Swarm Intelligence Clustering Algorithm based on Attractor

Qingyong Li, Zhiping Shi, Zhongzhi Shi

Institute of Computing Technology-Chinese Academy of Sciences – Beijing, China
E-mail: {liqy, shizp, shizz}@ics.ict.ac.cn

Abstract

Ant colonies behavior and their self-organizing capabilities have been popularly studied, and various swarm intelligence models and clustering algorithms also have been proposed. Unfortunately, the cluster number is often too high and convergence is also slow. We put forward a novel structure-attractor, which actively attracts and guides the ant's behavior, and implement an efficient strategy to adaptively control the clustering behavior. Our experiments show that swarm intelligence clustering algorithm based on attractor (**SICBA** for short) greatly improves the convergence speed and clustering quality compared with LF and also has many notable virtue such as flexibility, decentralization.

1 . Introduction

Swarm Intelligence emerged out of social insect collective behavior shows many interesting properties such as flexibility, robustness, decentralization and self-organization. Implementations of optimization and control algorithms based on swarm intelligence such as Ant Colony Optimization and Ant Colony Routing have been well known [1,2,3]. Clustering models and algorithms based on swarm intelligence, inspired by co-operative brood sorting by ants or other behavior, are also put forward, though they are still in a preliminary, proof-of -concept stage [4,5].

The swarm intelligence clustering models and algorithms have advantages in many aspects, such as no need of priori information, self-organization. However, the number of result cluster is often too high and the convergence is slow because of the ant's inefficient behaviors: randomly picking up items and dropping down items. Are there any methods to make ant to perform efficiently?

After some careful research, we believe that the algorithms inefficient performance is mainly because of the ant's inefficient moving. Especially in the first stage, the items is distributed sparsely, the probability for an ant to move to a place to pick up items or to drop down items is often small, so most of the ant's moving is inefficient. At the same time, the number of items loaded by an ant is an important factor for cluster number and accuracy.

In this paper, we put forward a new algorithm, named swarm intelligence clustering algorithm based on attractor (SICBA for short). SICBA attacks the problem in following two aspects:

Firstly, a novel, efficient structure, called attractor, is constructed. Simply, it is an item set converging the homogeneous items. Moreover, it contains not only local environment information, such as inner distance, but also global information, such as outer distance between attractors in the system. So it can actively attracts the ant to pick up dissimilar items or drop down similar items.

Secondly, an ant can pick up the farthest item or all items from an attractor controlled by a simple rule. Furthermore, a parameter is applied to form a strategy: first stage an ant is prior to pick up all items to coarsely but fast cluster; last stage the ant is mostly to pick up the farthest item to precisely partition. So it can distinctly improve the convergence speed and accuracy.

The paper is organized as follow: the following section introduces the related work; the next section describes the details of ***SICBA*** ; the experiments are showed in the section 4; at last we make our conclusion.

2. Related Work

Deneubourg et al [4] proposed an agent-based model to explain how ants manage to cluster the corpses of their dead nestmates. Artificial ants (or agents) are moving randomly on a square grid of cells on which some items are scattered. Each cell can only contain a single item. Whenever an unloaded ant encounters an item, this item is picked up with a probability which depends on an estimation of the density of items of the same type in the neighborhood. When a loaded ant encounters a free cell on the grid, the probability that this item is dropped also depends on an estimation of the local density of items of the same type.

Lumer and Faieta[5] (LF for short)extended the model of Deneubourg et al., using a dissimilarity-based evaluation of the local density, in order to make it suitable for data clustering. Unfortunately, the resulting number of clusters is often too high and convergence is slow. Therefore, a number of modifications were proposed, by Lumer and Faieta themselves as well as by others [6,7].

3. Swarm Intelligence Clustering Algorithm Based on Attractor

3.1 Basic Concept

Definition 1. Attractor is a data set which has similar items as a whole. The attractor can attract the ants to pick up the furthest item or all items from it according to a pick-attractive rate, and drop down an item or an item collect in it according to a drop-attractive rate. Furthermore, the attractor has it own status: active and inactive, if the attractor has not any item the attractor is considered as inactive, and it can't attract ant to pick up items or drop down items any longer; otherwise, it is considered as active and can attract any ant.

Definition 2. Inner distance is the average distance between the items and the attractor's centroid for an attractor.

Definition 3. Outer distance is the average distance with the other attractors in the system for an attractor.

Attractor's pick-attractive rate and drop-attractive rate are two very important features to control an ant to load items or unload items, differentiating the ant's randomly choosing method in [4, 5]. Pick-attractive rate (PAR) for an attractor is a numeric feature to indicate the attractor's affinity attracting the ant to pick up items, in other word, it indicate the probability for an ant to pick up a item or all the items from the attractor. If the PAR is greater, the probability for an ant to pick up items is also greater. The PAR value depends on the attractor's item number marked as C, if the number C is less than a threshold θ, the attractor is a small attractor whose PAR is determined by its inner distance, outer distance and item number, and if the inner distance, the outer distance and item number are less the PAR is greater; otherwise, if C is greater than θ the attractor is considered as big one and its PAR is effected by the maximal distance in the attractor and outer distance. The PAR is given by

$$PAR = \begin{cases} f_1(D_i)\times\alpha_1 + f_1(Do)\times\alpha_2 + T(C,\theta)\times\alpha_3 + \delta & if\ C<\theta \\ Dm\times\beta_1 + f_1(Do)\times\beta_2 & if\ C\geq\theta \end{cases}$$

(1)

where $D_i \in [0, 1]$, $D_o \in [0, 1]$ represent inner distance and outer distance, $f_1(x) = 1 - x$, and $T(C,\theta) = \dfrac{\theta}{C+\theta}$, $0 \le \alpha_1, \alpha_2, \alpha_3 \le 1\ and\ \alpha_1 + \alpha_2 + \alpha_3 = 1$, δ is a constant parameter to control the choosing strategy: if $\delta > 1$, the attractor which has few items will have the absolute priority to be chosen comparing with the attractor which has many items. Dm is the maximal distance in the attractor, and $0 \le \beta_1, \beta_2 \le 1\ and\ \beta_1 + \beta_2 = 1$.

Drop-attractive rate (DAR) for an attractor is a contrast feature to indicate the attractor's affinity attracting the ant to drop down items. In the same way, The DAR value firstly depends on C, if the number C is less than the threshold θ the attractor is small one and the DAR is always small; if $C \ge \theta$ the attractor is a big one and its DAR is determined by the distance between the items loaded by a ant and the attractor's items, and the outer distance also has slight influence. The DAR is given by

$$DAR = \begin{cases} C/T & if\quad C < \theta \\ (1-D)\times\beta_1 + Do\times\beta_2 & if\quad C \ge \theta \end{cases} \quad (2)$$

where T is the total number of items in the system, D is the distance between the loaded items by an ant and the attractor's items, $0 \le \beta_1, \beta_2 \le 1\ and\ \beta_1 + \beta_2 = 1$ β_1 is always greater than β_2.

3.2 Probability Conversion Function

Probability conversion function is a function which converts the attractor's quality and quantity characters into a pick up probability for an unloaded ant or drop down probability for an loaded ant. There are usually two related functions. One is for picking-up probability; another is for dropping probability.

Picking up stimulus An unloaded ant can perform the task: picking up an item or picking up all items. When the item number of the target attractor C is less than θ the ant picks up all the items. Obviously, the unloaded ant should pick up an entire attractor if the attractor is small, homogeneous and not isolated. The attractor is more homogeneous and coupling its inner distance and outer distance is less. So the $S_{pick-all}$ is given by

$$S_{pick-all} = f_1(D_i)\times\alpha_1 + f_1(Do)\times\alpha_2 \quad (4)$$

where $D_i \in [0, 1]$, $D_o \in [0, 1]$ represent inner distance and outer distance $0 \le \alpha_1, \alpha_2 \le 1\ and\ \alpha_1 + \alpha_2 = 1$.

While the item number C is greater than θ the attractor is considered as a big one, so only the most dissimilar item should be picked up. The stimulus $S_{pick-one}$ for picking up one item is mainly influenced by the furthest distance and slightly by the outer distance. $S_{pick-one}$ is given by

$$S_{pick-one} = Dm\times\beta_1 + f_1(Do)\times\beta_2 \quad (5)$$

where Dm is the maximal distance in the attractor, and $0 \le \beta_1, \beta_2 \le 1\ and\ \beta_1 + \beta_2 = 1$.

Drop down stimulus The stimulus for a loaded ant to drop its items L in an attractor is based on the local distance D between the loaded items and the attractor's items. If the ant loads an item collect the center of the loaded items is used to measure the similarity. Furthermore, we also consider the global factor – outer distance because the outer distance indicates the dependence among the attractors in the system. Stimulus for dropping down S_{drop} is given by

$S_{drop} = (1-D)\times\beta_1 + Do\times\beta_2$ (6)

where $0 \le \beta_1, \beta_2 \le 1 \, and \, \beta_1 + \beta_2 = 1$ β_1 is always greater than β_2.

3.3 Algorithm Description

Based on the above description we have the following algorithm.

Algorithm (**SICBA**: Swarm Intelligence Clustering Algorithm Based on Attractor)

1)Initialize θ, δ, MAXCYCLENUMBER ATTRACTORNUMBER and other parameters;

2. Run k-means algorithm assigned with ATTRACTORNUMBER clusters to form the original ATTRACTORNUMBER attractors;
3. Give ants initial attractors, initial states of ants are unloaded;
4. **WHILE**(cycle_counter< MAXCYCLENUMBER and NotConvergent) {
5. **FOR** (number of ants) {
6. **IF** the ant is unloaded, **THEN**

{calculate P_p;

Compare P_p with a random probability P_r, **IF** $P_p < P_r$ **THEN**

{ Not pick up anything;

According to PAR values, the unloaded ant moves to the greatest attractor;

}**ELSE**

{ Pick up the most farther items or all items;

Update the attractor's state according to the picking task;

According to DAR values, the loaded ant moves to the greatest attractor;

}}**ELSE** the ant is loaded

{ Calculate P_d;

Compare P_d with a random probability P_r, **IF** $P_d < P_r$ **THEN**

{ Not drop down;

According to DAR values, the loaded ant moves to the greatest attractor;

}**ELSE**

{ Drop down the load;

Update the attractor's state according to the dropping task;

According to PAR values, the loaded ant moves to the greatest attractor;

}}}}

4. Experimental Results

In this section, we'll demonstrate the experimental results about the performance of **SICBA** and the influence of the parameter δ for clustering performance.

The dataset IRIS chosen from UCI machine learning repository (http://www.ics.uci.edu/~mlearn/MLRepository.html) are used in this paper. IRIS database has 150 records with 4 attributes. All experiments are performed on a 800-MHz Pentium machine which 512 megabytes main memory, running on Windows 2000 professional. Programs are written in Windows/Vision C++ 6.0.

4.1 Comparison with FL

In this experiment we compare SICBA with LF about the clustering performance. Because here accuracy is little important we measure the clustering result number when run N cycles. The result is shown in the figure 1.

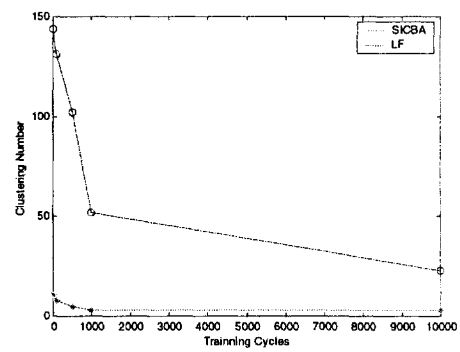

Fig. 1. Clustering Number with Training Cycle

Figure 1 shows that convergence speed of SICBA is faster than LF, especially in the first stage, SICBA quickly partitions dataset into a few clusters, however, the LF is very slow to be convergent. The last cluster number of SICBA is also better than LF. Figure 1 shows that the last number of SICBA is 4 a little greater than the real number 3, but LF is much greater than 3. Because SICBA initialize ATTRACTORNUMBER attractors with Kmeans, moreover, it picks up all items in the small attractor at first stage and picks up the farthest items in the big attractor at last stage, so SICBA can converge greatly faster than LF, furthermore, it also can get much better clustering result.

4.2 Influence of Parameter δ for the Clustering Performance

The parameter δ in formula (1) is a most important factor to affect the clustering performance, because it directly determines the picking up strategy. In this experiment the clustering performance is measured by clustering result number through 1000 times training. The influence shows in figure 2.

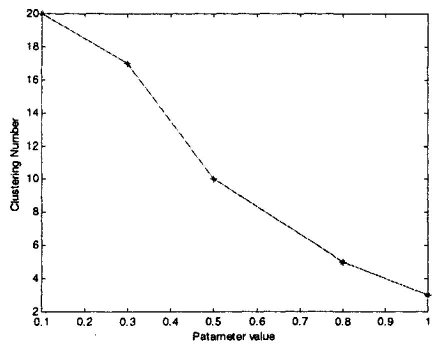

Fig.2. Influence of Parameter δ

Figure 2 shows that if value of δ is greater the clustering performance is better. From formula 1, we can easily get that δ directly effect the PAR value, that is to say, δ affects the probability for the small attractors to be chosen to pick up items. If δ is greater, the small attractor is more prior to be chosen than the big one, so SICBA is quicker to converge. As to last stage, most of the active attractors are all big one, and δ doesn't effect any longer. The probability of picking up item is mainly determine by the attractor's furthest item, so SICBA can precisely adjust the cluster and get better accuracy performance. Because the PAR values in the case of $C>\theta$ is always smaller than 1, obviously, δ greater than 1 is none meaning. In conclusion, when δ equals 1 SICBA can get the best performance and implement the efficient strategy: *small attractors prior.*

5. Conclusion

This paper put forward a novel structure in swarm intelligence clustering algorithm, named attractor, which contains not only the local information but also the global information, so it can actively attract the ant to pick up items or drop down items and avoid the ant aimless moving. Furthermore, we also implement an efficient strategy: *small attractors prior,* based on the attractor. The strategy makes SICBA pick up all items to coarsely but fast cluster in the first stage and pick up the furthest item to precisely fine-tune. The experiments prove that it can greatly improve the algorithm convergence speed and clustering quality compared with LF algorithm. Although SICBA has no advantages over classic kmeans algorithms on the aspect of space and time complexity, as a self-organization clustering algorithm, it has great advantages in robustness, visualization, flexibility and decentralization.

References

[1] Becker R., Holland O.E. and Deneubourg J.L. 'From local actions to global tasks: Stigmergy and collective robotics', in Brooks R. and Maes P. Artificial Life IV, MIT Press, 1994;

[2] E.Bonabeau, M.Dorigo,G.Theraulaz, Inspiration for optimization from social insect behaviour, Nature,vol 406,6 July 2000.

[3]Gianni Di Caro and Marco Dorigo, AntNet: Distributed Stigmergetic Control for Communications Networks, Journal of Artificial Intelligence Research 9(1998) 317-355;

[4] Deneubourg..J.L., Goss S.,Frank,N., Sendova-hanks, A.,Detrain C.,Chrerien L., The dynamics of collective sorting: robot-like ants and ant-like robots, in: Meyer J., Wilson S.W. (Eds.), Proceedings of the First International Conference on Simulation of Adaptive Behavior: From Animals to Animats, MIT Press/Bradford Books, Cambridge, MA, 1991, pp.356-363;

[5] E.Lumer, B.Faieta. Diversity and adaptation in populations of clustering ants . in J.-A.Meyer, S.W. Wilson(Eds.), Proceedings of the Third International Conference on Simulation of Adaptive Behavior: From Animals to Animats, Vol.3, MIT Press/ Bradford Books, Cambridge, MA, 1994, pp 501-508;

[6] J. Handl, B. Meyer. Improved Ant-Based Clustering and Sorting in a Document Retrieval Interface. Proc. of the 7th Int. Conf. on Parallel Problem Solving from Nature. 913-923 (2002).

[7] V. Ramos, F. Muge, P. Pina. Self-Organized Data and Image Retrieval as a Consequence of Inter-Dynamic Synergistic Relationships in Artificial Ant Colonies. Soft Computing Systems: Design, Management and Applications. 87, 500-509 (2002).