International Journal of Security and Its Application
Vol.11, No.3 (2017), pp.25-32
http://dx.doi.org/10.14257/ijsia.2017.11.2.03

An Optimized Clustering Algorithm based on Integration of Genetic Algorithm and Ant Colony Algorithm

Shucheng Xiao[1,a], Shanjing Chen[1,b], Shan Qiu[2], Zhendong Yang[1], Juan Xu[3], Jinlan Qiao[1,] Saixuan Yu[4]

[1] *Chongqing Logistical Engineering University, China;*
[2] *Chongqing University of Posts and Telecommunications, China.*
[3]*Fuel Supervision Division of Beijing Military Region,China*
[4] *Sichuan Winshare Vocational College, China.*
[a]*xiaosc@cqu.edu.cn,* [b] *chengshanjing_11@163.com*

Abstract

Genetic algorithm (GA) has good fast global searching ability and ant colony algorithm has good distributed parallelism and positive feedback ability. In this paper, an optimized clustering algorithm was proposed by integrating these two algorithms and making full use of their advantages and characteristics. The proposed algorithm could make clustering analysis more effectively through the automatic evolution rate matching optimization algorithm of the GA. According to simulation experiment, this algorithm was far superior to common clustering algorithms in term of optimization capability and time performance.

Keywords*: Clustering analysis; Genetic algorithm; ant colony algorithm; integrating algorithms*

1. Introduction

Clustering analysis is to divide data into different groups according to properties and the principle of maximum inter-group difference and minimum intra-group difference[1]. Typical clustering methods include partition methods (k-means[2] and PAM), hierarchical clustering methods and density-based clustering method[3-4].

Genetic algorithm[5] (GA) is a bionic optimized algorithm proposed by Professor John HoUmd from the America University of Michigan in 1975. It simulates the biologic evolution process based on the Darwinian's biological evolutionism of "survival of the fittest" and Mendelian's genetic heritable variation that "biological genetic evolution is mainly on chromosome and offspring is the ordered arrangement of parent genes on chromosome".

Ant colony algorithm is the latest bionic optimization algorithm that simulates intelligent behaviors of ant colony [6-9]. It was proposed by Dorigo M, an Italian scholar, in 1991. It is characteristic of strong robustness, excellent distributed computation mechanism and high compatibility with other methods[10].

Clustering method based on can be divided into two types according to principles[9]. One is data clustering based on the formation mechanism of ant colony and the other is clustering analysis based on pheromone using the foraging behavior of ant colony[11]. The first one was applied for clustering analysis in this paper. Deneubourg et al.[12] were the first ones to explore clustering method based on ant colony algorithm. They established a basic model and clustered data regarding random movement, pick or drop of ant colony according to the similarity between data object and surrounding object. Lumer E and Faieta B expanded this model to data analysis category and proposed the LF algorithm[13]. Ramos et al. [14-15]improved the LF algorithm from different perspectives and achieved some results.

Abbattista F et al.[16] put forward the first improvement strategy to integrate GA and ant colony algorithm, which was proved good in the simulation experiment using Oliver 30 TSP

ISSN: 1738-9976 IJSIA

and Eilon 50 TSP. Subsequently, people began to integrate ant colony algorithm and GA to solve multiple optimization problems in the discrete domain and continuous domain, which achieved good application effect. Dong et al.[17] used the combination of GA and ant colony algorithm to solve combinatorial explosion and NP problems, achieving satisfying result regarding optimization performance and time performance. The proposed algorithm targeted at clustering analysis. It integrates GA and ant colony algorithm, generates the initial clustering center of data object by the fast random global searching ability of GA and perfects the clustering structure through parallelism, positive feedback and high solving efficiency of the ant colony algorithm.

In this paper, Section 1 introduces the integration of GA and ant colony algorithm, including basic principle, analysis of GA and ant colony algorithm as well as the integral algorithm. Section 2 makes a comparative analysis on experimental results of the proposed algorithm, standard ant colony algorithm and fuzzy ant colony algorithm[18]. Section 3 is conclusion.

2. Integration of GA and Ant Colony Algorithm (GA2C2A)

2.1 Basic Principle and Design Idea of the GA2C2A

Although GA has fast large-scaled global searching ability, it often causes massive redundancy iterations when reaching to a certain extent, which results in poor utilization of feedback information in the system and thereby decreases solving efficiency. Due to random distribution of early data objects of the ant colony algorithm, ant behaviors like “pick” and “drop” are random, which requires long time to form effective clustering. GA has high convergence speed during the early searching period (t_0~t_a), but the search efficiency decreases significantly after t_a. On the contrary, the ant colony algorithm has low searching speed in t_0~t_a due to randomness of data and data movement, but its searching efficiency increases dramatically after a certain time. The basic idea of GA2C2A is to generate the initial clustering center of data object based on the fast global searching ability of GA in early period and perfect the clustering structure by taking advantages of positive feedbacks of the ant colony algorithm in the late period.

2.2 GA in the GA2C2A

(1) Problem description

Suppose the objective function is:

$$\min\ J = \sum_{r=1}^{p}\sum_{i=1}^{m_r}\left\|X_i^{(r)} - C_r\right\|^2 \tag{1}$$

Where the clustering center is $$C_r = \frac{1}{m_r}\sum_{i=1}^{m_r} X_i^{(r)}\ (i = 1,2,\cdots,m_r; r = 1,2,\cdots,P) \tag{2}$$

$$\sum_{r=1}^{p} m_r = N \tag{3}$$

Where m_r is number of samples of the r type; $X_i^{(r)}$ means that sample X_i belongs to the r type; N is sample size; P is number of clustering center (2≤P≤N-1).

(2) Chromosome structure

Suppose $\overline{Y} = (S_1 S_2 \cdots S_L)$ represents the chromosome structure, $\overline{Y}$ is $1\times L$ dimensional row vector; $S_l (1 \le l \le L)$ is the gene at the l^{th} position, and N is sample size. Then, chromosome requirements are as follows:

$$L = N; \tag{4}$$

$$S_l \in \{1,\cdots,P\}\ l = 1,\cdots,L; \tag{5}$$

$$\sum_{r=1}^{K}\delta(r)=P \text{ , where } \delta(r) = \begin{cases}1 & , r\in\{S_1\cdots S_L\}\\ 0 & , r\notin\{S_1\cdots S_L\}\end{cases} \qquad (6)$$

(3) Fitness function and GA operations

Fitness function (F) is defined $F=M/J$, where M is a constant and J is defined in (1). Therefore, individuals with smaller J have higher fitness

GA operations include selection, crossover and variation.

1) Selection rule: chromosomes are selected using the roulette strategy according to the fitness function. The best chromosome in every generation will be selected priori in the next generation and the same chromosome of every generation couldn't be selected for more than 2.

2) Crossover rule: crossover is controlled by loci-crossover which may make crossed chromosomes disagree with abnormal situations in (6). Therefore, the maximum allowed attempts (W) is defined. If it exceeds W, the matched chromosome will be abandoned and make another crossover.

3) Variation rule: variation uses inversion mutation method[19]. For example $\overline{Y}_{before}=a_1a_2a_3a_4a_5a_6\cdots a_L$, suppose there are breakages at intervals a_2a_3 and a_5a_6 and the breakage fragments are inserted at the directional order. Then, the inversed chromosome will become: $\overline{Y}_{before}=a_1a_2a_5a_4a_3a_6\cdots a_L$.

(4) GA description in the GA2C2A algorithm

GA in GA2C2A
1. Initialize population and set initial number of evolution generation, crossover rate and variation rate. 2. Selection. Calculate according to above formulas and select according to proportions of chromosome fitness. 3. Crossover. Any two individuals are chosen for crossover. 4. Variation. The variation position is produced randomly. 5. Get optimized chromosomes. Calculate fitness function values of every chromosome in each iteration, thus generating new offspring chromosome. End when meeting the ending condition and the clustering results will be acquired. Otherwise, turn to step2 repeat following steps.

2.3 Ant Colony Algorithm in GA2C2

(1) Introduction to standard ant colony algorithm (SACA)

SACA was proposed by Lumer E and Faieta B. The basic idea of SACA is that suppose ant moves randomly in a 2D plane with random distribution of data objects. In the beginning, ant chooses one data object randomly. The selection probability of this data object is calculated by its similarity in local region, which determines whether the ant "pick" or "drop" this data object. After limited iterations, similar data cluster together on the 2D plane, finally getting the clustering structure and cluster numbers. The probability for the ant to "pick up" the data object is determined by its similarity with object in current neighbor domain. Lower similarity means higher probability of "pick", while higher similarity means lower "pick". The probability for ant to "drop" the data object shows the opposite.

1) Similarity function

$$f(i)=\begin{cases}\frac{1}{s^2}\sum_j(1-d(i,j)/\alpha) & \text{if } f(i)>0,\\ 0 & \text{otherwise.}\end{cases} \qquad (7)$$

s^2 is the area of a local square surrounding the object i and α is the similarity

parameter, which is a constant.

2) Probability functions of “pick” and “drop”

Figure1 Performance comparison between our method and SACA and FACA algorithms in Iris dataset.

$$p_{pick}(i)=\left(\frac{k_p}{k_p+f(i)}\right)^2 \tag{8}$$

$$p_{drop}(i)=\begin{cases}2f(i) & if\ f(i)<k_d;\\ 1 & otherwise.\end{cases} \tag{9}$$

where k_p and k_d are threshold constants, valuing 0.1 and 0.15. $f(i)\in[0,1]$. For P_{pick}, if $f(i)<<k_p$, $P_{pick}\approx 1$; if $f(i)>>k_p$, $P_{pick}\approx 0$. For P_{drop}, if $f(i)<k_d$, $P_{drop}\approx 0$; if $f(i)\geq k_d$, the ant drops the data object.

(2) Description of ant colony algorithm in GA2C2A

(3) Integration of GA and ant colony algorithm

Ant colony algorithm in GA2C2A
1. Distribute ants in a 2D plane randomly. Initial clusters produced by GA are scattered in the plane according to cluster properties. Each cell has one or more objects. Initialize the maximum number of cycles (n) and nt_number; 2. for i=1,2,...,n; 3. for j=1,2,…,ant_number; Move ants If ants carry no object then if there’s an object in 8 neighbors of the ant, the ant picks up the data object at the probability of p_{pick} else the ant drops the data object at the probability of p_{drop} by calculating its similarity with 8 neighbors. 4. Mark the clustering model.

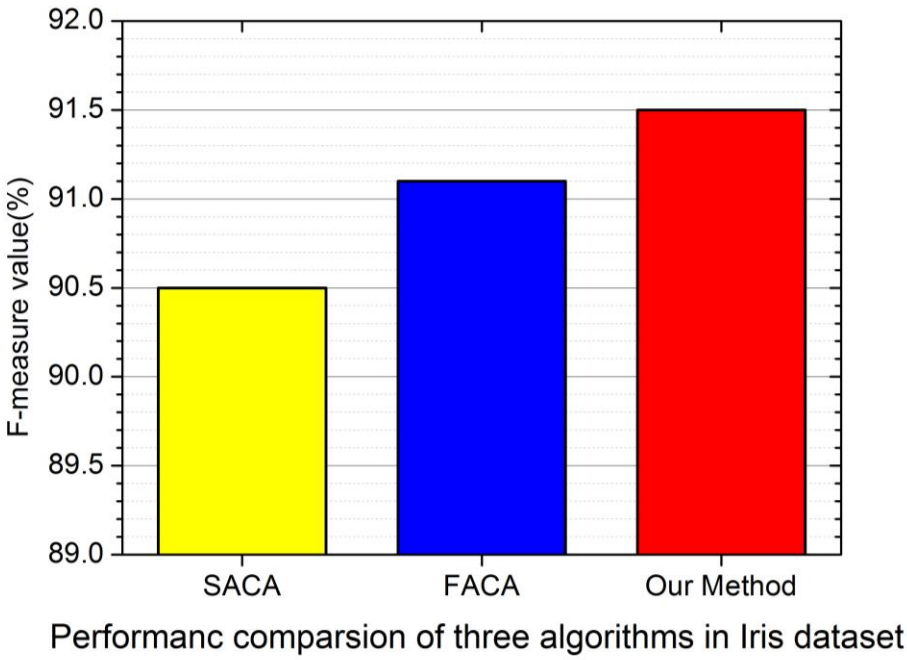

Figure 1. Performance Comparison between Our Method and SACA and FACA Algorithms in Iris Dataset

In Reference[17], the solving strategy of general optimization problems in integration of GA and ant colony algorithm was proposed. This strategy sets fixed number of iterations in GA, which will cause over early or late ending of the GA process and thereby couldn't ensure integration of these two algorithms at the best time. The integration strategy proposed based on Reference[20] can protect the best integration time. Main steps are:

1) Set the initial clustering center P according to practical situations. P determines the complexity of the mapping process of genetic clustering from chromosome of the solution to physical value of the solution.

2) Set the minimum and maximum numbers of genetic iterations.

3) Make statistics on evolution rate of offspring during the iteration process of GA and set the minimum evolution rate of offspring.

4) Within the preset number range of iterations, if evolution rate of offspring is smaller than, the optimization speed of the GA is relatively lower. It can end the iteration process of GA and enter into the ant colony algorithm.

3. Simulation experimental results

SACA, Fuzzy Ant Colony Algorithm(FACA)[21] and GA2C2A were tested by using 3 datasets in Table 1 (description of testing datasets). These datasets have classification tables that can be used in final performance evaluation.

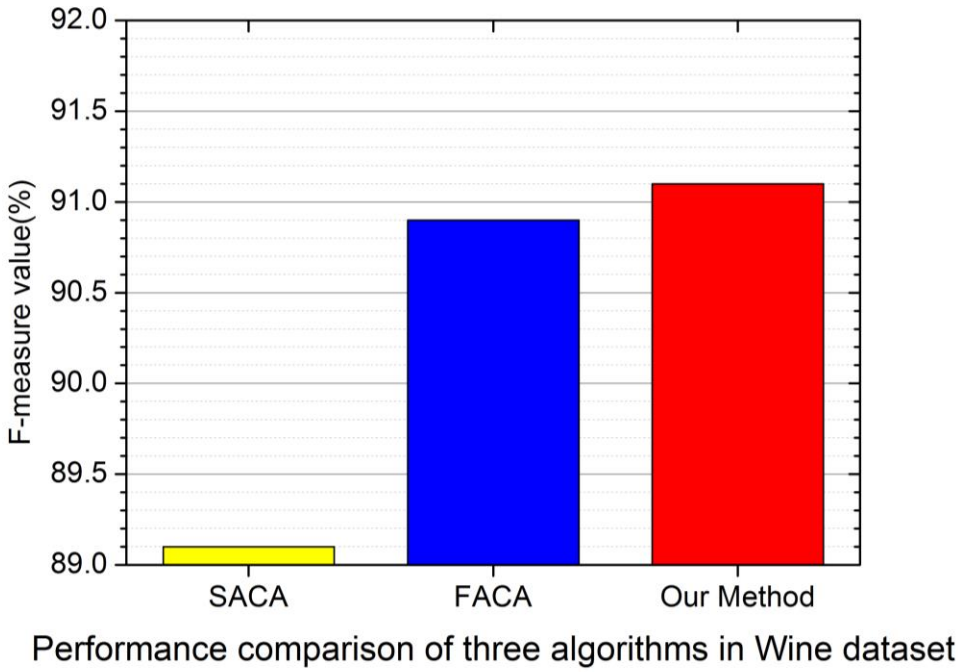

Figure 2. Performance Comparison between Our Method and SACA and FACA Algorithms in Wine Dataset

Table 1. The Description of Test Dataset

Data set	Entity quantity	Property quantity	Classify quantity
Iris	150	4	3
Wine	178	13	3
Glass	214	9	6

In this paper, F-measure, a common external evaluation method, was used the method in reference[22]. It combines the precision ratio and recall ratio in information retrieval. Precision ratio and recall ratio of the cluster j and related category i are defined as:

$$P = precision(i, j) = \frac{N_{ij}}{N_i} \tag{10}$$

$$R = recall(i, j) = \frac{N_{ij}}{N_j} \tag{11}$$

where N_{ij} is number of category i in the cluster j; N_j is number of objects in the cluster j; N_i is number of objects in the category i.

F-measure of category i is defined as:

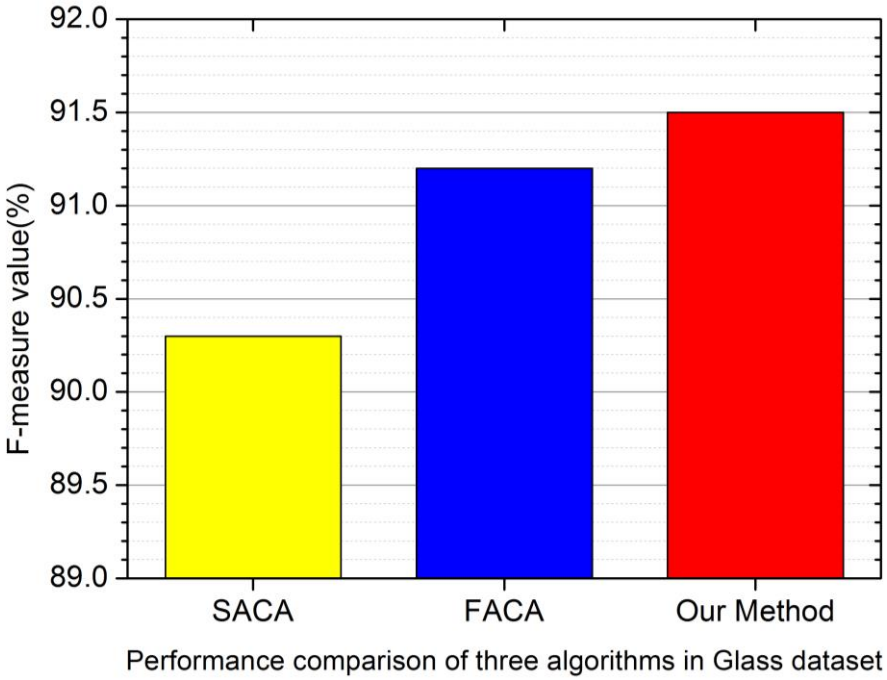

Figure 3. Performance Comparison between our Method and SACA and FACA Algorithms in Glass dataset

$$F(i) = \frac{2PR}{P + R} \tag{12}$$

The overall F-measure of clustering result F is calculated according to weighted average of F-measure of every category i as:

$$F = \sum_i (|i| \times F(i)) \tag{13}$$

We first make experiments based on Iris dataset. The average overall F-measure values of 50 tests are shown in Fig 1. As illustrated in Fig. 1, the F-measure value of our method is much higher than that of both SACA and FACA algorithms. Specifically, the F-measure value of our method is up to 91.5%, and increases by 1% and 0.4% in comparison to the SACA algorithm and FACA algorithm, respectively.

Secondly, we make experiments based on Wine dataset. The average overall F-measure values of 50 tests are shown in Fig 2. As illustrated in Fig. 2, the F-measure value of our method is higher than that of both SACA and FACA algorithms. Specifically, the F-measure value of our method is up to 91.1%, and improves by 2% and 0.2% in comparison to the SACA algorithm and FACA algorithm, respectively.

Finally, we make experiments based on Glass dataset. The average overall F-measure

values of 50 tests are shown in Fig 3. As illustrated in Fig. 3, the F-measure value of our method is higher than that of both SACA and FACA algorithms. Specifically, the F-measure value of our method is up to 91.5%, and improves by 1.2% and 0.3% in comparison to the SACA algorithm and FACA algorithm, respectively.

In conclusion, the F-measure values of our method are beyond 91% in terms of these three datasets. Moreover, the performance our method is much better than that of SACA algorithm and FACA algorithm in all the three datasets.

3. Conclusion and Future Work

Organic integration of GA and ant colony algorithm is to make full use of the fast global searching ability of GA and positive feedback convergence of the ant colony algorithm. In this paper, the initial clustering center in GA is determined according to statistics on evolution rate of offspring. Based on this initial clustering center, threshold for automatic starting the fuzzy ant colony algorithm after the optimization speed of GA declined is set. The simulation experiment demonstrates that the proposed algorithm has some advantages in optimization performance and time performance compared to SACA and fuzzy ant colony algorithm.

The difficulty of the proposed algorithm is the integration of GA and ant colony algorithm. Although current solution can meet requirements, it still needs to be perfected. In the future, we will study how to improve the accuracy of our method. Moreover, we will explore the convergence property, and improve the converging speed. In addition, we will research on the application of this optimization algorithm, such as wireless sensor network, artificial intellegence.

Moreover, we will collect real date traces to evaluate the performance of our work. Also, we compare the performance of our work with other clustering algorithm, such as k-means clustering algorithm[2, 23], Hierarchical clustering algorithm[24-25], Density-based clustering algorithm[26-27]. Through the experimental evaluations and comparisons, we aim at finding their advantages and disadvantages for these clustering algorithms. Thus, we can get the applicable range and conditions for each clustering algorithm.

References

[1] J. Van Ryzin, Classification and Clustering. Proceedings of an Advanced Seminar Conducted by the Mathematics Research Center, (**2014**), May 3–5, Madison, USA
[2] M. E. Celebi, A comparative study of efficient initialization methods for the k-means clustering algorithm, Expert Systems with Applications, vol. 40, pp. 200-210 (**2013**).
[3] L. Gong, Identification of activity stop locations in GPS trajectories by density-based clustering method combined with support vector machines, Journal of Modern Transportation, vol. 23, pp. 202-213(**2015**).
[4] L. Gong, Activity stop and non-activity stop identification in GPS trajectories utilizing density-Based clustering method and support vector machines,Transportation Research Board 94th Annual Meeting, (**2015**).
[5] S. Oreski and G. Oreski, Genetic algorithm-based heuristic for feature selection in credit risk assessment, Expert Systems with Applications, vol. 41, pp. 2052-2064 (**2014**).
[6] E. Elhamifar and R. Vidal, Sparse subspace clustering: Algorithm, theory, and applications, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, pp. 2765-2781 (**2013**).
[7] Y. Yao, A novel heterogeneous feature ant colony optimization and its application on robot path planning, IEEE Congress on Evolutionary Computation (CEC), (**2015**), pp. 522-528
[8] M. López-Ibánez, Ant Colony Optimization: A Component-Wise Overview, (**2015**).
[9] B. C. Mohan and R. Baskaran, A survey: Ant Colony Optimization based recent research and implementation on several engineering domain, Expert Systems with Applications, vol. 39, pp. 4618-4627 (**2012**)
[10] J. He and Z. Hou, Ant colony algorithm for traffic signal timing optimization, Advances in Engineering Software, vol. 43, pp. 14-18(**2012**).
[11] M. Reed, An ant colony algorithm for the multi-compartment vehicle routing problem, Applied Soft Computing, vol. 15, pp. 169-176(**2014**).
[12] J.-L. Deneubourg, The dynamics of collective sorting robot-like ants and ant-like robots, Proceedings of the first international conference on simulation of adaptive behavior on from animals to animats, (**1991**), pp. 356-363

[13] E. D. Lumer and B. Faieta, Diversity and adaptation in populations of clustering ants, Proceedings of the third international conference on Simulation of adaptive behavior, (**1994**), pp. 501-508.
[14] S. Zhongzhi, A clustering algorithm based on swarm intelligence, International Conferences on Info-tech and Info-net, (**2001**), pp. 58-66.
[15] V. Ramos and J. J. Merelo, Self-organized stigmergic document maps: Environment as a mechanism for context learning, Spanish Conference on Evolutionary and Bio-Inspired Algorithms, (**2002**), Merida, Spain.
[16] F. Abbattista, An evolutionary and cooperative agent's model for optimization, IEEE International Conference on Evolutionary Computation, (**1995**), pp. 668-671.
[17] Y.-F. Dong, Combination of genetic algorithm and ant colony algorithm for distribution network planning, International Conference on Machine Learning and Cybernetics, (**2007**), pp. 999-1002.
[18] P. M. Kanade and L. O. Hall, Fuzzy ants as a clustering concept, NAFIPS Conference, (**2003**), pp. 227-232
[19] K. Dasgupta, A genetic algorithm (ga) based load balancing strategy for cloud computing, Procedia Technology, vol. 10, pp. 340-347 (2013)
[20] W. Qing-Hong, An Ant Colony Algorithm With Mutation Features, Journal of Computer Research and Development, vol. 36, pp. 1240-1245 (**1999**)
[21] E. Amiri, Energy efficient routing in wireless sensor networks based on fuzzy ant colony optimization, International Journal of Distributed Sensor Networks, vol. 14(**2014**).
[22] H. Ayad and M. Kamel, Topic discovery from text using aggregation of different clustering methods, Conference of the Canadian Society for Computational Studies of Intelligence, (**2002**), pp. 161-175
[23] J. A. Hartigan and M. A. Wong, Algorithm AS 136: A k-means clustering algorithm, Journal of the Royal Statistical Society. Series C (Applied Statistics), vol. 28, pp. 100-108(**1979**)
[24] S. Bandyopadhyay and E. J. Coyle, An energy efficient hierarchical clustering algorithm for wireless sensor networks, IEEE conference INFOCOM, (**2003**), pp. 1713-1723.
[25] C. F. Olson, Parallel algorithms for hierarchical clustering, parallel computing, vol. 21, pp. 1313-1325 (**1995**).
[26] H. P. Kriegel, Density-based clustering, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 1, pp. 231-240 (**2011**).
[27] F. Cao, M. Estert, W. Qian, A. Zhou, Density-Based Clustering over an Evolving Data Stream with Noise, Proceedings of the 2006 SIAM International Conference on Data Mining, (**2006**), pp. 328-339.

