Postprint of K-Means Clustering Algorithm Based on Improved Flower Pollination Algorithm

Authors: Tao Zhiyong, Liu Xiaofang, Liu Ying, Wang and Zhang

Date: 2018-08-13T00:00:00+00:00

Abstract

To address the issue of the K-means clustering algorithm' s dependence on initial values and its tendency to fall into local optima, a K-means clustering algorithm based on improved flower pollination is proposed. This algorithm first employs sequences generated by chaotic mapping as the initial positions of the flower population, thereby ensuring the diversity and determinism of the flower population in the search space. Subsequently, the tabu search algorithm is introduced in the later search stage of flower pollination to avoid falling into local optimal solutions. Finally, the improved FPA algorithm is utilized to optimize the initial values of the K-means algorithm. Experimental results on five clustering datasets demonstrate that the average clustering accuracy of the improved algorithm is increased by 12.2% compared with the flower pollination clustering algorithm, proving that the proposed algorithm achieves better clustering performance on low-dimensional datasets.

Full Text

Preamble

Title: K-means Clustering Algorithm Based on Improved Flower Pollination

Authors: Tao Zhiyong[1], Liu Xiaofang[1],[2]†, Liu Ying[1], Wang Hezhang[1]

Affiliations: 1. School of Electronic & Information Engineering, Liaoning Technical University, Huludao, Liaoning 125105, China 2. Fuxin LiXing Technology Company Limited, Fuxin, Liaoning 123000, China

Abstract: To address the problem that the K-means clustering algorithm depends heavily on initial values and easily falls into local optima, this paper proposes a K-means clustering algorithm based on improved flower pollination. The algorithm first uses chaotic map sequences to initialize the positions of the flower population, ensuring both diversity and determinism of the population in the search space. Then, it introduces a tabu search algorithm during the later stages of flower pollination to avoid falling into local optimal solutions. Finally, the improved FPA algorithm is used to optimize the initial values of the K-means algorithm. Experimental results on five clustering datasets demonstrate that the improved algorithm increases the average clustering accuracy by 12.2% compared with the flower pollination clustering algorithm, proving that the proposed algorithm achieves better clustering performance on low-dimensional datasets.

Keywords: clustering; flower pollination; chaotic map; tabu search; K-means

0 Introduction

Clustering is a common and widely applied algorithm among various methods currently used for computer data analysis, particularly suitable for pattern recognition, data mining, and image analysis. This algorithm classifies results based on data similarity, grouping data with high similarity into the same category and separating data with low similarity into different categories. Due to its simple implementation and fast convergence, K-means has become a commonly used clustering algorithm. However, its clustering performance is highly sensitive to the initial clustering centers, and the random selection of these centers often causes the algorithm to fall into local optima. Consequently, many scholars have proposed improvements to the selection of initial clustering centers for K-means.

Duwairi et al. addressed the sensitivity of traditional K-means to initial center values by proposing an initialization method for spherical K-means. This approach introduces random perturbations to unknown solutions during the initialization phase and proposes a new evaluation metric for cluster compactness that measures the directional dispersion of vectors relative to cluster centers, with final clustering results determined based on this dispersion. Kumar et al. positioned clustering centers in high-density regions of the dataset while ensuring sufficient separation between centers, using kd-trees to identify density regions. This method improved K-means clustering performance but increased algorithmic time complexity. Bianchi et al. proposed a density-based approach for non-metric spaces that estimates clustering centers using the most representative models from the input data. This method is independent of dataset shape but involves substantial computational overhead. Consequently, researchers have considered using swarm intelligence algorithms to optimize initial clustering centers. For instance, Hu et al. replaced the random search in fruit fly algorithms with differential vectors from differential evolution algorithms, applying the hybrid approach to optimize K-means. Rahman et al. proposed combining genetic algorithms with K-means for automatic determination of clustering centers, but the weak local search capability and premature convergence of genetic algorithms resulted in low clustering precision. Wang et al. proposed an algorithm combining adaptive cuckoo search with K-means and implemented parallelization using the MapReduce programming model, though the inherent limitations of cuckoo search led to suboptimal clustering results and long execution times.

The flower pollination algorithm (FPA) is a recently proposed swarm intelligence optimization algorithm. The basic FPA has been improved by scholars due to certain shortcomings. Foreign researcher Draa conducted qualitative and quantitative analyses of the algorithm, while Sayed proposed a hybrid binary algorithm combining cloning techniques with FPA for feature selection, and Galvez developed a multimodal FPA enhanced with multimodal functionality to locate all possible optimal solutions in optimization problems. Domestic scholars have also extensively improved the FPA algorithm. Reference [12] integrated differential evolution strategies with FPA to enhance population diversity and improve global search capability. Reference [13] employed adaptive step size strategies during the global optimization phase and introduced the simplex method during local search to improve search capability. Reference [14] utilized Gaussian mutation to perturb global search for increased population diversity and incorporated Powell' s method to enhance local exploitation ability. Although these improvements enhanced the optimization capability of FPA, they still suffer from slow convergence speed and low optimization precision.

Based on these considerations, this paper proposes a flower pollination algorithm combining chaotic theory and tabu search. The algorithm first uses chaotic theory to initialize the flower population, increasing population diversity and accelerating iteration speed. Second, it introduces tabu search during the later search stages to avoid falling into local optima. Finally, the improved flower pollination algorithm is applied to K-means to determine clustering centers, thereby enhancing clustering effectiveness.

1.1 Clustering-Related Problems

The clustering problem can be formally defined as follows: For a dataset with n samples, partition it into k classes, denoted as $Y = \{Y_1, Y_2, \ldots, Y_n\}$ and $C = \{C_1, C_2, \ldots, C_k\}$. The objective of data clustering is to minimize the distance between data points and minimize the distance between data points and their cluster centers (C_j). This can be expressed as:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} \min_{j \in \{1,2,\ldots,k\}} \|Y_i - C_j\|^2$$

where Y_i represents a given data point and C_j represents a cluster center.

1.2 K-means Algorithm

The traditional K-means algorithm randomly selects k data points from the dataset as initial cluster centers. The remaining points are assigned to their nearest cluster center, and the mean of each cluster is computed as the new cluster center. This process repeats iteratively. The K-means algorithm proceeds as follows:

a) Randomly select k points as cluster centers.

b) Assign each remaining data point to its nearest cluster center.

c) Recalculate the cluster centers.

d) Repeat this process until the squared distance between each data point and its nearest cluster center is minimized.

1.3 Flower Pollination Algorithm

The flower pollination algorithm is a swarm intelligence optimization algorithm that simulates the pollination process of flowering plants in nature. The pollination process occurs in two forms: cross-pollination and self-pollination. Cross-pollination in nature requires external agents such as bees and insects, which follow Lévy flight patterns. This process corresponds to global search in the algorithm. Self-pollination does not require pollinators but relies on wind for pollen transfer, representing local search in the algorithm. The transition between global and local search is controlled by a switch probability $p \in [0, 1]$.

In reality, each flower can produce millions or more pollen gametes. To simplify the problem, the algorithm assumes each flowering plant has only one flower and each flower contains only one pollen grain, meaning one flower or pollen grain corresponds to one solution in the optimization problem.

The flower pollination algorithm operates under the following idealized conditions:

a) Biotic cross-pollination is performed by pollinators (birds, bees, etc.) that carry pollen and follow Lévy flights for global pollination.

b) Abiotic self-pollination represents the local search process in the algorithm.

c) Flower constancy refers to the reproduction probability, which is proportional to the similarity between two participating flowers.

d) The switch probability p determines the transition between global and local search. Due to factors such as wind and physical distance, the selection of p values is critical throughout the pollination process.

These idealized conditions can be mathematically formulated as follows:

When rand $> p$, the algorithm performs global pollination, implemented by:

$$x_i^{t+1} = x_i^t + \gamma L(\lambda)(g^* - x_i^t)$$

where x_i^t and x_i^{t+1} represent solutions at generations t and $t+1$, respectively, g^* denotes the current best solution in the population, γ is a scaling factor controlling step size (set to $\gamma = 1$ in this paper), and $L(\lambda)$ represents the Lévy flight displacement corresponding to flower individuals. The expression for $L(\lambda)$ is:

$$L \sim \frac{\lambda\Gamma(\lambda)\sin(\pi\lambda/2)}{\pi}\frac{1}{s^{1+\lambda}}, \quad (s \gg s_0 > 0)$$

where $\Gamma(\lambda)$ is the gamma function, and $\lambda = 3/2$ in this paper. The parameter s is determined by:

$$s = \frac{U}{|V|^{1/\lambda}}, \quad U \sim N(0, \sigma^2), V \sim N(0, 1)$$

where U and V are normally distributed random numbers, and σ is calculated as:

$$\sigma = \left(\frac{\Gamma(1+\lambda)\sin(\pi\lambda/2)}{\Gamma[(1+\lambda)/2]\lambda 2^{(\lambda-1)/2}}\right)^{1/\lambda}$$

When rand $< p$, the algorithm performs local pollination as shown in:

$$x_i^{t+1} = x_i^t + \varepsilon(x_j^t - x_k^t)$$

where x_j^t and x_k^t are pollen from different flowers of the same plant species, equivalent to two random solutions from the population, and $\varepsilon \in [0, 1]$ is a random number. This mechanism enhances population diversity and improves local search capability.

2 Improved Flower Pollination Algorithm

The basic flower pollination algorithm suffers from two main drawbacks: (a) lack of population diversity, and (b) slow convergence speed with susceptibility to local optima. To address drawback (a), this paper employs chaotic sequences to enhance diversity. For drawback (b), we introduce a tabu search table during the later search stages to enable pollen to find optimal solutions more rapidly.

2.1 Chaotic Optimization Strategy

Similar to common heuristic optimization algorithms, the flower pollination algorithm is sensitive to initial values. Random initialization during the population generation phase increases iteration counts during optimization, particularly when dealing with complex nonlinear and multimodal problems, thereby reducing overall speed and precision. Chaotic sequences can compensate for the uneven distribution caused by random initialization.

Chaotic sequences have been applied extensively in evolutionary algorithms, demonstrating feasibility for increasing population diversity and improving convergence speed. Due to their ergodicity, chaotic variable-based optimization search offers advantages over blind random search and can reduce the tendency of evolutionary algorithms to fall into local optima. The fundamental principle involves mapping unknown optimization variables into chaotic space $[0, 1]$, performing search operations using chaotic mapping rules, and then mapping the obtained solutions back to the original space. Initializing populations with chaotic sequences enables flower individuals to perform ergodic searches in the solution space, overcoming the uneven distribution problem of original flower population initialization.

Various chaotic mappings exist for generating chaotic sequences. This paper utilizes the Logistic map to generate the initial flower population, with the function form shown in Equation (7):

$$x_{i+1} = \alpha x_i(1 - x_i), \quad \alpha \in [0, 4]$$

where α is the chaotic coefficient. In Equation (7), x_i is a randomly initialized chaotic sequence with values in the range $[0, 1]$. The process of initializing the flower population using chaotic sequences is as follows: First, randomly initialize a population P with n pollen individuals. Then, use Equation (7) to generate a corresponding chaotic population CP for population P. Since this method still includes initially randomized individuals in the chaotic population, to minimize their impact on reducing solution precision and slowing convergence, the optimal solution obtained from each iteration undergoes chaotic mapping using Equation (7). The resulting chaotic solution is compared with the current optimal solution; if the chaotic solution is superior, it replaces the current optimal solution; otherwise, the current optimal solution is retained.

3 K-means Clustering Algorithm Based on Improved Flower Pollination

Based on the two improvements described above, we propose a K-means clustering algorithm based on improved flower pollination. The fundamental idea is as follows: The improved FPA algorithm performs one iteration of optimization, and the resulting new positions serve as initial points for the K-means algorithm. After one clustering operation, the newly obtained cluster centers update the flower population. The FPA and K-means algorithms execute alternately and repeatedly until termination conditions are met.

The computational steps of the improved FPA algorithm are described below:

a) Initialize all parameters.

b) For each of the n pollen individuals in population P, apply the Logistic Map using Equation (7) to calculate each pollen x_i.

c) Perform one K-means clustering on the initial flower population, then use Equation (1) to calculate the fitness value for each pollen. Record the current global optimal solution and its corresponding optimal value.

d) If rand $> p$, update the obtained solution using Equation (2) and perform boundary violation handling.

e) If rand $< p$, update the solution according to Equation (6) and perform boundary violation handling.

f) Compare the fitness values of the new solutions obtained in steps d) and e) with those of the unupdated solutions. If the new solutions have better fitness, replace the unupdated solutions with the new ones as the optimal solutions; otherwise, retain the unupdated solutions and their fitness values. Perform one K-means clustering on the new pollen and update the pollen with the newly formed cluster centers.

g) If $t > [N_{\text{iter}}/2]$, proceed to Step 8; otherwise, return to Step 4.

h) Apply the basic steps of the TS algorithm for local optimization of the new population.

i) Check termination conditions. If satisfied, output the clustering results; otherwise, return to Step 4.

4 Simulation Experiments and Results Analysis

To verify the superiority and effectiveness of the proposed algorithm, we conducted two groups of experiments. The first group tests algorithm effectiveness on five datasets, comparing our algorithm with the algorithm from reference [12] (DEFPA), standard FPA, differential evolution (DE), particle swarm optimization (PSO), and artificial bee colony (ABC) algorithms. The second group tests the clustering capability on datasets. The experimental environment: CPU is Intel Core i3-2350, memory is [specification not provided in original text].

4.1 Improved FPA Algorithm Performance Test

This section selects five datasets to validate the effectiveness of the proposed algorithm, including two artificial datasets and three real-world datasets (selected from the UCI Machine Learning Repository). Each algorithm was run 20 times on these datasets, and the best, worst, average, and standard deviation values were recorded in Tables 1, 2, 4, 5, and 6. Bold values indicate our algorithm outperforms the other two algorithms, while underlined values indicate other algorithms perform better. The convergence curves for each dataset are shown in Figures 2 through 6.

The first artificial dataset (art1) is a 3-dimensional, 5-class dataset containing 250 sample points, with each class following a normal distribution with means $U_1(85, 100)$, $U_2(70, 85)$, $U_3(55, 70)$, $U_4(40, 55)$, and $U_5(25, 40)$. Table 1 lists the algorithm comparisons for the art1 dataset, with convergence curves shown in Figure 2.

[Figure 1: see original paper] Tabu Search Flowchart

The tabu search (TS) algorithm is a heuristic algorithm for local optimization. Unlike common local optimization algorithms, TS employs tabu technology that prohibits revisiting previous solutions. A tabu list records visited local optima, and in subsequent searches, this information prevents or selectively avoids searching these points again. The process is described using the flowchart shown in Figure 1. The stopping condition is defined as the total number of cycles after each run, i.e., the maximum iteration steps.

Table 1 Algorithm Fitness Comparison on art1

As shown in Table 1, our algorithm achieves the best optimal and worst values among all five algorithms. Although the DEFPA algorithm shows smaller mean and standard deviation values, its optimal value is inferior. The six simulation curves in Figure 2 demonstrate that compared with the other five algorithms, our algorithm’ s fitness curve is smoother and converges faster. This improvement primarily stems from the chaotic strategy applied during the initial phase, which enhances global search capability and accelerates convergence speed, while the introduction of the tabu list ensures the algorithm avoids local optimal regions. Although the clustering improvement is not significant during the initial iterations compared to DEFPA, our algorithm demonstrates a trend toward finding the optimal solution within 25 iterations.

The second artificial dataset (art2) is a 2-dimensional, 4-class dataset containing 600 sample points, generated from four independent bivariate normal distributions as shown in Equation (8):

$$p_i(x) = \frac{1}{2\pi|\Sigma_i|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1} (x-\mu_i)\right), \quad i = 1, 2, 3, 4$$

where μ_i and Σ_i represent the mean vector and covariance matrix, respectively.

The mean vectors are $\mu_1 = [3,0]^T$, $\mu_2 = [0,3]^T$, $\mu_3 = [-3,0]^T$, $\mu_4 = [0,-3]^T$, and the covariance matrices are $\Sigma_1 = \Sigma_2 = \Sigma_3 = \Sigma_4 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$.

Table 2 Algorithm Fitness Comparison on art2

Table 2 shows that our algorithm achieves the best optimal, worst, mean, and standard deviation values among the six algorithms. The fitness value indicates clustering quality, with smaller values representing better clustering results. Our algorithm' s optimal fitness is 5.13 lower than that of the FPA algorithm, and its standard deviation approaches zero, demonstrating superior clustering performance and stability. As shown in Figure 3, our algorithm approaches the global optimum within 25 iterations and reaches the optimal solution between 25-50 iterations, indicating significantly enhanced global and local search capabilities that accurately determine cluster centers and improve clustering effectiveness. The PSO algorithm achieves a similar optimal value at 50 iterations but suffers from weak local search capability, making it prone to local optima. The DEFPA algorithm shows little fitness improvement between 25-50 iterations, indicating it has fallen into local optima. The ABC algorithm' s random selection of initial cluster centers leads to poor early clustering performance, and multiple iterations near local optima slow convergence. While the FPA algorithm performs well initially, its weak ability to locate cluster centers results in the highest clustering fitness.

Table 3 Dataset Attributes

Three real-world datasets were selected from the UCI repository. Their attributes are summarized in Table 3.

Table 4 Algorithm Fitness Comparison on Iris

Table 5 Algorithm Fitness Comparison on wine

Table 6 Algorithm Fitness Comparison on Heart

The algorithm comparison values for these three datasets are presented in Tables 4-6, with convergence behavior shown in Figures 4-6.

As shown in Table 4, our algorithm achieves the best optimal, worst, mean, and standard deviation values among the five algorithms. The mean fitness is 11 units lower than the PSO algorithm, indicating that the FPA algorithm with Logistic mapping and tabu search can accurately determine K-means cluster centers and improve clustering precision. Figure 4 reveals that during the initial phase, mapping solutions to chaotic intervals expands the search region for unknown solutions, causing rapid fitness changes. During later search stages, the tabu list records previously visited solutions, preventing the algorithm from being confined to local exploitation and avoiding local optima. While DE, DEFPA, PSO, and ABC algorithms can achieve the same optimal solution, PSO shows poor initial performance but converges extremely quickly. The FPA algorithm converges relatively slowly, and the ABC algorithm' s tendency toward premature convergence and weak local search capability result in higher fitness and difficulty reaching optimal values.

Table 5 shows that for the wine dataset, our algorithm outperforms all others in optimal, worst, mean, and standard deviation values. The standard deviation is 511.52 lower than the DE algorithm, indicating smaller differences in fitness across iterations and better clustering stability. Figure 5 demonstrates that although our algorithm' s initial fitness is higher than DEFPA, it matches DEFPA' s fitness by the 25th iteration and achieves lower fitness values later, converging to the optimum first during later iterations. While DEFPA achieves the same optimal solution, its convergence speed is slower. The FPA algorithm generally outperforms PSO, DE, and ABC algorithms. Although DE and ABC can obtain good optimal solutions, they converge extremely slowly. The PSO algorithm converges quickly to the optimal solution.

Table 6 indicates that our algorithm achieves the best optimal, worst, mean, and standard deviation values on the Heart dataset. The optimal value is 49.56 lower than the ABC algorithm, demonstrating that our algorithm can accurately determine cluster centers and achieve stable clustering results. Figure 6 shows that both the DE algorithm and our algorithm exhibit good convergence behavior, with our algorithm approaching the optimal solution within 25 iterations while DE converges slowly between 25-50 iterations, failing to escape local optimal regions. This occurs because DE is sensitive to parameter settings, and inappropriate parameters can lead to local optima problems. In contrast, our algorithm' s tabu list stores previously searched local solutions, avoiding repeated searches and escaping local regions. While DEFPA and PSO algorithms descend faster initially, they remain far from the optimal solution and require many additional iterations. The FPA algorithm has similar initial fitness to our algorithm but suffers from slow convergence due to weak global exploration and local exploitation capabilities. The ABC algorithm' s premature convergence and weak local search ability result in higher fitness and difficulty achieving optimal values.

[Figure 2: see original paper] Convergence Curves on art1
[Figure 3: see original paper] Convergence Curves on art2
[Figure 4: see original paper] Convergence Curves on Iris
[Figure 5: see original paper] Convergence Curves on wine
[Figure 6: see original paper] Convergence Curves on Heart

4.2 Clustering Experiments with the Improved Algorithm

The previous section presented extensive numerical simulation experiments demonstrating the effectiveness of our algorithm in solving clustering problems, showing it to be a faster-converging, more stable algorithm that avoids local optima. This section employs the six algorithms and five datasets from the previous section for clustering experiments, further analyzing algorithm performance by comparing average clustering accuracy after 50 iterations.

Table 7 Average Clustering Accuracy

As shown in Table 7, our algorithm achieves the highest clustering accuracy across all five datasets, demonstrating that the incorporation of chaotic sequences and tabu search enhances global search capability and enables effective escape from local optima, thereby improving clustering precision. Compared with the other five algorithms, our algorithm shows substantial improvements in clustering accuracy: 3.53% higher than the best PSO algorithm on art1, 1.72% higher than the best PSO algorithm on art2, 1.7% higher than the best DE algorithm on Iris, 0.28% higher than the best DEFPA algorithm on wine, and 4.89% higher than the best DEFPA algorithm on Heart. These results confirm that our algorithm possesses strong global search capability and can avoid local optima effectively, enabling pollen to find optimal solutions for clustering. While DEFPA shows good clustering performance on the wine dataset with significantly higher accuracy than DE, PSO, and ABC algorithms, FPA performs poorly on all datasets. DE achieves good clustering results on Iris with 84.71% accuracy, and PSO performs best on art1 with accuracy 11.38% higher than FPA. ABC’ s clustering results are similar to FPA.

5 Conclusion

This paper first proposes an improved flower pollination algorithm that incorporates chaotic sequences during initialization and introduces tabu search in later stages to address the problems of random initialization and susceptibility to local optima in the original algorithm. Subsequently, to address the issue that K-means is sensitive to initial cluster centers, which leads to imprecise and unstable clustering results, we combine the improved flower pollination algorithm with K-means clustering to propose an algorithm that can accurately determine cluster centers. Experimental results demonstrate that our algorithm improves clustering effectiveness, accelerates optimization capability, and avoids local optima problems. However, the algorithm exhibits poor performance in terms of time complexity and has high initial fitness values, with suboptimal clustering results for high-dimensional datasets. These limitations will be the focus of future research.

References

[1] Jain A K. Data clustering: 50 years beyond K-means [J]. Pattern Recognition Letters, 2010, 31(8): 651-666.

[2] Duwairi R, Abu M. A novel approach for initializing the spherical K-means clustering algorithm [J]. Simulation Modeling Practice and Theory, 2015, 54(5): 49-63.

[3] Kumar K M, Reddy A R M. An efficient k-means clustering filtering algorithm using density based initial cluster centers [J]. Information Sciences, 2017,

418: 286-301.

[4] Bianchi F M, Livi L, Rizzi A. Two density-based k-means initialization algorithms for non-metric data clustering [J]. Pattern Analysis and Applications, 2016, 19(3): 1-19.

[5] Hu Jixiong, Wang Chunzhi, Liu Chuan, et al. Improved K-means algorithm based on hybrid fruit fly optimization and differential evolution [C]//Proc of the 12th International Conference on Computer Science and Education. Piscataway, NJ: IEEE Press, 2017: 464-467.

[6] Rahman M A, Islam M Z. A hybrid clustering technique combining a novel genetic algorithm with K-means [J]. Knowledge-Based Systems, 2014, 71(71): 345-365.

[7] Wang Bo, Yu Xiangjun. Parallel K-means clustering algorithm based on adaptive cuckoo search [J]. Application Research of Computers, 2018, 35(3): 675-679.

[8] Yang Xinshe. Flower pollination algorithm for global optimization [C]//Proc of Unconventional Computing and Natural Computation. Berlin: Springer, 2012: 240-249.

[9] Draa A. On the performances of the flower pollination algorithm–Qualitative and quantitative analyses [J]. Applied Soft Computing, 2015, 34(C): 349-371.

[10] Sayed S A, Nabil E, Badr A. A binary clonal flower pollination algorithm for feature selection [J]. Pattern Recognition Letters, 2016, 77(C): 21-27.

[11] Galvez J, Cuevas E, Avalos O. Flower pollination algorithm for multimodal optimization [J]. International Journal of Computational Intelligence Systems, 2017, 10(2107): 627-646.

[12] Xiao Huihui, Wan Changxuan, Duan Yanming. Improved novel metaheuristic flower pollination algorithm [J]. Application Research of Computers, 2016, 33(1): 127-131.

[13] Xiao Huihui. A flower pollination algorithm based on simplex method and self-adaptive step [J]. Computer Engineering and Science, 2016, 38(10): 2126-2133.

[14] Xiao Huihui, Wan Changxuan, Duan Yanming, et al. Flower pollination optimization algorithm combined with Gauss mutation and Powell search method [J]. Journal of Frontiers of Computer and Technology, 2017, 11(3): 478-489.

[15] Wu Xiuli, Zhou Yongquan. Improved water wave optimization algorithm based on chaos optimization and simplex method [J]. Computer Science, 2017, 44(5): 218-225.

[16] Karimi E, Maleki H, Reza A. Tabu search algorithm to solve the intermodal terminal location problem [J]. Journal of Mathematical Extension, 2015, 9(1): 75-89.

[17] Niknama T, Amiri B. An efficient hybrid approach based on PSO ACO and k-means for cluster analysis [J]. Applied Soft Computing, 2010, 10(1): 183-197.

[18] Blake C L, Merz C J. UCI repository of machine learning databases [DB/OL]. [2018-04-29] http://archive.ics.uci.edu/ml/datasets.html.

Note: Figure translations are in progress. See original paper for figures.

Source: ChinaXiv —Machine translation. Verify with original.