An ACO-Based Clustering Algorithm With Chaotic Function Mapping

Lei Yang, South China Agricultural University, China*
Xin Hu, South China Agricultural University, China
Hui Wang, South China Agricultural University, China
Wensheng Zhang, Chinese Academy of Sciences, China
Kang Huang, South China Agricultural University, China
Dongya Wang, University of Exeter, UK

ABSTRACT

To overcome shortcomings when the ant colony optimization clustering algorithm (ACOC) deals with the clustering problem, this paper introduces a novel ant colony optimization clustering algorithm with chaos. The main idea of the algorithm is to apply the chaotic mapping function in the two stages of ant colony optimization: pheromone initialization and pheromone update. The application of chaotic mapping function in the pheromone initialization phase can encourage ants to be distributed in as many different initial states as possible. Applying the chaotic mapping function in the pheromone update stage can add disturbance factors to the algorithm, prompting the ants to explore new paths more, avoiding premature convergence and premature convergence to suboptimal solutions. Extensive experiments on the traditional and proposed algorithms on four widely used benchmarks are conducted to investigate the performance of the new algorithm. These experimental results demonstrate the competitive efficiency, effectiveness, and stability of the proposed algorithm.

KEYWORDS

ACO Clustering, Ant Colony Optimization, Chaotic Ant Colony, Data Mining

INTRODUCTION

Data mining is the most critical work in the era of big data. Cluster analysis is one of the most basic tasks of data mining (Kao & Cheng, 2006), which can divide a set of data objects into multiple groups. Data objects located in the same group indicated that they have close similarities. Otherwise, they will belong to different groups. (Ding et al. 2016; Yang et al.2004). By analyzing the similarity and dissimilarity between data in the data set, data objects are grouped or clustered (Hidayat, Fatichah, & Ginardi, 2016; Jabbar, Ku-Mahamud, & Sagban, 2018). Cluster analysis is also called unsupervised learning because class labels and even the number of classes of data objects are unknown before analyzing the data (Gonzalez-Pardo, Jung, & Camacho, 2017; Han, Pei, & Kamber, 2011). Although cluster analysis and classification prediction tasks are not equal, cluster analysis can be used as a prerequisite for classification (Baig, Shahzad, & Khan, 2013). That is, when a set of data objects was unknown about what kinds of labels it can be divided into, cluster analysis could be firstly used to divide the similar data objects into the same groups. And then, according to certain principles, class

DOI: 10.4018/IJCINI.20211001.oa20
*Corresponding Author

labels are affixed to those groups. If data is sufficient, class labels generated by the data set can be used for data classification, and the data set can be used as a training data set of the classification task.

Over the past two decades, group intelligence has attracted a great deal of interest among researchers because of its dynamic and flexible capabilities and its advantages in solving real-world nonlinear problems with high efficiency, and many group intelligence-based algorithms have been introduced for optimization in various areas of computer science (Anand Nayyar & Nayyar, 2018). Ant colony optimization algorithm is a swarm intelligence algorithm developed based on natural genetics and natural evolution of biological circles (Gonzalez-Pardo et al., 2017). As part of group intelligence, it solves complex combinatorial optimization problems by mimicking cooperative behavior among ants (Anand Nayyar, 2018). The algorithm has great global search ability and does not depend on the form of objective functions, so it is applied to solving the clustering problem (Menéndez, Otero, & Camacho, 2016; Monmarché, Slimane, & Venturini, 1999). At the same time, it has a particularly good ability to solve discrete, stochastic, dynamic problems (A Nayyar & Singh, 2016),and routing issues of sensor networks (Anand Nayyar & Singh, 2014). Basic analysis ant colony clustering algorithm (ACOC) aims to assign N data objects into K groups, by making the square of the Euclidean minimize between the data object and center of the corresponding group (Zhang Jianhua Jiang He, 2006). ACOC uses artificial ants (agent) to construct paths, each artificial ant starts with an empty string with length N, and each element in the string represents a data object in the data set. The value of this element object represents the grouping to which the corresponding data object is assigned. (Gao, Wang, Cheng, Inazumi, & Tang, 2016; Pei Zhenkui Li Hua, 2008).In order to improve the convergence rate, the principle of direct allocation is adopted in the initial stage of the ACOC algorithm, putting the ants on the data point at random and generating random global memory(Wang & Luo, 2019).In order to further improve the ACOC convergence and search ability, the variation factor of genetic algorithm was combined to improve the ant colony algorithm, and it enables the ant colony algorithm to generate genetic algorithm initial data in each iteration process, so as to improve the species diversity, expand the search scope of the solution and avoid getting into the local optimal solution dilemma(Wu, Yan, Zhang, & Shen, 2018).A hybrid algorithm for Big Data preprocessing ACO-clustering algorithm approach was proposed, which can help to increase search speed by optimizing the process. As the proposed method using ant colony optimization with clustering algorithm it will also contribute to reducing pre-processing time and increasing analytical accuracy and efficiency(Singh, Singh, & Pant, 2019).

In view of drawbacks of the basic ant colony clustering algorithm, such as running slowly, easy to fall into local optimal, unstable, and so on. This paper introduced a new ant colony clustering algorithm based on chaotic mapping function, named ACO-Based Clustering Algorithm with Chaos (ACOCC). In general, this algorithm has the next two main contributions:

(1) The sequence generated by chaotic mapping function is used to initialize the pheromone matrix, which aims to solve the problem of the incomplete searching state of ants caused by using the same tiny value as initial pheromones or using random number generators. So, artificial ants would distribute in different initial states as much as possible at the beginning of the algorithm.

(2) In the pheromone updating stage of ant colony local search, chaotic mapping function was introduced to generate slight perturbation factors for paths construction of artificial ants, so as to avoid premature and enable the ant to explore more nearby solution paths.

To verify the efficiency and effectiveness of the proposed algorithm, extensive experiments on four widely used benchmark data sets (Iris, Wines, Thyroid diseases, user knowledge modeling (UKD)) are conducted, also making wide comparisons with basic ant colony clustering algorithm. The results show that ACOCC has advantages compared with ACOC in the following aspect: quality of generated solutions, number of iterations, and stability.

The rest of this paper is organized as follows. Section 2 introduces the traditional ACOC. The details of our new ACOCC are described in Section 3. In Section 4, the performance of ACOCC is validated by extensive experiments. Finally, conclusions with future research directions are given in Section 5.

ACO CLUSTERING ALGORITHMS

Ant Colony Optimization

The visual abilities of many ant species in nature are only a rudimentary development, and many have no vision at all (Andries P. Engelbrecht, 2005; Dorigo & Gambardella, 1997), most of the communication between ants depends on the chemicals produced by ants, these chemicals are used to mark paths on the ground, such as the nest to the food source, by sensing these chemicals on the ground, ants can find the shortest path from nest to food. The behavior of these ants to release and sense chemicals on the path is a heuristic factor in the development of ant colony optimization (ACO) algorithms (Dorigo, Birattari, & Stützle, 2006), the application fields of ACO are extended from the TSP problem (Gao et al., 2016) to classification, clustering, association rules analysis and so on (Yu Hui Pei Zhenkui, 2010). For example, applying ACO on the optimization of routing protocols provides solutions for developing energy-efficient routing protocols for wireless sensor networks to improve the reliability and lifetime of sensor nodes (Anand Nayyar & Singh, 2015).

Cluster Analysis Based on ACO Algorithm

The data is regarded as the ants with different attributes, and the clustering center is food sources that ants are looking for, then data clustering can be regarded as a process of ants searching for the food source.

To construct a solution, artificial ants assigns a class label to each element in the string S (i.e., each data object in the data set) by using the information provided on the pheromone path. At the beginning of the algorithm, the pheromone matrix τ is initialized with a small value τ_0, and the value τ_{ij} on the path represents pheromone concentration of the data object *i* and group *j*. For the problem of "dividing *N* data objects into *K* groups," the size of the pheromone matrix is $N \times K$. As the algorithm iterates, the pheromone matrix will change accordingly.

Each artificial ant will construct a similar solution by using pheromone-based communication with other ants to obtain a near-optimal partition scheme for the given data set in each iteration. After generating *R* solution paths, the local search steps will be performed, which makes it possible to further improve the performance of these solutions. Then the pheromone matrix is updated according to the generated solution path. Artificial ants continue to build solution paths according to such steps until a certain number of iterations or certain conditions are reached.

ANT COLONY CLUSTERING ALGORITHM

Pheromone Initialization

At the beginning of the first iteration, each element of the pheromone matrix is initialized to the same initial value τ_0 or a sequence initialized by the random number generator. Artificial ants begin with an empty string *S* of length *N*, and then the solution path *S* will be gradually improved by ant colonies with the iteration, and the pheromone matrix will be updated according to the quality of solutions constructed by the artificial ants.

The formula for pheromone matrix initialization is shown below.

$$P_{ij} = \tau_0 \quad ; i = 1, \cdots, N, j = 1, \cdots, K \tag{1}$$

Construction of Solution Paths

Table 1. Relevant properties of the experimental data set

	Iris	Wine	Thyroid diseases	UKD
N(data set size)	150	178	215	258
n(number of attributes)	4	13	5	5
K(number of groups)	3	3	3	5

To build a complete solution path, artificial ants assign a class tag to a data object at each time, that is, each time a class tag is assigned to an element of the solution path S. In order to balance the relationship between the amount of pheromone (represented history information) and heuristic information (the desirability of state transition *xy*), artificial ants use one of the following two ways to select the class tags for the solution path.

(1) Selected by the probability of q_0: artificial ants select the class tag with the highest pheromone concentration from the pheromone matrix to the corresponding data object, where $q_0 (0 < q_0 < 1)$ is a predefined parameter.
(2) Selected by the probability of $1 - q_0$: a pseudo-random probability distribution (denoted as P_{ij}) is used to select a is assigned to the corresponding data object.

The next point selected from the existing pheromone matrix joins the existing solution path according to the state transition formula in the (2) method. The state transition probability is defined by the following formula (2).

$$P_{ij} = \frac{\tau_{ij}}{\sum_{k=1}^{K} \tau_{ik}} = 1, \ldots, \ K \tag{2}$$

Where P_{ij} is the normalized pheromone probability of the data element *i* belonging to group *j*.

Assessment of Solution Paths

In order to evaluate the quality of solution paths constructed by artificial ants, it is necessary to perform an objective function on solution paths of the problem. The objective function is defined as the sum of squares of the Euclidean distances of each data object to the center of its group. Suppose that a given data set of *N* data objects $\left\{x_1, x_2, x_3, \cdots, x_N\right\}$ is needed to cluster into *K* groups, and the path estimation function of this clustering problem is described as follows:

$$\min F = \sum_{j=1}^{K} \sum_{i=1}^{N} \sum_{v=1}^{n} \omega_{ij} x_{iv} - m_{jv}^{\ 2} \tag{3}$$

Where

$$\sum_{j=1}^{K} \omega_{ij} = 1,\ i = 1,\cdots,N \tag{4}$$

$$\sum_{i=1}^{N} \omega_{ij} \geq 1,\ j = 1,\cdots,K \tag{5}$$

Where x_{iv} is the value of the *v-th* attribute of data element *i*, *m* is a center points matrix with $N \times K$ size, m_{jv} is average value of the *v-th* attribute of all data objects in group *j*; ω is a weight matrix with $N \times K$ size, ω_{ij} is the associated weight indicated whether the data object x_i belongs to the group *j*, defined as

$$w_{i,j} = \begin{cases} 1,\ if 1 \in group\ j \\ 0, \quad otherwise \end{cases} \tag{6}$$

$$m_{jv} = \sum_{i=1}^{N} w_{ij} x_{iv} / \sum_{i=1}^{N} w_{ij} \qquad j = 1,\cdots,K, v = 1,\cdots,n \tag{7}$$

After obtaining the weight matrix, the central m_j of each class can be obtained by using the following formula (7).

$$m_{jv} = \sum_{i=1}^{N} w_{ij} x_{iv} / \sum_{i=1}^{N} w_{ij} \qquad j = 1,\cdots,K, v = 1,\cdots,n \tag{8}$$

Pheromone Update

Pheromones play an important role in solving problems, and the pheromone matrix will be updated at the end of each iteration. The strategies of pheromone update the reflect use of dynamic information generated by artificial ants. Pheromones is updated by the following formula:

$$\tau_{ij}\left(t+1\right) = \left(1-\rho\right)\tau_{ij}\left(t\right) + \sum_{l=1}^{L} \Delta\tau_{ij}^{l}, \quad i = 1,\cdots,N, j = 1,\cdots,K \tag{9}$$

Where ρ is a constant between 0 and 1, $\left(1-\rho\right)$ represents the volatilization rate of pheromones. Larger ρ indicates that the pheromone in the artificial ant's past solution path volatilized faster. If the group label *j* is assigned to the *i-th* element of the solution path, $\Delta\tau_{ij}^{l}$ is equal to $\frac{1}{F_l}$, otherwise $\Delta\tau_{ij}^{l}$ is 0.

THE ACO-BASED CLUSTERING ALGORITHM WITH CHAOTIC FUNCTION MAPPING

This section applies the ACOC to four common datasets and studies the performance and shortcomings of the ACOC

on datasets, and then a new algorithm is proposed: ACO-Based Clustering Algorithm with Chaos (ACOCC).

SUMMARY OF EXPERIMENTAL DATASETS

In this paper, experiments to compare the ACOC and ACOCC on the four public datasets are conducted. The relevant attributes of the experimental data are shown in Table 1, and the datasets are available from UCI's Machine Learning Data Warehouse (Bache & Lichman, 2013).

The Iris dataset consists of N = 150 data objects, belonging to K = 3 different irises, each sample is described by n = 4 attributes. The Wines dataset consists of N = 178 chemical analyses data of red wine, which originated from three different cultivars, and the type of the red wines was described by n = 13 continuous attributes. The Thyroid diseases dataset consist of N = 215 samples of patients with K = 3 thyroid diseases, and each data object is described by n = 5 attributes. The User Knowledge Modeling (UKD) dataset consist of N = 258 examples of the cognitive user model, and the types of user cognition are distinguished by using an intuitive cognitive classifier, and each sample is described by N = 5 attributes.

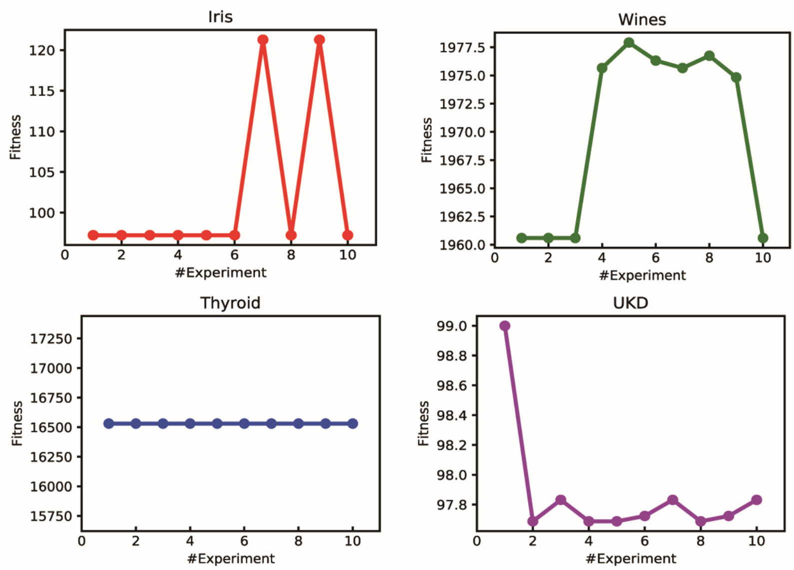
Table 2. The results of basic ACO clustering algorithm (ACOC) on four data sets

		1	2	3	4	5	6	7	8	9	10
Iris	fitness	97.222	97.222	97.222	97.222	97.222	97.222	121.277	97.222	121.277	97.222
	time	75.27	100.68	116.03	106.84	56.915	91.760	59.06	71.541	52.073	52.66
Wines	fitness	1960.59	1960.59	1960.59	1975.66	1977.92	1976.32	1975.66	1976.75	1974.83	1960.59
	time	103.68	257.26	200.11	142.26	160.19	92.92	126.87	156.98	112.44	98.83
Thyroid	fitness	16530.5	16530.5	16530.5	16530.5	16530.5	16530.5	16530.5	16530.5	16530.5	16530.5
	time	83.86	135.89	89.85	91.85	86.48	89.99	76.50	83.45	109.96	85.93
UKD	fitness	98.9992	97.6877	97.8315	97.6877	97.6877	97.7235	97.8316	97.6877	97.7235	97.8315
	time	1117.34	708.76	835.39	1144.78	1150.95	977.86	1158.54	953.71	963.70	742.95

Table 3. The statistics on 4 datasets of the basic ACO clustering algorithm (ACOC)

	fitness				time			
	optimal	worst	average	variance	optimal	worst	average	variance
Iris	97.222	121.27	102.03	10.1426	52.07	116.03	78.29	23.92
Wines	1960.59	1977.92	1969.95	8.0945	92.92	257.26	145.16	51.60
Thyroid	16530.5	16530.5	16530.5	0	76.50	135.90	93.38	17.27
UKD	97.6877	98.9992	97.8692	0.4021	708.72	1158.54	975.40	169.50

Figure 1. The fitness on 4 datasets of the basic ACO clustering algorithm (ACOC)



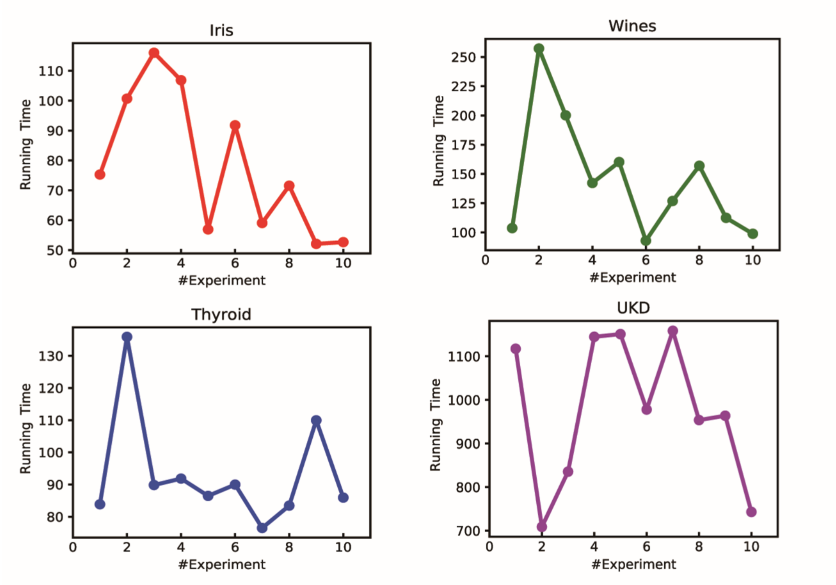
The Performance Analysis of The Basic ACO Clustering Algorithm (ACOC)

After finishing experiments, the performance of the ACOC is analyses. Table 2 shows the final fitness and running time (All time units in this article are seconds) of ACOC on ten times experiments.

As can be seen from Table 2, for the Iris dataset, with the ten times repeating experiments,the ACOC converges to the same optimal solution in eight times. But the other two(7 and 9) times converge to the same suboptimal solution, which indicates that the ACOC has not searched the path of the optimal solution in these two times, that is, they have fallen into the local optimal situation. The running times converged to the optimal solution of the algorithm starts from 52.66 to 116.03 in the other eight times. This is indicated the instability of the ACOC. Regarding the Wines dataset, the ACOC converges to the optimal solution four times in ten experiments, and the other six times converge to other different solutions. And running times converged to the optimal solution in the four times distributes from 98.83 to 200.1, it also greatly indicates the instability of the ACOC. For the Thyroid dataset, the ACOC stably converges to 16530.5, and its running times from 76.502 to 109.96 indicates the instability of the ACOC. For the UKD dataset, the optimal solution of the ACOC is 97.6877, the worst case is 98.9992, and the running time is from 742.945 to 1158.54, which indicates that the ACOC is very unstable.

As can be seen from Table 3, Figure 1 and Figure 2, optimal solution and running time of the ACOC have a large variance in the Iris dataset and Wines dataset, indicating that the ACOC is relatively unstable on finding solutions. The variance of the running time is large too in the Thyroid and UKD dataset, but the variance of the obtained optimal solution is small. This situation indicated that the artificial ant colony in the algorithm could not evenly search all the solutions in the solution space.

Figure 2. *The running time on 4 datasets of the basic ACO clustering algorithm (ACOC)*



According to the above problem happened ACOC, an improved ACO clustering algorithm based on chaotic mapping function (ACOCC) is proposed in this paper. The chaotic sequence is well known for its sensitivity to initial conditions, which makes it a good choice for the deterministic stochastic generator and other chaotic search function algorithms. These special applications and features make it become a candidate for improving evolutionary algorithms. And we attempt to improve the ACOC from two directions based on the chaotic mapping function.

CHAOTIC MAPPING FUNCTION

An Overview of Chaotic Mapping Function

The chaotic mapping function has two important characteristics, and one is abnormally sensitive to the initial conditions, another one is ergodicity, which means the chaotic motion can reach all the states in the relevant problem domain without repetition (Liu Daohua Li Gang, 2011). and these two characteristics help the chaos search optimization algorithm proposed (L. Li, 2013).

In this paper, the selected function is Logistic function which is a simple chaotic mapping function, and the function is:

$$Z_{n+1} = \mu Z_n \left(1 - Z_n\right); \quad n = 1, 2, 3, \cdots, Z \tag{10}$$

Where μ is the control parameter, and when μ = 4, the logistic function map is completely in the chaotic state (B. Li & Jiang, 1997). And we can obtain i chaotic variables, by giving i initial values with small difference in the equation (9) by using the properties of value-sensibility from chaotic function to initial values (Liu Lezhu Zhang Jiqian, 2013)). In this paper, all the control parameter μ and chaotic initial value Z_0 set as $Z_0 = 0.6, \mu = 3.8$.

ACO-BASED CLUSTERING ALGORITHM WITH CHAOS (ACOCC)

The ACO-Based Clustering Algorithm with Chaos (ACOCC) mainly includes two algorithms, and one is the ACO-Based Clustering Algorithm with Chaos initialization (ACOCI), another one is ACO-Based Clustering Algorithm with Chaos initialization and pheromone updating (ACOCIU). The ACOCI only uses chaotic mapping function when initializing the pheromone matrix. Based on AOCI, the ACOCIU additionally utilize chaotic mapping function in the pheromone updating stage.

ACOCI

The ACOC initialize the pheromone matrix to a tiny initial value called τ_0 when initializing the pheromone matrix or generate the same size of the pheromone matrix by using a random number generator. Whether using a chaotic mapping function or using a random generator to generate a random function is still controversial (Mombeini M., 2011), but using deterministic chaotic mapping has shown excellent experimental results compared with random number generator in the experiment (Bucolo, Caponetto, Fortuna, Frasca, & Rizzo, 2002).

In many other improved ACOC, it is a common method to make the algorithm more exploratory at present when insert random number generator which generate the random number to initialize the pheromone matrix in the first iteration and using chaotic mapping function has achieved good results instead of random number generator when initializing the pheromone matrix in other areas. Therefore, this paper will use the chaotic mapping function to initialize the pheromone matrix in the first iteration.

It can make the artificial ants as much as possible distribute to the different initial state by using the chaotic mapping function to initialize the pheromone matrix, and the artificial ants in the search process can achieve more states so that it can improve ergodicity of the search.

The formula of initializing pheromone matrix for the ACOC can be expressed as follows:

$$P_{ij} = \tau_0,\ i = 1,\ldots,K; j = 1,\ldots,N \tag{11}$$

or

$$P_{ij} = r_k,\ i = 1,\ldots,N; j = 1,\ldots,K; k = 1,\ldots,N \times K \tag{12}$$

Where r_k represents the k random number generated by the random number generator.

The formula when the ACOCI initializes the pheromone concentration is as follows:

$$P_{i,j} = Z_k\ , i = 1,\ldots,K; j = 1,\ldots,N; k = 1,\ldots,N \times K \tag{13}$$

$$Z_{k+1} = \mu Z_k \left(1 - Z_k\right), k = 1,\ldots,N \times K \tag{14}$$

Assuming that a set of data containing *N* data objects needs to be clustered into *K* different groups, the pseudo-code which ACOCI initialize the pheromone concentration can be described as Algorithms 1.

Local Search

The local search can help find good results when the heuristic information of the problem is not easy to obtain in many ACO algorithms. In these algorithms, local searches may be performed on all solutions built by artificial ants or may also be performed on a portion of them, such as in a percentage of the number of solutions on the local search (Shelokar, Jayaraman, & Kulkarni, 2004). In this paper, a local search is performed using 20% of the optimal solution for all artificial ants. Before making a local search, the solution set built by the artificial ant will be sorted ascending order according to the value of the objective function. The local search step is performed using the L solution paths of the highest fit (i.e., the lowest objective function).

Algorithm 1 A high-level description of improved initializing

1. Initialize $\tau = Z_0$
2. For i=1 to N
3. For j = 1 to K 4. $P_{ij} = \tau$ 5. $\tau = \mu\tau\left(1-\tau\right)$
6. End
7. End

The local search algorithms are as follows.

With the probability threshold p_{ls} between 0 and 1, the adjacent path of the solution path $S_k\left(k=1,\cdots,L\right)$ is generated by the following steps:

(1) $k=1$.
(2) Let S_t as a temporary solution path and assign $S_t = S_k\left(i\right), i=1,\cdots,N$.
(3) For each element of S_t, a random number r between *0* and *1* is generated. If $r \leq p_{ls}$, then an integer *j* that is between $\left[1,K\right]$ and not equal to $S_k\left(i\right)$ is generated, and then *j* is assigned to $S_t\left(i\right)$.
(4) Calculate the clustering centers and weights associated with S_t, and use the path quality formula to find its objective function F_t. If F_t is less than F_k, then replace S_k with S_t and replace F_k with F_t.
(5) k is incremented by *1*, if $k \leq L$, return to step (2), otherwise, it stops.

ACOCIU

Adding local search steps in the ACOC can help the algorithm to obtain better solutions path in some areas. However, the basic local search step may cause the pheromone on the solution path to accumulate too fast, leading to the early maturing of the algorithm finally which represents the algorithm may converge to a suboptimal solution rather than optimal solution early. And the reason for the early maturing of the algorithm is that some of the segments that do not exist in the local search are searched and given a higher pheromone concentration, which causes the artificial ants frequently select this path due to the pheromone concentration instead of the unexplored path, and then result in the optimal solution is not found or its likelihood is reduced.

In order to solve the problem of early convergence of the ACOC, this paper introduces a chaotic mapping function model in the pheromone updating step and tries to avoid stuck into local optimal situation by applying the three characteristics of the chaotic mapping function to the local search step.

The update pheromone function of the ACOC is shown in formula (8). For introducing the chaotic mapping function into the pheromone update step, the update pheromone formula will be replaced by the following formula:

$$\tau_{ij} = \left(1-\rho\right)\tau_{ij}\left(t\right) + \sum\nolimits_{k=1}^{L} \Delta\tau_{ij}^{k} + \varepsilon Z_{ij}, \quad i = 1,\cdots,N; j = 1,\cdots,K \tag{14}$$

$\Delta\tau_{ij}^{k}$ represents the pheromone produced by the artificial ant *k* in searching solution path, Z_{ij} means the perturbation factors produced by using the chaotic mapping function, $\rho\left(0 \le \rho \le 1\right)$ represents the volatilization rate of pheromone, and ε represents a tiny controlled constant.

The core idea of introducing chaotic mapping function in pheromone update step is to introduce chaotic mapping function for generating a scrambling factor, which introduces a perturbation factor in the solution path created by artificial ant, so that the artificial ants' path selection not only depend on the past "experience" as a guide but also add certain ergodicity to increase the artificial ants search space and avoid falling into the local optimal solution.

Figure 3. The convergence of the three algorithms on the data set Iris

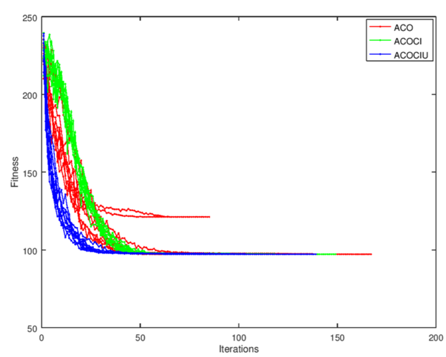


Figure 4. The convergence of the three algorithms on the data set Thyroid

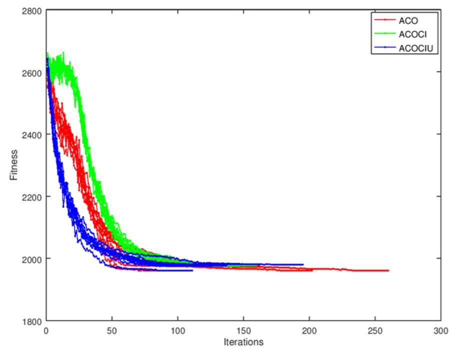


Figure 5. The convergence of the three algorithms on the data set Wines

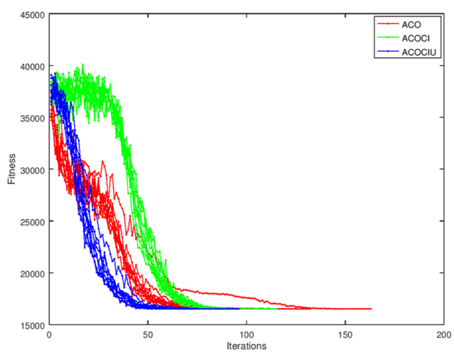
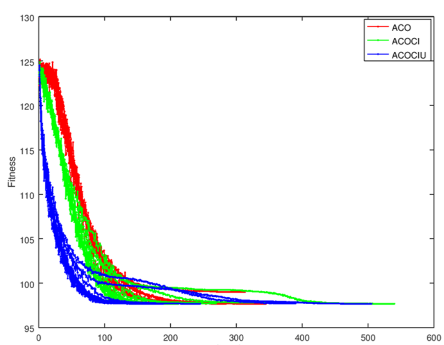


Figure 6. The convergence of the three algorithms on the data set UKD



EXPERIMENT AND RESULT ANALYSIS

The ACO-Based Clustering Algorithm with Chaotic Function Mapping proposed in this paper is implemented in MATLAB language. Our experiments using 4 publicly available data sets and compares the convergence, the fitting, the run time, the number of iterations, the stability of the ACOCC and ACOC algorithm, all the statistics are based on 30 independent runs, and the maximum number of iterations per experiment is 1000.

Comparison and Analysis of Algorithm Convergence

In order to compare the convergence performance of ACOC and ACOCC algorithm, we draw the graph of the convergence data of the ACO, ACOCI, and ACOCIU algorithms running ten times on the four data sets in this section, so as to intuitively see the differences.

Figure 3 shows the convergence of the three algorithms running ten times on the data set Iris, respectively. It can be seen that for the data set Iris, compared to the ACO, the convergence rate of the ACOCI is basically the same, and for the ACOCIU, the convergence speed is faster than that of the other two algorithms, and the number of iterations needed to achieve the convergence value is also relatively reduced.

Figure 4 shows the convergence of the three algorithms running ten times on the data set Thyroid, respectively. It can be seen that for the data set Thyroid, the ACOCI tends to converge to chaos at the beginning of the algorithm, and the solution decreases rapidly and converges, and finally converges to the same result of the ACO. And for the ACOCIU, the fitness of solution decreased rapidly in the process of solving, but some fluctuations in the falling process, indicating that the introduction of the chaotic mapping function makes the artificial ants explore space increases, in order to obtain the optimal solution under the same iteration is better than basic ant colony clustering, and the fluctuations in the iterative process can also make the artificial ants jump out of sub optimal solutions have been obtained solution to explore other solutions.

Figure 5 is the convergences of the ACO, the ACOCI, and the ACOCIU on the data set Wines. The red figure is the convergence of ACO, at the beginning stage of the algorithm, the rapid convergence of the ACO, but there is a stage of stagnation in the middle, follow-up and then continue to converge downward, indicating that the ACO trapped into a local optimal solution in the middle stage, and jump out of local optimal solution limit after a period of time.

The ACOCI approaches chaos at the beginning of the algorithms and converges to the optimal solution after sharp declines. It means that the ACOCI with the chaotic mapping function at the beginning of the algorithm makes the artificial ants accumulated enough path information in the beginning stage, which can quickly converge in the following.

The fitness of the ACOCIU is steadily descending in the operation process of the whole algorithm, and the descent speed is faster than the ACO. In the convergence stage, the ACOCI converges slowly, which shows that the introduction of chaotic mapping function in the pheromone initialization stage makes the search space of artificial ants larger so that the search time is longer. The ACOCIU convergences faster than the ACO, the reason is that the chaotic mapping function introduced at the pheromone update stage is conducive to the introduction of disturbance factors for artificial ants (Andries P. Engelbrecht, 2005), which can make the algorithm jump out of local optimal solution faster to get a better solution, so the convergence speed is faster.

Figure 6 shows the convergence of the three algorithms running ten times on the data set UKD, respectively. In the initial stage of the traditional ACOC algorithm, the lines of 10 overlapping part of the ACOC is thick, which illustrated the ACOC is not stable in the beginning stage, and there are many solution paths to search, the following solution path fitness decreased gradually, gradually convergence.

The line of the ACOCI is below the ACOC, which shows that the fitness of the solution path obtained by the improved algorithm is generally lower than that of the ACOC. And its convergence point is prior to the ACOC, which shows that the ACOCI has faster convergence speed than the ACOC.

The line of the ACOCIU is thin at the beginning of the algorithm, which indicates that the overlapping part is not much, that is, the path quality of the algorithm is little different at the beginning. But the slope of the line is much larger than that of the basic ant colony clustering algorithm. It is shown that the ACOCIU is better than the ACO in the process of solving the path to jump out of local optimal and find more optimal solutions. This is mainly due to the introduction of the chaotic mapping function in the pheromone update stage so that the artificial ants can introduce the disturbance factor in the search stage and search for more paths. In the vicinity of the convergence point of the algorithm, the convergence point of the ACOCIU is ahead of the convergence point of the ACO, but there are a few times stagnant running in the upcoming convergence, and then converge again after a certain period of time, which shows that the disturbance factor introduced by the ACOCIU for the artificial ants does effectively make the artificial ants find more solutions in the vicinity of the solution, which can effectively make the artificial ants find a better solution path with fast speed. So the convergence rate of the ACOCIU is faster than the ACO. But stagnation in the vicinity of the upcoming convergence point indicated that there are still a few operations sink into the local optimal solution for some time in the iterative process and then jumping out. This shows that the parameter setting of the improved ant colony clustering algorithm is not optimal, which may be a direction for future research.

Comparison and Analysis of Algorithmic Performance Statistics

Table 4 shows the fitting and time of the final convergence of the ACOCI running ten times on the data sets Iris, Wines, Thyroid, and User Knowledge Modeling. Table 5 shows the fitting and time of the final convergence of the ACOCIU running ten times on the data sets Iris, Wines, Thyroid, and User Knowledge Modeling.

Table 4. Experimental results of ACOCI

		1	2	3	4	5	6	7	8	9	10
Iris	fitness	97.2221	97.2221	97.2221	97.2221	97.2221	97.2221	97.2221	97.2221	97.2221	97.2221
	time	85.01	90.49	86.4	91.37	86.26	85.65	96.07	90.82	88.26	67.09
Wines	fitness	1975.66	1980.28	1980.28	1980.28	1975.66	1976.98	1980.28	1980.83	1976.98	1960.59
	time	116.92	146.57	117.12	191.72	123.73	132.09	111.14	119.70	134.64	109.16
Thyroid	fitness	16530.5	16530.5	16530.5	16530.5	16530.5	16530.5	16530.5	16530.5	16530.5	16530.5
	time	65.445	73.91	76.07	79.42	65.42	63.96	61.41	67.00	68.66	63.94
UKD	fitness	97.7234	97.6877	97.7234	97.6877	97.6877	97.6877	97.7235	97.7235	97.6877	97.7234
	time	495.833	521.396	503.053	1434.14	536.867	880.911	660.152	505.732	804.622	1196.35

Table 5. Experimental results of ACOCIU

		1	2	3	4	5	6	7	8	9	10
Iris	fitness	97.2221	97.2221	97.2221	97.2221	97.2221	97.2221	97.2221	97.2221	97.2221	97.2221
	time	94.760	101.93	91.993	90.426	103.37	80.737	98.974	89.974	93.285	86.360
Wines	fitness	1975.66	1977.92	1980.83	1976.98	1976.98	1975.66	1977.92	1976.98	1976.98	1976.85
	time	138.832	146.548	144.565	148.340	143.772	144.521	148.606	155.361	140.661	157.829
Thyroid	fitness	16530.5	16530.5	16530.5	16530.5	16530.5	16530.5	16530.5	16530.5	16530.5	16530.5
	time	94.401	85.258	87.005	88.652	95.655	91.426	87.111	91.980	89.309	87.087
UKD	fitness	97.7234	97.6877	97.6877	97.828	97.7234	97.7234	97.7235	97.8174	97.7235	97.8315
	time	862.066	2069.26	705.226	830.87	670.962	832.233	1017.37	1126.05	859.758	689.602

In order to compare the overall performance of the ACOCC and the ACOC. The following statistics are used to compare the experimental data of the two algorithms, which are the optimal fitting degree, the worst fitting degree, the average fitting degree, the shortest running time, maximum run time, average run time, convergence fit variance. Table 6 shows the statistics of the ACOC and the ACOCC in the data set Iris.

The experimental data show that the ACOCC can find better solutions than the ACOC in searching the solution path, the better stability, and the comparable running time. Table 8 and Table 9 respectively show the Statistics of the ACOC and the ACOCC on the data set Wines and Thyroid. Figure 7 to Figure 10 shows the running time and the fitting curve of the ACOC, the ACOCI, and the ACOCIU.

Table 6. The Statistics of Three Algorithms on the Data Set Iris

algorithms			fitness			time		
	optimal	worst	average	variance	shortest	maximum	average	variance
ACOC	97.222	121.27	102.03	10.142	52.073	116.03	78.285	23.916
ACOCI	97.222	97.222	97.222	0	80.737	103.37	93.181	6.9689
ACOCIU	97.222	97.222	97.222	0	67.090	96.070	86.742	7.690

Table 7. The Statistics of Three Algorithms on the Data Set Wines

algorithms	fitness				time			
	optimal	worst	average	variance	shortest	maximum	average	variance
ACOC	1960.59	1977.92	1969.95	8.094	92.92	257.26	145.16	51.60
ACOCI	1960.59	1980.83	1976.78	6.057	109.17	191.72	130.28	5.97
ACOCIU	1960.59	1980.83	1976.78	6.056	109.16	191.72	130.27	24.45

Table 8. The Statistics of Three Algorithms on the Data Set Thyroid

algorithms	fitness				time			
	optimal	worst	average	variance	shortest	maximum	average	variance
ACOC	16530.5	16530.5	16530.5	0	76.5	135.90	93.38	17.27
ACOCI	16530.5	16530.5	16530.5	0	61.46	79.43	68.53	3.46
ACOCIU	16530.5	16530.5	16530.5	0	61.41	79.42	68.52	5.95

Table 9. The Statistics of Three Algorithms on the Data Set UKD

algorithms	fitness				time			
	optimal	worst	average	variance	shortest	maximum	average	variance
ACOC	97.687	98 .999	97.869	0.402	708.71	1158.54	975.39	169.49
ACOCI	97.687	97.723	97.705	0.018	495.83	1434.14	753.90	413.00
ACOCIU	97.687	97.723	97.705	0.018	495.83	1434.14	753.90	329.68

For the data set Iris, it can be seen from Figure 7 that the ACOC is unstable on the obtained optimal solution, the solution cannot reach the optimal solution twice, and its run time fluctuation range is larger. And algorithm ACOCI and the algorithm ACOCIU can stably obtain the optimal solution, and the running time fluctuation range is smaller than ACOC.

For the data set Thyroid, it can be seen from Figure 9 that ACOC, ACOCI, and ACOCIU are able to obtain the optimal solution stably. But the operation time images of the ACOC are most above the ACOCI, all above the ACOCIU, and the fluctuation range is larger than the ACOCI and the ACOCIU.

For the data set UKD, it can be seen from Figure 10 that the ACOC algorithm cannot find the optimal solution a few times, the ACOCI and the ACOCIU can find out the optimal solution stably. But the operation time images of the ACOC are most above the ACOCI and the ACOCIU, which shows that the running time of the ACOC is longer than the running time of the ACOCI and the ACOCIU.

According to the above experimental data and charts of the ACOC, the ACOCI, the ACOCIU on Iris, Wines, Thyroid, UKD. The ACOCI and the ACOCIU are better than the ACOC either in the optimal solution or in the run time or algorithm stability. We can see that the ACOCC is a better algorithm.

There are four data sets using in this paper and data description attribute respectively from 3 to 13. As mentioned above, the result of the ACOCC is better than that of the ACOC. Therefore, the improved ACOCC is a better, effective, and more promising clustering algorithm for cluster analysis.

Figure 7. Experimental Results of Three Algorithms on Data Set Iris

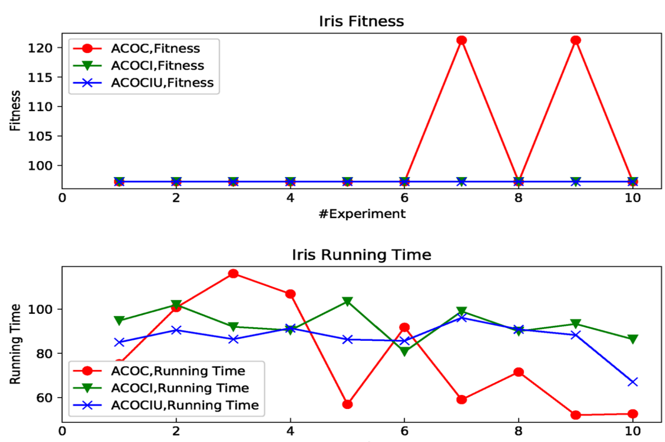


Figure 8. Experimental Results of Three Algorithms on Data Set Wine

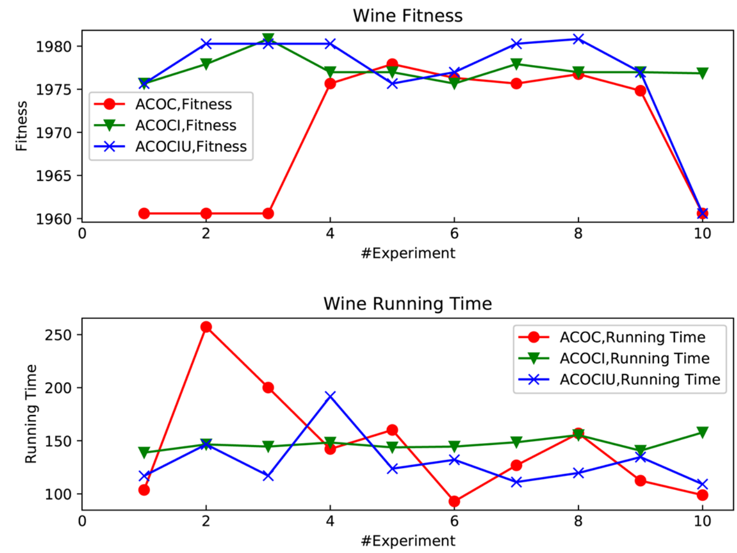


Figure 9. Experimental Results of Three Algorithms on Data Set Thyroid

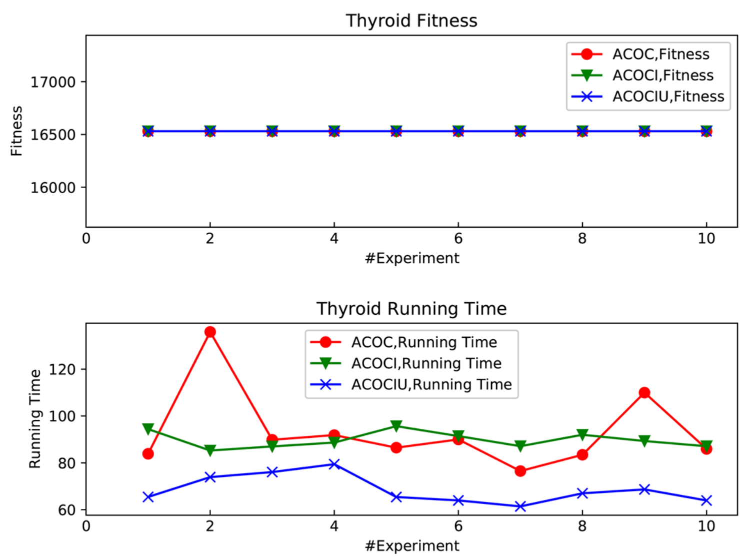
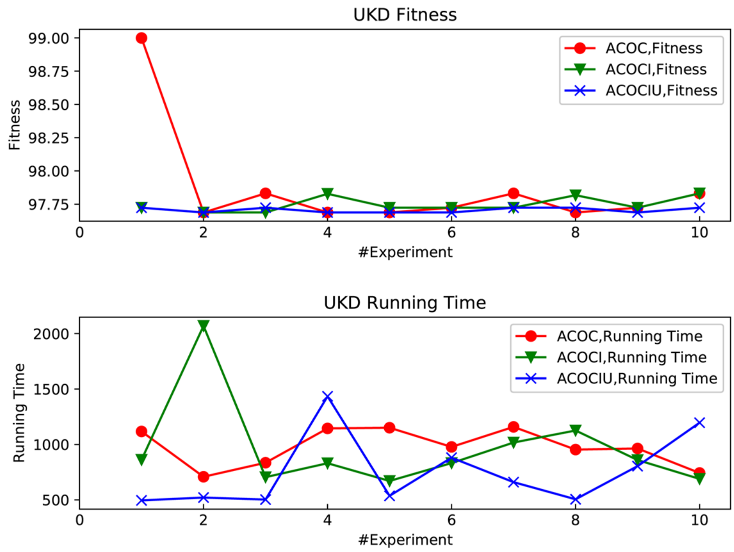


Figure 10. Experimental Results of Three Algorithms on Data Set UKD



CONCLUSION AND FUTURE WORK

Conclusion

In order to overcome shortcomings of the ACOC applied in clustering problems (running time is slow, easy to fall into the local optimal solution, the algorithm is unstable, and so on), this paper introduces a novel algorithms ACOCC, in which the pheromone matrix is initialized by chaos sequence generating mapping function to avoid using a same small value as the initial pheromone or using the random number generator to generate an initial sequence. This could help artificial ants initialized as much as possible in different initial states at the beginning of the algorithms to solve the problem that ant search state is not complete. In order to solve the ACOC easy to fall into the local optimal solution, the chaotic mapping function is multiplied by a small control variable as the superposition of pheromone in the pheromone update stage of local search to construct a small disturbance factor, so that artificial ants jump out of the local optimal solution of the path to explore its nearby solution paths.

The improved ant colony clustering algorithm based on chaotic mapping function can significantly reduce the running time, iteration times and increase the stability of clustering algorithms in large data sets. In order to evaluate the performance of the ACOCC, we implement and compare the fitness, the number of iterations, the optimal solution, and stability between the new algorithm and the basic ant colony clustering algorithm in four public data sets. The experimental results show that the improved ant colony clustering algorithm based on chaotic mapping function has some advantages over the basic ant colony clustering algorithm in the quality of the solution, the number of iterations, the optimal solution, and stability.

FUTURE WORK

Then there are several areas worth further studying:

(1) This paper only selected four data sets to do a comparison of different algorithms. In the next step, we will select more data sets and conduct the algorithm experiments to study the universality of the algorithm.
(2) To further study how to set the parameters of the improved ant colony clustering algorithm to optimize the clustering effect.
(3) The improved ant colony clustering algorithm based on chaotic mapping function is used to perform the parallel computation to improve the operational efficiency of data clustering.

ACKNOWLEDGMENT

This work was partially supported by the Natural Science Foundation of Guangdong Province of China (Grant No.2020A1515010691), Science and Technology Project of Guangdong Province of China (Grant No.2018A0124), and National Natural Science Foundation of China (Grant Nos. 61573157 and 61703170). The authors also gratefully acknowledge the reviewers for their helpful comments and suggestions that helped to improve the presentation.

REFERENCES

Bache, K., & Lichman, M. (2013). *UCI machine learning repository*. Academic Press.

Baig, A. R., Shahzad, W., & Khan, S. (2013). Correlation as a heuristic for accurate and comprehensible ant colony optimization based classifiers. *IEEE Transactions on Evolutionary Computation*, *17*(5), 686–704. doi:10.1109/TEVC.2012.2231868

Bucolo, M., Caponetto, R., Fortuna, L., Frasca, M., & Rizzo, A. (2002). Does chaos work better than noise? *IEEE Circuits and Systems Magazine*, *2*(3), 4–19. doi:10.1109/MCAS.2002.1167624

Dorigo, M., Birattari, M., & Stützle, T. (2006). Ant colony optimization artificial ants as a computational intelligence technique. *IEEE Computational Intelligence Magazine*, *1*(4), 28–39. doi:10.1109/CI-M.2006.248054

Dorigo, M., & Gambardella, L. M. (1997). Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, *1*(1), 53–66. doi:10.1109/4235.585892

Engelbrecht, A. P. (2005). *Fundamentals of computational swarm intelligence*. John Wiley & Sons.

Gao, S., Wang, Y., Cheng, J., Inazumi, Y., & Tang, Z. (2016). Ant colony optimization with clustering for solving the dynamic location routing problem. *Applied Mathematics and Computation*, *285*, 149–173. doi:10.1016/j.amc.2016.03.035

Gonzalez-Pardo, A., Jung, J. J., & Camacho, D. (2017). {ACO}-based clustering for Ego Network analysis. *Future Generation Computer Systems*, *66*, 160–170. doi:10.1016/j.future.2016.06.033

Han, J., Pei, J., & Kamber, M. (2011). *Data mining: concepts and techniques*. Elsevier.

Hidayat, D. T., Fatichah, C., & Ginardi, R. V. (2016). Pattern reduction enhanced ant colony optimization clustering algorithm. *2016 International Seminar on Application for Technology of Information and Communication (ISemantic)*. doi:10.1109/isemantic.2016.7873858

Jabbar, A. M., Ku-Mahamud, K. R., & Sagban, R. (2018). Ant-based sorting and ACO-based clustering approaches: A review. *ISCAIE 2018 - 2018 IEEE Symposium on Computer Applications and Industrial Electronics*, 217–223. doi:10.1109/ISCAIE.2018.8405473

Kao, Y., & Cheng, K. (2006). An ACO-based clustering algorithm. *International Workshop on Ant Colony Optimization and Swarm Intelligence*, 340–347. doi:10.1007/11839088_31

Li, B., & Jiang, W. (1997). Chaos optimization method and its application. *Control Theory & Applications*, *4*, 613–615.

Li, L. (2013). *Chaotic ant colony algorithm and its application*. Beijing: Chinese Science and Technology Press.

Liu & Li, . (2011). Ant Colony Hybrid Optimization Method Based on chaos technology. *Computer Engineering and Applications*, *47*(13), 42–45.

Liu & Zhang, . (2013). A modified chaotic ant colony optimization algorithm. *Wuli Xuebao*, *62*(17), 170501. doi:10.7498/aps.62.170501

Menéndez, H. D., Otero, F. E. B., & Camacho, D. (2016). Medoid-based clustering using ant colony optimization. *Swarm Intelligence*, *10*(2), 123–145. doi:10.1007/s11721-016-0122-5

Mombeini, M., & Ali, N. M. (2011). Ant Colony Optimization with Chaos. *Recent Advances in Electrical Engineering and Electronic Devices*, 217–223.

Monmarché, N., Slimane, M., & Venturini, G. (1999). On Improving Clustering in Numerical Databases with Artificial Ants. In Advances in Artificial Life (pp. 626–635). doi:10.1007/3-540-48304-7_83

Nayyar, A. (2018). Advances in Swarm Intelligence for Optimizing Problems in Computer Science. Advances in Swarm Intelligence for Optimizing Problems in Computer Science. doi:10.1201/9780429445927

Nayyar, A. (2018). Introduction to Swarm Intelligence. Advances in Swarm Intelligence for Optimizing Problems in Computer Science. doi:10.1201/9780429445927-3

Nayyar, A., & Singh, R. (2014). A Comprehensive Review of Ant Colony Optimization ({ACO}) based Energy-Efficient Routing Protocols for Wireless Sensor Networks. *International Journal of Wireless Networks and Broadband Technologies*, *3*(3), 33–55. doi:10.4018/ijwnbt.2014070103

Nayyar, A., & Singh, R. (2015). A Comprehensive Review of Ant Colony Optimization (ACO) based Energy-Efficient Routing Protocols for Wireless Sensor Networks. *International Journal of Wireless Networks and Broadband Technologies*, *3*(3), 33–55. doi:10.4018/ijwnbt.2014070103

Nayyar, A., & Singh, R. (2016). Ant Colony Optimization — Computational swarm intelligence technique. *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, 1493–1499.

Pei Zhenkui Li Hua, S. J. H. J. (2008). Ant colony clustering algorithm research and application. *Computer Engineering and Design*, *29*(19), 5009–5013.

Shelokar, P. S., Jayaraman, V. K., & Kulkarni, B. D. (2004). An ant colony approach for clustering. *Analytica Chimica Acta*, *509*(2), 187–195. doi:10.1016/j.aca.2003.12.032

Singh, N., Singh, D., & Pant, B. (2019). ACOCA: Ant Colony Optimization Based Clustering Algorithm for Big Data Preprocessing. *International Journal of Mathematical*, *Engineering and Management Sciences*, *4*, 1239–1250. doi:10.33889/IJMEMS.2019.4.5-098

Wang, X., & Luo, K. (2019). LF Ant Colony Clustering Algorithm with Global Memory. *Computer Engineering and Applications, 20*.

Wu, S., Yan, L., Zhang, Q., & Shen, X. (2018). Research on Clustering Algorithm Based on Improved Ant Colony Algorithm. *Computer & Digital Engineering, 9*.

Yu & Pei, . (2010). An improved ant colony clustering algorithm. *Journal of Zhengzhou University Science Edition*, *42*(3), 59–62.

Zhang & He, . (2006). A survey of ant colony clustering algorithms. *Computer Engineering and Applications*, *42*(16), 171–174.

Lei Yang received the B.S. degrees in computer science from Henan normal University, China, in 2001, M.S. degrees in computer science from Shaanxi normal University, China, in 2006, and the Ph.D. degree in computer science from the South China agricultural university at Guangzhou, in 2016. Now he is a Lecturer in Department of computer science and engineering at the South China agricultural university. His current research interests include ant colony optimization, the evolutionary multi-objective optimization, machine learning, data mining and applications.

Xin Hu received the B.S. degrees in new energy science and engineering form Lanzhou University of Technology, China, in 2017. He is currently pursuing the M.S. degree with the department of computer science and engineering, South China agricultural university. His current research interests include evolutionary multi-objective optimization and machine learning and data mining and applications.

Hui Wang is a doctoral graduate student at the College of Mathematics and Informatics, South China Agricultural University. She got a master's degree from the School of Information Engineering, Jiangxi University of Science and Technology in 2017. Her research interests include evolutionary algorithms, intelligent computing, multi-objective optimization, and wireless sensor networks.

Wensheng Zhang is the researcher of institute of automation, Chinese academy of sciences, doctoral supervisor, member of the academic committee of the institute, director of major projects. His interests include Artificial intelligence, machine learning, big data knowledge mining, probability graph model representation and reasoning, Deep Learning, precision perception and intelligent control, 3d digital physical simulation, embedded video image processing.

Kang Huang received the B.E. degrees in electronic and information engineering from Hunan Institute of Science and Technology, China, in 2016. Now he is currently pursuing the M.Agr. degrees in Agricultural Engineering and Information Technology, South China Agricultural University. His current research interests include gene expression programming, differential evolution, and the applications of evolutionary computations.

Dongya Wang received the B.Eng. in computer science and technology from Zhengzhou University, China, and M.Sc. in computer science from City University of Hong Kong, HK, in 2015. He is currently working towards his Ph.D. degree in constrained multi-objective optimisation at the University of Exeter.