Journal of Zhejiang University-SCIENCE A (Applied Physics & Engineering)
ISSN 1673-565X (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

A new algorithm based on metaheuristics for data clustering*

Tsutomu SHOHDOHJI[1], Fumihiko YANO[2], Yoshiaki TOYODA[3]

([1]*Department of Computer and Information Engineering, Faculty of Engineering, Nippon Institute of Technology,*
Gakuendai 4-1, Miyashiro-Machi, Saitama 345-8501, Japan)

([2]*Division of Integrated Sciences, J. F. Oberlin University, Tokiwa 3758, Machida, Tokyo 194-0294, Japan*)

([3]*Aoyama Gakuin University, Fuchinobe 5-10-1, Sagamihara, Kanagawa 252-5258, Japan*)

E-mail: shodoji@nit.ac.jp; yano@obirin.ac.jp; toyoda@1965.jukuin.keio.ac.jp

Received Oct. 28, 2010; Revision accepted Oct. 29, 2010; Crosschecked Oct. 29, 2010

Abstract: This paper presents a new algorithm for clustering a large amount of data. We improved the ant colony clustering algorithm that uses an ant's swarm intelligence, and tried to overcome the weakness of the classical cluster analysis methods. In our proposed algorithm, improvements in the efficiency of an agent operation were achieved, and a new function "cluster condensation" was added. Our proposed algorithm is a processing method by which a cluster size is reduced by uniting similar objects and incorporating them into the cluster condensation. Compared with classical cluster analysis methods, the number of steps required to complete the clustering can be suppressed to 1% or less by performing this procedure, and the dispersion of the result can also be reduced. Moreover, our clustering algorithm has the advantage of being possible even in a small-field cluster condensation. In addition, the number of objects that exist in the field decreases because the cluster condenses; therefore, it becomes possible to add an object to a space that has become empty. In other words, first, the majority of data is put on standby. They are then clustered, gradually adding parts of the standby data to the clustering data. The method can be adopted for a large amount of data. Numerical experiments confirmed that our proposed algorithm can theoretically applied to an unrestricted volume of data.

Key words: Metaheuristics, Ant colony clustering, Data clustering, Swarm intelligence
doi:10.1631/jzus.A1001030 **Document code:** A **CLC number:** TP301

1 Introduction

Classification of an enormous volume of data enables us to determine structure and relativity within the data and identify useful information. The cluster analysis technique is one of popular data classification techniques. However, this technique has several drawbacks: it tends to revert to localized best-fit solutions; it requires prior specification of the number of clusters; and, a natural classification of data is difficult. Lumer and Faieta (1994) have proposed a new technique called "ant colony clustering (ACC)", which is modeled on the intelligence of an ant swarm. Swarm intelligence is the result of information gathering by a very large number of individuals who perform simple processing and influence each other. Swarm intelligence is also characterized by the ability to achieve optimum processing results from a large crowd operating as a unified whole. ACC can potentially address the shortcomings of existing cluster analysis techniques through the use of swarm intelligence. The purpose of this study is to modify the ACC algorithm to enhance efficiency and enable more accurate clustering.

2 Ant colony clustering

2.1 Outline of the ACC algorithm

The ACC algorithm is a clustering algorithm that imitates the burial action of ants. The behavior of ants, as they collect their companions' corpses together in

* Project (No. 18510132) supported by the Japan Society for the Promotion of Science Grants-in-Aid for Scientific Research

one location, is applied to the algorithm. Fig. 1 (Bonabeau *et al*., 1999) shows the appearance of the change of the mass of ant corpses according to the passage of time. In the ACC algorithm, an ant corresponds to an agent and an ant corpse corresponds to an object. In this algorithm, the burial behavior of artificial ants is represented as data clustering. The artificial ants (agents) have change place where to carry the corpses (data) by object classification. Moreover, the lattice space with some area is prepared as a place for clustering. The agent moves it at some new locations if neither an object in it nor similar objects in surroundings. On the other hand, the agent acts according to the rule of putting it if there are some similar objects. Also, the agent can move at random in the search space (Shohdohji *et al*., 2007).

The clustering procedures in the ACC algorithm are as follows:

Step 1: Initialization. Arrange objects with agents.

Step 2: State confirmation of site. Confirm state of site surrounding the agents.

Step 3: Selection of action. The action (either pick up or put down) is determined in accordance with the situation.

Step 4: Movement. Move to a site that can be moved at random.

Step 5: End of procedure determination.

All agents must execute Steps 2 through 4. The procedures end when all steps as provided beforehand have been executed.

2.2 Deficiencies in the ACC algorithm

The ACC algorithm forms the cluster by moving objects on the search field. Therefore, a lot of problems that accompany the concept of physical distance on the search field remain unsolved.

1. Number of necessary steps

The ACC algorithm requires an enormous number of steps to complete clustering. If hundreds of datasets are involved, the number of steps required for clustering can reach several millions, depending on the parameter settings. This is thought to be largely attributable to the agent’s locomotion strategy. The agent frequently shuttles back and forth within a constant range since it moves between eight adjacent sites at random with an equal probability. This is inefficient to reach remote locations. Moreover, it is considered as a highly unnatural operation, despite having been observed as a natural behavior in the ant world.

2. Clustering accuracy

Clustering accuracy is governed by the distance between clusters and the size of clusters. The potential influence on clustering accuracy constitutes a problem. It is thought that the initial formation of clustering influences the final clustering configuration. In a place with many similar objects, there is a high probability that the agent will put down the object. Moreover, a cluster of a significant large initial size will often have an influence. It is thought that these factors influence the final clustering configuration. The followings are the possible phenomena:

(1) Objects are mixed together.

(2) Objects are surrounded by a certain cluster and cannot escape.

These phenomena are thought to be attributable to the distance between clusters. Short distances between different types of clusters are difficult to identify. However, above phenomena are hard to resolve and integrate when the same kind of clusters are mutually parted.

(3) Restriction of search domain by area.

When the clustering field is limited, a limitation is imposed on the search domain. It is necessary to the collection of objects in remote locations as well as to

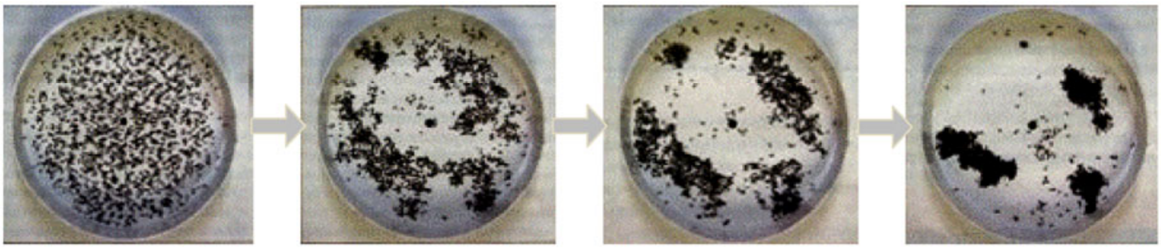


Fig. 1 Clustering process of ant corpses according to the passage of time (Bonabeau *et al*., 1999)

the preparation of a field with sufficient area to allow time for the integration of clusters, in a field that is wide relative to the quantity of data. Furthermore, data in excess of the field area cannot be processed. The problem associated with distance is not solved only by preparing a wide field. Meanwhile, the efficiency of clustering decreases greatly.

3 Our proposed algorithm

Algorithms that describing phenomena such as the division of labor by agents and distribution of global information are regarded as improvements to the current ACC technique. The algorithm proposed in this study contains a number of improvements that are designed to preserve the essential characteristics of swarm intelligence.

3.1 More efficient ant movement

The efficiency improvement of ant movement is a function to improve the efficiency of clustering by making the agent's movement more natural. Based on observations of actual ants, we added the following improvements to agent movements. Table 1 shows the differences in agent movement between the ACC algorithm (Lumer and Faieta, 1994) and our proposed algorithm.

1. Movement in three directions forward

We assumed the probability of going straight to be 50% to reproduce the natural movement of an ant, and the probability for the diagonal left side and right side to be 25%. That is, we designed the algorithm to take the action to which an artificial ant gave priority to straight advancement. These probabilities are based on observations of actual ant's movements. When the range within which the ant can recognize objects is made 3×3 sites forward, and the agent puts the object on the site, the traveling direction will be reversed. This provides visual confirmation of the road that has already been traveled to the rear side. Moreover, advancing requires the object to be placed ahead, where there are already many similar objects. To avoid this situation, it was decided to allow the ant to move only in three directions forward.

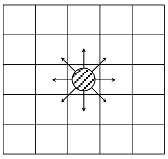
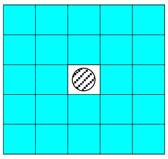
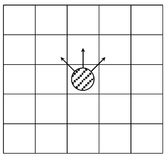
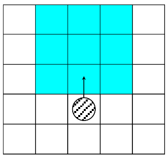
2. Object priority movement

In the natural world, ants do not move completely at random, but rather are continuously searching out the direction of travel with their feelers. Ants are thought to exhibit this searching behavior irrespective of whether there are obstructions in the traveling direction or whether they have been asked for something that does not exist. It was assumed that the object had been confirmed beforehand and existed in view of the ant when moving, so that the ant moved to the object's location by priority. We therefore believe that the efficiency of object searching has improved.

3. Unification of probability equation

In the previous choice judgment method, the decision of where to put down the object was made after the decision of whether to pick up the object, and

Table 1 Comparison of two algorithms in agent movement*

Algorithm	Direction of movement	Ant's view	Probability of picking up an object	Probability of putting down an object
ACC algorithm (Lumer and Faieta, 1994)	 Surrounding 8 sites	 Surrounding 5×5 sites	$P_\mathrm{p}=\left(\frac{k_1}{k_1+f}\right)^2$, $f=\frac{1}{s^2}\sum_{O_j}\left(1-\frac{d(O_i,O_j)}{\alpha}\right)$	$P_\mathrm{d}=\begin{cases}2f, \text{if } f<k_2,\\ 1, \text{ otherwise}\end{cases}$
Our proposed algorithm	 Forward 3 sites	 Forward 3×3 sites	$f=\frac{1}{s^2}\sum_{n=1}^{s^2}P_n$, $P_j=1-\frac{d(O_i,O_j)^2}{\alpha}$	

* In the probability equations, s is the range (s×s) on the view site, d is the degree of similarity (distance) between objects, O_i is an object that an agent is carrying, O_j is a compared object, and k_1, k_2, etc. are parameters

the respective probabilities were determined. In this method, when each agent (artificial ant) picks up the object, non-resemblance is judged in the same space, and when the agent places it, it might be judged that it is reversely resembles. This is very unnatural. In addition, the fixed criteria would be given when many parameters and thresholds were set in k_1, k_2, etc., which could cause the swarm intelligence characteristic to decrease. To overcome these problems, two expressions to calculate the similarity degree of data were integrated into one expression and some improvements were added.

3.2 Condensation of clusters

Condensation of clusters unites objects together under specific conditions. This technique states that the ratio of the area that the cluster on the field occupies will decrease. In the ACC algorithm, the distance of the object on the field has no meaning and only the distinction whether it is the same kind of clusters is important. According to the ACC algorithm, an object group deemed to be the same kind of cluster is compactly condensed. As a result, it becomes possible to move objects in each cluster since two or more sets of objects (i.e., a cluster) are treated as one object. This improves both clustering efficiency and precision. Moreover, the small cluster size virtually eliminates the problem of distance between clusters. Therefore, there is a further advantage of no need to prepare an unnecessarily wide field relative to the number of objects. The process of cluster condensation is described below.

1. Conditions. The cluster condensation proposed in this study is a function that performs additional processing after the decision about where the agent puts the object. First of all, the algorithm compares the object that exists on the agent's view site with the carried object and determines the degree of similarity. When a similar decision goes out to all objects in the view site, it will be condensed into the cluster. If the object that exists in the agent's view is one cluster, it will resemble all of the objects. This approach is based on the principle that the object carried by the agent should belong to the cluster.

2. Condensation processing. Actual processing of cluster condensation is an operation that unites "object of the highest degree of similarity in the view site" and "object that the agent carries". In this case, the attribute value of the cluster (hereafter, condensation object) rendered as an object by cluster condensation can be obtained from Eq. (1):

$$x(O_i+O_j)=\frac{x(O_i)n(O_i)+x(O_j)n(O_j)}{n(O_i)+n(O_j)}, \qquad (1)$$

where x is the attribute value of the object and n is the number of objects united thus far.

3.3 Improved version of cluster condensation

The improved version of the cluster condensation algorithm contains partial modifications to the cluster condensation movement previously described. Objects deemed to have a constant frequency, similar to the technique for determining the degree of similarity in the improved algorithm, are condensed mutually. We believe that this enables a higher speed cluster condensation. The procedure is described below.

1. Comparison of objects in view site. Comparison of the object in the view site (O_j) and the possession object (O_i) is performed as per the previous algorithm. The similarity judgment frequency of objects O_j and O_i is counted as one when objects O_j and O_i are judged similar. The operation described above is executed for all objects that exist on the view site.

2. Condensation processing. If a similar count reaches a constant frequency when objects are compared, objects O_j and O_i are united. Condensation processing at this time is the same as the standard cluster condensation.

3. Relocation processing. An object will be relocated if condensation processing has not been completed when the comparison of objects ends and if something is judged to resemble one of the objects that already exist. The improved algorithm differs greatly from the previous algorithms in that objects are not put in a place but are arranged on the site within the field that has become vacant at random. This is designed to minimize the influence of dispersion on the initial placement in the location where the cluster is formed and the object positioned.

3.4 Addition of objects

The addition of objects involves adding new datasets in turn as additional targets, until the number of objects existing on the field by cluster condensation to decrease. Suppose that 10 000 datasets are to be classified. The datasets are not all classified at the same time, but rather are classified incrementally by adding groups of 500–1000 datasets to the field. This constitutes a major flaw in the current ACC algorithm. The addition of this function has effectively lifted the restriction on the quantity of data that can be processed. This function also reduces the processing load by enabling classification with a narrow field and a small number of agents. To boost classification accuracy, it is useful to classify a large amount of data using a small number of agents. Despite the improved efficiency of classification, however, the level of accuracy is still in question, since the number of objects being transported increases in direct proportion to the number of agents. This can be illustrated by imagining the extreme example of classifying 100 objects by handling 100 agents. As objects can be added at any time, this algorithm can be adjusted to a dynamically changing database with a parallel distributed processing.

4 Numerical experiment

4.1 Numerical experiment

We verified the efficiency of the proposed algorithm on the condition of Table 2. Fig. 2 shows the attribute value data used for the numerical experiment. These data have given arbitrary variance as becoming two clusters. Four algorithms were used in the experiment: ACC algorithm, efficiency improvement of movement (EIM), cluster condensation (CC), and the improved version of cluster condensation (CC2). We compared the number of steps after clustering as recorded by each algorithm.

4.2 Results and discussion of the numerical experiment

Table 3 shows the results of numerical experiment. It is understood that the number of steps has declined dramatically, when make into efficiency of movement. In the ACC algorithm, the same kinds of clusters have become to multiple forms in the early stage. And then, it is difficult to progress to reach the optimal clustering. If the movement of agents is inefficient, it takes time to reach the optimal solution and difficult to combine between clusters that have intervals. It is understood that the improvement of the agent's operation greatly contributes from the above-mentioned to the efficiency of clustering.

Table 2 Parameters used in the numerical experiment

Parameter	Value
Number of experiments	10
Number of ants	30
Number of objects	400
Field size (grid)	100×100
Threshold of ACC algorithm	k_1=0.1, k_2=0.1, α=0.5
Threshold of CC2 algorithm	α=0.1

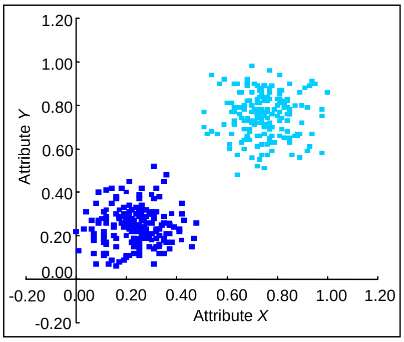


Fig. 2 Scatter diagram of data used for the numerical experiment

Table 3 Number of steps after clustering of each algorithm

Algorithm	Number of steps ($\times 10^3$)		
	Maximum	Minimum	Average
ACC	7420	1860	4410
EIM	1016	430	657
CC	223	128	177
CC2	39.7	35.6	37.1

ACC: ant colony clustering; EIM: efficiency improvement of movement; CC: cluster condensation; CC2: improved version of cluster condensation

As a result, it can be seen that the improved efficiency of movement contributes significantly to the rate of clustering. In addition, the algorithm has been further improved by adding the cluster condensation function. The problem with objects from different types of clusters being left in the cluster is not resolved solely through more efficient agent movement. The improved algorithm classifies data at faster speed than the ACC algorithm. However, it was repeatedly shown that these clusters did not become a single cluster at the final stage in the new algorithm when formed at the position where the same type of clusters parted.

These problems are considered to be almost completely resolved since the cluster size decreases as the cluster condenses. Furthermore, clustering was completed more quickly in the CC2. Since the CC2 provides virtually the same level of accuracy, it can be used to generate condensation decisions more quickly than the ACC algorithm. The dispersion of measurements has also been largely eliminated.

4.3 Summary of numerical experiment

Classification speed and accuracy were found to be significantly better following the improvements in agent movement and cluster condensation. However, several new problems were identified. For instance, the number of objects on the field decreases too greatly when the condensation of the cluster reaches the critical limit. Because the cluster is formed with a few objects in such a state, the distinction of the cluster is difficult. In some cases, there occurs a phenomenon whereby the cluster does not exist on the field, causing the agent to pick up all objects that should form into a single cluster. This can be attributed to an excessive drop in the number of objects. It is thought that this phenomenon can be avoided by ensuring that the number of objects remains above a fixed minimum.

5 Conclusions

Algorithm performance increased markedly through the introduction to the ACC algorithm of more efficient movement, improved cluster condensation, and additional object functions. Despite the performance improvement, we have identified three key issues that still need to be addressed:

1. The reliance on monitoring to determine the timing of clustering completion.

2. The necessity for similar level space beforehand.

3. The lack of definition of the range considered to be a cluster.

A random value in each agent is also provided to make the best use of swarm intelligence and methods (such as selectively deciding the action from a current experience of the agent). Incorporating concepts such as the genetic algorithm and the immune algorithm would undoubtedly produce an even more efficient algorithm. Similarly, the approach of classifying data while sharing information involves parallel distributed processing using two or more clients; i.e., ACC swarm clustering, which could potentially double the benefits of swarm intelligence via distribution of processing load. In the near future, we hope to resolve the remaining problems and further refine the algorithm into a general-purpose algorithm suitable for application to real-life problems.

Acknowledgments

The authors thank Mr. Natsuki SAMURA (Panasonic System Networks Software Co., Ltd., Japan) for his helpful comments and suggestions.

References

Bonabeau, E., Dorigo, M., Theraulaz, G., 1999. Swarm Intelligence: From Natural to Artificial Systems. Oxford University Press, USA.

Lumer, E.D., Faieta, B., 1994. Diversity and Adaptation in Populations of Clustering Ants. Proceedings of the 3rd International Conference on the Simulation of Adaptive Behavior, p.501-508.

Shohdohji, T., Samura, N., Yano, F., Toyoda, Y., 2007. An Improvement of Ant Colony Clustering Algorithm Based on Ant Behavior. Proceedings of the 37th International Conference on Computers and Industrial Engineering, p.13-21.