognition effect of the clustering method in each cluster. Based on the known external classification criteria for measuring the clustering effectiveness, the F-measure can be calculated as shown in Equation 36.

$$F = \frac{2*P*R}{P+R} \tag{36}$$

where P stands for *precision*. $precision = \frac{TP}{TP+FP}$. R represents *recall*. $recall = \frac{TP}{TP+FN}$.

The NMI index evaluates the clustering results against the standard results based on mutual information. Generally, the larger the NMI value is, the more similar the two clustering results are. The definition of the NMI index is shown as follows:

$$NMI(X; Y) = \frac{2I(X; Y)}{H(X) + H(Y)} \tag{37}$$

where $I(X; Y)$ denotes the mutual information, which is a measure in information theory that can be regarded as the amount of information contained in one variable about another variable.

$$I(X; Y) = \sum_{n=1}^{K_1}\sum_{m=1}^{K_2} \frac{P_{nm}^{12}}{N} \log\left(\frac{\frac{P_{nm}^{12}}{N}}{\frac{p_n^1}{N} \times \frac{p_m^2}{N}}\right) \tag{38}$$

P_{nm}^{12} is the number of samples in which the n_{th} cluster of p^1 is the same as the m_{th} subset of p^2. $H(X)$ and $H(Y)$ represent the information entropy of clusters X and Y, respectively.

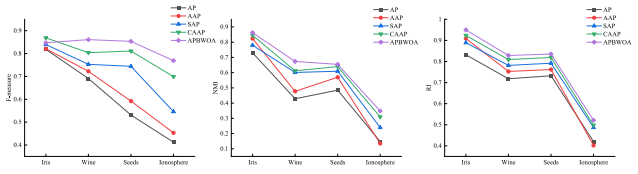
$$H(X) = -\sum_{n=1}^{K} \frac{p_n}{N} \log\left(\frac{p_n}{N}\right) \tag{39}$$

where K represents the number of clusters, N contains the sizes of the samples, and p_n stands for the number of samples in the n_{th} cluster.

The Rand index evaluates the clustering results by using the principles of permutation and combination. The larger the RI value is, the better the clustering effect is.

$$RI = 2\left[\frac{f_{00} + f_{11}}{N(N-1)}\right] \tag{40}$$

f_{00} is the number of data points with different labels that belong to different clusters. f_{11} is the number of data points with the same labels that belong to the same cluster.

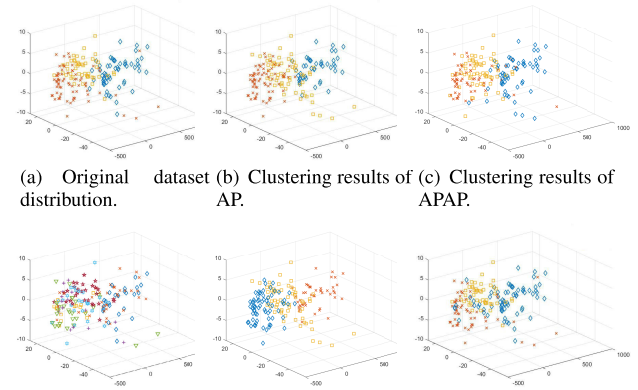


(a) Comparison results in terms of the F-measure. (b) Comparison results in terms of the NMI index. (c) Comparison results in terms of the RI.

FIGURE 5. Comparison results of four algorithms.

3) RESULTS AND ANALYSIS

In the experiments, the clustering results are evaluated with reference to the real divisions. Figure 5 shows the comparison results of different algorithms on the five UCI datasets. On the Iris dataset, the four algorithms do not differ much from each other. This is because the Iris dataset has a good clustering structure, so the traditional algorithms can also achieve good results. When dealing with the latter three datasets, APBWOA can handle data points that cannot be distinguished by Euclidean distance due to the improvement of the similarity measure. The results indicate that APBWOA has the feature of adapting to complex structured datasets, while the accuracy of the algorithm is also enhanced.

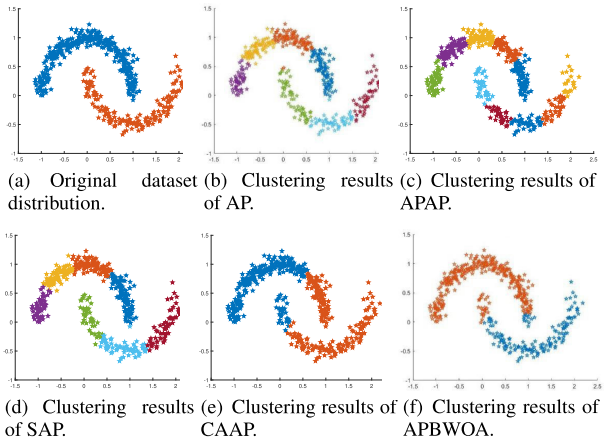


(a) Original dataset distribution. (b) Clustering results of AP. (c) Clustering results of APAP.

(d) Clustering results of SAP. (e) Clustering results of CAAP. (f) Clustering results of APBWOA.

FIGURE 6. Comparison results on the Wine dataset.

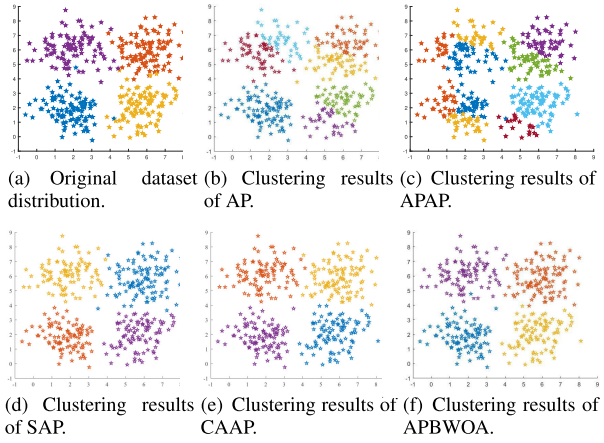
There are 178 samples with thirteen dimensions in the Wine dataset. The samples can be divided into three categories, with the number of samples in the categories being 59, 71, and 48. PCA is applied to reduce the dataset to three dimensions, and the samples are represented by different shapes. The visualization results are shown in Figure 6. The figure shows that affinity propagation classifies the lower three and the two upper left samples belonging to the second class into the third class when processing the outlier points. Obviously, the APAP, SAP, and CAAP cannot handle outliers effectively either. However, the APBWOA correctly classifies the points, which indicates that the algorithm has certain robustness and noise immunity. In addition, when processing the linearly inseparable data points, the AP algorithm classifies 13 points belonging to the third class into the first class and 18 points belonging to the second class into the third class. Furthermore, the APAP misclassified 11 points in the third class and 10 points in the other clusters. The SAP produced up to 7 redundant clusters. The results show that both the CAAP and APBWOA have fewer errors. In comparison, the APBWOA incorrectly classifies only 12 samples belonging to the second class into the first class and correctly classifies those in the third class. The performance results indicate that the APBWOA has a superior ability to handle the analysis of linearly inseparable problems.



(a) Original dataset distribution. (b) Clustering results of AP. (c) Clustering results of APAP.

(d) Clustering results of SAP. (e) Clustering results of CAAP. (f) Clustering results of APBWOA.

FIGURE 7. Comparison results on the two-moon dataset.



(a) Original dataset distribution. (b) Clustering results of AP. (c) Clustering results of APAP.

(d) Clustering results of SAP. (e) Clustering results of CAAP. (f) Clustering results of APBWOA.

FIGURE 8. Comparison results on the aggregation dataset.

Among the artificial datasets, the two-moon dataset contains 500 data points randomly generated by two semicircles. There are two diverse clusters, and the distribution is shown in Figure 7(a). In the aggregation dataset, there are 400 data points randomly generated by four mixed Gaussian distributions, and these can be seen in Figure 8(a). From the clustering results, it is obvious that the traditional AP algorithm performs poorly in handling such density-sensitive data distributions, which can be seen in Figure 7(b) and 8(b). Consequently, the AP algorithm generates more clusters than are present in the actual situation. The reason for this is that

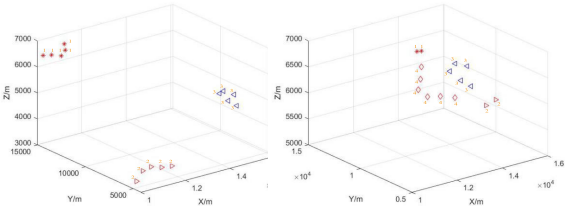
the Euclidean distance measure does not satisfy the global consistency requirement for cluster partitioning, resulting in the failure of the AP algorithm to give accurate results when handing complex clusters. For this reason, the Euclidean distance-based APAP, SAP, and CAAP also did not achieve satisfactory results on the two-moon dataset, which can be seen in Figures 7(c), 7(d) and 7(e). Both the APAP and SAP produce up to seven redundant clusters. Furthermore, the CAAP is able to find the optimal preference via optimization, but it cannot accurately reflect the true flow structure between data points. In contrast, the APBWOA uses the weighted Mahalanobis distance to quantify node similarity, and this measure has higher accuracy for identification purposes. On the aggregation dataset, the SAP, CAAP, and APBWOA all obtained better results, which is clearly reflected in Figures 8(c), 8(d) and 8(e). However, after further comparison, the APBWOA is better at handling outlier points. As observed in Figure 8(f), the proposed algorithm clearly divides the dataset into four classes, and the clustering result for the boundary points is basically consistent with the original dataset divisions. On this basis, the stability and accuracy of the APBWOA in handling complex structures are verified.

TABLE 5. The target status information.

Batch	X/m	Y/m	Z/m	Speed	Course	Party
1	13952	13120	6503	996	120	A
2	14191	13202	6478	945	120	A
3	13281	11251	6456	973	120	A
4	12805	10312	6359	952	120	A
5	12312	9423	6302	956	120	A
6	12401	8733	6212	973	60	A
7	12626	8007	6268	947	60	A
8	12925	7256	6271	967	60	A
9	13855	6299	6103	932	60	A
10	14167	6026	6201	934	60	A
11	14013	9013	6315	912	270	B
12	14024	9612	6583	914	270	B
13	14030	10145	6387	930	270	B
14	14048	8501	6632	960	270	B
15	14028	8090	6286	927	270	B

B. EXPERIMENTS ON A SIMULATED SCENARIO

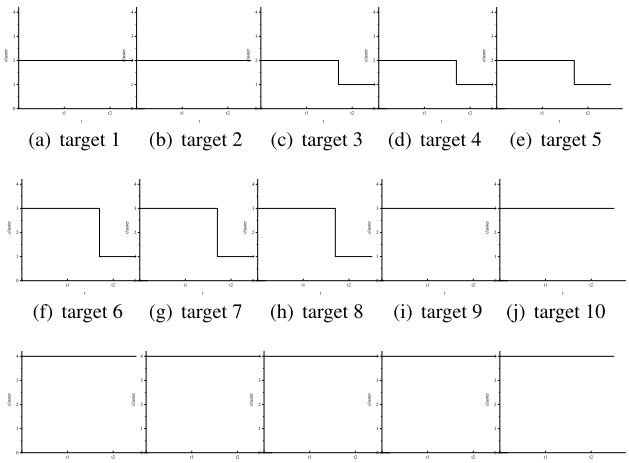
Suppose that a certain moment t, aircraft 1-5 of Party A enter the battlefield from the northwest and fly along a line 60 degrees southeast. Simultaneously, aircraft 6-10 of Party A enter the battlefield from the southwest and fly along a line 60 degrees northeast. In addition, aircraft 11-15 of Party B enter the battlefield from the east side, and they are distributed in a ‘‘zigzag‘‘ arrangement. At moment t1, due to the temporary order, aircraft 1-2 fly along the direction of 30 degrees southeast, while aircraft 9-10 fly along the direction of 30 degrees northeast. At the same time, Party B remains unchanged, cruising on the established route. At moment t2, aircraft 3, 4, 5, 6, 7 and 8 merge into a group, while Party A completes semi-encirclement of Party B. The battlefield target status information at moment t is shown in Table 5.



(a) Clustering view at time t. (b) Clustering view at time t2.

FIGURE 9. The three-dimensional view of the simulated scenario.

In Figure 9(a), the red side represents Party A, and the blue side represents Party B. The clusters to which the targets belong are shown with orange numbers. At moment t, the targets of Party A are divided into two groups in total, with targets 1-5 in group 1 approaching from the northwest and targets 6-10 in group 2 approaching from the southwest. The targets in group 3, Party B, are located on the east side of the map and contain a total of 5 targets from 11-15. After the simulation runs for a period of time, the motion state of each target changes at moment t2. Then, three targets each from the original cluster 1 and cluster 2 separated from their respective original clusters. These 6 targets generated a formation and are coded as group 4. While the remaining targets continue to fly according to the original established route, the formation sequence remains unchanged. The result is accurately embodied by the APBWOA, and no error occurs for the clustered targets, as shown in Figure 9(b).



(a) target 1 (b) target 2 (c) target 3 (d) target 4 (e) target 5

(f) target 6 (g) target 7 (h) target 8 (i) target 9 (j) target 10

(k) target 11 (l) target 12 (m) target 13 (n) target 14 (o) target 15

FIGURE 10. Target grouping results at all times.

It can be seen from Figure 9 to 10 that targets 1-5, 6-10, and 11-15 are divided into 3 formations at time t. At time t1, the headings of aircraft 3, 4, 5, 6, 7, and 8 are adjusted. The formation of aircraft 1-5 is split into two sequences, 1-2 and 3-5, and the formation of aircraft 6-10 is split into 6-8 and 9-10. At time t2, aircraft 3, 4, 5, 6, 7, and 8 merge in formation. The battlefield targets are divided into four formations: 1-2, 3-8, 9-10, and 11-15. The aircraft in Party

A formed a semi-encircling trend of Party B, which proves the correctness and effectiveness of the algorithm.

V. CONCLUSION

In this paper, we propose an affinity propagation method based on an improved whale optimization algorithm for the effective swarming of spatial targets to solve the battlefield target grouping problem. Chaotic maps and nonlinear inertia weights are introduced into the whale optimization algorithm, and the algorithm achieves a balance between local search capability and global exploitation capability according to the change in weights from large to small. Subsequently, the weighted Mahalanobis distance is used instead of the Euclidean distance in the AP algorithm, and the preference search is combined with the silhouette index. Experiments on different datasets and simulated scenarios show that compared with AP, APAP, SAP, CAAP. APBWOA is more accurate and effective in terms of clustering, thus providing an effective solution for the spatial clustering of battlefield targets.

Although some improvements to affinity propagation have been made, there are still problems to be solved. For instance, affinity propagation is a hard-partitioned clustering algorithm. Consequently, there is a clear division boundary in the clustering process. In such a case, all data samples are strictly divided into different classes. However, many objects studied in real life do not have strict class attributes and subordination relations. Additionally, there are overlaps in attributes and class relations. Nevertheless, the soft clustering algorithm represented by the fuzzy c-means algorithm can solve such problems effectively. This gives us great inspiration.

In future work, the APBWOA can be combined with soft clustering algorithms to obtain more realistic clustering results than those obtained here. Additionally, the APBWOA can be introduced to solve real-world problems in many aspects, such as image segmentation, business planning, and traffic recognition. Furthermore, it can be extended to communication, finance, and other applications.

REFERENCES

[1] J. Tang, W. Bi, F. Liu, and W. Zhang, "Exploring urban travel patterns using density-based clustering with multi-attributes from large-scaled vehicle trajectories," *Phys. A, Stat. Mech. Appl.*, vol. 561, Jan. 2021, Art. no. 125301.

[2] M. H. Dangisso, D. G. Datiko, and B. Lindtjørn, "Identifying geographical heterogeneity of pulmonary tuberculosis in southern Ethiopia: A method to identify clustering for targeted interventions," *Global Health Action*, vol. 13, no. 1, Dec. 2020, Art. no. 1785737.

[3] S. T. Tayalı, "A novel backtesting methodology for clustering in mean–variance portfolio optimization," *Knowl.-Based Syst.*, vol. 209, Dec. 2020, Art. no. 106454.

[4] E. Biabiany, D. C. Bernard, V. Page, and H. Paugam-Moisy, "Design of an expert distance metric for climate clustering: The case of rainfall in the lesser antilles," *Comput. Geosci.*, vol. 145, Dec. 2020, Art. no. 104612.

[5] M. Yuan and R. Choudhary, "A two-step clustering framework for locally tailored design of residential heating policies," *Sustain. Cities Soc.*, vol. 63, Dec. 2020, Art. no. 102431.

[6] G. Zhang, C. Zhang, and H. Zhang, "Improved K-means algorithm based on density canopy," *Knowl.-Based Syst.*, vol. 145, pp. 289–297, Apr. 2018.

[7] L. Xi and F. Zhang, "An adaptive artificial-fish-swarm-inspired fuzzy c-means algorithm," *Neural Comput. Appl.*, vol. 32, pp. 16891–16899, Dec. 2020.

[8] Y. Chen, L. Zhou, N. Bouguila, C. Wang, Y. Chen, and J. Du, "BLOCK-DBSCAN: Fast clustering for large scale data," *Pattern Recognit.*, vol. 109, Jan. 2021, Art. no. 107624.

[9] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, Feb. 2007.

[10] Y. Li, C. Guo, and L. Sun, "Fast clustering by affinity propagation based on density peaks," *IEEE Access*, vol. 8, pp. 138884–138897, 2020.

[11] H. Cui, L. Wu, Z. He, S. Hu, K. Ma, L. Yin, and L. Tao, "Exploring multidimensional spatiotemporal point patterns based on an improved affinity propagation algorithm," *Int. J. Environ. Res. Public Health*, vol. 16, no. 11, p. 1988, Jun. 2019.

[12] Y. Han, H. Wu, M. Jia, Z. Geng, and Y. Zhong, "Production capacity analysis and energy optimization of complex petrochemical industries using novel extreme learning machine integrating affinity propagation," *Energy Convers. Manage.*, vol. 180, pp. 240–249, Jan. 2019.

[13] C. Wang, "A sample entropy inspired affinity propagation method for bearing fault signal classification," *Digit. Signal Process.*, vol. 102, Jul. 2020, Art. no. 102740.

[14] L. Wang, Z. Hao, X. Han, and R. Zhou, "Gravity theory-based affinity propagation clustering algorithm and its applications," *Tehnički vjesnik.*, vol. 25, no. 4, pp. 1125–1135, 2018.

[15] G. Kun, W.-Z. Guo, Q.-R. Qiu, and Q.-S. Zhang, "Community detection algorithm based on local affinity propagation and user profile," *J. Commun.*, vol. 36, no. 2, p. 68, 2015.

[16] L. Sun, R. Liu, J. Xu, S. Zhang, and Y. Tian, "An affinity propagation clustering method using hybrid kernel function with LLE," *IEEE Access*, vol. 6, pp. 68892–68909, 2018.

[17] S. Taheri and A. Bouyer, "Community detection in social networks using affinity propagation with adaptive similarity matrix," *Big Data*, vol. 8, no. 3, pp. 189–202, Jun. 2020.

[18] M. Wang, W. Zhang, W. Ding, D. Dai, H. Zhang, H. Xie, L. Chen, Y. Guo, and J. Xie, "Parallel clustering algorithm for large-scale biological data sets," *PLoS ONE*, vol. 9, no. 4, Apr. 2014, Art. no. e91315.

[19] J. Jiang, Z. Wang, T. Chen, C.-C. Zhu, and B. Chen, "Adaptive ap clustering algorithm and its application on intrusion detection," *J. Communications.*, vol. 36, no. 11, pp. 118–126, 2015.

[20] P. Li, H. Ji, B. Wang, Z. Huang, and H. Li, "Adjustable preference affinity propagation clustering," *Pattern Recognit. Lett.*, vol. 85, pp. 72–78, Jan. 2017.

[21] Y. Lu and Y. Li, "A novel fault diagnosis method for circuit breakers based on optimized affinity propagation clustering," *Int. J. Electr. Power Energy Syst.*, vol. 118, Jun. 2020, Art. no. 105651.

[22] I. Dagher, S. Mikhael, and O. Al-Khalil, "Gabor face clustering using affinity propagation and structural similarity index," *Multimedia Tools Appl.*, vol. 80, no. 3, pp. 4719–4727, 2020.

[23] Y. Han, C. Fan, Z. Geng, B. Ma, D. Cong, K. Chen, and B. Yu, "Energy efficient building envelope using novel RBF neural network integrated affinity propagation," *Energy*, vol. 209, Oct. 2020, Art. no. 118414.

[24] L.-C. Wang, Y.-S. Chao, S.-H. Cheng, and Z. Han, "An integrated affinity propagation and machine learning approach for interference management in drone base stations," *IEEE Trans. Cognit. Commun. Netw.*, vol. 6, no. 1, pp. 83–94, Mar. 2020.

[25] U. Maulik, S. Bandyopadhyay, and A. Mukhopadhyay, *Multiobjective Genetic Algorithms for Clustering: Applications in Data Mining and Bioinformatics*. Cham, Switzerland: Springer, 2011.

[26] A. H. Halim, I. Ismail, and S. Das, "Performance assessment of the metaheuristic optimization algorithms: An exhaustive review," *Artif. Intell. Rev.*, vol. 54, pp. 2323–2409, Oct. 2021.

[27] J.-S. Chou and D.-N. Truong, "A novel Metaheuristic optimizer inspired by behavior of jellyfish in ocean," *Appl. Math. Comput.*, vol. 389, Jan. 2021, Art. no. 125535.

[28] R. Zhou, Q. Liu, J. Wang, X. Han, and L. Wang, "Modified semi-supervised affinity propagation clustering with fuzzy density fruit fly optimization," *Neural Comput. Appl.*, vol. 1, pp. 1–18, Nov. 2020.

[29] B. Jia, B. Yu, Q. Wu, C. Wei, and R. Law, "Adaptive affinity propagation method based on improved cuckoo search," *Knowl.-Based Syst.*, vol. 111, pp. 27–35, Nov. 2016.

[30] Y. Liu, J. Liu, Y. Jin, F. Li, and T. Zheng, "An affinity propagation clustering based particle swarm optimizer for dynamic optimization," *Knowl.-Based Syst.*, vol. 195, May 2020, Art. no. 105711.

[31] S. F. Hussain and S. Iqbal, "CCGA: Co-similarity based co-clustering using genetic algorithm," *Appl. Soft Comput.*, vol. 72, pp. 30–42, Nov. 2018.
[32] J. Chen, X. Qi, L. Chen, F. Chen, and G. Cheng, "Quantum-inspired ant lion optimized hybrid k-means for cluster analysis and intrusion detection," *Knowl.-Based Syst.*, vol. 203, Sep. 2020, Art. no. 106167.
[33] L. Wang, Q. Ji, and X. Han, "Adaptive semi-supervised affinity propagation clustering algorithm based on structural similarity," *Tehnicki Vjesnik/Tech. Gazette*, vol. 23, no. 2, pp. 425–435, 2016.
[34] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, 2016.
[35] F. S. Gharehchopogh and H. Gholizadeh, "A comprehensive survey: Whale optimization algorithm and its applications," *Swarm Evol. Comput.*, vol. 48, pp. 1–24, Aug. 2019.
[36] J. Zhang, L. Hong, and Q. Liu, "An improved whale optimization algorithm for the traveling salesman problem," *Symmetry*, vol. 13, no. 1, p. 48, Dec. 2020.
[37] P. Sun, H. Liu, Y. Zhang, L. Tu, and Q. Meng, "An intensify atom search optimization for engineering design problems," *Appl. Math. Model.*, vol. 89, pp. 837–859, Jan. 2021.
[38] M. Liu, X. Yao, and Y. Li, "Hybrid whale optimization algorithm enhanced with Lévy flight and differential evolution for job shop scheduling problems," *Appl. Soft Comput.*, vol. 87, Feb. 2020, Art. no. 105954.
[39] J. Luo, H. Chen, A. A. Heidari, Y. Xu, Q. Zhang, and C. Li, "Multi-strategy boosted mutative whale-inspired optimization approaches," *Appl. Math. Model.*, vol. 73, pp. 109–123, Sep. 2019.
[40] Q. Fan, Z. Chen, Z. Li, Z. Xia, J. Yu, and D. Wang, "A new improved whale optimization algorithm with joint search mechanisms for high-dimensional global optimization problems," *Eng. Comput.*, vol. 1, pp. 1–28, Jan. 2020.
[41] W. Guo, T. Liu, F. Dai, and P. Xu, "An improved whale optimization algorithm for feature selection," *Comput., Mater. Continua*, vol. 62, no. 1, pp. 337–354, 2020.
[42] M. S. Turgut, H. M. Sağban, O. E. Turgut, and O. T. Özmen, "Whale optimization and sine–cosine optimization algorithms with cellular topology for parameter identification of chaotic systems and Schottky barrier diode models," *Soft Comput.*, vol. 25, no. 2, pp. 1365–1409, Jan. 2021.
[43] A. A. Heidari, I. Aljarah, H. Faris, H. Chen, J. Luo, and S. Mirjalili, "An enhanced associative learning-based exploratory whale optimizer for global optimization," *Neural Comput. Appl.*, vol. 32, pp. 5185–5211, Jan. 2020.
[44] S. Chakraborty, A. Kumar Saha, S. Sharma, S. Mirjalili, and R. Chakraborty, "A novel enhanced whale optimization algorithm for global optimization," *Comput. Ind. Eng.*, vol. 153, Mar. 2021, Art. no. 107086.
[45] A. A. Ewees, M. A. Elaziz, and D. Oliva, "A new multi-objective optimization algorithm combined with opposition-based learning," *Expert Syst. Appl.*, vol. 165, Mar. 2021, Art. no. 113844.
[46] V. K. R. A. Kalananda and V. L. N. Komanapalli, "A combinatorial social group whale optimization algorithm for numerical and engineering optimization problems," *Appl. Soft Comput.*, vol. 99, Feb. 2021, Art. no. 106903.
[47] Y. Meng, J. Liang, F. Cao, and Y. He, "A new distance with derivative information for functional k-means clustering algorithm," *Inf. Sci.*, vols. 463–464, pp. 166–185, Oct. 2018.
[48] I. Dokmanic, R. Parhizkar, J. Ranieri, and M. Vetterli, "Euclidean distance matrices: Essential theory, algorithms, and applications," *IEEE Signal Process. Mag.*, vol. 32, no. 6, pp. 12–30, Nov. 2015.
[49] P. C. Mahalanobis, *On the Generalized Distance in Statistics*. New Delhi, India: National Institute of Science of India, 1936.
[50] R. L. Haupt and S. E. Haupt, *The Continuous Genetic Algorithm*. Wiley, 2004.
[51] H. Chen, W. Li, and X. Yang, "A whale optimization algorithm with chaos mechanism based on quasi-opposition for global optimization problems," *Expert Syst. Appl.*, vol. 158, Nov. 2020, Art. no. 113612.
[52] K. Chen, F. Zhou, and A. Liu, "Chaotic dynamic weight particle swarm optimization for numerical function optimization," *Knowl.-Based Syst.*, vol. 139, pp. 23–40, Jan. 2018.
[53] G. Kaur and S. Arora, "Chaotic whale optimization algorithm," *J. Comput. Design Eng.*, vol. 5, no. 3, pp. 275–284, Jul. 2018.
[54] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Proc. IEEE Int. Conf. Evol. Comput. World Congr. Comput. Intell.*, May 1998, pp. 69–73.
[55] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.
[56] M. A. Elaziz and D. Oliva, "Parameter estimation of solar cells diode models by an improved opposition-based whale optimization algorithm," *Energy Convers. Manage.*, vol. 171, pp. 1843–1859, Sep. 2018.
[57] L. Wang, Y. Zhang, and S. Zhong, "Typical process discovery based on affinity propagation," *J. Adv. Mech. Design, Syst., Manuf.*, vol. 10, no. 1, 2016, Art. no. JAMDSM0001.
[58] A. Asuncion and D. Newman, *UCI Machine Learning Repository*. Irvine, CA, USA: Univ. of California, 2007.

YUXIAN DUAN was born in Weihai, Shandong, China, in 1992. He is currently pursuing the M.S. degree with the College of Air and Missile Defense, Air Force Engineering University. His research interests include command and control systems, situation awareness, and intelligent information processing.

CHANGYUN LIU was born in 1973. He received the B.S. and M.S. degrees from Air Force Engineering University (AFEU), in 1996 and 2002, respectively, and the Ph.D. degree in information and communication engineering from Xidian University, in 2014. He is currently a Professor with the College of Air and Missile Defense, AFEU. His research interests include signal processing, target tracking, and time-frequency analysis.

SONG LI was born in Anhui, China, in 1977. He received the Ph.D. degree from Air Force Engineering University (AFEU). He is currently an Associate Professor with the College of Air and Missile Defense, AFEU. His main research interests include command automation information processing and air defense auxiliary decision.