Research Article

An Improved Animal Migration Optimization Algorithm for Clustering Analysis

Mingzhi Ma,[1] Qifang Luo,[1,2] Yongquan Zhou,[1,2] Xin Chen,[1] and Liangliang Li[1]

[1]*College of Information Science and Engineering, Guangxi University for Nationalities, Nanning 530006, China*
[2]*Guangxi High School Key Laboratory of Complex System and Intelligent Computing, Nanning 530006, China*

Correspondence should be addressed to Qifang Luo; l.qf@163.com

Animal migration optimization (AMO) is one of the most recently introduced algorithms based on the behavior of animal swarm migration. This paper presents an improved AMO algorithm (IAMO), which significantly improves the original AMO in solving complex optimization problems. Clustering is a popular data analysis and data mining technique and it is used in many fields. The well-known method in solving clustering problems is k-means clustering algorithm; however, it highly depends on the initial solution and is easy to fall into local optimum. To improve the defects of the k-means method, this paper used IAMO for the clustering problem and experiment on synthetic and real life data sets. The simulation results show that the algorithm has a better performance than that of the k-means, PSO, CPSO, ABC, CABC, and AMO algorithm for solving the clustering problem.

1. Introduction

Data clustering is the process of grouping data into a number of clusters. The goal of data clustering is to make the data in the same cluster share a high degree of similarity while being very dissimilar to data from other clusters. It is a main task of exploratory data mining and a common technique for statistical data analysis used in many fields, including machine learning, pattern recognition, image analysis, information retrieval, and bioinformatics. Cluster analysis was originated in anthropology by Driver and Kroeber in 1932 and introduced to psychology by Zubin in 1938 and Tryon in 1939 and famously used by Cattell beginning in 1943 [1] for trait theory classification in personality psychology. Many clustering methods have been proposed; they are divided into two main categories: hierarchical and partitional. The k-means clustering method [2] is one of the most commonly used partitional methods. However, the results of k-means solving the clustering problem highly depend on the initial solution and it is easy to fall into local optimal solutions. Zhang et al. have proposed an improved k-means clustering algorithm called k-harmonic means [3]. But the accuracy of the results obtained by the method is not high enough.

In recent years, many studies have been inspired by animal behavior phenomena for developing optimization techniques, such as firefly algorithm (FA) [4], cuckoo search (CS) [5], bat algorithm (BA) [6], artificial bee colony (ABC) [7], and particle swarm optimization (PSO) [8]. Because of its advantages of global, parallel efficiency, robustness, and universality, these bioinspired algorithms have been widely used in constrained optimization and engineering optimization [9, 10], scientific computing, automatic control, and clustering problem [11–21]. Niknam et al. have proposed an efficient hybrid evolutionary algorithm based on combining ACO and SA for clustering problem [15, 16] in 2008. In 1991, Colorni et al. have presented ant colony optimization (ACO) algorithm based on the behavior of ants seeking a path between their colony and a source of food [22]. Then Shelokar et al. and Kao and Cheng have solved the clustering problem using the ACO algorithm [17, 18] in 2004 and 2006. Eberhart and Kennedy have proposed particle swarm optimizer (PSO) algorithm which simulates the movement of organisms in a bird flock or fish school [8] in 1995 and the algorithm also has been adopted to solve this problem by Omran et al. and van der Merwe and Engelbrecht [19, 23] in 2005 and 2003. Kao et al. have presented a hybrid approach according to combination

of the k-means algorithm, Nelder-Mead simplex search, and PSO for clustering analysis [14] in 2008. Niknam et al. have presented a hybrid evolutionary algorithm based on PSO and SA (simulated annealing algorithm, 1989 [24]) to solve the clustering problem [13] in 2009. Zou et al. have proposed a cooperative artificial bee colony algorithm to solve the clustering problem and experiment on synthetic and real life data sets to evaluate the performance [11] in 2010. Niknam and Amiri have proposed an efficient hybrid approach based on PSO, ACO, and k-means called PSO-ACO-K approach for clustering analysis [12] in 2010. The artificial bee colony (ABC) algorithm is described by Karaboga [25] in 2005 and it has been adopted to solve clustering problem by Karaboga and Ozturk [20] in 2011. Voges and Pope have used an evolutionary-based rough clustering algorithm for the clustering problem [21] in 2012. Chen et al. have used monkey search algorithm for clustering analysis [26] in 2014.

Animal migration algorithm (AMO) is a new bioinspired intelligent optimization algorithm by simulating animal migration behavior proposed by Li et al. [27] in 2013. AMO simulates the widespread migration phenomenon in the animal kingdom, through the change of position, replacement of individual, and finding the optimal solution gradually. AMO has obtained good experimental results on many optimization problems. This paper presents an algorithm to improve the performance of AMO. We proposed a new migration method to modify the performance of AMO, the migration process based on shrinking animals living area operator; this method guarantees AMO rapid convergence to global optimum. By means of selecting the better solution space around the current solution, it improves search ability and accelerates convergence velocity, and it has more chance to find the global optima.

The structure of the paper is as follows. In Section 2, the traditional method k-means for clustering is presented. In Section 3, the original AMO algorithm is introduced. Section 4 describes our proposed novel approach of migration process. Section 5 elaborates the improved AMO and some biological foundations of animal behaviors are explained. Section 6 illustrates experiments and discusses the results. Section 7 studies the extent of different size of shrinkage coefficient impact of the proposed algorithm. At the end of the paper, we conclude it with future directions and developments with the improved AMO.

2. The k-Means Clustering Algorithm

The target of data clustering is grouping data into a number of clusters; k-means is one of the simplest unsupervised learning algorithms that solve the clustering problem. It is proposed by MacQueen in 1967 [28]. The procedure follows a simple and easy way to classify a given data set $D = \{x_1, x_2, \ldots, x_n\}$ through a certain number of clusters $G_1, G_2, \ldots, G_K$ (assume K clusters) fixed a priori; each data vector is a p-dimensional vector, satisfying the following conditions [29, 30]:

(1) $G_i = \varnothing$, $i = 1, 2, \ldots K$;

(2) $G_i \cap G_j = \varnothing$, $i, j = 1, 2, \ldots K, i \neq j$;

(3) $\bigcup_{i=1}^{K} G_i = \{x_1, x_2, \ldots, x_n\}$.

The k-means clustering algorithm is as follows.

(1) Set the number of clusters K and the data set $D = \{x_1, x_2, \ldots, x_n\}$.

(2) Randomly choose K points $c_1, c_2, \ldots, c_K$ as the cluster centroids from $\{x_1, x_2, \ldots, x_n\}$.

(3) Assign each object x_i to the group that has the closest centroid. The principle of division is as follows: if $d(x_i - c_i) < d(x_i - c_k)$, $k = 1, 2, \ldots, K$ and $i \neq k$. The data x_i will be divided into classified collection G_i.

(4) When all objects have been assigned, recalculate the positions of the K centroids $c_1^*, c_2^*, \ldots, c_K^*$:

$$c_i^* = \frac{1}{|G_i|} \sum_{x_j \in G_j} x_j, \quad i = 1, 2, \ldots, K, \tag{1}$$

where $|G_i|$ is the number of the points in the classified collection G_i.

(5) Repeat Steps (2) and (4) until the centroids no longer move.

The main idea is to define K centroids, one for each cluster. These centroids should be placed in a cunning way because of different location causing different result. So the better choice is to place them as much as possible far away from each other. In this study, we will use Euclidian metric as a distance metric. The expression is given as follows:

$$d\left(x_i, c_j\right) = \sqrt{\sum_{k=1}^{p} \left(x_{ik} - c_{jk}\right)^2}. \tag{2}$$

Finally, this algorithm aims at minimizing an objective function, in this case a squared error function. The objective function is as follows:

$$f\left(X, C\right) = \sum_{i=1}^{n} \min \left\{\|x_i - c_k\|^2 \mid k = 1, 2, \ldots, K\right\}. \tag{3}$$

3. Animal Migration Optimization (AMO)

Animal migration algorithm can be divided into animal migration process and animal updating process. In the migration process, the algorithm simulates how the groups of animals move from current position to a new position. During the population updating process, the algorithm simulates how animals are updated by the probabilistic method.

3.1. Animal Migration Process. During the animal migration process, an animal should obey three rules: (1) avoid collisions with your neighbors; (2) move in the same direction as your neighbors; and (3) remain close to your neighbors. In order to define concept of the local neighborhood of an individual, we use a topological ring, as has been illustrated in Figure 1. For the sake of simplicity, we set the length of the neighborhood to be five for each dimension of the individual. Note that, in our algorithm, the neighborhood topology is static and is defined on the set of indices of vectors.

If the index of animal is i, then its neighborhood consists of animal having indices $i - 2, i - 1, i, i + 1, i + 2$, if the index of animal is 1, the neighborhood consists of animal having indices $NP - 1, NP, 1, 2, 3$, and so forth. Once the neighborhood topology has been constructed, we select one neighbor randomly and update the position of the individual according to this neighbor, as can be seen in the following formula:

$$X_{i,G+1} = X_{i,G} + \delta \cdot \left(X_{\text{neiborhood},G} - X_{i,G} \right), \qquad (4)$$

where $X_{\text{neiborhood},G}$ is the current position of the neighborhood, δ is produced by using a random number generator controlled by a Gaussian distribution, $X_{i,G}$ is the current position of ith individual, and $X_{i,G+1}$ is the new position of ith individual.

3.2. Population Updating Process. During the population updating process, the algorithm simulates how some animals leave the group and some join in the new population. Individuals will be replaced by some new animals with a probability Pa. The probability is used according to the quality of the fitness. We sort fitness in descending order, so the probability of the individual with best fitness is $1/NP$ and the individual with worst fitness, by contrast, is 1, and the process can be shown in Algorithm 1.

In Algorithm 1, $r_1, r_2 \in [1, \ldots, NP]$ are randomly chosen integers, $r_1 \neq r_2 \neq i$. After producing the new solution $X_{i,G+1}$, it will be evaluated and compared with the $X_{i,G}$, and we choose the individual with a better objective fitness:

$$X_i = \begin{cases} X_{i,G} & \text{if } f\left(X_{i,G}\right) \text{ is better than } f\left(X_{i,G+1}\right), \\ X_{i,G+1} & \text{otherwise.} \end{cases} \qquad (5)$$

4. The New Migration Process Method

In AMO, algorithm uses migration process and population updating process to find a satisfactory solution. The proposed algorithm used a new migration process by establishing a living area by the leader animal (the individuals with best fitness value) and animals migrate from current locations into this new living area to simulate animal migration process.

At first, there are NP animals that live in living area, as shown in Figure 2(a), moving, eating, drinking, reproducing, and so on; some individuals move randomly and their position be updated, and then we calculate the best position of animals by fitness function and record it. But the amount of food or water gradually diminished as the time wore on, as shown in Figure 2(b), and some animals migrate from the current areas which have no food and water to a new area with abundant food and water, as shown in Figure 2(c). In Figure 2, the green parts represent the living areas with abundant food and water; animals can live in these areas. And the yellow parts represent the areas that lack food or water; animals can no longer live in these areas, and they must migrate to a new living area (the green parts in Figure 2(c)). We shrink the living area after a period of time (as shown in Figures 2(a) and 2(c)), and then animals migrate to the new living area ceaselessly. As a rule of thumb, the globally optimal

solution always nearby is the current best solution; in IAMO, the animals living area is smaller and smaller (by formula (6)) after each iteration, and the individuals get closer and closer to the globally optimal solution, so we can accelerate the convergence velocity and precision of the algorithm to some extent.

The boundary of the living area is established by

$$\text{low} = X_{\text{best}} - R, \qquad \text{up} = X_{\text{best}} + R, \\ R = \rho \cdot R, \qquad (6)$$

where X_{best} is the leader animal (the current best solution), low and up are the lower and upper bound of the living area, R is living area radius, ρ is shrinkage coefficient, $\rho \in (0, 1)$, and low, up, and R are all $1 \times D$ row vector. In general, the original value of R depends on the size of the search space. As iterations go on, a big value of R improves the exploration ability of the algorithm and a small value of R improves the exploitation ability of the algorithm.

5. The IAMO Algorithm for Solving Clustering Problem

5.1. Initializing the Population. During the initialization process, the algorithm begins with initializing a set of NP animal positions $X_1, X_2, X_3, \ldots, X_{NP}$; each animal position X_i is a $1 \times (K \times D)$-dimensional vector, where K is the number of clustering center and D is the dimension of the test set $G_{N \times D}$. The cluster centers $x_i^* = (x_{i1}, x_{i2}, \ldots, x_{iD})$, $(i = 1, 2, \ldots, K)$, each center x_i^* is $1 \times D$-dimensional vector, and the lower bound of the centers is the minimum of each column in test set $G_{N \times D}$, namely, $a_i^* = \min\{G_1, G_2 \ldots, G_D\}$, and the upper bound of the centers is $b_i^* = \max\{G_1, G_2 \ldots, G_D\}$. So we can initialize the position of an individual $X_i = x_1^*, x_2^*, \ldots, x_K^* = (x_{11}, x_{12}, \ldots, x_{1D}), (x_{21}, x_{22}, \ldots, x_{2D}), \ldots, (x_{K1}, x_{K2}, \ldots, x_{KD}) = x_{11}, x_{12}, \ldots, x_{KD}$, and then the lower and upper bounds of the solution space are $a = (a_1^*, a_2^*, \ldots, a_K^*)$ and $b = (b_1^*, b_2^*, \ldots, b_K^*)$.

Animals are randomly and uniformly distributed between the prespecified lower initial parameter bound a and the upper initial parameter bound b. So the jth component of the ith vector is as follows:

$$x_{i,j} = a_j + \text{rand}_{i,j}[0, 1] \cdot \left(b_j - a_j\right), \\ i = 1, \ldots, NP, \quad j = 1, \ldots, K \times D, \qquad (7)$$

where $\text{rand}_{i,j}[0, 1]$ is a uniform distribution random number between 0 and 1.

5.2. Animals Migration. During the migration process, because of animals hunting, foraging, or drinking in the living area, some parts of the living area are lacking food or water or climate condition change, and some animals migrate from the current living area to the new living area which has abundant food and water or climate condition suitable for living. We assume that there is only one living area, and animals out of the new living area would be migrating into the new living area, as depicted in Section 4. We calculate the

```
(1) For i = 1 to NP do
(2)    For j = 1 to D do
(3)        If rand > P_a then
(4)            X_{i,G+1} = X_{r1,G} + rand · (X_{best,G} - X_{i,G}) + rand · (X_{r2,G} - X_{i,G})
(5)        End If
(6)    End For
(7) End For
```

ALGORITHM 1: Population updating process.

```
(1)  Begin
(2)     Set the generation counter G, living area radius R, shrinkage coefficient ρ, and randomly initialize X_i
         with a population of NP animals in solution space
(3)     Evaluate the fitness for each individual X_i, record the best individual X_best
(4)  While stopping criteria is not satisfied do
(5)     Establish a new living area by low = X_best - R, up = X_best + R
(6)     Animals migrate into the new living area
(7)  For i = 1 to NP do
(8)     For j = 1 to D do
(9)         Select randomly r1 ≠ r2 ≠ i
(10)        If rand > P_a then
(11)            X_{i,G+1} = X_{r1,G} + rand · (X_{best,G} - X_{i,G}) + rand · (X_{r2,G} - X_{i,G})
(12)        End If
(13)    End For
(14) End For
(15) For i = 1 to NP do
(16)    Evaluate the offspring X_{i,G+1}
(17)        If X_{i,G+1} is better than X_i then
(18)            X_i = X_{i,G+1}
(19)        End If
(20) End for
(21)    Memorize the best solution achieved so far
(22)    R = R · ρ
(23) End while
(24) End
```

ALGORITHM 2: An improved animal's migration optimization algorithm (IAMO).

distance between cluster centers $x_1^*, x_2^*, \cdots, x_K^*$ and text data set, then we classify test data set into K categories according to the distance, and, finally, we can obtain the fitness according the fitness function:

$$f(X,G) = \sum_{i=1}^{N} \min\left\{\left\|G_i - x_k\right\|^2 \mid k = 1, 2, \ldots, K\right\}. \quad (8)$$

According to the fitness function, we obtain the best individual X_{best}, and the new living area can be established by X_{best} and R.

5.3. Individuals in Population Updating. During the population updating process, algorithm simulates some animals that are preyed by their enemies or some animals leave the group and some join in the group from other groups or some new animals are born. In IAMO, we assume that the number of available animals is fixed, and every animal will be replaced by Pa, as shown in Section 3.2.

Specific implementation steps of the improved animals migration optimization algorithm (IAMO) can be shown as in Algorithm 2.

6. Numerical Simulation Experiments

All of the algorithm was programmed in MATLAB R2008a, numerical experiment was set up on AMD Athlon(tm)II ∗4 640 processor and 2 GB memory.

The experimental results comparing the IAMO clustering algorithm with six typical stochastic algorithms including the PSO [31], CPSO [32], ABC [20], CABC [11], AMO [27], and k-means algorithms are provided for two artificial data sets and eight real life data sets (*Iris*, teaching assistant evaluation (TAE), wine, seeds, StatLog (heart), Hagerman's
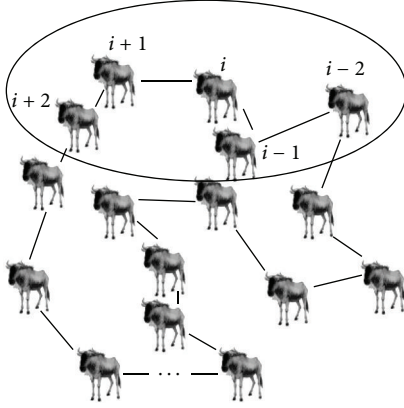
FIGURE 1: The concept of the neighborhood of an animal.

survival, balance scale, and Wisconsin breast cancer) which are selected from the UCI machine learning repository [33].

Artificial Data Set One ($N = 250, d = 3, and K = 5$). This is a three-featured problem with five classes, where every feature of the classes was distributed according to Class 1-Uniform (85, 100), Class 2-Uniform (70, 85), Class 3-Uniform (55, 70), Class 4-Uniform (40, 55), and Class 5-Uniform (25, 40) [12, 14]. The data set is illustrated in Figure 3.

Artificial Data Set Two ($N = 600, d = 2, and K = 4$). This is a two-featured problem with four unique classes. A total of 600 patterns were drawn from four independent bivariate normal distributions, where classes were distributed according to

$$N_2 \left(\mu = \begin{pmatrix} m_i \\ m_i \end{pmatrix}, \Sigma = \begin{bmatrix} \begin{bmatrix} 0.5 & 0.05 \\ 0.05 & 0.5 \end{bmatrix} \end{bmatrix} \right), \qquad (9)$$

where $i = 1, 2, 3, 4, \ldots > 4, m_1 = -3, m_2 = 0, m_3 = 3$, and $m_4 = 6$.

μ and Σ are mean vector and covariance matrix, respectively [12, 14]. The data set is illustrated in Figure 4.

Iris Data ($N = 150, d = 4, and K = 3$). This data set with 150 random samples of flowers from the *Iris* species *setosa*, *versicolor*, and *virginica* collected by Anderson [34]. From each species, there are 50 observations for sepal length, sepal width, petal length, and petal width in cm. This data set was used by Fisher [35] in his initiation of the linear-discriminant-function technique [11, 12, 33].

Teaching Assistant Evaluation ($N = 151, d = 5, and K = 3$). The data consist of evaluations of teaching performance over three regular semesters and two summer semesters of 151 teaching assistant (TA) assignments at the Statistics Department of the University of Wisconsin-Madison. The scores were divided into 3 roughly equal-sized categories ("low," "medium," and "high") to form the class variable [33].

Wine Data ($N = 178, d = 13, and K = 3$). This is the wine data set, which is also taken from MCI laboratory. These data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wines. There are 178 instances with 13 numeric attributes in wine data set. All attributes are continuous. There is no missing attribute value [11, 12, 33].

Seeds Data ($N = 210, d = 7, and K = 3$). This data set consists of 210 patterns belonging to three different varieties of wheat: Kama, Rosa, and Canadian. From each species, there are 70 observations for area A, perimeter P, compactness C ($C = 4 * pi * A/P^2$), length of kernel, width of kernel, asymmetry coefficient, and length of kernel groove [33].

StatLog (Heart) Data ($N = 270, d = 13, and K = 2$). This data set is a heart disease database similar to a database already present in the repository (heart disease databases) but in a slightly different form [33].

Hagerman's Survival ($N = 306, d = 3, and K = 2$). The data set contains cases from a study that was conducted between 1958 and 1970 at the University of Chicago's Billings Hospital on the survival of patients who had undergone surgery for breast cancer. It records two survival status patients with the age of patient at time of operation, patient's year of operation, and number of positive axillary nodes detected [33].

Balance Scale Data ($N = 625, d = 4, and K = 3$). This data set was generated to model psychological experimental results. Each example is classified as having the balance scale tip to the right, to the left, or balanced. The attributes are the left weight, the left distance, the right weight, and the right distance. The correct way to find the class is the greater of (left-distance * left-weight) and (right-distance * right-weight). If they are equal, it is balanced [33].

Wisconsin Breast Cancer ($N = 683, d = 9, and K = 2$). It consists of 683 objects characterized by nine features: clump thickness, cell size uniformity, cell shape uniformity, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli, and mitoses. There are two categories in the data: malignant (444 objects) and benign (239 objects) [11, 12, 33].

Here we set the parameters of AMO and IAMO as follows. The population size of the AMO and IAMO is 100. In IAMO, the original living area radius $R = 0.3(b - a)$ and shrinkage coefficient $\rho = 0.92$. For the PSO, inertia weight $w = 0.729$, acceleration coefficients $c1 = 2, c2 = 2$, and population size $M = 100$. The population size of the CPSO is 20. The population size of the ABC and CABC are 50 and 10, respectively. In order to compare with other algorithms, the maximum generations of all algorithms are 100.

For every data set, each algorithm is applied 20 times individually with random initial solution. For the Art1 and Art2 data set, once the randomly generated parameters are determined, the same parameters are used to test the performance of three algorithms. We ranked each algorithm according to the mean result. The results are kept four digits after the decimal point. The mean value, the best value, the worst value, the standard deviation, and the rank value are recorded in Tables 1, 2, 3, 4, 5, 6, 7, 8, 9, and 10.
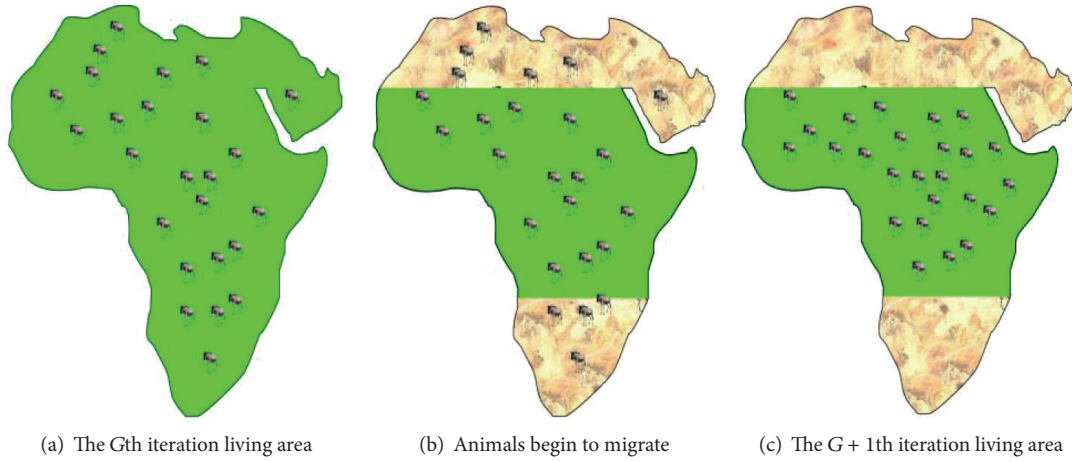
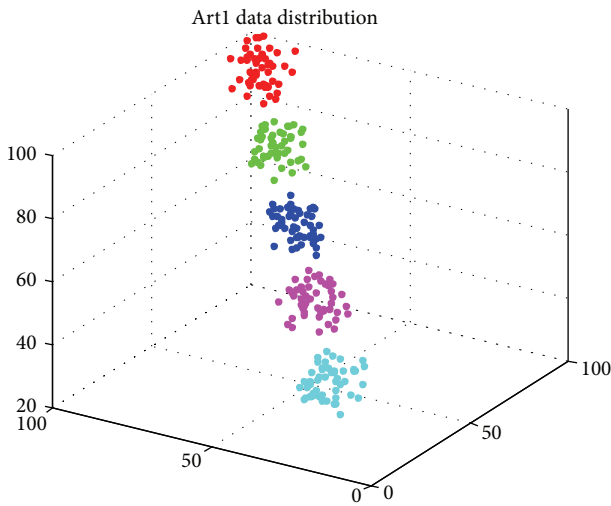(a) The Gth iteration living area   (b) Animals begin to migrate   (c) The $G + 1$th iteration living area

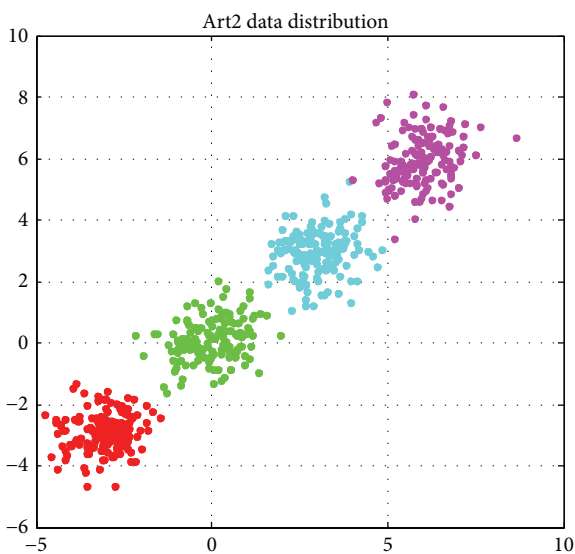FIGURE 2: Animals migration process.

FIGURE 3: The distribution image of Art1.

TABLE 1: Results obtained by the algorithms for 20 different runs on Art1 data.

Algorithm	Mean	Best	Worst	Standard deviation	Rank
k-means	2106.6059	1720.7627	2456.0468	358.8518	7
PSO	1991.3796	1721.6508	2448.1993	192.7566	5
CPSO	1860.5699	1718.6937	2417.5543	305.7752	4
ABC	1718.5496	1718.2939	1718.9832	0.1955	3
CABC	1718.4434	1718.2544	1720.3302	0.5488	2
AMO	2062.1954	1974.4561	2199.1275	54.8921	6
IAMO	1718.2540	1718.2538	1718.2540	$1.0990e - 05$	1

TABLE 2: Results obtained by the algorithms for 20 different runs on Art2 data.

Algorithm	Mean	Best	Worst	Standard deviation	Rank
k-means	644.2100	514.6623	896.4104	181.1884	7
PSO	514.7965	513.9191	523.8308	2.3765	5
CPSO	513.9046	513.9035	513.9085	$1.3621e - 03$	4
ABC	513.9037	513.9035	513.9045	$3.2720e - 04$	3
CABC	513.9037	513.9035	513.9696	$1.5235e - 04$	2
AMO	525.0879	516.1389	546.2325	8.4842	6
IAMO	513.9035	513.9035	513.9035	$9.3111e - 06$	1

FIGURE 4: The distribution image of Art2.

Tables 1–10 show that IAMO is very precise than other algorithms in solving the ten data sets. As seen from the results, the IAMO algorithm provides the best value and small standard deviation in comparison with other methods. For the Art1 and Art2 data set in Tables 1 and 2 which were randomly generated, IAMO obtained the best mean and smallest standard deviation compared to other algorithms. The mean

TABLE 3: Results obtained by the algorithms for 20 different runs on *Iris* data.

Algorithm	Mean	Best	Worst	Standard deviation	Rank
k-means	102.8412	99.4582	123.4678	8.7805	7
PSO	98.3345	96.6567	104.2224	2.2431	5
CPSO	96.9721	96.6580	97.5211	$2.966e-01$	4
ABC	96.6659	96.6566	96.7547	$2.1388e-02$	3
CABC	96.6561	96.6555	96.6599	$1.1685e-03$	2
AMO	99.0055	97.0751	100.5484	1.1202	6
IAMO	96.6555	96.6555	96.6555	$1.2155e-06$	1

TABLE 4: Results obtained by the algorithms for 20 different runs on TAE data.

Algorithm	Mean	Best	Worst	Standard deviation	Rank
k-means	1533.5441	1503.9418	1605.1206	29.7491	7
PSO	1501.5895	1490.2455	1529.2453	16.3170	6
CPSO	1499.8073	1492.1980	1532.4523	17.4859	5
ABC	1491.4434	1490.9775	1492.4754	0.5128	3
CABC	1491.3099	1490.9276	1497.3575	2.7356	2
AMO	1499.0215	1493.3564	1509.4512	6.8524	4
IAMO	1491.0900	1490.9321	1492.5707	0.4482	1

TABLE 5: Results obtained by the algorithms for 20 different runs on wine data.

Algorithm	Mean	Best	Worst	Standard deviation	Rank
k-means	17853.2412	16374.4353	18484.3458	1003.6327	7
PSO	16307.0584	16302.8364	16314.0563	5.8844	5
CPSO	16304.9829	16299.0526	16312.1291	5.9887	3
ABC	16305.6668	16299.1970	16321.9535	9.8439	4
CABC	16292.1982	16292.1858	16292.2094	$1.0563e-02$	2
AMO	16359.7965	16319.9935	16400.5533	30.0025	6
IAMO	16292.1855	16292.1849	16292.1862	$5.0627e-04$	1

TABLE 6: Results obtained by the algorithms for 20 different runs on seeds data.

Algorithm	Mean	Best	Worst	Standard deviation	Rank
k-means	313.4977	313.1428	313.7343	$2.6879e-01$	5
PSO	326.5250	318.3185	335.2944	6.0131	7
CPSO	312.1138	311.9116	312.3788	0.2899	4
ABC	312.0382	311.8520	312.2110	$6.7210e-02$	3
CABC	311.7980	311.7980	311.7982	$1.4865e-04$	2
AMO	319.3922	313.8100	327.9267	3.1572	6
IAMO	311.7980	311.7980	311.7980	$3.3686e-05$	1

TABLE 7: Results obtained by the algorithms for 20 different runs on StatLog (heart) data.

Algorithm	Mean	Best	Worst	Standard deviation	Rank
k-means	10695.7851	10682.5524	10703.8544	8.2080	6
PSO	10760.0684	10644.8965	11024.2641	25.9460	7
CPSO	10651.1354	10624.2168	10747.7609	55.0264	4
ABC	10627.4760	10626.7154	10629.6472	1.0354	3
CABC	10622.9904	10622.9824	10623.6762	$1.7830e-02$	2
AMO	10675.1758	10658.6325	10696.1824	17.6918	5
IAMO	10622.9824	10622.9824	10622.9825	$5.0093e-05$	1

TABLE 8: Results obtained by the algorithms for 20 different runs on Hagerman's survival data.

Algorithm	Mean	Best	Worst	Standard deviation	Rank
k-means	2640.1277	2610.5245	3180.5211	126.7500	7
PSO	2567.4479	2566.9899	2569.1620	0.9587	6
CPSO	2567.3233	2566.9889	2567.8257	0.4578	5
ABC	2566.9892	2566.9888	2566.9895	$2.3919e-04$	2
CABC	2566.9903	2566.9888	2566.9955	$2.8709e-03$	3
AMO	2566.9985	2566.9907	2567.0096	$9.1598e-03$	4
IAMO	2566.9888	2566.9888	2566.9888	$2.3022e-06$	1

value of IAMO obtained is 1718.2540 in solving Art1, while ABC and CABC obtained 1718.5496 and 1718.4434, and IAMO gives 4 orders of magnitude better than ABC and CABC. Same to solving Art2, IAMO obtained 513.9035, while CPSO, ABC, and CABC obtained 513.9046, 513.9037, and 513.9037, respectively, but the standard deviation of IAMO is at least 2 orders of magnitude better than them. For *Iris* data set, the mean value, the optimum value, and the worst value of IAMO are all 96.6555 and the standard deviation is $1.2155e-06$ which revealed the robustness of IAMO. CABC also sought the best solution 96.6555, but the standard deviation is bigger than IAMO when the best solutions of AMO, PSO, CPSO, ABC, and k-means are 97.0751, 96.6567, 96.6580, 96.6566, and 99.4582, respectively. Table 4 shows the results of algorithms on the TAE data set. The mean value

of IAMO is 1491.0900 which is smaller than that of AMO, PSO, CPSO, ABC, CABC, and k-means within 20 runs. For wine data set, IAMO reached the mean value 16292.1855 while CABC reached the mean value 16292.1982. The best value and worst value of IAMO are 16292.1849 and 16292.1862, which are also better than 16292.1858 and 16292.2094 obtained by CABC, and the standard deviation value of IAMO is also the smallest one. Table 6 provides the results of algorithms on the

TABLE 9: Results obtained by the algorithms for 20 different runs on balance scale data.

Algorithm	Mean	Best	Worst	Standard deviation	Rank
k-means	1426.7522	1423.8570	1433.8423	3.1208	5
PSO	1430.1546	1426.4237	1447.6403	9.3451	7
CPSO	1424.8260	1423.5525	1425.5460	0.9048	4
ABC	1423.9238	1423.8308	1425.9821	$4.5452e-02$	3
CABC	1423.9109	1423.8206	1425.2445	$1.4053e-02$	2
AMO	1429.1499	1427.9670	1430.8177	1.1641	6
IAMO	1423.8204	1423.8204	1423.8204	$4.9763e-06$	1

TABLE 10: Results obtained by the algorithms for 20 different runs on cancer data.

Algorithm	Mean	Best	Worst	Standard deviation	Rank
k-means	2981.2564	2976.4687	2988.4277	4.8661	5
PSO	3001.2685	2969.7475	3114.3658	63.9370	6
CPSO	2964.8268	2964.4167	2965.2941	0.3773	3
ABC	2965.5369	2964.9734	2966.4785	0.8081	4
CABC	2964.4138	2964.3870	2964.5222	$6.0177e-02$	2
AMO	3001.6328	2974.0974	3050.9651	32.7322	7
IAMO	2964.3870	2964.3870	2964.3870	$2.2260e-05$	1

seeds data set; the IAMO algorithm and CABC algorithm are superior to those obtained by the others. Although IAMO and CABC reached the same mean value 311.7980, the standard deviation of IAMO is 1 order of magnitude better than CABC. On StatLog (heart) data set results given in Table 7, IAMO gets the best value is 10622.9824 and the same as CABC, while the mean values of the two algorithms are 10622.9824 and 10622.9904, so the IAMO is better than CABC algorithm. For Hagerman's survival data set, the optimum value 2566.9888 can be obtained by IAMO, ABC, and CABC, but the standard deviations of ABC and CABC are $2.3919e-04$ and $2.8709e-03$ which is worse than that of $2.3022e-06$ obtained by IAMO. The standard deviation of PSO is a little bigger than that of CPSO. For balance scale data set in Table 9, as seen from the results, the mean, best, and worst ones are all 1423.8204, which reflect the stable characteristics of IAMO. The three best algorithms in this test data are IAMO, CABC, and ABC, and the best results of them are 1423.8204, 1423.8206, and 1423.8308. For Wisconsin breast cancer data set in Table 10, the mean value, the best value, and the worst value are all 2964.3870 which are obviously superior to k-means, PSO, CPSO, ABC, and AMO.
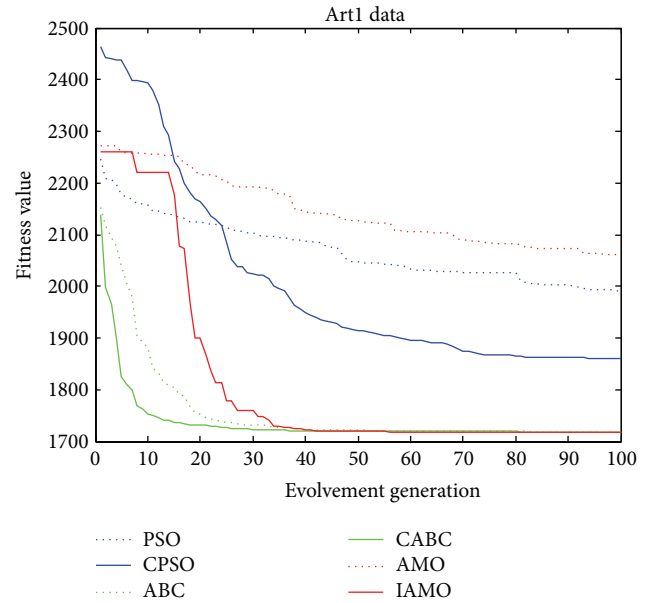

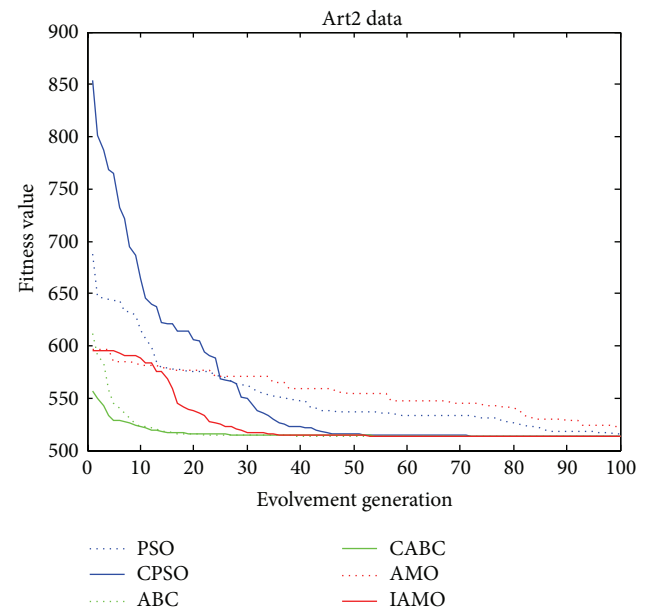FIGURE 5: The convergence curve of the Art1 data.

FIGURE 6: The convergence curve of the Art2 data.

As seen from Table 1 to Table 10, we can conclude that although the convergence rate is not quick enough at the beginning of the iteration compared to ABC and CABC, the final results are the best compared to other algorithms in all test data sets. The most results of ABC and CABC are better than PSO and CPSO, and the k-means algorithm is the worst for most of test data sets.

Figures 5, 6, 7, 8, 9, 10, 11, 12, 13, and 14 show the convergence curves of different data sets for various algorithms. Figures 15 and 16 show the original data distribution of *Iris* data set and the clustering result by IAMO algorithm.
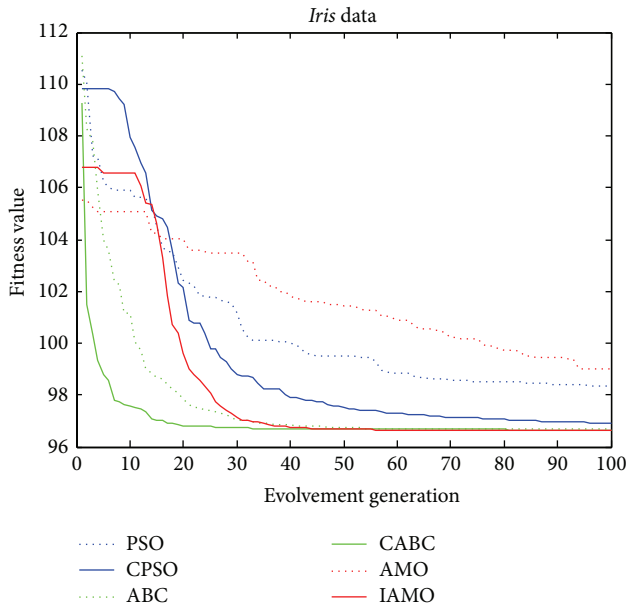
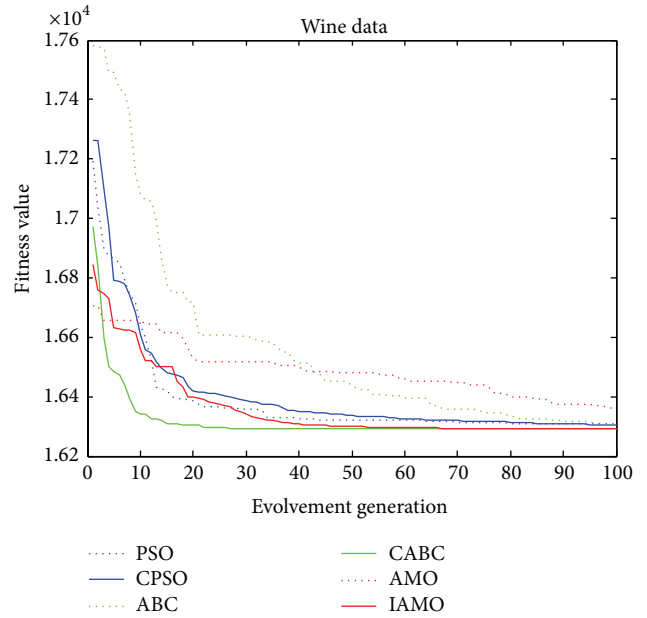FIGURE 7: The convergence curve of the *Iris* data.

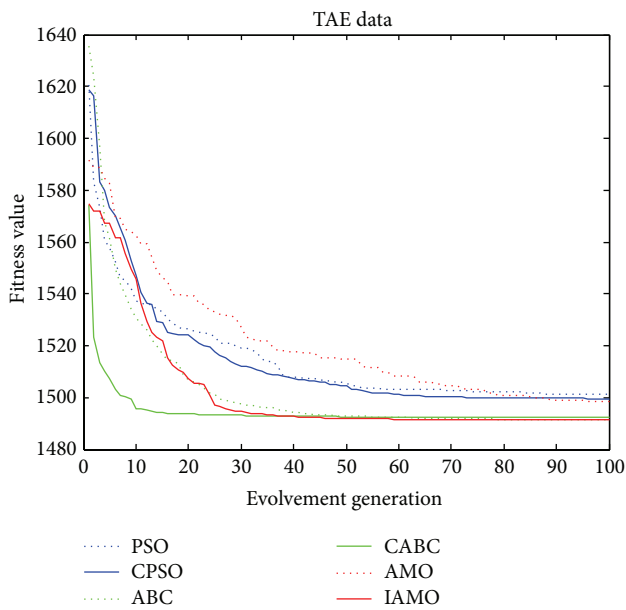FIGURE 9: The convergence curve of the wine data.

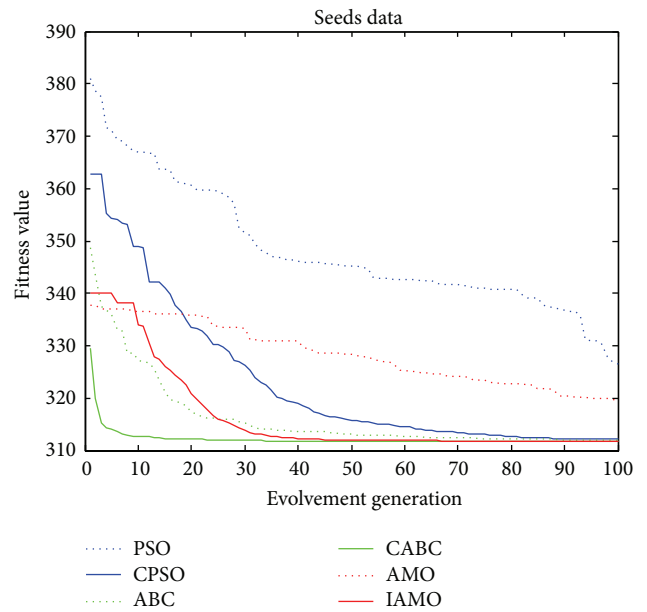FIGURE 8: The convergence curve of the TAE data.

FIGURE 10: The convergence curve of the seeds data.

7. Living Area Radius Evaluation

The performance and results of the proposed algorithms are greatly affected by the size of living area. At the beginning of the iteration, a big value of R improves the exploration ability of the algorithm and; at the end of iteration, a small value of R improves the exploitation ability of the algorithm. We adopted a fixed shrinking coefficient $\rho = 0.92$ to change the living area radius after each iteration, as shown in formula (6). To study the extent of R impacts on the proposed algorithm, we selected Art1 data set and *Iris* data set, using different ρ to evaluate the performance of the proposed algorithm.

Figure 17 shows the results of an experiment on Art1; we can conclude that if we choose ρ between 0.6 and 0.9, it has a better convergence precision than that of $\rho = 0.99$ or $\rho = 0.40$. If we choose $\rho = 0.40$, IAMO algorithm plunges into local optima, and if we choose $\rho = 0.99$, the IAMO algorithm has a very low convergence rate. And likewise in Figure 18, for *Iris* test data set, IAMO algorithm quickly converged at global optimum before 30 iterations if we choose $\rho = 0.80$, while IAMO could not escape from poor local optima and to global optimum if we choose $\rho = 0.70$, $\rho = 0.60$, or $\rho = 0.40$.
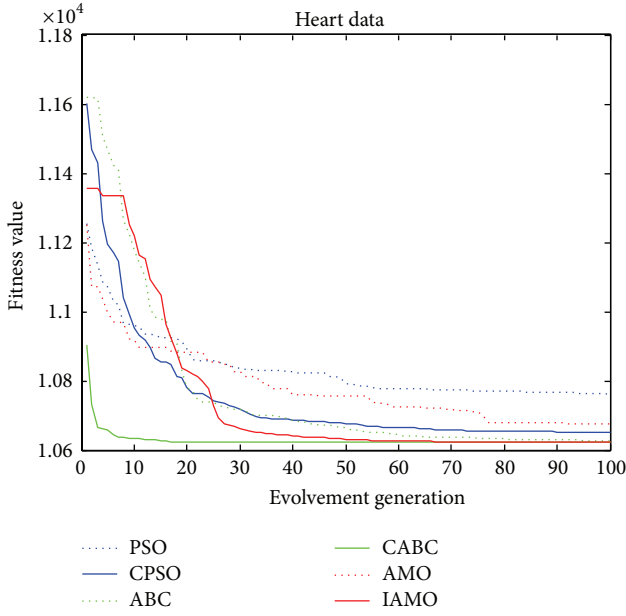
FIGURE 11: The convergence curve of the heart data.
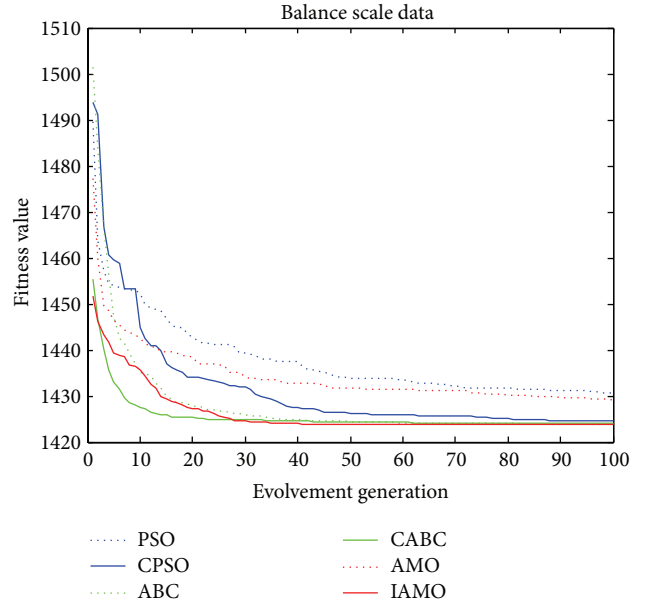

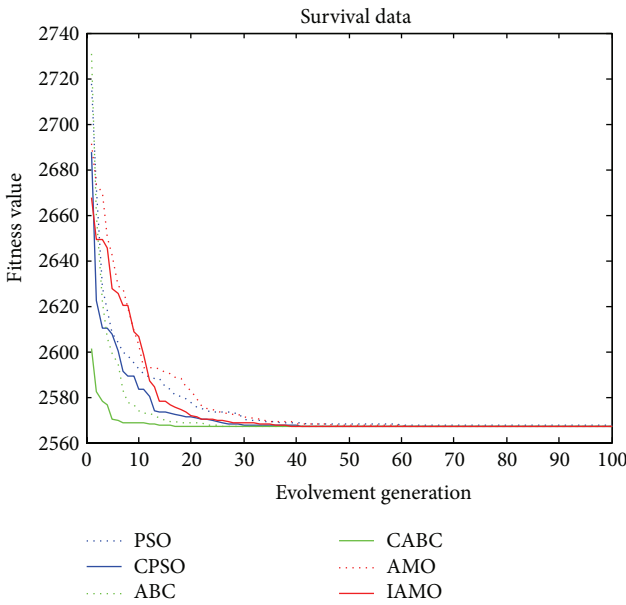FIGURE 13: The convergence curve of the balance scale data.

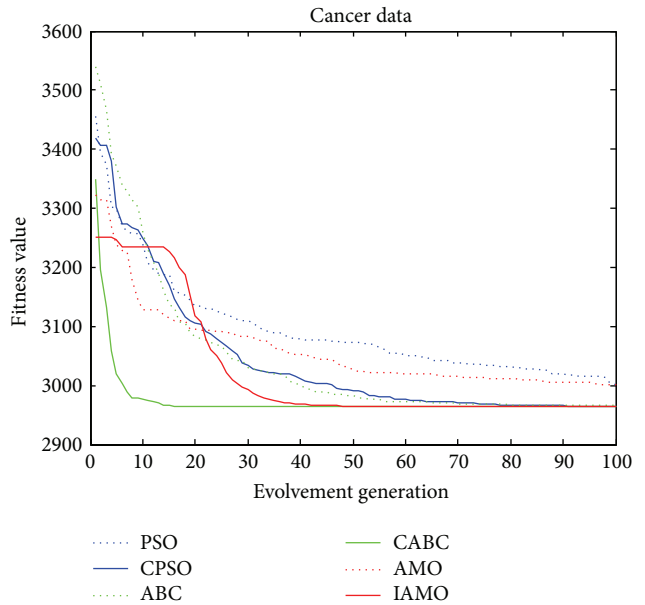

FIGURE 12: The convergence curve of the survival data.

FIGURE 14: The convergence curve of the cancer data.

So the best ρ for solving *Iris* data set must exist between 0.7 and 0.99.

The results suggest that a proper ρ can greatly improve the algorithm convergence velocity and convergence precision, and an improper ρ may lead the IAMO fall into local optimum.

8. Conclusions

In this paper, to improve the deficiencies of the AMO algorithm, we improved the algorithm by using a new migration

method based on shrinking animals living area. By 10 typical standard test data sets simulation, the results show that IAMO algorithm generally has strong global searching ability and local optimization ability and can effectively avoid the deficiencies that conventional algorithms easily fall into local optimum. IAMO has improved the convergence precision of AMO and rank 1st in all test data sets; therefore, it is very practical and effective to solve clustering problems. At last, how to define a proper and unified radius of living area needs to be considered in subsequent work.
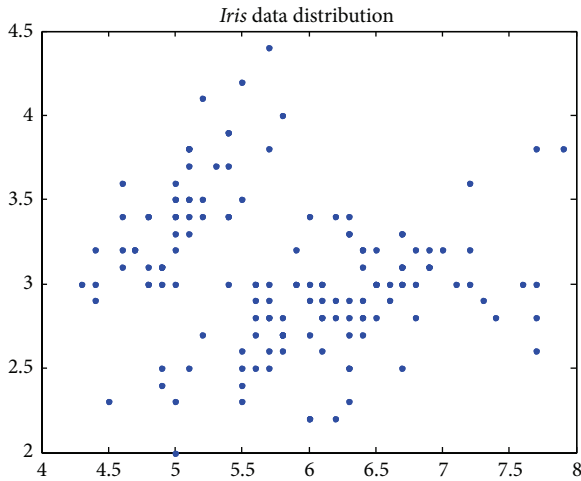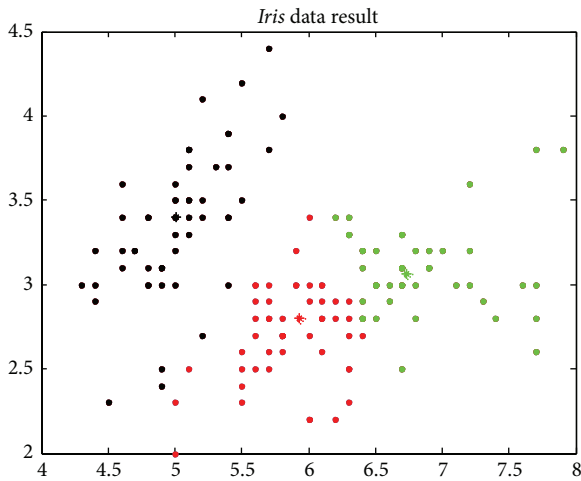
FIGURE 15: The *Iris* data distribution.

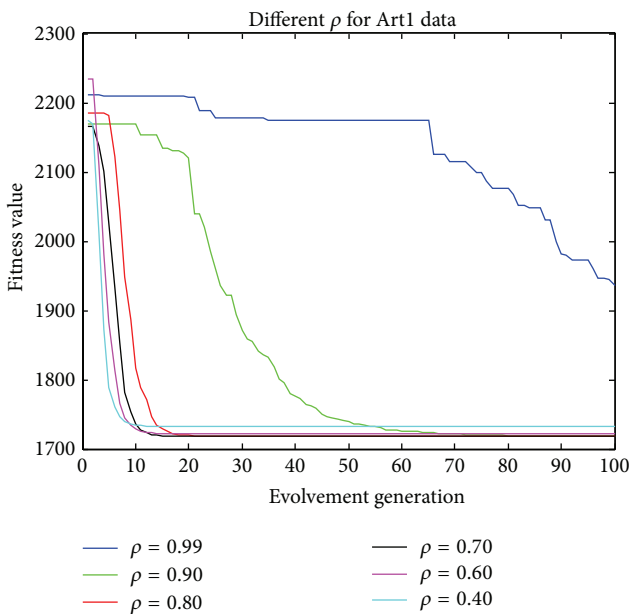FIGURE 16: The *Iris* data clustering result.

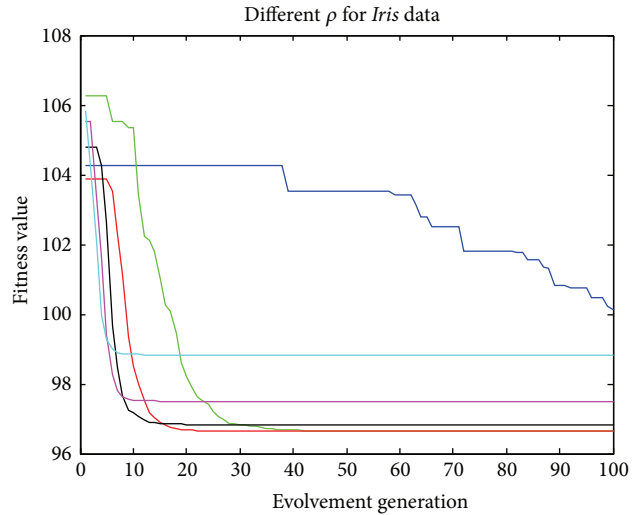FIGURE 17: The convergence curve of the Art1 with different ρ.

FIGURE 18: The convergence curve of the *Iris* with different ρ.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.
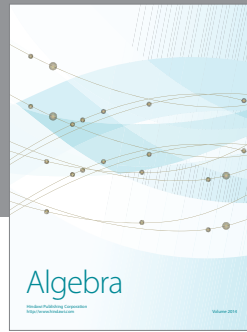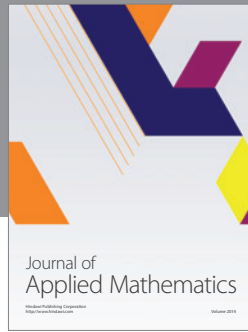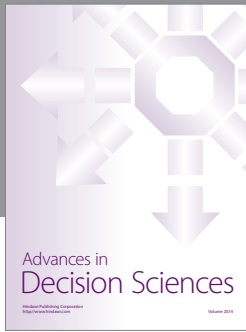
Acknowledgments

References

[1] R. B. Cattell, "The description of personality: basic traits resolved into clusters," *Journal of Abnormal and Social Psychology*, vol. 38, no. 4, pp. 476–506, 1943.

[2] K. R. Zalik, "An efficient k-means clustering algorithm," *Pattern Recognition Letters*, vol. 29, no. 8, pp. 1385–1391, 2008.

[3] B. Zhang, M. Hsu, and U. Dayal, "K-harmonic means—a data clustering algorithm," Tech. Rep. HPL-1999-124, Hewlett-Packard Laboratories, 1999.

[4] X.-S. Yang, *Nature-Inspired Metaheuristic Algorithms*, Luniver Press, 2008.

[5] X.-S. Yang and S. Deb, "Cuckoo search via Lévy flights," in *Proceedings of the World Congress on Nature and Biologically Inspired Computing (NABIC '09)*, pp. 210–214, IEEE, December 2009.

[6] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," in *Nature Inspired Cooperative Strategies for Optimization*, vol. 284 of *Studies in Computational Intelligence*, pp. 65–74, Springer, Berlin, Germany, 2010.

[7] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony

(ABC) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, 2007.

[8] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the 6th International Symposium on Micro Machine and Human Science*, pp. 39–43, Nagoya, Japan, October 1995.

[9] A. H. Gandomi, X.-S. Yang, A. H. Alavi, and S. Talatahari, "Bat algorithm for constrained optimization tasks," *Neural Computing and Applications*, vol. 22, no. 6, pp. 1239–1255, 2013.

[10] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, "Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems," *Engineering with Computers*, vol. 29, no. 1, pp. 17–35, 2013.

[11] W. Zou, Y. Zhu, H. Chen, and X. Sui, "A clustering approach using cooperative artificial bee colony algorithm," *Discrete Dynamics in Nature and Society*, vol. 2010, Article ID 459796, 16 pages, 2010.

[12] T. Niknam and B. Amiri, "An efficient hybrid approach based on PSO, ACO and *k*-means for cluster analysis," *Applied Soft Computing Journal*, vol. 10, no. 1, pp. 183–197, 2010.

[13] T. Niknam, B. Amiri, J. Olamaei, and A. Arefi, "An efficient hybrid evolutionary optimization algorithm based on PSO and SA for clustering," *Journal of Zhejiang University: Science A*, vol. 10, no. 4, pp. 512–519, 2009.

[14] Y.-T. Kao, E. Zahara, and I.-W. Kao, "A hybridized approach to data clustering," *Expert Systems with Applications*, vol. 34, no. 3, pp. 1754–1762, 2008.

[15] T. Niknam, J. Olamaei, and B. Amiri, "A hybrid evolutionary algorithm based on ACO and SA for cluster analysis," *Journal of Applied Sciences*, vol. 8, no. 15, pp. 2695–2702, 2008.

[16] T. Niknam, B. Bahmani Firouzi, and M. Nayeripour, "An efficient hybrid evolutionary algorithm for cluster analysis," *World Applied Sciences Journal*, vol. 4, no. 2, pp. 300–307, 2008.

[17] P. S. Shelokar, V. K. Jayaraman, and B. D. Kulkarni, "An ant colony approach for clustering," *Analytica Chimica Acta*, vol. 509, no. 2, pp. 187–195, 2004.

[18] Y. Kao and K. Cheng, *An ACO-Based Clustering Algorithm*, Springer, Berlin, Germany, 2006.

[19] M. Omran, A. P. Engelbrecht, and A. Salman, "Particle swarm optimization method for image clustering," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 19, no. 3, pp. 297–321, 2005.

[20] D. Karaboga and C. Ozturk, "A novel clustering approach: Artificial Bee Colony (ABC) algorithm," *Applied Soft Computing Journal*, vol. 11, no. 1, pp. 652–657, 2011.

[21] K. E. Voges and N. K. L. Pope, "Rough clustering using an evolutionary algorithm," in *Proceedings of the 45th Hawaii International Conference on System Sciences (HICSS '12)*, pp. 1138–1145, IEEE, January 2012.

[22] A. Colorni, M. Dorigo, and V. Maniezzo, *Distributed Optimization by Ant Colonies*, Elsevier Publishing, Paris, France, 1991.

[23] D. W. van der Merwe and A. P. Engelbrecht, "Data clustering using particle swarm optimization," in *Proceedings of the Congress on Evolutionary Computation (CEC '03)*, vol. 1, pp. 215–220, Canberra, Australia, December 2003.

[24] E. H. L. Aarts and J. H. Korst, *Simulated Annealing and Boltzmann Machines*, John Wiley & Sons, 1989.

[25] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Tech. Rep. TR06, Erciyes University Press, Erciyes, Turkey, 2005.

[26] X. Chen, Y. Zhou, and Q. Luo, "A hybrid monkey search algorithm for clustering analysis," *The Scientific World Journal*, vol. 2014, Article ID 938239, 16 pages, 2014.

[27] X. Li, J. Zhang, and M. Yin, "Animal migration optimization: an optimization algorithm inspired by animal migration behavior," *Neural Computing and Applications*, vol. 24, no. 7-8, pp. 1867–1877, 2014.

[28] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, pp. 281–297, University of California Press, Berkeley, Calif, USA, 1967.

[29] X. Chen and J. Zhang, "Clustering algorithm based on improved particle swarm optimization," *Journal of Computer Research and Development*, pp. 287–291, 2012.

[30] X. Liu, Q. Sha, Y. Liu, and X. Duan, "Analysis of classification using particle swarm optimization," *Computer Engineering*, vol. 32, no. 6, pp. 201–213, 2006.

[31] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, December 1995.

[32] F. van den Bergh and A. P. Engelbrecht, "A cooperative approach to participle swam optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 225–239, 2004.

[33] C. L. Blake and C. J. Merz, UCI Repository of Machine Learning Databases, http://archive.ics.uci.edu/ml/datasets.html.

[34] E. Anderson, "The irises of the gaspe peninsula," *Bulletin of the American Iris Society*, vol. 59, pp. 2–5, 1935.

[35] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, part 2, Article ID 179188, pp. 179–188, 1936.

Advances in
Operations Research

Advances in
Decision Sciences

Journal of
Applied Mathematics

Algebra

Journal of
Probability and Statistics

The Scientific
World Journal

International Journal of
Differential Equations

International Journal of
Combinatorics

Advances in
Mathematical Physics

Submit your manuscripts at
http://www.hindawi.com

Hindawi

Journal of
Complex Analysis

Journal of
Mathematics

Mathematical Problems
in Engineering

Abstract and
Applied Analysis

Discrete Dynamics in
Nature and Society

International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces

International Journal of
Stochastic Analysis

Journal of
Optimization