

Research Article

A Feature Weighted Fuzzy Clustering Algorithm Based on Multistrategy Grey Wolf Optimization

Yongli Liu , Zhonghui Wang , and Hao Chao 

School of Computer Science and Technology, Henan Polytechnic University, Jiaozuo 454003, Henan, China

Correspondence should be addressed to Yongli Liu; yongli.buaa@gmail.com

Received 18 May 2021; Revised 9 July 2021; Accepted 26 August 2021; Published 7 September 2021

Academic Editor: Peng-Yeng Yin

Copyright © 2021 Yongli Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Traditional fuzzy clustering is sensitive to initialization and ignores the importance difference between features, so the performance is not satisfactory. In order to improve clustering robustness and accuracy, in this paper, a feature-weighted fuzzy clustering algorithm based on multistrategy grey wolf optimization is proposed. This algorithm cannot only improve clustering accuracy by considering the different importance of features and assigning each feature different weight but also can easily obtain the global optimal solution and avoid the impact of the initialization process by implementing multistrategy grey wolf optimization. This multistrategy optimization includes three components, a population diversity initialization strategy, a nonlinear adjustment strategy of the convergence factor, and a generalized opposition-based learning strategy. They can enhance the population diversity, better balance exploration and exploitation, and further enhance the global search capability, respectively. In order to evaluate the clustering performance of our clustering algorithm, UCI datasets are selected for experiments. Experimental results show that this algorithm can achieve higher accuracy and stronger robustness.

1. Introduction

Clustering technology is widely used in data mining, pattern recognition, machine learning, and image processing [1–4]. Existing algorithms can be divided into hard clustering and fuzzy clustering. Fuzzy C-Means (FCM) [5] is the representative algorithm of fuzzy clustering. This algorithm constructs an objective function based on the intraclass distance according to the principle of intraclass compactness and obtains the final clustering result by optimizing the objective function. The FCM algorithm is simple and efficient, but its disadvantages are obvious. The FCM algorithm uses the Lagrange multiplier method to solve the iterative formula of membership degree and centroid. This method is not only sensitive to initialization but also cannot guarantee that the clustering result is globally optimal. Accordingly, Zhang et al. [6] used the particle swarm optimization (PSO) algorithm to find the global best quality center and obtained the clustering results. Jiang et al. [7] proposed a multiview clustering algorithm using PSO with double weights where PSO was used to find the centroid in each iteration.

The grey wolf optimizer algorithm has the following advantages: (1) strong global search ability; (2) faster convergence speed and better local search ability; (3) simple principle, few parameters, and easy to operate and realize. Gupta and Deep [8] proposed an improved algorithm RW-GWO based on random walk to improve grey wolf's search ability, and the algorithm is employed to find the optimal setting for the directional overcurrent relay problem which is a highly complex problem. In order to alleviate the problem of premature convergence due to the stagnation at suboptimal solutions in classical GWO, an improved leadership-based GWO called GLF-GWO [9] is proposed. The GLF-GWO algorithm enhances the search efficiency of leading hunters in GWO and provides better guidance to accelerate the search process of GWO. Yichu [10] proposed a Fuzzy C-Means clustering algorithm based on grey wolf optimization (GWOFM), which has better clustering stability and robustness. However, the GWOFM algorithm, just like the traditional FCM algorithm, ignores the importance difference between features. Keller et al. [11] proposed a feature weighted fuzzy

clustering algorithm, which gives different weights to each feature and achieves better clustering effect. Similarly, Zhou et al. [12] calculated the weight of each feature by using the maximum information entropy principle, proposed the EWFCM algorithm, and improved the accuracy of clustering.

Based on the above analysis, this paper proposes a feature weighted Fuzzy C-Means Clustering algorithm based on multistrategy grey wolf optimization (MSGWO-WFCM). This algorithm not only gives different weights to each feature but also uses GWO to find the global optimal centroid, which not only achieves the global best but also solves the sensitive problem of the initialization process.

The rest of this paper is organized as follows. Section 2 introduces some algorithms related to our work. In Section 3, we improve GWO and propose the multistrategy GWO. In Section 4, we detail the feature weighted clustering algorithm based on MSGWO. Section 5 demonstrates the effectiveness of our algorithm by carrying out experiments. Finally, the conclusions are given in Section 6.

2. Related Work

2.1. The WFCM Algorithm. Given the dataset X with N samples, expressed as $X = \{x_1, \dots, x_N\}$, where each sample has D -dimensional features, and v_1, \dots, v_k are K centroids. Let U be a $K \times N$ matrix, whose element u_{ci} is the membership of the i th object to the c cluster. The WFCM algorithm introduces weight matrix W to distinguish the importance of different features. The objective function and constraints of this algorithm are shown as following equations:

$$J_{\text{WFCM}}(U, V, W) = \sum_{c=1}^K \sum_{i=1}^N u_{ci}^m \sum_{d=1}^D w_{cd}^t x_{id} - v_{cd}^2, \quad (1)$$

$$\sum_{c=1}^K u_{ci} = 1, u_{ci} \in [0, 1], \quad (2)$$

$$\sum_{d=1}^D w_{cd} = 1, w_{cd} \in [0, 1], \quad (3)$$

where w_{cd} is the weight of the d th feature in the c th category and m and t are two parameters to control the distribution of membership degree and feature weight, respectively. The Lagrange multiplier method is used to solve membership degree and centroid, and the Lagrange function is constructed as follows:

$$L_{\text{WFCM}}(U, V, W) = J_{\text{WFCM}} + \sum_{i=1}^N \lambda_i \left(\sum_{c=1}^K u_{ci} - 1 \right) + \sum_{c=1}^K \mu_c \left(\sum_{d=1}^D w_{cd} - 1 \right). \quad (4)$$

By calculating the partial derivative of equation (4), the updating formulas of membership degree, centroid, and feature weight can be obtained, as shown in following equations:

$$u_{ci} = \left[\sum_{p=1}^K \left(\frac{\sum_{d=1}^D w_{cd}^t x_{id} - v_{cd}^2}{\sum_{d=1}^D w_{pd}^t x_{id} - v_{pd}^2} \right)^{1/m-1} \right]^{-1}, \quad (5)$$

$$v_{cd} = \frac{\sum_{i=1}^N u_{ci}^m x_{id}}{\sum_{i=1}^N u_{ci}^m}, \quad (6)$$

$$w_{cd} = \left[\sum_{p=1}^D \left(\frac{\sum_{i=1}^N u_{ci}^m x_{id} - v_{cd}^2}{\sum_{i=1}^N u_{ci}^m x_{ip} - v_{cp}^2} \right)^{1/t-1} \right]^{-1}. \quad (7)$$

The WFCM algorithm completes the clustering process by iterating the updating formula of membership, centroid, and feature weight alternately.

2.2. Original Grey Wolf Optimization. The GWO algorithm is inspired by the social hierarchy and hunting behaviour of grey wolf population [13]. Grey wolf is a predator at the top of the food chain. Most of them like to live in groups. The average size of each population is 5–12. Grey wolf population has a very strict social ruling class, which is divided into four levels from top to bottom, α wolf, β wolf, δ wolf, and ω wolves. The α wolf is mainly responsible for hunting, choosing habitat, and other decisions; the β wolf is the second leader, who is mainly responsible for assisting the leader to manage the group; the δ wolf is mainly responsible for reconnaissance, taking care of the old and the weak, hunting, etc. And, the ω wolves are at the bottom of the population. Although the ω wolves are subordinate to other grey wolves, they are indispensable to balance the internal relationship of the population.

The hunting process of wolves includes the following stages:

2.2.1. Encircling Prey. In this stage, grey wolves surround their prey during hunting. The mathematical formula is as follows:

$$\begin{aligned} \vec{D} &= \left| \vec{C} \cdot \vec{X}_p(t) - \vec{X}(t) \right|, \\ \vec{X}(t+1) &= \vec{X}_p(t) - \vec{A} \cdot \vec{D}, \end{aligned} \quad (8)$$

where t represents the current iteration, \vec{A} and \vec{C} are two coefficient vectors, \vec{X}_p is the vector position of the target, and \vec{X} represents the position vector of a grey wolf. \vec{A} and \vec{C} are obtained from the following formula:

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a}, \quad (9)$$

$$\vec{C} = 2 \cdot \vec{r}_2. \quad (10)$$

In the iterative process, the value of \vec{a} decreases linearly from 2 to 0, and \vec{r}_1 and \vec{r}_2 are two random vectors in the interval (0, 1).

2.2.2. Hunting. Hunting is usually guided by the α wolf, and the β and δ wolves may also go hunting. In order to simulate the hunting behaviour of the grey wolf, we assume that α (the

best candidate solution), β , and δ wolves better understand the potential location of prey. Therefore, the first three optimal solutions obtained so far are preserved, and other grey wolves (including the ω wolves) update its own position according to these optimal solutions. The specific mathematical model is as follows:

$$\begin{aligned}\bar{D}_\alpha &= \left| \bar{C}_1 \cdot \bar{X}_\alpha - \bar{X} \right|, \\ \bar{D}_\beta &= \left| \bar{C}_2 \cdot \bar{X}_\beta - \bar{X} \right|,\end{aligned}\quad (11)$$

$$\begin{aligned}\bar{D}_\delta &= \left| \bar{C}_3 \cdot \bar{X}_\delta - \bar{X} \right|, \\ \bar{X}_1 &= \bar{X}_\alpha - \bar{A}_1 \cdot \left(\bar{D}_\alpha \right), \\ \bar{X}_2 &= \bar{X}_\beta - \bar{A}_2 \cdot \left(\bar{D}_\beta \right), \\ \bar{X}_3 &= \bar{X}_\delta - \bar{A}_3 \cdot \left(\bar{D}_\delta \right),\end{aligned}\quad (12)$$

$$\bar{X}_{t+1} = \frac{\bar{X}_1 + \bar{X}_2 + \bar{X}_3}{3}.\quad (13)$$

2.2.3. Attacking Prey (Exploitation). When the prey stops moving, the wolf begins to attack. In order to approach the prey, with the decrease of the a value, the fluctuation range of A also decreases. In other words, in the iterative process, when the value of a decreases from 2 to 0, A takes a random value within $[-a, a]$. When the value of A is within the range $[-1, 1]$, the next position of the wolf can be any position between the current position and the prey position. When $|A| < 1$, the wolves attack prey.

2.2.4. Search for Prey (Exploration). In this stage, the grey wolves search for prey based on the locations of α , β , and δ . At first, they separate from each other, and then, they will gather to attack prey. In order to simulate divergence mathematically, use random values greater than 1 or less than -1 to force the grey wolf to deviate from its prey, which can improve the exploration ability of this algorithm and make it realize global search.

2.3. Some Improved Versions of GWO. In recent years, in order to further improve the optimization accuracy and search efficiency, many researchers have tried to improve GWO. Saremi et al. [14] improved GWO by introducing the dynamic evolutionary population operator, which enhanced the local search ability and accelerated the convergence speed. Jayabarath et al. [15] embedded crossover and mutation operators to improve GWO and helped solve economic scheduling problems. In order to improve the population diversity of GWO, a levy-embedded GWO (LGWO) [16], by combining Levy flight and greedy selection strategy with the improved hunting stage, was proposed. A hybrid algorithm of the biogeography optimization

algorithm and GWO is proposed [17] to help GWO jump out of the local optimum. In order to overcome the premature convergence problem of GWO, Wang et al. [18] combined the basic GWO with the Gaussian GED algorithm and proposed the GEDGWO algorithm. Besides, a memory-based grey wolf optimizer (mGWO) [19] is also proposed to make the balance between exploitation and exploration more stable.

3. Multistrategy Grey Wolf Optimization (MSGWO)

This paper improves GWO from three aspects: (1) in the aspect of population initialization, the strategy of population diversity initialization is proposed to improve the population diversity; (2) the nonlinear adjustment strategy of the convergence factor is proposed to balance exploration and exploitation better; (3) a general reverse learning strategy is proposed to further enhance the global search ability.

3.1. Population Diversity Initialization Strategy. When solving the function optimization problem, GWO generates initialization population randomly. The stochastic method cannot guarantee that the initial population covers the decision space of the problem well, and it is easy to lose the diversity of the population. To solve this problem, we propose a semiuniform and semirandomized initialization method. In this method, half of the population still uses the random initialization method, and the other half is generated by the method of global homogenization and then local random. In this initialization method, the search space $X_i (i = 1, 2, \dots, K)$ is evenly divided into several equal length subspaces whose size equals to one half of the population, and the initial individual value needs to be randomly generated in the randomly selected subspace. Each subspace has and only has one chance to generate individuals, which makes the GWO algorithm retain the randomness of the population on the basis of overall uniformity and avoids the problem of overconcentration of initialization positions caused by random population initialization. Therefore, the population can be relatively evenly distributed in the whole search space, which ensures the randomness and diversity of the population [20]. Algorithm 1 gives the initialization flow of semihomogenization and semirandomization.

3.2. Nonlinear Adjustment Strategy of Convergence Factor. It can be seen from equation (12) that parameter A plays a crucial role in coordinating the global and local exploitation capabilities of the GWO algorithm. When $|A| > 1$, the group will expand the search scope, and GWO has strong exploration ability. And, when $|A| < 1$, the group will narrow the search scope and search in local areas, and therefore, GWO has strong exploitation ability. Equation (9) shows that A changes with the convergence factor a , whose value decreases linearly from 2 to 0 with the increase of iteration times. As we all know, the optimization process of GWO is a complex nonlinear change process, and the linear change of convergence factor a obviously cannot reflect the actual

optimization search process. Wei [21] proposed that the convergence factor a changes nonlinearly with the number of iterations in the grey wolf optimization algorithm with nonlinear adjustment strategy of the control parameter value. The standard function test results show that this nonlinear strategy has better optimization performance than the linear strategy. Inspired by the inertia weight update in the improved PSO algorithm [22], we propose a nonlinear change convergence factor update method:

$$a(t) = a_{\text{initial}} - (a_{\text{initial}} - a_{\text{final}}) \times \left(\frac{t}{t_{\text{max}}} \right)^k, \quad (14)$$

where a_{initial} and a_{final} are the initial and terminal values of convergence factor a , respectively, t is the current number of iterations, t_{max} is the maximum number of iterations, and k ($k > 0$) is the nonlinear adjustment coefficient. In equation (14), the convergence factor a changes nonlinearly with the increase of the number of iterations, which guarantees the effective balance of the global search and local search ability of our algorithm.

3.3. Generalized Opposition-Based Learning. Opposition based learning (OBL) [23] can improve the search performance. Based on OBL, Wang et al. [23] proposed the concept of generalized opposition-based learning (GOBL), and the experiment shows that the strategy has more advantages.

In order to coordinate the exploration and exploitation ability, this paper implements a GOBL strategy for all individuals in the current population. By combining the opposite population with the current population, the excellent individuals are selected into the next generation population to enhance the diversity of the population, which can effectively reduce the probability of the algorithm falling into the local optimum.

3.4. MSGWO. The steps of the MSGWO algorithm mainly include the following:

Step 1: set the parameters, including the size of the population N , the dimension d , the maximum number of iterations t_{max} , the initial value of the convergence factor a_{initial} , the terminal value of the convergence factor a_{final} , and the adjustment coefficient k , and initialize a , A , and C

Step 2: generate N individuals as the initial population in the search space, and let $t = 1$

Step 3: calculate the fitness value of each individual grey wolf, and select the best three grey wolf X_{α} , X_{β} , and X_{δ}

Step 4: if $t < t_{\text{max}}$, update the position of each individual in the group; otherwise, the algorithm ends.

Step 5: implement the GOBL strategy for all individuals in the current population, update the position of each individual, and select the positions of the top three fitness values as the positions of X_{α} , X_{β} , and X_{δ} , respectively

Step 6: calculate the value of convergence factor a according to equation (14), and then, calculate the values of A and C according to equations (9) and (10).

Step 7: if the convergence condition is met, this algorithm will end; otherwise, let $t = t + 1$ and return to Step 4.

The pseudocode of the MSGWO algorithm is listed as Algorithm 2.

4. MSGWO-WFCM

This paper proposes a feature weighted fuzzy clustering algorithm based on multistrategy grey wolf optimization (MSGWO-WFCM). In the process of clustering, the MSGWO algorithm is used instead of the Lagrange multiplier method to find the optimal centroids, which can ensure that our algorithm is not only easy to find the global optimal solution but also insensitive to the initialization process.

4.1. Fitness Value Function. Fitness value function is the benchmark for evaluating the quality of individuals. The larger the function value is, the better the individual is, and vice versa. In GWO, this function is used to judge the grey wolf level. Specifically, the α , β , and δ wolves with highest fitness remain, and guide ω wolves to search for prey. In this paper, the fitness function is set as

$$f(V_i) = \frac{1}{J_{\text{WFCM}}(U_i, V_i, W_i)}. \quad (15)$$

4.2. Algorithm Steps. The implementation steps of the MSGWO-WFCM algorithm are as follows:

Step 1: set the parameters, including population size N , dimension d , maximum number of iterations t_{max} and adjustment coefficient k , and initialize a , A , and C

Step 2: generate N individuals as the initial population in the search space, and let $t = 1$

Step 3: if $t < t_{\text{max}}$, calculate the fitness value f of each individual according to equation (15), and select the three wolves X_{α} , X_{β} , and X_{δ} with the least fitness; otherwise, this algorithm will end

Step 4: update the values of parameters a , A , and C

Step 5: update the position of each individual according to equations (11)–(13)

Step 6: implement the GOBL strategy for all individuals in the current population, and update the location of each individual

Step 7: recalculate the fitness value f of each individual according to equation (15)

Step 8: if the convergence condition is satisfied, the algorithm will end; otherwise, let $t = t + 1$ and return to Step 3

The pseudocode of the MSGWO-WFCM algorithm is given as Algorithm 3.

Input: population size n , dimension d of search space, interval $[l_j, u_j]$ of variables $X_j(j \in [1:d])$ in search space.
Output: initial population

- (1) for $j = 1$ to d
- (2) for $i = 1$ to $n/2$
- (3) randomly generate a value from $[l_j, u_j]$ to assign to X_j^i
- (4) end for
- (5) $\Delta j = (u_j - l_j) / (n/2)$
- (6) $\nabla j = \{[l_j, l_j + \Delta j], [l_j + \Delta j, l_j + 2\Delta j], \dots, [l_j + (n/2 - 1)\Delta j, u_j]\}$
- (7) for $(n/2) + 1$ to n
- (8) randomly select a subinterval from the set ∇j , and randomly generate a value to assign to X_j^i
- (9) update set ∇j : delete the subinterval selected in Step 8 from set ∇j
- (10) end for
- (11) end for
- (12) output the initial population $\{X^1, X^2, \dots, X^n\}$

ALGORITHM 1: Semihomogenization and semirandomization.

- (1) set the parameters
- (2) generate N individuals as initial population in search space according to algorithm 1
- (3) calculate the fitness value of each wolf
- (4) determine the values of X_{α} , X_{β} , and X_{δ} and let $t = 0$
- (5) while $t < t_{\max}$
- (6) for $i = 1$ to N
- (7) update the position of the i th grey wolf according to equations (11)–(13)
- (8) end for
- (9) implement the GOBL strategy for all individuals in the current population to update the position of each individual
- (10) calculate the fitness value of each wolf
- (11) update and save X_{α} , X_{β} , and X_{δ}
- (12) calculate the value of convergence factor a according to equation (14), and then, calculate the values of A and C according to equations (9) and (10)
- (13) $t = t + 1$
- (14) end while
- (15) return X_a

ALGORITHM 2: MSGWO.

5. Experiments

5.1. MSGWO Algorithm Experiments

5.1.1. Benchmark Functions. In order to evaluate the effectiveness of the MSGWO algorithm, eight benchmark test functions (as Table 1) are selected for experiments. In Table 1, f_1 – f_5 are unimodal test functions and f_6 – f_8 are multimodal test functions. Each algorithm runs 30 times independently, and its average value is used to reflect the convergence accuracy, and the standard deviation is used to reflect the stability.

5.1.2. Experimental Results and Analysis. The MSGWO algorithm is used to carry out numerical experiments on above eight standard test functions. The results are compared with those of PSO [24], GWO, HGSO [25], AO [26], AOA [27], and MRFO [28] algorithms in terms of the average value and standard deviation. To achieve a fair comparison, the iteration number and population size of all optimizers are set to 500 and 30, respectively. The values used for the main controlling parameters of the comparative algorithms can be seen in Table 2. The

analysis has been performed on MATLAB 2018a platform on a computer with a Windows 10 64 bit professional and 16 GB RAM.

Table 3 shows the experimental results on 8 standard test functions. It can be seen from Table 3 that, for functions f_1 , f_3 , f_6 , and f_7 , MSGWO converges to the theoretical optimal value 0. For functions f_4 and f_5 , the average value of the MSGWO algorithm in 30 experiments is very close to the global optimal solution. In addition, compared with other algorithms, the standard deviations of MSGWO on eight standard functions are the minimum, and seven of them are all 0, indicating that the stability of the MSGWO algorithm is better. In the case of the same population size and iteration times, compared with PSO, GWO, HGSO, AO, AOA, and MRFO algorithms, the MSGWO algorithm has better average and standard deviation and, therefore, has advantages in stability and optimization performance.

The diversity analysis in MSGWO can be done by comparing the diversity curves of classical GWO and MSGWO. These diversity curves are drawn by considering the average distance between the solutions in each iteration. To calculate the average distance, the Euclidean distance $\| \cdot \|$

Input: dataset $X = \{x_1, \dots, x_N\}$, number of clusters K , parameters m and t , number of population n , initial value a_{initial} and terminal value a_{final} of the convergence factor, threshold parameter ξ , maximum iterations t_{max} and fitness value $\text{ft}(V_t)$ of t generation population

Output: clustering result vector q

- (1) Use the strategy of population diversity to initialize n populations. The centroid matrix of the i th population is V_i ($i = 1, \dots, n$);
- (2) Initialize the weight matrix W_i corresponding to the i -th particle;
- (3) Let $t = 1$;
- (4) for $i = 1$ to n
- (5) Update u_{ci} with equation (5);
- (6) Update w_{cd} with equation (7);
- (7) Update the fitness value $\text{f1}(V_i)$ of the i th population with equation (15);
- (8) end for
- (9) while $t < t_{\text{max}}$
- (10) Select the best three wolves x_α , x_β , and x_δ according to the fitness value;
- (11) Calculate the value of convergence factor a according to equation (14), and calculate the values of A and C according to equations (9) and (10);
- (12) Update the position of each individual according to equations (11)–(13);
- (13) Implement the GOBL strategy for all individuals in current population, and update the position of each individual;
- (14) $t = t + 1$;
- (15) for $i = 1$ to n
- (16) Update u_{ci} with equation (5);
- (17) Update w_{cd} with equation (7);
- (18) Update the fitness value $\text{ft}(V_i)$ of the i th population with equation (15);
- (19) end for
- (20) if $(\min \text{ft}(V_i) - \min \text{ft-1}(V_i)) < \xi$
- (21) break;
- (22) end if
- (23) end while
- (24) Output membership matrix U
- (25) $q(i) = \arg \max_{1 \leq c \leq K} u_{ci}, i = 1, \dots, N$

ALGORITHM 3: MSGWO-WFCM.

between two solutions $X = (x_1, x_2, \dots, x_d)$ and $Y = (y_1, y_2, \dots, y_d)$ is calculated as follows:

$$\|X - Y\|_2 = \sqrt{\sum_{j=1}^d (x_j - y_j)^2}. \quad (16)$$

From the diversity curves drawn in Figure 1, it can be analyzed that the average distance between the search agents in MSGWO is less than classical GWO, which shows the better balance between the exploration and exploitation and a better convergence rate of MSGWO. It can also be observed that the leading hunters are improved through nonlinear convergence factor strategy as compared to classical GWO because MSGWO provides better solution to these functions and the average distance between the search agents is less than classical GWO.

Due to space limitation, Figure 2 only shows the convergence curves of MSGWO and comparison algorithms for six functions with fixed number of iterations. It can be clearly seen that, compared with PSO, GWO, HGSO, AO, AOA, and MRFO, our MSGWO algorithm has faster convergence speed and higher convergence accuracy.

5.1.3. Influence of Parameters. In GWO, the nonlinear adjustment parameter k controls the change of the convergence factor and, therefore, has a great influence on the performance.

In this section, five different values ($k = 1/3, k = 1/2, k = 1, k = 2$, and $k = 3$) are selected to analyse the influence on the performance of MSGWO through numerical experiments. The experimental results of MSGWO with different adjustment coefficient k are given in Table 4.

It can be seen from Table 3 that when $k = 1/3$, the optimization performance of MSGWO is the best. For functions f_6, f_7 , and f_8 , the value of k has little effect on MSGWO. For functions f_1, f_3 , and $k = 1/3, 1/2$, and 1 , the MSGWO algorithm can converge to the theoretical optimal value 0 , which is better than the results of the other two groups. For functions f_2, f_4 , and $k = 1/3$ and $1/2$, the MSGWO algorithm can converge to the theoretical optimal value 0 , which is better than the results of the other three groups; for the other three groups, the smaller the value of k , the closer the MSGWO algorithm can converge to the theoretical optimal value. For function f_5 , the smaller the value of k , the closer the MSGWO algorithm can converge to the theoretical optimal value. The experimental results show that the parameter k has a certain influence on the results of MSGWO.

5.2. MSGWO-WFCM Clustering Experiments

5.2.1. Experimental Preparation. In order to evaluate the clustering effect of our MSGWO-WFCM algorithm, FCM, GWOFCM, HPSOFCM [29], WFCM, EWFCM, and

TABLE 1: Benchmark test functions.

Function	Dim	Formula	Range	F_{\min}
<i>Sphere</i>	30	$f_1(x) = \sum_{i=1}^n x_i^2$	[-100,100]	0
<i>Schwefel 2.22</i>	30	$f_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	[-10,10]	0
<i>Schwefel 1.2</i>	30	$f_3(x) = \sum_{i=1}^n (\sum_{j=1}^n x_j)^2$	[-100,100]	0
<i>Schwefel 2.21</i>	30	$f_4(x) = \max\{ x_i , 1 \leq x_i \leq n\}$	[-100,100]	0
<i>Quartic</i>	30	$f_5(x) = \sum_{i=1}^n ix_i^4 + \text{random}[0, 1]$	[-1.28,1.28]	0
<i>Rastrigin</i>	30	$f_6(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	[-5.12,5.12]	0
<i>Criewank</i>	30	$f_7(x) = 1/4000 \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(x_i/\sqrt{i}) + 1$	[-600,600]	0
<i>Ackley</i>	30	$f_8(x) = -20 \exp(-0.2\sqrt{1/n \sum_{i=1}^n x_i^2} - \exp(1/n \sum_{i=1}^n \cos(2\pi x_i)) + 20 + e)$	[-32,32]	0

TABLE 2: Parameter values for the comparative algorithms.

Algorithm	Parameter	Value
PSO	Cognitive and social constant	(C1, C2) 2, 2
	Inertia weight	Linear reduction from 0.9 to 0.1
	Velocity limit	10% of dimension range
GWO	Convergence parameter (a)	Linear reduction from 2 to 0
	M_1	0.1
HGSO	M_2	0.2
	a	1
	β	1
	K	1
AO	a	0.1
	δ	0.1
AOA	a	5
MRFO	μ	0.5
MSGWO	S	2
	Convergence parameter (a)	Nonlinear reduction from 2 to 0
	k	1/3

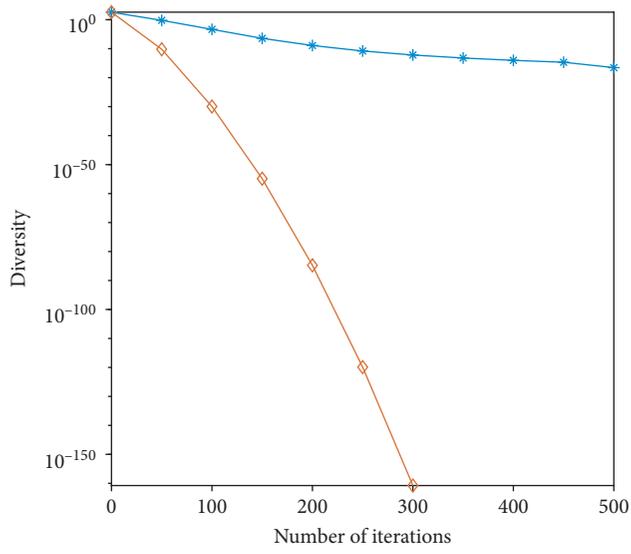
TABLE 3: Comparison of the optimization results of seven algorithms on eight test functions.

Function		PSO	GWO	HGSO	AO	AOA	MRFO	MSGWO
f_1	Ave	2.38E-28	6.86E-28	4.03E-183	6.81E-108	2.42E-08	0.00E+00	0.00E+00
	Std	3.02E-04	7.16E-28	0.00 E+00	3.73E-107	7.36E-08	0.00E+00	0.00E+00
f_2	Ave	3.84E-02	9.13E-17	5.69E-93	7.39E-72	0.00 E+00	6.56E-207	0.00E+00
	Std	5.11E-02	5.51E-17	3.06E-92	3.88E-71	0.00 E+00	0.00E+00	0.00E+00
f_3	Ave	9.01E+01	1.49E-05	6.29E-147	2.59E-109	9.68E-04	0.00E+00	0.00E+00
	Std	3.26E+01	3.85E-05	3.32E-147	1.42E-108	7.69E-04	0.00E+00	0.00E+00
f_4	Ave	1.08E+00	8.85E-07	3.11E-89	3.71E-52	3.50E-15	2.00E-198	0.00E+00
	Std	2.04E-01	1.22E-06	1.69E-88	2.03E-51	1.92E-14	0.00E+00	0.00E+00
f_5	Ave	1.61E-01	1.80E-03	1.60E-04	1.00E-04	7.11E-05	1.32E-04	6.65E-05
	Std	5.34E-02	2.11E-04	1.00E-03	1.16E-04	5.92E-05	1.44E-04	5.51E-05
f_6	Ave	5.13E+01	1.48E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Std	1.38E+01	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_7	Ave	7.60E-03	3.40E-03	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Std	1.03E-02	7.10E-03	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_8	Ave	2.49E-01	9.91E-14	1.13E-15	8.88E-16	8.88E-16	8.88E-16	8.88E-16
	Std	4.95E-01	1.32E-14	9.01E-16	0.00E+00	8.88E-16	0.00E+00	0.00E+00

SFWFCM [30] are selected for comparative experiments. In our experiment, the values used for the main parameters of the clustering algorithms can be seen in Table 5.

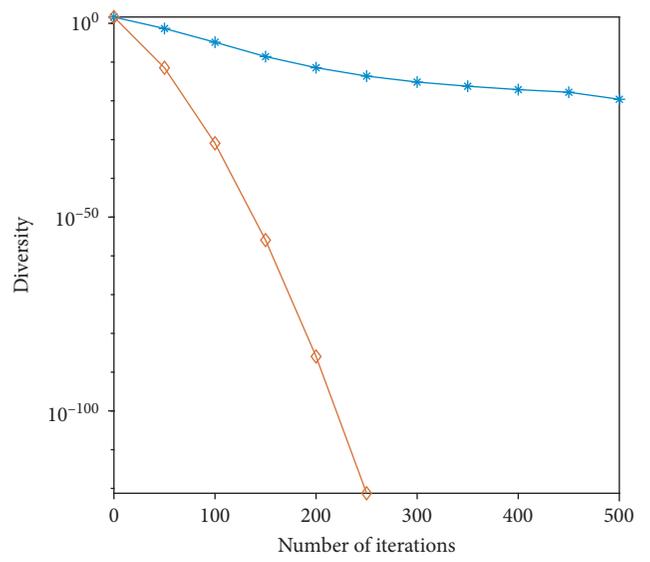
Six standard datasets are selected from UCI database, and the information of each dataset is shown in Table 6. The accuracy is selected as the evaluation index for the experimental results.

In the six datasets, Iris is a classic dataset in machine learning, and it contains three categories, each of which contains 50 samples. The Hab dataset comes from a study on the survival rate of patients with breast cancer surgery. The Ion dataset is a radar dataset collected by the system in Goose Bay, Labrador. The TSE dataset was provided by students from Gazi University in Turkey. The ecoli dataset



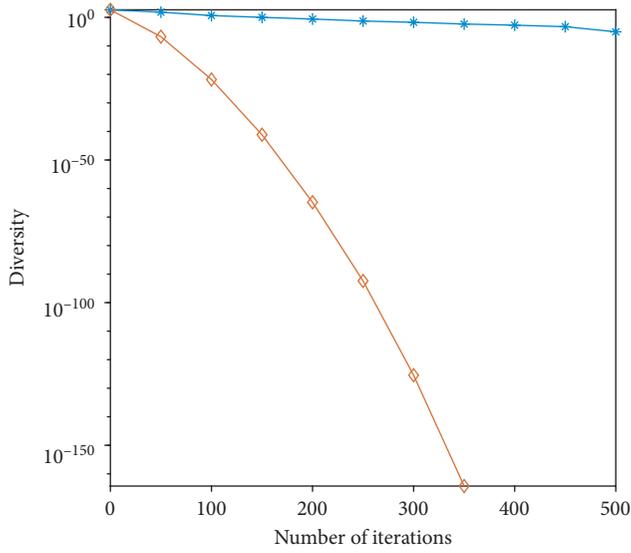
—*— GWO
—◇— MSGWO

(a)



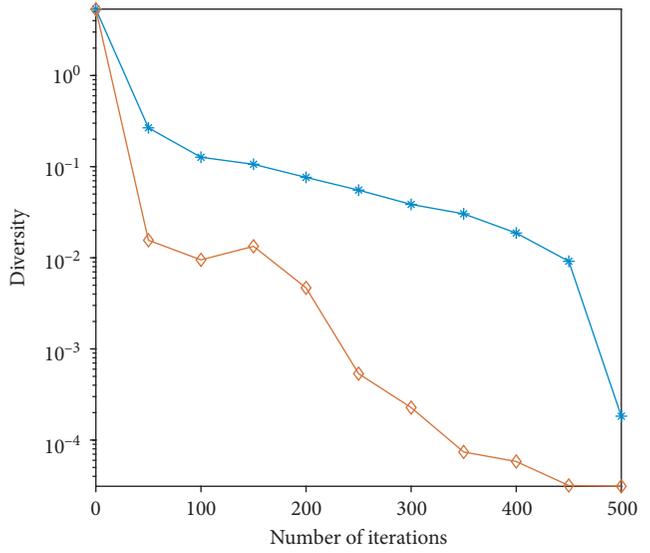
—*— GWO
—◇— MSGWO

(b)



—*— GWO
—◇— MSGWO

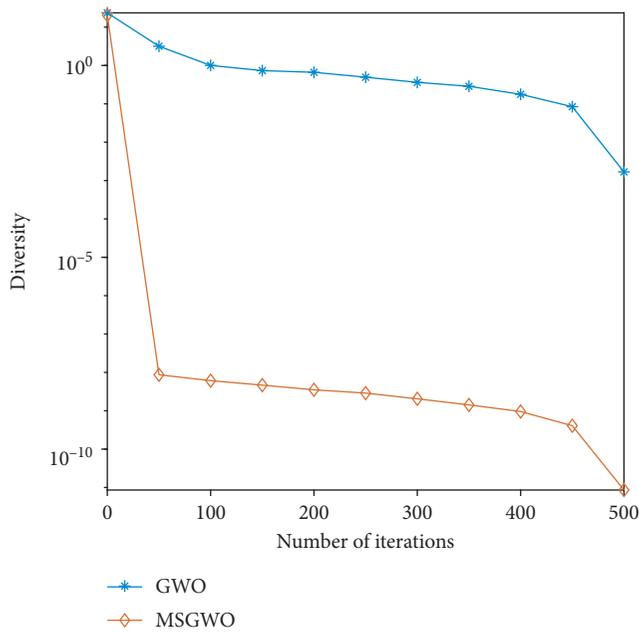
(c)



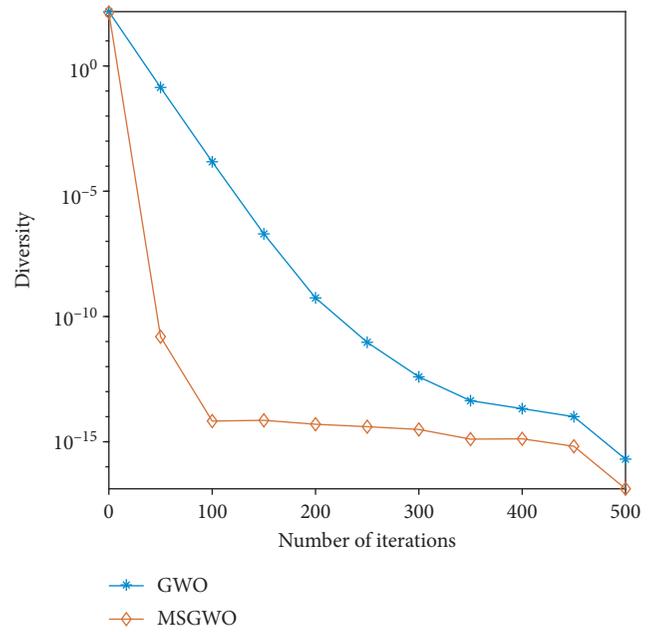
—*— GWO
—◇— MSGWO

(d)

FIGURE 1: Continued.



(e)



(f)

FIGURE 1: Diversity analysis of six test functions. (a) *Sphere*. (b) *Schwefel 2.22*. (c) *Schwefel 1.2*. (d) *Quartic*. (e) *Rastrigin*. (f) *Ackley*.

contains 336 samples, and each sample consists of 7 features. The pendigits dataset is a digital dataset created by 250 samples from 44 authors.

5.2.2. Experimental Results and Analysis. When initializing the centroids, C samples are randomly selected as the centroid matrix. Each algorithm is run for 10 times, and the average results are taken as the final experimental results, which are shown in Table 7.

It can be seen from Table 7 that the accuracy of the MSGWO-WFCM algorithm is the highest on the Iris dataset, 6.8% higher than FCM, 9% higher than WFCM, 4.6% higher than EWFCM, and 9.67% higher than the SFWFCM algorithm. On the Hab dataset, the accuracy of the MSGWO-WFCM algorithm is only 1.51% higher than that of the WFCM algorithm and 1.77% higher than that of the EWFCM algorithm, but 24.97% higher than FCM and 24.22% higher than GWOFCM. On the Ion dataset, the accuracy of the MSGWO-WFCM algorithm is 1.2% higher than that of EWFCM, 4.42% higher than that of SFWFCM, 6.64% higher than that of GWOFCM, and 11.11% higher than that of WFCM. On the TSE dataset, the accuracy of the MSGWO-WFCM algorithm is equal to that of WFCM, 0.8% higher than EWFCM, 3.51% higher than FCM, 10.2% higher than HPSOFCM, and 14.79% higher than the GWOFCM algorithm. On the ecoli dataset, the accuracy of MSGWO-WFCM is 22.6% higher than WFCM, 2.26% higher than FCM, 0.68% higher than HPSOFCM, and 6.81% higher than the GWOFCM algorithm. On the pendigits dataset, the accuracy of MSGWO-WFCM is 2.57% higher than WFCM, 3.93% higher than GWOFCM,

3.76% higher than EWFCM, 6.6% higher than FCM, 11.06% higher than SFWFCM, and 16.79% higher than the HPSOFCM algorithm. It is obvious that the accuracy of the MSGWO-WFCM algorithm is better than that of the contrast algorithms.

In order to evaluate the ability of the MSGWO-WFCM algorithm to find the optimal solution, experimental results of seven algorithms running for 10 times are shown as Figure 3.

Figure 3(a) shows the results on Iris dataset. From Figure 3(a), we can see that the MSGWO-WFCM algorithm is more stable than WFCM, EWFCM, and SFWFCM and more accurate than the FCM, GWOFCM, and HPSOFCM algorithm. From Figure 3(b), it can be seen that the accuracy of the WFCM algorithm and MSGWO-WFCM algorithm is much higher than the FCM and GWOFCM algorithm on the Hab dataset, and the MSGWO-WFCM algorithm is more stable than HPSOFCM, EWFCM, and SFWFCM. The results on the Ion dataset are illustrated as Figure 3(c). This group of results shows that the MSGWO-WFCM algorithm is superior to the contrast algorithms in terms of accuracy and stability. From Figure 3(d), the results on the TSE dataset, it can be seen that the accuracy of the MSGWO-WFCM algorithm is higher than that of the FCM, GWOFCM, HPSOFCM, and SFWFCM algorithm and equivalent to the WFCM and EWFCM algorithm. From Figure 3(e), although the MSGWO-WFCM algorithm has some fluctuation on the ecoli dataset, its accuracy is still higher than that of the other six algorithms. In Figure 3(f) of the pendigits dataset, the MSGWO-WFCM algorithm also has some fluctuations, but it has less fluctuation than WFCM, and its accuracy is higher than that of other five algorithms.

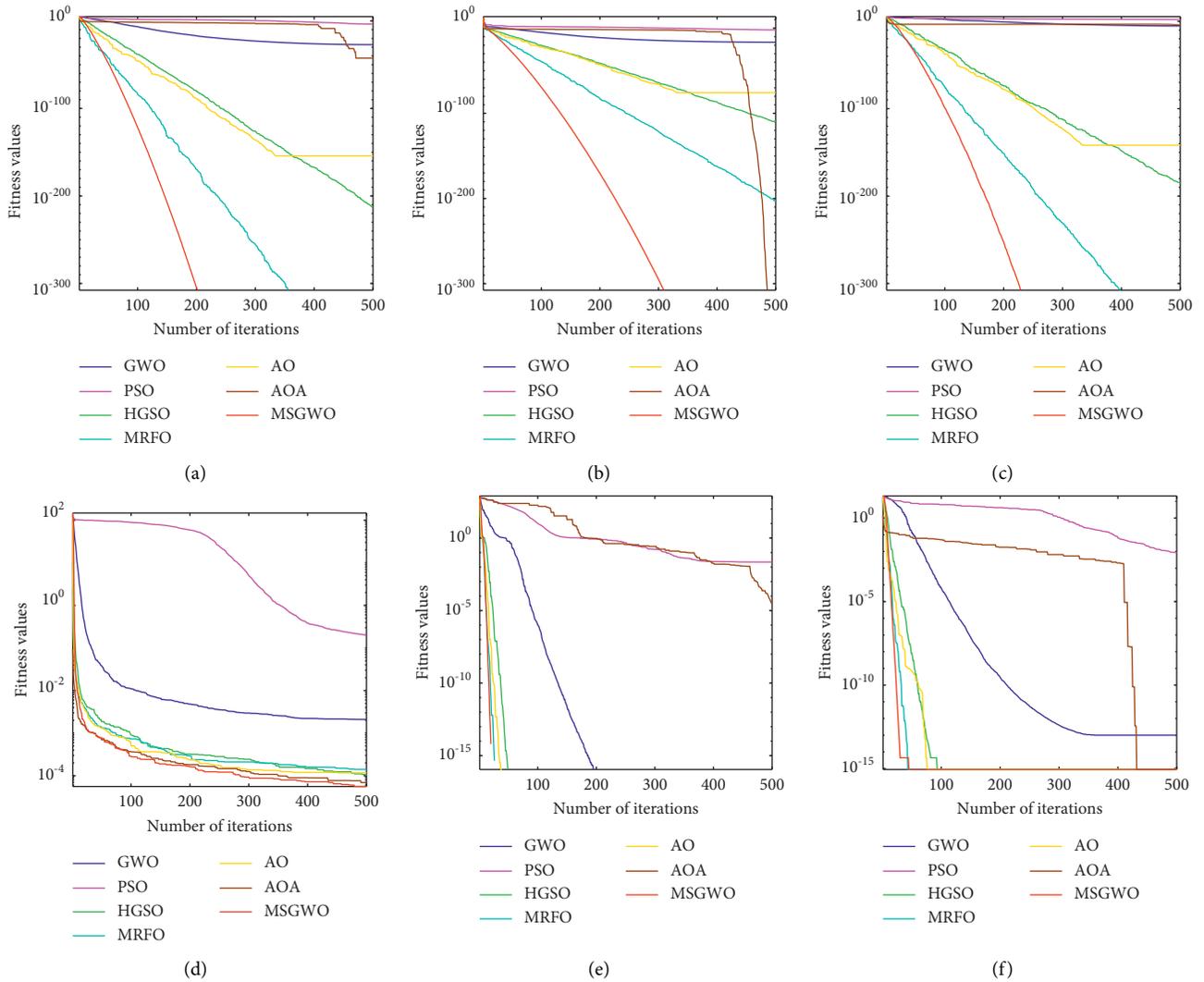


FIGURE 2: Convergence curves of six test functions. (a) Sphere. (b) Schwefel 2.22. (c) Schwefel 1.2. (d) Quartic. (e) Ackley. (f) Criewank.

TABLE 4: Comparison of optimization performance of different k values in MSGWO.

Function		$k = 1/3$	$k = 1/2$	$k = 1$	$k = 2$	$k = 3$
f_1	Ave	0.00E+00	0.00E+00	0.00E+00	0.00E+00	4.39E-257
	Std	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_2	Ave	0.00E+00	0.00E+00	3.89E-247	1.09E-165	1.76E-130
	Std	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.83E-130
f_3	Ave	0.00E+00	0.00E+00	0.00E+00	2.09E-250	5.37E-188
	Std	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_4	Ave	0.00E+00	0.00E+00	1.38E-234	1.78E-152	1.92E-117
	Std	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_5	Ave	6.65E-05	7.12E-05	9.05E-05	1.23E-04	1.38E-04
	Std	5.51E-05	7.22E-05	7.84E-05	1.17E-04	1.12E-04
f_6	Ave	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Std	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_7	Ave	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Std	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_8	Ave	8.88E-16	8.88E-16	8.88E-16	8.88E-16	8.88E-16
	Std	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00

TABLE 5: Parameter values for the clustering algorithms.

Algorithm	Parameter	Value
FCM	m	1.5
	Convergence parameter (a)	Linear reduction from 2 to 0
GWOFCM	m	1.5
	Population	50
	Cognitive and social constant	($C1, C2$) 0.5, 1.5
HPSOFCM	Inertia weight	Linear reduction from 0.9 to 0.1
	Population	50
	m	1.5
WFCM	t	1.5
	m	1.5
EWFCM	m	1.5
	μ	0.5
SFWFCM	p	-2
	q	2
	m	1.5
MSGWO-WFCM	Convergence parameter (a)	Nonlinear reduction from 2 to 0
	k	1
	m	1.5
	Population	50
	t	1.5

TABLE 6: Information of the datasets.

Datasets	Iris	Hab	Ion	TSE	Ecoli	Pendigits
Samples	150	306	351	5820	336	10992
Attributes	4	3	34	28	7	16
Clusters	3	2	2	5	8	10

TABLE 7: Experimental result.

	Iris	Hab	Ion	TSE	Ecoli	Pendigits
FCM	88.67	51.96	70.94	78.32	53.54	71.53
GWOFCM	88.40	52.71	68.29	67.04	48.99	74.20
HPSOFCM	89.33	51.96	70.51	71.63	55.12	61.34
WFCM	86.47	75.42	63.82	81.63	33.20	75.56
EWFCM	90.87	75.16	73.73	81.03	55.36	74.37
SFWFCM	85.80	67.79	69.86	67.92	54.88	67.07
MSGWO-WFCM	95.47	76.93	74.93	81.83	55.80	78.13

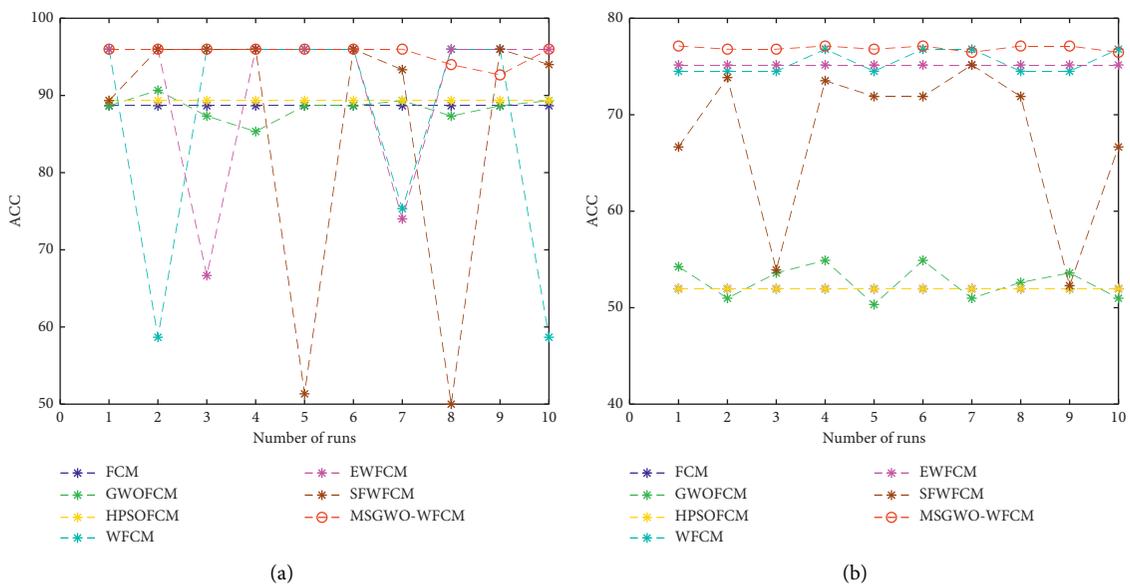


FIGURE 3: Continued.

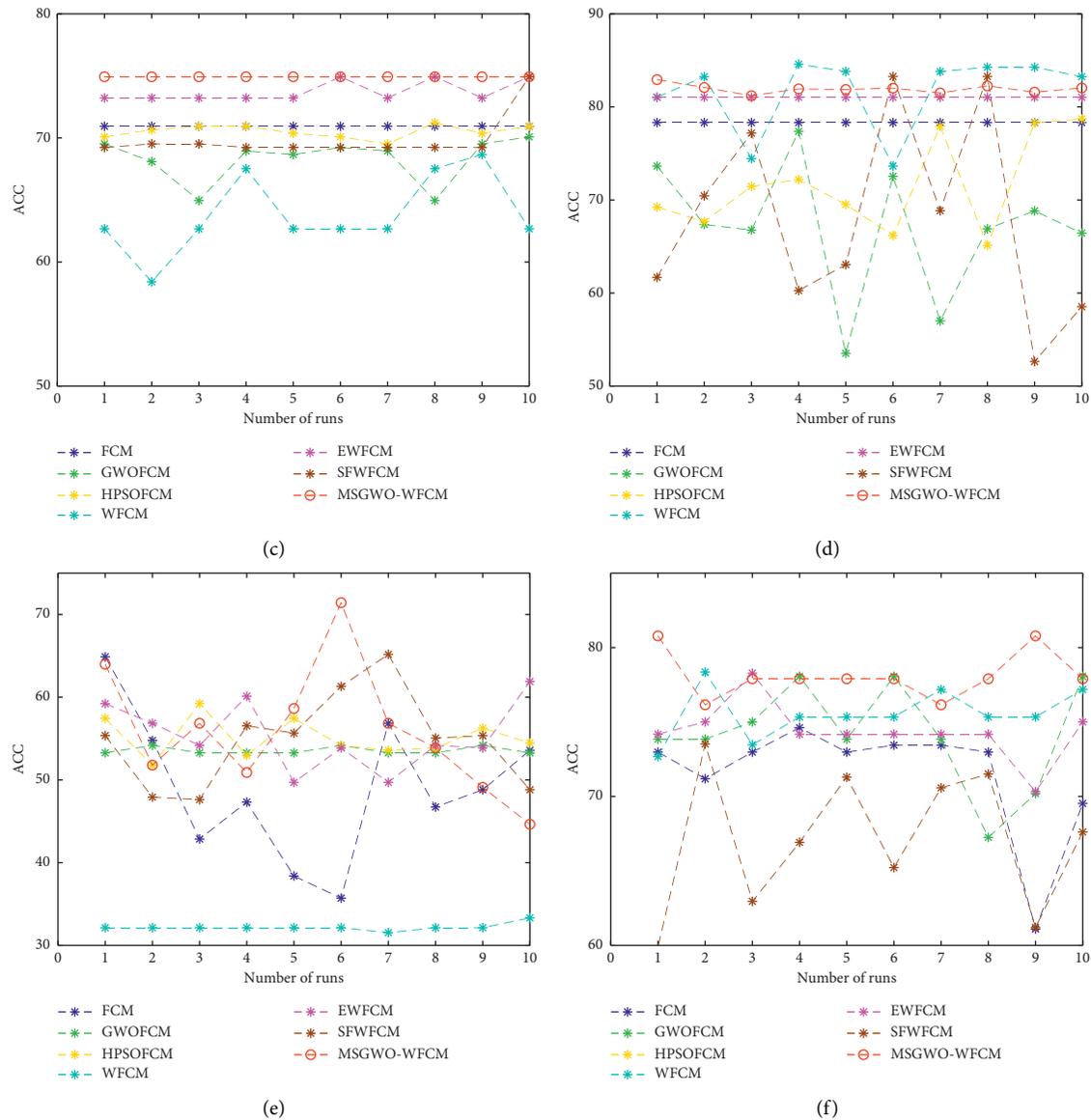


FIGURE 3: Results of ten runs. (a) Iris dataset. (b) Hab dataset. (c) Ion dataset. (d) TSE dataset. (e) Ecoli dataset. (f) Pendigits dataset.

6. Conclusions

This paper proposes a multistrategy grey wolf optimization algorithm (MSGWO). First, the population diversity initialization strategy is introduced to enhance the population diversity; second, the convergence factor nonlinear adjustment strategy can be introduced to better balance exploration and exploitation; finally, the reverse learning strategy further enhances the global search capability. And, the results show that it has better convergence speed and convergence accuracy.

In order to overcome the shortcomings of the traditional fuzzy clustering, on the one hand, the differences between different features are considered, and different weights are assigned; on the other hand, MSGWO is used to update the centroid to ensure the global optimality of the clustering results and effectively alleviate the impact of

the initialization process. Experimental results show that the performance of MSGWO-WFCM in terms of the accuracy and robustness is better than the comparison algorithms.

In future, we will explore the practical application of the proposed methods in different fields, such as image segmentation, text mining, and medical problems. Furthermore, we will introduce other search strategies and/or splitting operator for GWO to enhance the guiding search ability of GWO.

Data Availability

All the datasets used in this paper are derived from the UCI (University of California Irvine) Machine Learning Repository. Please visit <https://archive.ics.uci.edu/ml/datasets.php>.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The authors would like to thank members of the IR and DM Research Group from Henan Polytechnic University for their invaluable advice that makes this paper successfully completed. The authors would like to thank the support of National Science Fund Subsidized Project under Grant no. 61872126.

References

- [1] M. G. Rahman and M. Z. Islam, "Missing value imputation using a fuzzy clustering-based EM approach," *Knowledge and Information Systems*, vol. 46, no. 2, pp. 389–422, 2016.
- [2] K. Zhou, S. Yang, and Z. Shao, "Household monthly electricity consumption pattern mining: a fuzzy clustering-based model and a case study," *Journal of Cleaner Production*, vol. 141, pp. 900–908, 2017.
- [3] A. S. Sabitha, D. Mehrotra, and A. Bansal, "Delivery of learning knowledge objects using fuzzy clustering," *Education and Information Technologies*, vol. 21, no. 5, pp. 1329–1349, 2016.
- [4] F. B. Abdelaziz and O. Limam, "Multiobjective fuzzy clustering with coalition formation: the case of brain image processing," *INFOR: Information Systems and Operational Research*, vol. 55, no. 1, pp. 52–69, 2017.
- [5] J. C. Bezdek, R. Ehrlich, and W. Full, "FCM: the fuzzy c-means clustering algorithm," *Computers and Geosciences*, vol. 10, no. 2, pp. 191–203, 1984.
- [6] L. Zhang, C. Zhou, M. Ma, X. Liu, and C. Sun, "Fuzzy C-mean clustering based on particle swarm optimization," *Journal of Jilin University (Science Edition)*, vol. 44, no. 2, pp. 219 pages, 2006.
- [7] B. Jiang, F. Qiu, L. Wang, and Z. Zhang, "Bi-level weighted multi-view clustering via hybrid particle swarm optimization," *Information Processing and Management*, vol. 52, no. 3, pp. 387–398, 2016.
- [8] S. Gupta and K. Deep, "A novel random walk grey wolf optimizer," *Swarm and evolutionary computation*, vol. 44, pp. 101–112, 2019.
- [9] S. Gupta and K. Deep, "Enhanced leadership-inspired grey wolf optimizer for global optimization problems," *Engineering with Computers*, vol. 36, no. 3, pp. 1–24, 2019.
- [10] Y. Jiang, "Reader behavior analysis based on gray wolf optimization clustering algorithm," *Journal of Jishou University (Natural Sciences Edition)*, vol. 41, no. 4, pp. 38 pages, 2020.
- [11] A. Keller and F. Klawonn, "Fuzzy clustering with weighting of data variables," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 8, no. 6, pp. 735–746, 2000.
- [12] J. Zhou, L. Chen, C. L. P. Chen, Y. Zhang, and H.-X. Li, "Fuzzy clustering with the entropy of attribute weights," *Neurocomputing*, vol. 198, pp. 125–134, 2016.
- [13] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.
- [14] S. Saremi, S. Z. Mirjalili, and S. M. Mirjalili, "Evolutionary population dynamics and grey wolf optimizer," *Neural Computing and Applications*, vol. 26, no. 5, pp. 1257–1263, 2015.
- [15] T. Jayabarathi, T. Raghunathan, B. R. Adarsh, and P. N. Suganthan, "Economic dispatch using hybrid grey wolf optimizer," *Energy*, vol. 111, pp. 630–641, 2016.
- [16] A. A. Heidari and P. Pahlavani, "An efficient modified grey wolf optimizer with Lévy flight for optimization tasks," *Applied Soft Computing*, vol. 60, pp. 115–134, 2017.
- [17] X. Zhang, Q. Kang, J. Cheng, and X. Wang, "A novel hybrid algorithm based on biogeography-based optimization and grey wolf optimizer," *Applied Soft Computing*, vol. 67, pp. 197–214, 2018.
- [18] X. Wang, H. Zhao, T. Han, H. Zhou, and C. Li, "A grey wolf optimizer using Gaussian estimation of distribution and its application in the multi-UAV multi-target urban tracking problem," *Applied Soft Computing*, vol. 78, pp. 240–260, 2019.
- [19] S. Gupta and K. Deep, "A memory-based grey wolf optimizer for global optimization tasks," *Applied Soft Computing*, vol. 93, Article ID 106367, 2020.
- [20] C. Xie, X. Zou, X. Xia, and Z. Wang, "A multi-objective particle swarm optimization algorithm integrating multiply strategies," *Acta Electronica Sinica*, vol. 43, no. 8, pp. 1538 pages, 2015.
- [21] Z. L. Wei, H. Zhao, M. D. Li, and Y. Wang, "A grey wolf optimization algorithm based on nonlinear adjustment strategy of control parameter," *Journal of Air Force Engineering University (Natural Science Edition)*, vol. 17, no. 3, pp. 68–72, 2016.
- [22] S. Saxena and M. Pohit, "A new non linear inertia weight Approach in PSO for faster rigid image registration," in *Proceedings of the 2019 6th International Conference on Signal Processing and Integrated Networks (SPIN)*, Noida, India, March 2019.
- [23] H. Wang, Z. Wu, S. Rahnamayan, Y. Liu, and M. Ventresca, "Enhancing particle swarm optimization using generalized opposition-based learning," *Information Sciences*, vol. 181, no. 20, pp. 4699–4714, 2011.
- [24] G. Venter and J. Sobieszcanski-Sobieski, "Particle swarm optimization," *AIAA Journal*, vol. 41, no. 8, pp. 1583–1589, 2003.
- [25] F. A. Hashim, E. H. Houssein, M. S. Mabrouk, W. A. Atabany, and S. Mirjalili, "Henry gas solubility optimization: a novel physics-based algorithm," *Future Generation Computer Systems*, vol. 101, pp. 1–12, 2019.
- [26] L. Abualigah, D. Yousri, M. Abd Elaziz, A. A. Ewees, M. A. A. Al-Qaness, and A. H. Gandomi, "Aquila Optimizer: a novel meta-heuristic optimization Algorithm," *Computers and Industrial Engineering*, vol. 157, Article ID 107250, 2021.
- [27] L. Abualigah, A. Diabat, S. Mirjalili, M. Abd Elaziz, and A. H. Gandomi, "The arithmetic optimization algorithm," *Computer Methods in Applied Mechanics and Engineering*, vol. 376, Article ID 113609, 2021.
- [28] W. Zhao, Z. Zhang, and L. Wang, "Manta ray foraging optimization: an effective bio-inspired optimizer for engineering applications," *Engineering Applications of Artificial Intelligence*, vol. 87, Article ID 103300, 2020.
- [29] S. Chen, Z. Xu, and Y. Tang, "A hybrid clustering algorithm based on fuzzy c-means and improved particle swarm optimization," *Arabian Journal for Science and Engineering*, vol. 39, no. 12, pp. 8875–8887, 2014.
- [30] K. Li, Y. Gao, and Z. Cao, "Fuzzy clustering algorithm based on the automatic variable weights of samples and features," *Journal of Harbin Engineering University*, vol. 39, no. 9, pp. 1554–1560, 2018.