A Novel Dynamic Clustering Method by Integrating Marine Predators Algorithm and Particle Swarm Optimization Algorithm

N. WANG¹, J. S. WANG², (Member, IEEE), L. F. ZHU², H. Y. WANG², AND G. WANG²

¹College of Economics and Management, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China
²School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan 114051, China

Corresponding author: J. S. Wang (wang_jiesheng@126.com)

This work was supported in part by the Project National Natural Science Foundation of China under Grant 71772082 and Grant 71472080, and in part by the Basic Scientific Research Project of Institution of Higher Learning of Liaoning Province under Grant 2017PWDF10.

ABSTRACT Data clustering is the process of identifying natural groupings or clusters based on a certain similarity measure in multi-dimensional data. Aiming at the dynamic clustering problem where the number of clusters cannot be determined in advance, a hybrid dynamic clustering method based on the marine predators algorithm (MPA) and particle swarm optimization (PSO) algorithm was proposed. The position update strategy of the PSO algorithm was used to make up for the lack of MPA in global searching. The fixed-length coding strategy with the real number coding method was used to deal with the variable length clustering optimization problem, and the unfeasible solution processing strategy and the penalty function strategy are adopted to improve the performance of the algorithm and achieve simultaneous optimization of the number of clusters and cluster centers. The proposed MPA-PSO algorithm with PSO algorithm, MPA, Differential Evolution (DE) algorithm, Spotted Hyena Optimizer (SHO), Lightning Searching Algorithm (LSA) and Equilibrium Optimizer (EO) are adopted to carry out the clustering simulation experiments on four artificial data sets and six real data sets (Iris, Wine, Wisconsin breast cancer, Vowel, Seeds, and Wdbc) in UCI databases. Three performance indicators (the number of clusters, ARI and Accuracy) are used to evaluate the clustering results. The experimental results show that the proposed method can not only successfully find the correct number of clusters, but also obtain stable results for most test problems.

INDEX TERMS Dynamic clustering, marine predator algorithm, particle swarm optimization algorithm.

I. INTRODUCTION

Data clustering is the process of identifying natural groupings or clusters in multi-dimensional data based on a certain similarity measure (such as Euclidean distance) [1], [2]. A cluster is usually identified by a cluster center or center of gravity [3]. In recent years, clustering has been widely used in the fields of engineering, computer science, biology and medicine, social science and economics [4]. Since the data may have different shapes and sizes, it is usually not known how many clusters should be formed [5]. In the solution to the number of clusters, the cluster validity functions can be used to obtain the effective number of clusters. At present, many effective cluster validity indexes have been proposed, such as Dunn index [6], DB Index [7], CH index [8], CS index [9] only focusing on the structure of the data set. Some cluster validity indexes also consider the geometric structure of the data set and the degree of membership, such as the Xie and Beni index (XB) [10], the SC index (SC) proposed by Bensaid et al. [11], VK index based on XB index proposed by Kwon [12], VT index proposed by Tang et al. [13], Pakhira-Bandyopadhyay-Maulik index (PBMF) [14], Partition coefficient and exponential separation (PCAES) index [15], CWB index proposed by M. Ramze Rezae [16], VWL index proposed by Chih-Hung Wu et al. [17].

Although the above cluster validity functions are widely used, many researches are developing clustering algorithms that do not require a predefined number of clusters. This research direction is called automatic clustering or dynamic clustering. In recent years, many clustering algorithms have...
been developed. The optimization ability of genetic algorithm was used to automatically find the number of clusters and the appropriate clustering of any data set [18]. In order to encode a variable number of clusters, a string encoding method consisting of real numbers and uncorrelated symbols is adopted, and then a muti-objective clustering algorithm MIE-MOCK was proposed to improve the accuracy of clustering. The algorithm uses randomly selected crossover operator and mutation operator to increase search diversity instead of a single crossover operator and mutation operator [19]. By using the so-called multi-information exchange operator to increase search diversity, this algorithm can ultimately improve clustering quality. The Empire Competition Algorithm (ICA) was proposed for automatic clustering problems, and a new method based on random and homogeneity-based merge-splitting methods was proposed to change the number of clusters [20]. The experimental results show that the algorithm has advantage in convergence speed and the solution quality. A new symmetry-based genetic clustering algorithm was proposed to automatically evolve the number of clusters from the data set and divide the clusters appropriately. Then a new point symmetry-based clustering validity index (Sym index) was proposed to measure the effectiveness of clustering [21]. Hong Yu et al. proposed an effective automatic clustering method and extended the rough set model of decision theory to clustering [22]. The automatic clustering problem with muti-objective optimization (MOO) problem was studied and two clustering validity indexes are optimized at the same time [23]. Cong Liu et al. designed three adaptive coding schemes. The fixed-length chromosomes were used to deal with variable-length optimization problems so as to automatically detect the number of clusters [24]. A muti-elite particle swarm optimization (MEPSO) method was proposed for clustering complex data sets and linear inseparable data sets. The kernel function can cluster the linearly inseparable data in the original input space into isomorphic groups of the transformed high-dimensional feature space [25]. A new locally adjusted differential evolution algorithm (DELA) was proposed to find the best optimal number of clusters [26]. An improved differential evolution (DE) algorithm was proposed to solve the automatic clustering problem and the best solution effect and acceleration factor were proposed to improve the convergence of the solution algorithm [27]. Mahamed proposed a dynamic clustering method based on particle swarm optimization (DCPSO) algorithm, which was applied to the image segmentation of synthetic images and natural images [28]. R. J. Kuo et al. proposed a dynamic clustering method (DCGA) based on particle swarm optimization (PSO) and genetic algorithm (GA) [29]. Shih-Ming Pan proposed an automatic clustering algorithm framework (called ETSAs), which does not require the user to give every possible value of the required parameters (including the number of clusters) [30]. The simulation results show that ETSA is superior in finding the correct number of clusters.

This paper proposes a hybrid MPA-PSO algorithm that combines marine predator algorithm (MPA) and particle swarm optimization (PSO) algorithm to solve dynamic clustering problems. The fixed length coding strategy with the real number coding method was used to deal with the variable length clustering optimization problem to realize automatic clustering, that is to find the optimal number of clusters and clustering centers. The effectiveness of the proposed method is verified by simulation experiments between the MPA-PSO algorithm and several other intelligent optimization algorithms on four artificial data sets and six UCI data sets.

II. HYBRID ALGORITHM BY INTEGRATING MARINE PREDATORS ALGORITHM AND PARTICLE SWARM OPTIMIZATION ALGORITHM (MPA-PSO)

A. MARINE PREDATORS ALGORITHM

The main inspiration of the Marine Predator Algorithm (MPA) is the extensive foraging strategy of marine predators, namely Levy flight and Brown motion and the optimal encounter rate strategy in the interaction of predator and predator organisms [31]. Marine predators use Levy strategy for environments with low prey concentration, and Brown motion for areas with abundant prey. The speed ratio $v$ from prey to predator represents a trade-off between Levy flight and Brown motion.

1) In the low speed ratio ($v = 0.1$), the best strategy for the predator is the Levy flight strategy. The prey is either in Brown motion or Levy flight.

2) In the unit speed ratio ($v = 1$), if the prey moves in Levy flight, the best strategy for the predator is the Brown motion, otherwise scenarios depend on the size of the system.

3) In the high-speed ratio ($v ≥ 10$), the best strategy for the predator is to be completely immobile. In this case, the prey is either Brown motion or Levy flight.

The mathematical model of the MPA algorithm is introduced in details. Firstly initialize a set of solutions in the searching space.

$$X_0 = X_{\min} + rand(X_{\max} - X_{\min})$$ (1)

where, $X_{\min}$ and $X_{\max}$ are the lower and upper bounds of the variable, and $rand$ is a uniform random vector in the range $[0, 1]$.

According to the survival of the fittest theory, the top predators in nature are more talented in foraging. Therefore, the best solution is designated as the top predator to construct a matrix called the elite matrix. This matrix array is used to find the prey based on the position information of the prey.

$$Elite = \begin{bmatrix} X_{1,1} & X_{1,2} & \cdots & X_{1,d} \\ X_{2,1} & X_{2,2} & \cdots & X_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ X_{n,1} & X_{n,2} & \cdots & X_{n,d} \end{bmatrix}_{n \times d}$$ (2)
where, \( \vec{X}_1 \) represents the top predator vector and it is copied \( n \) times to construct the elite matrix; \( n \) is the number of searching agents, and \( d \) is the searching dimension.

Another matrix with the same dimension as the elite matrix is called the prey matrix, and the predator updates its position based on it, that is to initially create the initial prey matrix, and build the elite matrix with the fittest (predator). Its prey matrix is defined as follows.

\[
\text{Prey} = \begin{bmatrix}
X_{1,1} & X_{1,2} & \cdots & X_{1,d} \\
X_{2,1} & X_{2,2} & \cdots & X_{2,d} \\
\vdots & \vdots & \ddots & \vdots \\
X_{n,1} & X_{n,2} & \cdots & X_{n,d}
\end{bmatrix}
\]  
(3)

where, \( X_{i,j} \) represents the \( i \)th dimension of the \( j \)th prey.

In the iterative process of the main loop of the MPA algorithm, the optimization process is divided into three stages according to the speed ratio. The model is described as follows.

1) HIGH SPEED RATIO STAGE (\( \nu \geq 10 \))

The exploration stage at this stage can be expressed as follows.

While \( \text{Iter} < \frac{1}{3} \text{Max}\_\text{Iter} \)

\[
\text{stepsiz}_{ei} = \overrightarrow{R}_{B} \otimes (\text{Elite}_{i} - \overrightarrow{R}_{B} \otimes \text{Pre}_{ey}_{i}) \quad i = 1, \ldots, n
\]  
(4)

\[
\text{Pre}_{ey}_{i} = \text{Pre}_{ey}_{i} + P \ast \overrightarrow{R} \otimes \text{stepsiz}_{ei}
\]  
(5)

where, \( \overrightarrow{R}_{B} \) is a vector of random numbers, and these numbers are based on the normal distribution to represent the Brown motion; \( \otimes \) is termwise multiplication; \( P = 0.5 \) is a constant, and \( \overrightarrow{R} \) is a vector of uniform random numbers in the scope \([0, 1]\). This situation occurs in the first third of the iteration, the step size or moving speed and the exploration capability are higher; \( \text{Iter} \) is the current iteration, and \( \text{Max}\_\text{Iter} \) is the maximum iteration.

2) UNIT SPEED RATIO STAGE (\( \nu = 1 \))

This part occurs in the middle stage of optimization, and the exploration attempts to be converted to exploitation. At this stage, both exploration and exploitation are important. Therefore, half of the individuals are designated for exploration and the other half are designated for exploitation. At this stage, the prey is responsible for hunting, and the predator is responsible for prospecting.

While \( \frac{1}{3} \text{Max}\_\text{Iter} < \text{Iter} < \frac{2}{3} \text{Max}\_\text{Iter} \)

For the first half of the population:

\[
\text{stepsiz}_{ei} = \overrightarrow{R}_{L} \otimes (\text{Elite}_{i} - \overrightarrow{R}_{L} \otimes \text{Pre}_{ey}_{i}) \quad i = 1, \ldots, n/2
\]  
(6)

\[
\text{Pre}_{ey}_{i} = \text{Pre}_{ey}_{i} + P \ast \overrightarrow{R} \otimes \text{stepsiz}_{ei}
\]  
(7)

For the latter half of the population:

\[
\text{stepsiz}_{ei} = \overrightarrow{R}_{B} \otimes (\overrightarrow{R}_{B} \otimes \text{Elite}_{i} - \text{Pre}_{ey}_{i}) \quad i = n/2, \ldots, n
\]  
(8)

\[
\text{Pre}_{ey}_{i} = \text{Elite}_{i} + P \ast \overrightarrow{CF} \otimes \text{stepsiz}_{ei}
\]  
(9)

where, \( \overrightarrow{R}_{L} \) is a random number vector based on the Lévy distribution, representing the Lévy flight. At this stage, the first half of the prey moves with the Lévy strategy, while the other half moves with the Brown motion strategy.

\( \overrightarrow{CF} \) is considered as an adaptive parameter, which is used to control the step length of the predator’s movement, and the specific expression is shown in Eq. (10).

\[
\overrightarrow{CF} = (1 - \frac{\text{Iter}}{\text{Max}\_\text{Iter}})^{2 \frac{\text{Iter}}{\text{Max}\_\text{Iter}}}
\]  
(10)

3) LOW SPEED RATIO STAGE (\( \nu = 0.1 \))

At low speed ratios or when the predator is moving faster than the prey, this happens in the last stage of the optimization process, which is mainly related to high development capabilities. At low speed ratio conditions, the best strategy for the predator is Lévy flight.

While \( \text{Iter} > \frac{2}{3} \text{Max}\_\text{Iter} \)

\[
\text{stepsiz}_{ei} = \overrightarrow{R}_{L} \otimes (\overrightarrow{R}_{L} \otimes \text{Elite}_{i} - \text{Pre}_{ey}_{i}) \quad i = 1, \ldots, n
\]  
(11)

\[
\text{Pre}_{ey}_{i} = \text{Elite}_{i} + P \ast \overrightarrow{CF} \otimes \text{stepsiz}_{ei}
\]  
(12)

Another cause of changes in the behavior of marine carnivores is environmental problems, such as vortex formation or fish gathering device (FADs) effect. Sharks spend more than 80% of their time near the FADs, and for the remaining 20% of the time, they may make longer jumps in different dimensions to find an environment where another prey is distributed. FADs is considered a local optimal solution, and its role is to capture these points in the searching space. These longer jumps are taken into account during the simulation, which avoids the local optimal stagnation. Therefore, the mathematical expression of the FADs effect can be represented as:

\[
\overrightarrow{Per}_{yi} = \begin{cases}
\overrightarrow{Per}_{yi} + \overrightarrow{CF}[\overrightarrow{X}_{\min} + \overrightarrow{R} \otimes (\overrightarrow{X}_{\max} - \overrightarrow{X}_{\min})] \otimes \overrightarrow{U} & \text{if } r \leq \text{FADs} \\
\overrightarrow{Per}_{yi} + [\text{FADs}(1 - r) + r] \cdot (\overrightarrow{Per}_{y1} - \overrightarrow{Per}_{y2}) & \text{if } r > \text{FADs}
\end{cases}
\]  
(13)

where, \( \overrightarrow{U} \) is a binary vector containing an array of 0 and 1. Its construction method is described as follows. Generate a random vector in the scope \([0, 1]\). If the array is less than 0.2, change it to array 0. If the array is greater than 0.2, change its array to 1. \( r \) is a uniform random number in the scope \([0, 1]\). The probability that FADs will affect the optimization process is \( \text{FADs} = 0.2 \).
FIGURE 1. Flowchart of MPA-PSO algorithm.

MPA saves memory by preserving the old location of the prey. After updating the current solution, compare the fitness values of the current solution and the old solution, and exchange them if the fitness of the old solution is better than the current solution. The pseudo code of MPA is described as follows.

Initialize the searching agents (Prey) populations \( i = 1, \ldots, n \)

While termination criteria are not met

Calculate the fitness and construct the Elite matrix

If \( \text{Iter} < \frac{1}{4} \text{Max Iter} \)

Update prey based on Eq. (5)

Else if \( \frac{1}{4} \text{Max Iter} < \text{Iter} < \frac{2}{3} \text{Max Iter} \)

For the first half of the populations \( (i = 1, \ldots, n/2) \)

Update prey based on Eq. (7)

For the other half of the populations \( (i = n/2, \ldots, n) \)

Update prey based on Eq. (9)

Else if \( \text{Iter} > \frac{2}{3} \text{Max Iter} \)

Update prey based on Eq. (12)

End If

Save accomplish memory and update Elite matrix

Apply FADs effect and update based on Eq. (13)

Determine the optimal solution

Update the parameters of MPA

Iter<\text{Max Iter} 

\( \text{Iter}<\frac{1}{3} \text{Max Iter} \)

Yes

No

\( \frac{1}{3} \text{Max Iter}<\text{Iter}<\frac{2}{3} \text{Max Iter} \)

Yes

No

\( \frac{2}{3} \text{Max Iter}<\text{Iter} \)

Yes

No

Complete memory saving and elite update

Apply FADs effect and update according to formula (13)

End

Update position according to formula (11-12)

Update position according to formula (9)

Update position according to formula (6-7)

Update position according to formula (14-15)
Save accomplish memory and update Elite matrix
End While

B. PARTICLE SWARM OPTIMIZATION (PSO) ALGORITHM

The Particle Swarm Optimization (PSO) algorithm is a global random searching algorithm based on swarm intelligence proposed by Kennedy and Eberhart, which is inspired by simulating the migration and swarming behavior of birds in the foraging process [32]. The velocity and position update formula of the PSO algorithm is described as follows.

\[ V_{id}(t + 1) = \omega \cdot V_{id}(t) + c_1 \cdot \phi_1 \cdot (P_{id}(t) - X_{id}(t)) + c_2 \cdot \phi_2 \cdot (P_{gd}(t) - X_{id}(t)) \]  (14)
\[ X_{id}(t + 1) = X_{id}(t) + V_{id}(t + 1) \]  (15)

where, \( t \) is the number of iterations; \( i \) is the number of particles; \( d \) is the number of dimensions; \( c_1 \) and \( c_2 \) are learning factors; \( \phi_1 \) and \( \phi_2 \) are random numbers of [0, 1]; \( \omega \) is the inertia weight; \( P_{id} \) is the local optimal solution found by the \( i \)-th particle so far, and \( P_{gd} \) is the best particle in the entire community or a certain neighbor so far.

C. HYBRID ALGORITHM BY INTEGRATING MARINE PREDATORS ALGORITHM AND PARTICLE SWARM OPTIMIZATION ALGORITHM (MPA-PSO)

As a global optimization algorithm, MPA divides the number of iterations to perform different functions. Global exploration is performed in the first 1/3 of the iteration, half of the particle exploration and half of the particle exploitation in the middle 1/3, and partial development of the last 1/3. As the first step to find the best solution, the exploration ability should be good to give the next step of local development an optimal direction to find the optimal solution, but the initial global searching ability effect is not very ideal. The PSO algorithm is an effective swarm intelligence optimization algorithm, and its global searching and local searching ability can be confirmed. Therefore, this article combines MPA and PSO algorithm to improve the optimization efficiency of MPA, and replaces the position update strategy of PSO algorithm with the first stage strategy of MPA. The flowchart of the proposed MPA-PSO algorithm is shown in Fig. 1.

III. DYNAMIC CLUSTERING METHOD BASED ON MPA-PSO ALGORITHM

A. CODING METHOD

In the automatic clustering problem, there are usually two main problems, namely the determination of the number of clusters and the clustering centers. In order to adapt to these two goals, this paper adopts fixed-length coding in the real-number coding method [33]. Each example in this coding strategy consists of two parts, the latter part is responsible for representing the number of clusters, and the other part represents the cluster centers. In this method, for \( n \) particle, each particle contains \( d \)-dimensional data, the maximum number of clusters \( K_{max} \) is specified, and the form of each particle \( X_i(t) \) in the \( t \)-th iteration is a matrix with \( K_{max} \) rows and \( d + 1 \) columns. The single particle coding strategy is shown in the Fig. 2.

![FIGURE 2. Single particle coding strategy.](image)

The \( m_{ij} \) in the first column is the cluster center, and each cluster center contains \( d \)-dimensional data. The last column is \( K_{max} \) positive floating-point numbers in [0,1], each of which can determine whether the corresponding cluster is activated (that is, whether it is actually used to classify data), which is named as the activation threshold. When \( Y_{ij} > 0.5 \), its corresponding cluster center is activated. Fig. 3 is an example of a single particle encoding, that is to say that the position coordinates of a set of specific parameters. The cluster center contains 3 dimensions, \( K_{max} = 5 \), among which the 1, 2, and 5 thresholds are greater than 0.5, and the corresponding cluster center is activated and can be used to partition the data set (marked in bold in the Fig. 3). The pros and cons of the particle division can be judged by appropriate clustering validity index.

![FIGURE 3. Example of single particle encoding.](image)

B. FITNESS FUNCTION

In this study, the PBMF index, a product form of validity index proposed by Pakhira et al. [14], was used. It consists of three parts. The first part decreases as the number of clusters increases. The second part is the ratio of the sum of the distances between the data set as a whole and divided into \( k \) categories. The numerator is fixed and it estimates the compactness of \( k \) classes. As \( k \) increases, this factor increases the strength of the PBM index. The third part is the maximum distance between \( k \) classes, which represents the separation between classes. When \( k \) increases, the first part
N. Wang et al.: Novel Dynamic Clustering Method by Integrating MPA and PSO Algorithm

FIGURE 4. Scatter plot of artificial data sets.

decreases, and the other two parts increase. Therefore, these three factors compete with each other and balance each other, which promotes the closeness and separation of clusters. The maximization of the PBMF index ensures that a small number of compact clusters are formed when there is a large gap between at least two clusters. The PBM index is defined as follows.

$$V_{PBMF} = \left( \frac{1}{k} \times \frac{E_1}{E_k} \times D_k \right)^2 = \left( \sum_{j=1}^{n} u_{ij} \|x_j - v_i\| \times \max_{i,j=1,...,k} \|v_i - v_j\| \right)^2 \quad (16)$$

where, $u_{ij}$ represents the membership function. Because this article is a hard classification, in this article $u_{ij}$ is a constant and equal to 1 (if $x_j$ is in the $i$-th cluster) or 0. This paper uses the reciprocal of the PBMF exponent as the fitness function, so it is the problem of finding the minimum objective function.

C. UNFEASIBLE SOLUTION PROCESSING STRATEGY

In any heuristic method, unfeasible solutions are universal. For automatic clustering, if the number of active clusters is less than 2, that is to say that the cluster size is less than 2, an unfeasible solution will appear. In this mechanism, if the number of clusters of a particle is less than 2, select 2 to $K_{max}$ random bits and activate the corresponding clusters, which can increase the randomness of the algorithm.
The pseudo code of the unfeasible solution processing strategy is described as follows.

Given particle \( X_i(t) \), \( N \) is the number of clusters that are activated.

If \( N < 2 \)

1. Generate integer number \( a (2 < a < K_{\text{max}}) \)
2. Sort the threshold and its corresponding cluster center from largest to smallest.
3. The first \( i \) thresholds of the sorted particles are changed to 1, that is, \( i \) cluster centers are activated.

\( N = \) i

End If

In addition, all data points are assigned to the nearest activated cluster, and then the size of the cluster is evaluated. If the cluster has more than two data points, then it is valid. Here our practical penalty strategy is to add a real number \( \alpha \) to the fitness function, so that the fitness value of this particle will change and the clustering effect will be worsened, thereby effectively avoiding the unfeasible solution from being selected as the optimal solution.

D. ALGORITHM FLOWCHART

Step 1: Initialize each particle to contain \( K_{\text{max}} \) randomly selected cluster centers and \( K_{\text{max}} \) activation thresholds in \([0,1]\).

Step 2: Use the rules described in Section 3.1 to find the active cluster center of each particle and calculate the fitness value of each particle.

Step 3: For \( t = 1 : t_{\text{max}} \)

1. Use Eq. (6)-(9) and Eq. (11)-(15) to update the position and obtain the activated cluster centers.
2. Calculate and compare the distance between the activated multiple cluster centers and the data samples to obtain the corresponding cluster division.
3. When encountering an unfeasible solution, correct it through repair and punishment strategies.

Step 4: Use the cluster centers and divisions obtained by the optimal particles (the particles that produce the maximum fitness function value) at time \( t = t_{\text{max}} \) as the final solution.

IV. SIMULATION EXPERIMENTS AND RESULT ANALYSIS

A. PERFORMANCE INDICATORS

There are three methods for testing the clustering effect: the number of clusters, the adjusted Rand index and the Accuracy rate.

1) ADJUSTED RAND INDEX

Rand index \([34]\) needs to be given actual category information \( C \). Suppose \( K \) is the clustering result, \( a \) represents the logarithm of elements of the same category in both \( C \) and \( K \), and \( b \) represents the logarithm of elements of different categories in both \( C \) and \( K \), then the Rand index can be defined as:

\[
RI = \frac{a + b}{C_2^{\text{samples}}} \tag{17}
\]

where, \( C_2^{\text{samples}} \) is the total logarithm of elements that can be formed in the data set, and the value range of \( RI \) is \([0, 1]\). The larger the \( RI \) value, the more consistent the clustering result is with the real situation.

For random results, \( RI \) does not guarantee that the score is close to zero. In order to be sure that the index should be close to zero in the case of random clustering results, the adjusted rand index \( (ARI) [35] \) was proposed, it has a higher degree of discrimination.

\[
ARI = \frac{RI - E[RI]}{\max(RI) - E[RI]} \tag{18}
\]

The value range of \( ARI \) is \([-1, 1]\). The larger the value, the more consistent the clustering result is with the real situation.
FIGURE 5. ARI under different algorithms on different data sets.
In a broad sense, $ARI$ measures the degree of similarity between two data distributions.

2) ACCURACY
The definition of accuracy is the ratio of the number of correctly classified samples to the total number of samples for a given test data set. That is, the accuracy of the test data set when the loss function is 0-1 loss [36].

B. EXPERIMENTAL DATA SETS
This paper selects 4 kinds of artificial data sets and 6 kinds of real data sets in UCI database for carrying out the clustering simulation experiments. The selected UCI database data sets include Iris, Wine, Wisconsin breast cancer, Vowel, Seeds and Wdbc. Four artificial data sets have linear inseparable clusters with different shapes. Table 1 shows the data points of the simulation experiment data sets. The scatter graphs of the artificial data sets are shown in Fig.4 and information such as quantity, number of clusters and data dimensions are listed in Table 1.

C. SIMULATION EXPERIMENTS AND RESULT ANALYSIS
This paper compares the improved MPA-PSO algorithm with several swarm intelligence optimization algorithms in simulation experiments, including Particle Swarm Optimization (PSO) Algorithm [32], Marine Predators Algorithm (MPA) [31], Differential Evolution (DE) Algorithm [37], Spotted Hyena Optimizer (SHO) [38], Lightning Search Optimization Algorithm (LSA) [39] and Equilibrium Optimizer (EO) [40]. The specific parameter settings of each algorithm are listed in Table 2. Set the maximum number of clusters $K_{\text{max}} = 20$. Six kinds of real data sets and four
kinds of artificial data sets in UCI database are adopted for carrying out the clustering experiments. The effectiveness of random algorithms largely depends on the choice of the initial solutions, so all the algorithms in this paper adopt randomly generated initial solutions, and for each data set, the algorithms are executed ten times to test the effectiveness of various algorithms. Three performance indicators (cluster number, ARI and Accuracy) are used to evaluate the clustering results. Running results are shown in Table 3-12.

It can be seen from Table 3 that although the MPA-PSO algorithm divides the Iris data correctly, it slightly loses the accuracy and ARI in DE algorithm. So the MPA-PSO algorithm has the best clustering effect compared with other algorithms. In other words, the MPA-PSO algorithm still achieves a good clustering effect. Seen form Table 4, the MPA-PSO algorithm has obtained good results not only for clustering but also for clustering effect on the Wine data set. Seen from Table 5, all algorithms other than SHO in 10 times simulation experiments have been correctly clustered, and it is worth noting that the clustering effect of the MPS-PSO algorithm is the best. The MPA-PSO algorithm has shown significant effects in the Vowel data division in Table 6, and the obtained clustering accuracy and ARI are also better than other algorithms, but the shortcoming is stability. The correct division of Seeds and Wdbc data by the MPA-PSO algorithm also shows its superiority. In the division of the artificial data sets, it can be seen that in the automatic clustering of the Data2_2 data set in Table 9, except for the PSO algorithm and SHO, the accuracy of the remaining algorithms has reached 100%. While for the Data3_2 data set, the MPA-PSO algorithm in automatic clustering shows high accuracy and
TABLE 5. Results obtained by different algorithms on Cancer data set.

|                | Clustering number | ARI       | Accuracy  |
|----------------|------------------|-----------|-----------|
| MPA-PSO        | 2≤0              | 0.7283±0.0085 | 0.9282±0.0024 |
| PSO            | 2≤0              | 0.5482±0.0086 | 0.8746±0.0025 |
| MPA            | 2≤0              | 0.4907±0.0552 | 0.8555±0.0179 |
| DE             | 2≤0              | 0.5818±0.0585 | 0.8846±0.0188 |
| SHO            | 2.1±0.3          | 0.5131±0.1798 | 0.848±0.0669  |
| LSA            | 2≤0              | 0.5478±0.0074 | 0.8744±0.0002 |
| EO             | 2≤0              | 0.543±0.0015  | 0.8731±0.0003 |

TABLE 6. Results obtained by different algorithms on Vowel data set.

|                | Clustering number | ARI       | Accuracy  |
|----------------|------------------|-----------|-----------|
| MPA-PSO        | 5.6429±0.9715    | 0.327±0.0625 | 0.4914±0.0747 |
| PSO            | 3±1              | 0.2654±0.0356 | 0.4346±0.0407 |
| MPA            | 2±0              | 0.2515±0.0006 | 0.3946±0.0063 |
| DE             | 2±0              | 0.2451±0.0058 | 0.4±0        |
| SHO            | 2±0              | 0.2764±0.0179 | 0.4065±0.0072 |
| LSA            | 2±0              | 0.2518±0     | 0.4±0        |
| EO             | 2±0              | 0.2518±0     | 0.4±0        |

TABLE 7. Results obtained by different algorithms on Seeds dataset.

|                | Clustering number | ARI       | Accuracy  |
|----------------|------------------|-----------|-----------|
| MPA-PSO        | 2.8±0.4          | 0.506±0.0915 | 0.7291±0.108 |
| PSO            | 2.2±0.4          | 0.4196±0.0753 | 0.6508±0.0735 |
| MPA            | 2±0              | 0.3795±0.0026 | 0.614±0     |
| DE             | 2±0              | 0.393±0.0262  | 0.6205±0.0097 |
| SHO            | 2±0              | 0.4247±0.0427 | 0.6381±0.0241 |
| LSA            | 2±0              | 0.4005±0.0231 | 0.6236±0.0118 |
| EO             | 2±0              | 0.3816±0     | 0.614±0     |

TABLE 8. Results obtained by different algorithms on Wdbc data set.

|                | Clustering number | ARI       | Accuracy  |
|----------------|------------------|-----------|-----------|
| MPA-PSO        | 2±0              | 0.635±0.0321 | 0.8989±0.0097 |
| PSO            | 2.1±0.3          | 0.349±0.1899 | 0.7684±0.1209 |
| MPA            | 2±0              | 0.4557±0.0877 | 0.8409±0.03 |
| DE             | 2±0              | 0.4402±0.0902 | 0.8352±0.0308 |
| SHO            | 2.2±0.4          | 0.562±0.1016  | 0.8513±0.0748 |
| LSA            | 2±0              | 0.3951±0.0631 | 0.8197±0.0225 |
| EO             | 2±0              | 0.3951±0.0631 | 0.8197±0.0225 |

TABLE 9. Results obtained by different algorithms on Data2_2 data set.

|                | Clustering number | ARI       | Accuracy  |
|----------------|------------------|-----------|-----------|
| MPA-PSO        | 2±0              | 1±0       | 1±0       |
| PSO            | 2.7±1.1874       | 0.9075±0.1675 | 0.925±0.1434 |
| MPA            | 2±0              | 1±0       | 1±0       |
| DE             | 2±0              | 1±0       | 1±0       |
| SHO            | 2.3±0.6403       | 0.8672±0.1836 | 0.9214±0.1187 |
| LSA            | 2±0              | 1±0       | 1±0       |
| EO             | 2±0              | 1±0       | 1±0       |

TABLE 10. Results obtained by different algorithms on Data3_2 data set.

|                | Clustering number | ARI       | Accuracy  |
|----------------|------------------|-----------|-----------|
| MPA-PSO        | 3±0              | 0.8444±0.0029 | 0.9473±0.0009 |
| PSO            | 3.4±0.6633       | 0.8362±0.022 | 0.9323±0.0268 |
| MPA            | 3±0.6325         | 0.7667±0.1539 | 0.8849±0.1096 |
| DE             | 3.2±0.4          | 0.8384±0.0225 | 0.9397±0.0113 |
| SHO            | 2.8±0.9798       | 0.6268±0.1655 | 0.7733±0.1221 |
| LSA            | 3.1±0.3          | 0.8435±0.0002 | 0.944±0.009 |
| EO             | 3.1±0.7          | 0.7991±0.1157 | 0.9105±0.0853 |

stability. In the clustering of data sets Data5_2 and Data4_3, the MPA-PSO algorithm not only performs correctly in the number of clusters, but also has better clustering effect than other algorithms.

Fig. 5 shows the convergent trend of the ARI indicator in 300 iterations under seven algorithms on different data sets. Those curves are the obtained average values by running ten
The MPA-PSO algorithm is compared with six algorithms of number of clusters and can obtain better clustering results. Six kinds of real data sets in UCI database are adopted to characteristics. Four kinds of artificial synthetic data sets and find the appropriate number of clusters according to data to solve the problem of pre-setting the number of clusters, PSO algorithm to make up for the shortcomings of MPA’s optimization algorithms on different data sets. The simulation times. Fig. 6 is a comprehensive display of the accuracy of all data sets, MPA-PSO algorithm can get better clustering results.

V. CONCLUSION

This paper firstly proposes the combination of MPA and PSO algorithm to make up for the shortcomings of MPA’s lack of randomness, and then adopts MPA-PSO algorithm to solve the problem of pre-setting the number of clusters, and find the appropriate number of clusters according to data characteristics. Four kinds of artificial synthetic data sets and six kinds of real data sets in UCI database are adopted to verify that the MPA-PSO algorithm can obtain the correct number of clusters and can obtain better clustering results. The MPA-PSO algorithm is compared with six algorithms of PSO, MPA, DE, SHO, LSA, and EO. The simulation results show that the MPA-PSO algorithm can get better clustering results.

REFERENCES

[1] A. K. Jain, M. N. Murty, and P. J. Flynn, “Data clustering: A review,” ACM Comput. Surv., vol. 31, no. 3, pp. 264–293, 1999.
[2] A. K. Jain, R. P. W. Duin, and J. Mao, “Statistical pattern recognition: A review,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 22, no. 1, pp. 4–37, Jan. 2000.
[3] C. Y. Lee and E. K. Antonsson, “Dynamic partitional clustering using evolution strategies,” in Proc. 3rd Asia–Pacific Conf. Simulated Evol. Learn., Oct. 2000, pp. 2716–2721.
[4] A. Ducournau, A. Bretto, S. Rital, and B. Laget, “A reductive approach to hypergraph clustering: An application to image segmentation,” Pattern Recognit., vol. 45, no. 7, pp. 2788–2803, Jul. 2012.
[5] G. Hamerly and C. Elkan, “Learning the K in K-means,” in Proc. 7th Annu. Conf. Neural Inf. Process. Syst., 2003, pp. 281–288.
[6] J. C. Dunn, “A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters,” J. Cybern., vol. 3, no. 3, pp. 32–57, Jan. 1973.
[7] D. L. Davies and D. W. Bouldin, “A cluster separation measure,” IEEE Trans. Pattern Anal. Mach. Intell., vol. PAMI-1, no. 2, pp. 224–227, Apr. 1979.
[8] T. Calinski and J. Harabasz, “A dendrite method for cluster analysis,” Commun. Statist. Theory Methods, vol. 3, no. 1, pp. 1–27, 1974.
[9] C.-H. Chou, M.-C. Su, and E. Lai, “A new cluster validity measure and its application to image compression,” Pattern Anal. Appl., vol. 7, no. 2, 205–220, Jul. 2004.
[10] N. L. Xie and G. Beni, “A validity measure for fuzzy clustering,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 13, no. 8, pp. 841–847, Aug. 1991.
[11] A. M. Bensaid, L. O. Hall, J. C. Bezdek, L. P. Clarke, M. L. Silbiger, J. A. Arrington, and R. F. Murtagh, “Validity-guided (re)clustering with applications to image segmentation,” IEEE Trans. Fuzzy Syst., vol. 4, no. 2, pp. 112–123, May 1996.
[12] S. H. Kwon, “Cluster validity index for fuzzy clustering,” Electron. Lett., vol. 34, no. 22, p. 2176, 1998.
[13] Y. Tang, F. Sun, and Z. Sun, “Improved validation index for fuzzy clustering,” in Proc. Amer. Control Conf., Jun. 2005, pp. 1120–1125.
[14] M. K. Pakhira, S. Bandyopadhyay, and U. Maulik, “Validity index for crisp and fuzzy clusters,” Pattern Recognit., vol. 37, no. 3, pp. 487–501, Mar. 2004.
[15] K.-L. Wu and M. S. Yang, “A cluster validity index for fuzzy clustering,” Pattern Recognit. Lett., vol. 26, no. 9, pp. 1275–1291, 2005.
[16] M. R. Rezaee, B. P. F. Lelieveldt, and J. H. C. Reiber, “A new cluster validity index for the fuzzy C-mean,” Pattern Recognit. Lett., vol. 19, pp. 237–246, Mar. 1998.
[17] C.-H. Wu, C.-S. Ouyang, L.-W. Chen, and L.-W. Lu, “A new fuzzy clustering validity index with a median factor for centroid-based clustering,” IEEE Trans. Fuzzy Syst., vol. 23, no. 3, pp. 701–718, Jun. 2015.
[18] S. Bandyopadhyay and U. Maulik, “Genetic clustering for automatic evolution of clusters and application to image classification,” Pattern Recognit., vol. 35, no. 6, pp. 1197–1208, Jun. 2002.
[19] C.-W. Tsai, W.-L. Chen, and M.-C. Chiang, “A modified multiobjective EA-based clustering algorithm with automatic determination of the number of clusters,” in Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC), Oct. 2012, pp. 2833–2838.
[20] Z. Aliniya and S. A. Mirroshandel, “A novel combinatorial merge-split approach for automatic clustering using imperialist competitive algorithm,” Expert Syst. Appl., vol. 117, pp. 243–266, Mar. 2019.
[21] S. Bandyopadhyay and S. Saha, “A point symmetry-based clustering technique for automatic evolution of clusters,” IEEE Trans. Knowl. Data Eng., vol. 20, no. 11, pp. 1441–1457, Nov. 2008.
[22] H. Yu, Z. Liu, and G. Wang, “An automatic method to determine the number of clusters using decision-theoretic rough set,” Acoust. Bull., vol. 55, nos. 1–2, pp. 101–115, 2014.
[23] S. Saha and S. Bandyopadhyay, “A symmetry based multiobjective clustering technique for automatic evolution of clusters,” Pattern Recognit., vol. 43, no. 3, pp. 738–751, Mar. 2010.
[24] C. Liu, A. Zhou, and G. Zhang, “Automatic clustering method based on evolutionary optimisation,” IET Comput. Vis., vol. 7, no. 4, pp. 258–271, Aug. 2013.

TABLE 11. Results obtained by different algorithms on Data5_2 data set.

| Clustering number | ARI        | Accuracy   |
|-------------------|------------|------------|
| MPA-PSO           | 5±0        | 0.957±0.0046 | 0.9824±0.002 |
| PSO               | 4.2±1.3266 | 0.7722±0.2133 | 0.7904±0.2105 |
| MPA               | 2.6±0.6633 | 0.329±0.1606 | 0.5048±0.1246 |
| DE                | 3.5±0.5    | 0.5429±0.1652 | 0.6712±0.1108 |
| SHO               | 3.1±1.5133 | 0.4291±0.1684 | 0.5404±0.1891 |
| LSA               | 3.6±1.1136 | 0.5931±0.2969 | 0.7044±0.223  |
| EO                | 2.8±0.6    | 0.3665±0.1488 | 0.542±0.1153  |

TABLE 12. Results obtained by different algorithms on Data4_3 data set.

| Clustering number | ARI        | Accuracy   |
|-------------------|------------|------------|
| MPA-PSO           | 3.833±0.3727 | 0.959±0.1005 | 0.9583±0.0932 |
| PSO               | 3.5±0.8062 | 0.7641±0.2528 | 0.8378±0.1587 |
| MPA               | 2.5±0.5    | 0.5222±0.1905 | 0.625±0.125   |
| DE                | 3.1±0.3    | 0.738±0.0853 | 0.7741±0.0747 |
| SHO               | 2.7±0.9    | 0.4788±0.186  | 0.6223±0.1411 |
| LSA               | 3.2±0.6    | 0.7608±0.1923 | 0.8±0.15     |
| EO                | 3±0        | 0.7127±0      | 0.75±0       |
[25] S. Das, A. Abraham, and A. Konar, “Automatic kernel clustering with a multi-elitist particle swarm optimization algorithm,” Pattern Recognit. Lett., vol. 29, no. 5, pp. 688–699, Apr. 2008.

[26] J. Arellano-Verdejo, E. Alba, and S. Gómez-Calderon, “Efficiently finding the optimum number of clusters in a dataset with a new hybrid differential evolution algorithm: DELA,” Soft Comput., vol. 20, no. 3, pp. 895–905, Mar. 2016.

[27] R. J. Kuo and F. E. Zuñiga, “An improved differential evolution with cluster decomposition algorithm for automatic clustering,” Soft Comput., vol. 23, pp. 8957–8973, Aug. 2018.

[28] M. G. H. Omran, A. Salman, and A. P. Engelbrecht, “Dynamic clustering using particle swarm optimization with application in image segmentation,” Pattern Anal. Appl., vol. 8, p. 332, Feb. 2005.

[29] R. J. Kuo, Y. J. Syu, Z.-Y. Chen, and F. C. Tien, “Integration of particle swarm optimization and genetic algorithm for dynamic clustering,” Inf. Sci., vol. 195, pp. 124–140, Jul. 2012.

[30] S.-M. Pan and K.-S. Cheng, “Evolution-based tabu search approach to automatic clustering,” IEEE Trans. Syst., Man, Cybern. C, Appl. Rev., vol. 37, no. 5, pp. 827–838, Sep. 2007.

[31] A. Faramarzi, M. Heidarinejad, S. Mirjalili, and A. H. Gandomi, “Marine predators algorithm: A nature-inspired Metaheuristic,” Expert Syst. Appl., vol. 152, Aug. 2020, Art. no. 113377.

[32] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in Proc. Int. Conf. Neural Netw., Nov./Dec. 1995, pp. 1942–1948.

[33] A. José-García and W. Gómez-Flores, “Automatic clustering using nature-inspired metaheuristics: A survey,” Appl. Soft Comput., vol. 41, pp. 192–213, Apr. 2016.

[34] W. M. Rand, “Objective criteria for the evaluation of clustering methods,” J. Amer. Stat. Assoc., vol. 66, no. 336, pp. 846–850, Dec. 1971.

[35] L. Hubert and P. Arabie, “Comparing partitions,” J. Classification, vol. 2, no. 1, pp. 193–218, Dec. 1985.

[36] H. Li, Statistical Learning Method. Beijing, China: Tsinghua Univ. Press, 2012.

[37] R. Storn and K. Price, “Differential evolution—A simple and efficient heuristic strategy for global optimization over continuous spaces,” J. Glob. Optim., vol. 11, pp. 341–359, Dec. 1997.

[38] G. Dhiman and V. Kumar, “Spotted hyena optimizer: A novel bio-inspired based metaheuristic technique for engineering applications,” Adv. Eng. Softw., vol. 114, pp. 48–70, Dec. 2017.

[39] H. Shareef, A. A. Ibrahim, and A. H. Mutlag, “Lightning search algorithm,” Appl. Soft Comput., vol. 36, pp. 315–333, Nov. 2015.

[40] A. Faramarzi, M. Heidarinejad, B. Stephens, and S. Mirjalili, “Equilibrium optimizer: A novel optimization algorithm,” Knowl.-Based Syst., vol. 191, Mar. 2020, Art. no. 105190.

[41] J. S. WANG (Member, IEEE) received the B.Sc. and M.Sc. degrees in control science and engineering from the University of Science and Technology Liaoning, China, in 1999 and 2002, respectively, and the Ph.D. degree in control science and engineering from the Dalian University of Technology, China, in 2006. He is currently a Professor and a Doctor’s Supervisor with the School of Electronic and Information Engineering, University of Science and Technology Liaoning. His main research interests include modeling of complex industry process, intelligent control, and computer integrated manufacturing.

[42] L. F. ZHU received the B.Eng. degree in electrical engineering and automation from the Liren College, Yanshan University, Qinhuangdao, China, in 2018. She is currently pursuing the M.Eng. degree with the School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, China. Her research interests include data mining, clustering, and meta-heuristic algorithms.

[43] H. Y. WANG received the B.Eng. degree in measuring technology and instrument from the University of Science and Technology Liaoning, China, in 2019, where he is currently pursuing the M.Eng. degree with the School of Electronic and Information Engineering. His main research interests include meta-heuristic algorithms and machine learning.

[44] G. WANG received the B.Eng. degree in automation from the University of Science and Technology Liaoning, Anshan, China, in 2020, where he is currently pursuing the M.Eng. degree with the School of Control Science and Engineering. His research interests include machine learning and intelligent optimization algorithm.

N. WANG is currently pursuing the Ph.D. degree with the College of Economics and Management, Nanjing University of Aeronautics and Astronautics, Nanjing, China. He is currently a Researcher with the China Center for Aerospace Science and Technology International Communications, Beijing, China. His main research interests include meta-heuristic algorithms and machine learning.