An Optimized Ant System For Clustering With Elitist Ant And Local Search

Ming-Pan LI[a], Min YAO[b]

[1]*Department of Computer Science ZheJiang University HangZhou 310000, China*
[2]*Department of Computer Science ZheJiang University, HangZhou 310000, China*

[a]*kinglmp@163.com* [b]*myao@zju.edu.cn*

Abstract: Clustering analysis is an important field in data mining, and also one of the current research hotspots in computer science. This paper focus on some classical data clustering algorithms and swarm intelligence, especially ant colony optimization, trying to combine these two kinds of algorithms and improve the efficiency and accuracy of data clustering. This paper proposes a new ant colony optimization data clustering algorithm, named ant colony clustering algorithm with elitist ant and local search (ACC-EAL). This algorithm adopts a new pheromone incremental calculation method, making the distances among the clusters tend to increase, and the clusters get denser. Meanwhile local search provides the ants more opportunity to find optimal solution and the elite ant strategy makes the ants with optimal solutions contribute more to the pheromone increment.

1.Introduction

Ant system (AS) algorithm is a kind of heuristic bionic evolutionary system based on swarm intelligence, and is proposed by Italian scholar M.Dorigo first to solve the TSP problem[1-4]. The experimental results prove that ant algorithm is very robust and has the ability to find the optimal solution, but on the other hand it has the defect of low convergence speed and the tendency to stagnate. Later L.M. Gambardella and M. Dorigo put forward an improved ant algorithm, that is, Ant-Q, which replace the stochastic proportional choice rule of AS by pseudo random proportional state transition rule in Ant-Q, making the algorithm get better balance between knowledge exploration and knowledge exploitation. And based on Ant-Q M.Dorigo proposed the ant colony system (ACS)[5,6] . M.Dorigo and other scholars further develop ant algorithms into a general used optimization technique, named ant colony optimization (ACO), and all the algorithms that match the ACO framework is called ant optimization algorithm. At the moment ant colony optimization algorithm has been used in solving combination optimal problem, function optimization problem, robot path planning, data mining, network routing, and has achieved good results. At present some well-studied ant system includes ant optimization algorithm, the Rank-based Version of Ant System proposed by Bullnhemier[7] , the ant nest and ant eggs classification model raised by Deneubourg, and the MAX-MIN Ant System (MMAS) by T.Stützle and H.Hoos[8-9], and the Best-Worst Ant System (BWAS) by O.Cordón[10], etc.

Overall, the present study of ant colony algorithm is still in its infancy, many ideas are still budding, the theoretical system has not been well established, and the convergence has not yet been rigorously proven, but there is no doubt that the algorithm has a great advantage and a bright outlook in solving complex optimization problems, especially discrete optimization.

Data mining aims to extract knowledge from data, and data mining is an interdisciplinary research field, the goal of data mining is to discover knowledge that is accurate and more important, is comprehensible for the users. There are several different aspects of data mining, that is, classification, regression, dependence modeling and clustering, etc. Clustering aims to discover some groups or organizations of data samples in a given dataset, often by quantifying the similarities and dissimilarities between two different objects. And the samples in a cluster are similar with the samples in the same cluster while are dissimilar with the samples not in the different cluster. There are many cluster algorithms are based on swarm intelligence, such algorithms are often distributed, and well self-organized, robust and scalable. Currently most swarm intelligence based clustering algorithms are ant clustering algorithms and particle clustering algorithms. Ant based clustering models include

the ant nest and ant eggs classification model raised by Deneubourg, and the ants task assignment models put forward by Bonabeau, and the ant nests classification model by Lumer and Faieta. There are many other models such as models based on information entropy, fuzzy neural network. All these models and algorithms, although have made great progress, is still flawed, for example, some models need to set a lot of parameters. Another type of clustering algorithms attempts to combine classical clustering algorithm and ant optimization algorithm, such as combining fuzzy c-means clusters and ACO [11], also achieved remarkable results.

In this paper we proposed an ant colony optimization (ACO) algorithm for the cluster task of data mining. In this task, the goal is to assign each case to one cluster, based on the similarity of the case with others. The rest of this paper is organized as follows. In section II we present a brief overview of ant colony optimization, we will introduce the characteristic and basic framework of ACO algorithms. In section III we will introduce our new cluster algorithm based on ant colony. And in Section IV we will introduce some experiments and show the performance of our algorithm compared with some classical cluster algorithm. Finally, section V concludes this paper and points directions of the future research.

2. Ant Colony Optimization

2.1 An overview of ACO

ACO algorithms are stochastic search procedures. Their central component is the pheromone model, which is used to probabilistically sample the search space [12]. An ACO algorithm is a system based on agents which simulate the natural behavior of real ants, such as foraging, nest, and task assignment, etc. This kind of system as a new metaheuristic was proposed to solve combinatorial optimization problem, and is proven robust, scalable and versatile. In ACO, each path followed by an ant is associated with a candidate solution for a given problem. And how much pheromone an ant deposited on a path is proportional to the quality of the candidate solution for the problem correspond to the path; Another idea behind ACO is that when an ant need to make a choice between more than two paths, the path with larger amount of pheromone is more likely to be chosen.

In general, the ACO approach attempts to solve an optimization problem by repeating the following two steps [11]:

1) Candidate solutions are constructed using a pheromone model, that is, a parametrized probability distribution over the solution space;

2) The candidate solutions are used to modify the pheromone values in a way that is deemed to bias future sampling toward high quality solutions.

2.2 The framework of basic ACO algorithms

We have to know that ACO is a very general definition, which is a desirable feature making the algorithm robust but also makes the theoretical analysis about ACO complicated. Actually, we can capture the framework of basic ACO algorithm like follows. At first, the pheromone values are initiated to a constant positive value. Then, a series of constructive heuristic steps constructing and searching candidate solutions follow, in each iteration, each ant probabilistically constructs solutions for the optimization problem according to the present pheromone model, and then for every valid solution, a local search is optionally implemented to find a better solution around this candidate solution. And before the next iteration starts, the pheromone value must be updated according to the current valid solutions. When the iteration stop condition met, output the best-so-far solution. We summarize the process above in a clearer way as follows.

The framework of basic ACO algorithms:

Pheromone initiation. At the beginning of the algorithm, first initiate the pheromone values to a constant positive value.

Constructing solution. A main component of any ACO algorithms is heuristically and constructively construct the candidate solutions. And a solution construction starts from an empty solution space, and at each construction step, the solution space will be extended by adding new feasible component to it.

Local search. This step is optional. A local search step may be used by an ant to improve the solution it has got. This step may help improve the performance of the whole algorithm.

Apply pheromone update rule. The goal of updating the pheromone value is to deposit more pheromone on the path of higher quality.

If the iteration stop condition met, output the best-so-far solution, otherwise go back to *b.* to execute the iteration again,

2.3 Pheromone Update Rule

The pheromone update rule is one of the keys to obtain a satisfactory performance in ACO algorithms, and it helps search for the candidate solutions and influence the convergence speed. M.Dorigo once proposed three different methods of calculating the pheromone increment corresponding to three different algorithm models, that is, ant-cycle system, ant-density system, and ant quantity system [13]. In ant-cycle system, the pheromone update rule between sample *i* and sample *j* is

$$\Delta\tau_{i,j}^{k} = Q / L_{K} \quad (1)$$

where Lk is the path length that ant k traveled during last

iteration, so that the amount of pheromone will depend on the quality of the solution, and the amount of pheromone on the optimal path will increase gradually. While in ant-density system, the pheromone increment between sample i and sample j could be calculated by the following formula.

$$\Delta\tau_{ij} = Q / d_{ij} \tag{2}$$

where dij represent the distance between sample i and sample j, so that the short edge in the graph will be favorable. And in ant-quantity system, the pheromone increment is a constant value Q which is uncorrelated with the path.

Obviously the three models above are different from each other. Some scholars have used the three models to solve the TSP problem, and there are 20 cities and 20 ants in the problem, every experiment performs 3000 iterations. And finally the result shows that ant-cycle is obviously better than the other two [14]. The reason for the difference between the three models is the different feedback information, ant-cycle can get the global information and are more likely to find the shorter path, while the other two models just use local information, and their search process will not benefit from the optimal solution. In this paper, we will propose a new pheromone update rule, and will be discussed later.

3 An Improved Aco Cluster Algorithm

In our algorithm, we try to fuse the ACO algorithm and some classical cluster algorithm, and put some improvement proposal for the ACO algorithm. Our algorithm is somehow like k-means algorithm, the basic process is just like k-means. In k-means algorithm, a sample is assigned to a group depends on the distance from the sample to the cluster centers, while in our algorithm, which cluster a sample belongs to is decided by the pheromone model, and is probabilistically determined. And we also adopt the concept of *elitist ant*, make the ants that find optimal solution have a greater influence on the pheromone model. Moreover we also use a local search strategy, which make the ants more likely to find the optimal solution. And in our algorithm, we introduced a new pheromone update rule and a new method to calculate the heuristic function, which makes the clusters become denser and get far away from each other. Later we will explain the details about our ACO cluster algorithm based on elitist ant and local search.

The idea of our new pheromone update rule and the local search strategy are inspired by the work in [17] and [18].

3.1 New pheromone update rule

In our algorithm, we use elitist ants to update pheromone values, for every elitist ant, if sample x does not belong to cluster C_i, then $\Delta\tau_{xi}^{k} = 0$, otherwise,

$$\Delta\tau_{xi}^{k} = \frac{Q * minCenterDist(k)}{\frac{1}{\#C_i}\sum_{x\in C_i} Dist(x, C_i)} \tag{3}$$

where Q is a constant positive number, and $\#C_i$ represent the number of samples in cluster C_i, *minCenterDist(k)* represents the minimum distance between any two different clusters in the solution of ant k, $Dist(x,C_i)$ represents the distance between sample x and the cluster center of cluster C_i. We can easily inference from formula (3) that this definition of pheromone increment will make the clusters denser and get far away from each other, that the samples in a cluster will be more similar with the ones in the same cluster while distinct from the samples in the different clusters.

A. The pheromone volatile factor

It is common sense that the pheromone will evaporate, so in ACO algorithm, the pheromone update rule includes a factor that represent the pheromone evaporation speed. After each iteration the pheromone value is updated as follows.

$$\tau(t+1) = (1-\rho)\tau_{xi}(t) + \Delta\tau \tag{4}$$

And ρ is the volatile factor. The volatile factor influences the search capability and the convergence of the ACO algorithm, so carefully choose a reasonable volatile factor is very important. If the volatile value is too large and the problem is of large scale, some region unsearched will has a very low pheromone level and may even become zero, which will terribly reduce the search performance of the algorithm. While with a too small volatile factor will make the searched solution may be searched again, and slow down the convergence speed of the algorithm. Generally speaking, 0.3 or so will be reasonable for the volatile factor, in this paper we choose the volatile factor to be 0.2.

3.2 The heuristic function

In our ACO cluster algorithm, the probability of sample x to be assigned to cluster C_i will be [15]

$$P(x, C_i) = \frac{\tau_{x,C_i}^{\alpha}\eta_{x,C_i}^{\beta}}{\sum_{j=0}^{K}(\tau_{x,Cj}^{\alpha}\eta_{x,C_j}^{\beta})} \tag{5}$$

where K is the total number of clusters and

$$\eta_{x,C_i} = \frac{\kappa}{Dist(x, C_i)} \tag{6}$$

In equation (5) and (6), η is the value of a problem-dependent function for sample x and cluster C_i, the higher is

that value of η, the more likely x be assigned to cluster C_i. And κ is a constant to adjust the heuristic function, κ should be properly set according to the problem and initial pheromone level, in our experiment, κ is set to be 2000. We can see from equation (6) that a sample is more likely to be classified into the nearer clusters.

In equation (5), the pheromone heuristic factor α reflects the importance of the information (the pheromone) obtained by the ants in the search process, and the expectation heuristic factor β reflects the importance of the heuristic information for the search process. α and β balance the relative weight of the pheromone versus the heuristic information, they are relevant and contradictory. β reflects the strength of the local information, the bigger β is, the more likely the ant will choose the local shortest path, which will speed up the convergence, but will reduce the randomness of the search process and will be trapped into a local optimum. On the other hand, the heuristic factor α reflects the strength of the randomness factors influence the search process, the bigger α is, the more likely the ant will choose the path that has already be searched and also reduce the randomness of the search process and also will trap the algorithm into a local optimum. The appropriate value of α and β can be determined by simulation experiments. In this paper, we choose α =1.5, and β =4.5.

3.3 The Local Search Strategy

We have mentioned that local search will help the ACO algorithms to get better performance, in our algorithm, we also adopted the local search strategy, and our local search strategy is like follows.

For an ant a, if the solution a gets for the cluster task is $solution_a = [c_1,c_2,c_3\ldots\ldots,c_N]$, and c_i is a number between 1 and K which indicate which cluster sample i belongs to. For every i from 1 to N, generate a random number r_i, and r_i is greater than 0 and less than 1, if r_i is greater than the local search threshold value P, then we need to assign a random number between 1 and K inclusive to c_i, and after all the samples is processed, we recalculate the cluster center. And if the new solution is better than the original one, then replace the original solution with the new solution.

3.4 The Ant System With Elitist Ant And Local Search

We introduced our new pheromone update rule and heuristic function above, in this section the details of the algorithm will be discussed.

Our algorithm complies with the framework of the basic ACO algorithms. At first, assume that the number of clusters is *K,* then for each ant, we need to randomly choose *K* cluster centers. After that we need to initiate the initial pheromone level from every sample to each cluster center, assume that the number of sample is *N,* then the data structure is a *N* by *K* matrix.

After the initiation process, then comes the main loop, for each ant, it need to assign each sample to different clusters, the ant use equation (5) to calculate the probability of each sample belong to the clusters, and every sample will be assigned to one cluster according to the following strategy: For each cluster sorted by the corresponding probability decreasingly, generate a random number *r* between 0 and 1, if *r* is smaller than the probability, then the sample is assigned to the corresponding cluster. If after the loop the sample is not assigned to any cluster, then assign the sample to the cluster with the largest probability.

After every sample is assigned to one of the clusters, the ant need to recalculate the cluster center for each cluster C_i, just in the follow method

$$\mu_i = \frac{\sum_{j=1}^{N} 1\{c_j == i\} x_j}{\sum_{j=1}^{N} 1\{c_j == i\}} \quad (7)$$

where *1{condition c}* return 1 if the condition is true and 0 otherwise. After the above step, we need to rank the ants according to the quality of their solution, the method to measure the quality of the solution is

$$f(X, C) = \sum_{i=1}^{N} (x_i - c_i)(x_i - c_i)^T \quad (8)$$

After ranking the ants, the next step is local search, after which we re-rank the ants according to their new solution, finally we use the elitist ants to update the pheromone level, the method is adding up the pheromone increment and the amount of volatile, just like follows.

$$\tau_{xi}(t+1) = (1-\rho)\tau_{xi}(t) + \sum_{k=1}^{m} \Delta\tau_{xi}^{k} \quad (9)$$

Then all the non-elitist ants need to update their cluster center again. The algorithm repeats the above process until the termination conditions is satisfied.

From what has been discussed above, our algorithm can be described and can be performed by the following steps:

Algorithm: *ant colony clustering algorithm with elitist ant and local search (ACC-EAL)*.

Input: the dataset to be clustered and the number of clusters.

Output: the dataset marked with the cluster label.

Algorithm details:

Initiation. We need to initiate the pheromone level, the number of ants, the number of clusters, and for every ant, needs to initiate *K* cluster centers randomly.

Cluster the data. For every ant and every data sample, assign each data sample to one cluster according to $p(x,c_i)$, we should calculate $p(x,c_i)$ as equation (5).

Recalculate the new cluster centers, using equation (7).

Rank all the ants according to the qualities of their solutions, using equation (8) to measure the solutions.

Local search. For the top 10% elitist ants, perform local search, the strategy has been introduced above.

Pheromone update. Update the pheromone level using equation (3) and equation (9).

For the non-elitist ants, update the cluster centers randomly.

If the termination condition is satisfied, then output the so-far-best solution and exit, otherwise, go to step 3.

In our ACO cluster algorithm with elitist ants and local search strategy, we adopt the elitist ant strategy, making the optimal solutions contribute more to the pheromone increment, and the local search strategy helps the algorithm converge more quickly. And we create a new pheromone update rule, and intuitively we can figure out that the denser the clusters are and the further the clusters are away from each other, the more the pheromone level will be incremented.

4 Experimental Result and Discussion

We implemented our ACO based cluster algorithm on three datasets, the output of the algorithm is a vector that indicate which cluster the corresponding data sample belongs to. And we use the Hungarian algorithm to match the output cluster label with the real label and calculate the accuracy of our algorithm. The result shows that our algorithm performs much better than the classical k-means algorithm.

4.1 Dataset

We use 3 different kinds of datasets, and all the dataset are from UCI machine learning repository. The first dataset we use is *Iris*, which is perhaps the best known database to be found in the pattern recognition literature. The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other 2; the latter are NOT linearly separable from each other. *Iris* contains 3 kind of iris flowers, that is, *Setosa*, *Versicolour* and *Virginica*, and *Iris* consist of 150 data samples, and each sample include 4 features, that is, sepal length, sepal width, petal length and petal width. The second dataset is *Wine*, which is the result 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. *Wine* consists of 179 samples, and each sample includes 13 attributes, such as the content level of alcohol and malic acid, flavonoid and the color intensity, etc. And all the features are continuous. The third dataset is *Vehicle silhouettes*, which is used to classify a given silhouette as one of four types of vehicle, using a set of features extracted from the silhouette. The vehicle may be viewed from one of many different angles. This dataset has 946 samples and each sample includes 18 features extracted from the silhouettes by the HIPS (Hierarchical Image Processing System) extension BINATTS, which extracts a combination of scale independent features utilizing both classical moments based measures such as scaled variance, skewness and kurtosis about the major/minor axes and heuristic measures such as hollows, circularity, rectangularity and compactness. [16]

4.2 Experimental Result

At first we mainly explore the relationship between the ant colony size and the clustering accuracy. Table I shows how the clustering accuracy change with the ant colony size.

TABLE I. Clustering Accuracy Of Different Ant Colony Size

Datase/ Ant colony size	3	5	8	10	15	20	25	30	40	50
Iris	88.90	89.40	89.60	90.03	90.16	90.60	89.63	89.93	90.00	89.56
Wine	69.69	69.75	70.00	70.47	70.55	70.64	70.16	70.19	70.08	70.16
Vehicle silhouettes	45.00	45.11	45.24	45.98	46.09	45.96	45.98	45.98	46.09	45.86

We also draw the curve to observe the relationship between the clustering accuracy and the and colony size more clearly.

We can see from Table I and Figure 1 that when the ant colony size is relatively small, the clustering accuracy will grow as the ant colony size get larger. But when the ant colony size is big enough, the grow of ant colony size helps little with increasing the clustering accuracy. We will explore the reason later.

We also compare our algorithm with the classical k-means algorithm, the following table shows that our algorithm is much better that the k-means.

TABLE II. Acc-Eal Vs K-Means

Dataset/ algorithm	ACC-EAL	k-means
Iris	90.60	81.61
Wine	70.64	64.04
Vehicle silhouettes	46.09	44.79

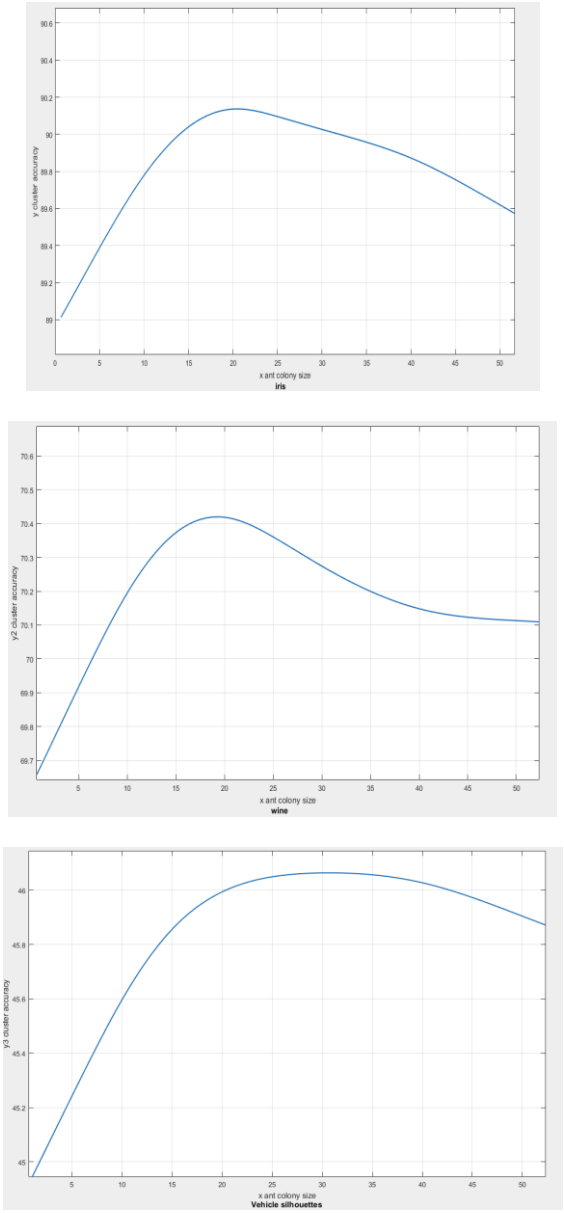

Figure1 The Figure Of Table 1

On the whole, we can see that compared with the k-means algorithm, our algorithm gets a higher clustering accuracy.

ACO algorithm is a kind of stochastic search algorithm, and just like other simulated evolvement algorithm, ACO algorithm finds the optimal solution from the solution space constituted by multiple feasible solutions. In this process, not only the adaptive ability of each individual, but also cooperation between individuals is needed. The ant colony can orderly converge toward the optimal solution in the complex search process, which benefit a lot from the exchange of information and cooperation between the individuals. When the ant colony is relatively large, many ants will get the similar solution, and it doesn't help to improve the cluster accuracy just by increasing the ant colony size.

Conclusion

This paper proposed a cluster algorithm called ACC-EAL based on ACO. The goal of ACC-EAL is to discover clusters in dataset. In ACC-EAL we introduced a new pheromone update rule and heuristic function, and we also adopted the local search and elitist ant strategy. All the improvements aim to make the cluster denser inside and the distance among clusters become further. And ACC-EAL is well suited for parallel and distributed implementation to improve the program efficiency, because the ants are non-interfering during the search process.

ACO proves to be an efficient tool to solve optimization problem, but it is not as mature as genetic algorithm and neural network, so I think further research is very urgent, such as further studying the behavior of real ants, and studying ACO mathematically, and optimizing ACO to speed up the convergence, etc.

Acknowledgment

This paper is the partial achievement of project 2014BAD10B02 supported by the National Support Program.

References

1. M. Dorigo, "Optimization learning and natural algorithms", PhD Thesis, Dipartimento Elettronica, Politecnico di Milano, Italy, 1992.
2. M. Dorigo, L. M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem", IEEE Transactions on Evolutionary Computation, Vol. 1, No. 1, pp.53-66, 1997.
3. M.Dorigo, L. M. Gambardella, "Ant colonies for the traveling salesman problem", BioSystems, Vol. 43, No. 1, 1997, pp. 73-81.
4. L. M. Gambardella, M. Dorigo, "Solving symmetric and asymmetric TSPs by ant colonies", Proceedings of the IEEE International Conference on Evolutionary Computation, IEEE Press, Piscataway, pp.622-627, 1996.
5. Luca M. Gambardella, M.Dorigo. "Ant-Q: A Reinforcement Learning approach to the traveling salesman problem", Proceedings of ML-95, Twelfth Intern. Conf. on Machine Learning, Morgan Kaufmann, 1995, 252–260.
6. Marco Dorigo, Luca Maria Gambardella. "A STUDY OF SOME PROPERTIES OF ANT-Q", Proceedings of PPSN IV–Fourth International Conference on Parallel Problem Solving From Nature, H.–M. Voigt, W. Ebeling, I. Rechenberg and H.–S. Schwefel (Eds.),

Springer-Verlag, Berlin, 656–665.
7. B.Bullnheimer,R.F.Hartl,and C.Strauss.A new rank-based version of the ant system:A computational study[J].Central Europe on Journal for Operations Research and Economics,7(1) :25-38,1999.
8. T.Stützle and H.Hoos.The MAX-MIN ant system and local search for
9. the traveling salesman problem[C].In T.Baeck,Z.Michal ewicz,and X.Yao,editors Proceedings of IEEE-ICEC-EPS' 97, IEEE International
10. Conference on Evolutionary Computation and Evolutio nary Programming Conference,pages 309-314.IEEE Press,1997.
11. T.Stützle and H.Hoos.Improvements on the ant system: Introducing MAX-MIN ant system[C]. In Proceedings of the International Conference on Artificial Neural Networks and Genetic Algorithms, pages 245-249.Springer Verlag,Wien,1997.
12. O.Cordón,I.Fernández de Viana,F.Herrera,and L.Moren o.A New ACO model intergrating evolutionary comput ation concepts:The best-worst ant system[C]. Proc of the 2nd International Workshop on Ant Algorithms, 2000: 22-29.
13. C.Immaculate Mary,Dr. S.V. Kasmir Raja, Dean. Improved Fuzzy C-Means Clusters With Ant Colony Optimization. International Journal of Computer Science & Emerging Technologies (E-ISSN: 2044-6004) 1Volume 1, Issue 4, December 2010
14. Marco Dorigo, Christian Blum. Ant colony optimization theory: A survey. Theoretical Computer Science 344 (2005) 243–278
15. HAN-CHEN HUANG. "THE APPLICATION OF ANT COLONY OPTIMIZATION ALGORITHM IN TOUR ROUTE PLANNING", Journal of Theoretical and Applied Information Technology, 2013
16. JIAN-FENG YANG. "The Ant Colony Algorithm And Its Application Research". CNKI, 2007:23-26
17. Mr.Pankaj K. Bharne, Mr.V. S. Gulhane, Miss. Shweta K. Yewale:" Data Clustering Algorithms B- ased On Swarm Intelligence".IEEE. 2011
18. http://archive.ics.uci.edu/ml
19. Sara Saatchi, Chih Cheng Hung. "Hybridization of the Ant Colony Optimization with the K-Means Algorithm for Clustering". SCIA 2005.
20. [18] P.S. Shelokar, V.K. Jayaraman, B.D. Kulkarni*. "An ant colony approach for clustering". Analytica Chimica Acta 2004.