TELKOMNIKA Indonesian Journal of Electrical Engineering
Vol.12, No.3, March 2014, pp. 2118 ~ 2126
DOI: http://dx.doi.org/10.11591/telkomnika.v12i3.3973

Application of Parallel Annealing Particle Clustering Algorithm in Data Mining

Xue-Feng Jiang
School of computer engineering Shenzhen Polytechnic, Shenzhen, 518055, China
e-mail: xfj_2013@163.com

Abstract

With development of the computer technology, the large-scale calculation problems are often appeared in the network, it needs a lot of system resources and support of hardware, it often bring troubles in engineering optimization, so it needs apply the method such as the group's global optimization method and its improved algorithm to obtain reliable results in the computer system. In the study, it proposes a kind of particle swarm optimization based on parallel annealing parallel clustering algorithm, it is a new global optimization algorithm and it is especially suitable for continuous variable problem. In the engineering field, it can be used in large-scale computational problems; it is based on the method of group, and has parallelism abilty. In the parallel particle swarm optimization algorithm, the particle swarm can reduce the consumption of calculation time. The experimental results show that the Multipoint interface (MPI) communication used in annealing parallel particle swarm optimization algorithm not only can reduce the computing time of particle swarm, but also improve the clustering quality, stronger effectiveness algorithm is verfied.

Keywords*: particle clustering algorithm, parallel annealing, data mining, application*

1. Introduction

According to the huge data produced by commercial industrial information and finance, etc., various kinds of data algorithms have become the main trend of data mining. In the commercial field, the data mining can be applied to obtain information in the competition of today's global market, e.g., the retailer can use data mining technology to analyze the purchase mode of customers, in the mail business, the data mining technology can be used in the market analysis, in addition it also can be used in the fraud testing of credit card. Now with the developing of e-commerce, it produces a large number of Web data, these data are need of mature and complicated data mining technology.

The latest progress in related technology, which makes it become possible for the application of large-scale data in many scientific fields. The produce speed of this kind of data is much faster than the artificial analysis speed of them. As known that the calculation simulation program running on the best performance computer can produce trillions of byte data in a few hours, which will take a few weeks for one person to find useful information from these data. The data mining technology is expected to develop the corresponding tools, which can analyze the large number of data set generated by simulation programs, and help engineers and scientists grasp dynamic physical process and all kinds of causal relationships between the potential mechanisms. Recently there appeared some other applications of the data mining technology, e.g., in the field of genome analysis and understanding of gene activity and in analysis of astrophysics field such as classification of the stars and the Milky Way, etc. [1-5]. However the mining large data set requires huge computational resources, because the running of mining algorithm in computer will spend much time, in order to reduce the calculation time, one effective method is the sampling method, but in some cases, it may lead to inaccuracy of data model, even that the mining models is useless as in the contour recognition and abnormal point identification [6, 7]. Another way is to adopt the parallel computing and parallel data mining algorithm which can provide the most effective way in the large data set mining [8, 9].

In the studies, [10, 11] some meaningful research directions in the parallel data mining are introduced, e.g., optimization of data mining structure, the neural network, analysis and design of the training control system, etc. Now the particle swarm optimization has been widely

Received July 22, 2013; Revised September 13, 2013; Accepted October 15, 2013

used in all kinds of problems of data processing, as the particle swarm algorithm needs less parameter, and the parameters settings of the particle swarm algorithm are suitable for most of the data processing, problems [12-14].

2. Parallel Annealing Particle Swarm Optimiation Algorithm

Although in recent years, a lot of improved particle swarm optimization (PSO) algorithm have been appeared in order to improve its performance [15], but the improved algorithm is serial calculation it influenced the potential parallelism ability of particle swarm optimization algorithm, so in order to improve the performance of particle swarm optimization algorithm, the improved algorithm of parallel computing is needed. In the study, particle swarm optimization calculation is adopted in the parallel calculation, learning and communication mode of the parallel particle swarm optimization algorithm are presented. As each parallel process of particle swarm can not solve the problem of local minima, the simulated annealing algorithm is adopted in order to improve the diversity of particle swarm and avoid the occurrence of the phenomenon of local convergence.

Design of Parallel Algorithm: According to the analysis above mentioned, optimization algorithm of particle swarm can be easily decomposed and parallel implemented in the multiple computers, thus the speed of optimization of the groups may be increased or decreased according to the performance of the computers' processors, so it can be able to search solution space more accurately and reduce the possibility of being trapped in local minima. The algorithm has the scalability, which means that the essence of this algorithm is concurrency operation, but it will bring additional cost of communication.

The tracking processes of particle swarm are in all relatively independent parallel process. There exists different distribution of different parameters in each process, in order to ensure the diversity in serial particle swarm optimization algorithm, the calculation of the algorithm is performed independently. So the solving of particle swarm optimization problem is the obstacle in improving computational efficiency of in parallel calculation, a synchronous way is adopted to update parallel particle swarm and optimize individual particle and groups of fitness in the study.

when the asynchronous parallel computing method in the particle swarm optimization algorithm is adopted, the convergence of particle swarm to the local minimum problem can be solved, because parallel computing algorithm can improve the calculation speed and accelerate the convergence of particle swarm. Although the different parameters can keep the diversity of population, but it can not improve convergence of the algorithm essentially.

The simulated annealing process is based on the Metropolis algorithm, and its temperature is dropped slowly, i.e., the temperature plays the role of regulating parameters, if the reduction of temperature is assumed as well as speed of logarithms, the simulated annealing process will converge to the lowest energy state energy function Combined with the Metropolis algorithm in each parallel process of simulation, the annealing algorithm is designed as follows.

The Markov chain is used in the simulation of the heat balance and form the related random algorithm. It is the method applying the Monte Carlo stochastic to simulate the large number of atoms at a given temperature balance state. The random variable X_n represents the state of any Markov chain x_i when the time is n, the new randomly generated state x_j, it denotes another random variable Y_n, it can make the hypothesis that the generated new state meets the symmetry conditions as below (1):

$$p(Y_n = x_j / X_n = x_i) = p(Y_n = x_i / X_n = x_j) \quad (1)$$

Where ΔE represents the generated energy difference from state $X_n = x_i$ to state $Y_n = x_j$ in the system. If the energy difference is negative, then the transfer can lead to a lower energy state and the transfer can be accepted. The new state will be accepted as the starting point of the algorithm in the next step. If energy difference ΔE is positive, the algorithm is

based on probability method in the process. In the first step, choose the random number ξ evenly distributed in the unit interval [0,1], if $\xi < \exp(-\Delta E / T)$, where T represents the operating temperature. The transfer will be accepted and set $X_{n+1} = Y_n$, otherwise transfer will be refused and set the $X_{n+1} = X_n$, then old configuration is put to use in the next step of the algorithm.

The calculated purpose is to find a low energy system; its state is a Markov chain process. When the temperature T is near to the zero, just shown as the $F = \langle E \rangle - TH$, the system free energy F is approach to average energy $\langle E \rangle$, as $T \to 0$, the $\langle E \rangle$ will be deduced to minimum value of the average energy, in other words, the low-energy state in the sequence at low temperature will get the stronger support. But this strategy is not a good method, because in the low temperature, the convergence speed of the Markov chain is very slow. So the better method to improve the computational efficiency is making the random system work at high temperature, then convergence speed will be fast and then as the temperature drops slowly, the system will keeps the balance.

Then two related elements are designed, one is the transition probability of the iterative process in algorithm in new temperature; another is the annealing schedule which determines the drop speed of the temperature.

In order to achieve finite time convergence of the particle swarm in each parallel process through applying simulated annealing algorithm, the simulated annealing schedule and the temperature with finite sequence value and the number of transfer under different temperature can be adopted. simulated annealing schedule and the related parameters are set as follows:

(a) Initial temperature value

The initial value of the temperature is set high enough to make all the proposed transfer can be accepted by the simulated annealing algorithm.

(b) Drop of temperature

Cooling temperature is the key parameters jump out of local minima, as the cooling temperature is directly influenced by the accepted standard, so cooling temperature can be adjusted automatically through the fitness of the particle group.

(c) Temperature of the ultimate value

If the temperature is dropped for the provisions times, it still does not reach the expected times, the system annealing will be stopped or require the lower value of the acceptance ratio and the acceptance ratio is defined as the number of transfer.

In the early operation state of the Particle swarm algorithm, in general the ratio of local maximum adaptive value and the individual average maximum value are large, if the value of annealing temperature is set higher, the particle can be moved freely. With the operation of the particle swarm optimization, the annealing temperature t is decreased automatically.

So for each temperature, enough times of the transfer will make each of the tests have more accepted transfer. In the approximation of the global optimal solution, if the temperature drop rate is slow enough, the accepted deterioration solution probability will be decreased, the particle group will form the lowest energy ground state.

According to the annealing temperature t_{ij}, the acceptance criteria of the simulated annealing transition probability can be designed.

$$p_{ij} = \begin{cases} 1 & f(x_y) < f(x_{i(j+1)}) \\ \exp\left(-\dfrac{f(x_{ij}) - f(x_{i(j+1)})}{t_{ij}}\right) = \exp\left(-\dfrac{f(x_{ij}) - f(x_{i(j+1)})}{f(p_y Best)\sqrt{f(x_y)} - 1}\right) & f(x_y) \geq f(x_{i(j+1)}) \end{cases} \quad (2)$$

According to the particle swarm evolution of adaptive value and the transition probability of the accepting particle, it not only receives the optimal solution, but also can accept deterioration solution and jump out of local minima. When the adaptive value of the new particle is increased, the system must accept new particle as new particles; when the adaptive value is decreased,

from the type (2), the simulated annealing transition probability P_{ij} can be calculated, if the P_{ij} is the random number between 0 and 1. The system also will accept the new particles, or receive the original particle as the new particle. The evolutions of the probability of accepting deterioration solution P_{ij} approaches the zero gradually, the algorithm can jump from local minima and find the global optimal solution, it can guarantee the convergence of algorithm.

Simulated annealing algorithm transition probability can satisfy the sufficient conditions of heat balance. According to the type of structure, the transfer probability is symmetric nonnegative, so for most of the Markov chain, the transition probability satisfies the below three conditions:

(a) Negative condition $P_{ij}>0$ for all (i, j).

(b) Normalized condition $\Sigma p_{ij}=1$ for all i.

(c) Symmetry condition $p_{ij} = p_{ji}$ for all (i, j).

The more details of the transition probability p_{ij} and the balance principle is needed to clarify the details of balance principle, the consideration is as below:

(a) When $\Delta E<0$

The hypothesis that from state X_i to transfer state $X_{i(j+1)}$, the energy change ΔE is negative, from the formula (2), $P_{ij}=1$ can be obtained, transition probability formula of symmetry can be used to get $P_{ij}=P_{ji}$, so when $\Delta E<0$ the balance principle can be satisfied.

(b) When $\Delta E>0$

If make the hypothesis that from state X_i to transfer state r $X_{i(j+1)}$, the energy change ΔE is positive, according to the formula (2) it can be find that:

$$\exp(-\frac{\Delta E}{T}) = \exp(-\frac{f(x_{ij}) - f(x_{i(j+1)})}{t_{ij}}) < 1 \quad (3)$$

Through applying the symmetry of transition probability formula $P_{ij} = P_{ji}$, it can be found that $\Delta E>0$ meets balance principle.

3. Algorithm of Parallel Anneling Particle Cluster Class

3.1. Parallel Clustering Algorithm Design

The data parallel calculation method is adopted in previous parallel clustering algorithm just shown as Figure 1, the magnitudes of the data block stored in each parallel node and exchange the calculation results through the parallel communication method. If the data classification is not reasonable e.g., the data belongs to the same kind is divided to different nodes, or belong to the different kind is divided to the same node, all these will reduce the clustering quality.

How to design the parallel clustering algorithm in the cluster environment is a notable research direction. We will discuss how to design the parallel clustering algorithm and reduce the computation time and improve the clustering quality. Due to the high complexity of the clustering algorithm, the convergence speed is slow, so in the group environment of parallel clustering algorithm, the idea of parallel tasks is designed as Figure 2.

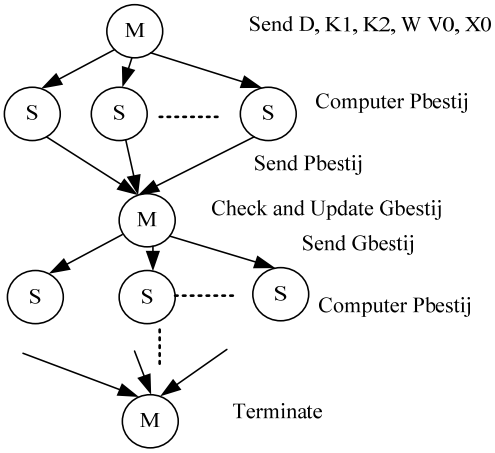

Figure 1. Synchronous Parallel of PSO Algorithm

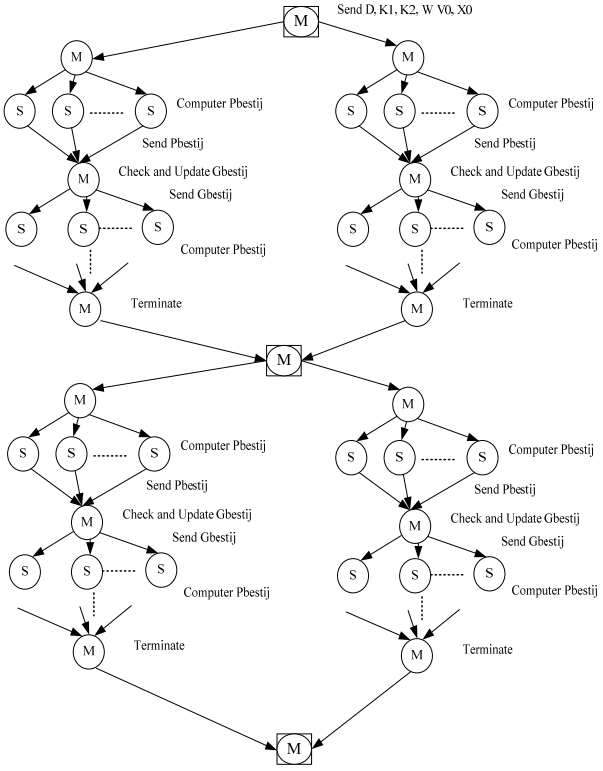

Figure 2. Particial Asynchronous Parallel of PSO Algorithm

3.2. Design of Parallel Clustering Algorithm

In the clustering algorithm, it assumes that the MC is the number and the data can be divided into many kind of particles and clustering center, so each particle x_i can be structured as $x_i = (m_{i1}....m_{ij}...m_{lN_c})$, among them , m_{ij} represents the number of j clustering center and the i particle among C_{ij} class, so one group of particle represents more candidate center of particle swarm, the fitness is very easy to obtained through using quantization error.

$$f_{ij} = \frac{\sum_{j=1}^{M_c}\left[\sum_{\forall Z_p \in C_{ij}} d(z_{p,}, m_j)/C_{ij}\right]}{M_c} \tag{4}$$

Among them, the d is assigned to one kind of sample point, the distance between sample point and Euclidean center is defined as follows:

$$d(z_p, m_j) = \sqrt{\sum_{k=1}^{M_d}(z_{pk} - m_{jk})^2} \tag{5}$$

$$f_{ij} = \frac{\sum_{j=1}^{M_c}\left[\sum_{\forall Z_p \in C_{ij}} d(z_{p,}, m_j)/C_{ij}\right]}{M_c} \tag{6}$$

$$d(z_p, m_j) = \sqrt{\sum_{k=1}^{M_d} (z_{pk} - m_{jk})^2} \tag{7}$$

Among them, M_d represents the input sample dimension, namely that the number of samples in given frequency. Thus it puts forward a kind of global optimization of the parallel algorithm for clustering. Through applying theoptimization algorithm of the parallel annealing particle swarm, sample data can be clustered according to the following methods:

1. The data is transferred from the Master machine to the Slaver machine;
2. Initialization from Machine slaver and each particle will randomly form MC clustering center;
3. Set the numbers as local or global iteration rate, asynchronous parallel way times clustering is performed in the machine Slaver.
   (1) For each particle i do;
   (2) For each sample data pz do;
   (3) Calculate the Euclidean distance from the sample to the C_{ij};
   (4) Compare the update local optimal position of particle swarm in the machine Slaver;
   (5) The simulated annealing to new generation particle swarm is performed in the machine Slaver use annealing progress and probability transfer formula;
4. Host master executive blocking synchronization to get the all fitness results from machine Slaver parallel particle;
5. By comparing host master update global optimal position judge if the new global optimal position of fitness is satisfied with iterative termination conditions, the conditions are satisfied, the calculation will be terminated or be turned to the next step;
6. Host master will broadcast the global optimal position.
7. The machine Slaver gets the global optimal particle position and turn to the step3.

4. Experimental and Comparative Analysis

The clustering problem for the testing is as follows:

Test data 1: This problem follows the clustering rule, the number of the randomly generated data among [-1, 1] is about 60000, The data is about 2M bytes and if the $x_1, x_2 \sim U(-1,1)$, $x_1 > 0.7$ or $x_1 \le 0.3$ and $x_2 \ge -0.2 - x_1$ then x_1, x_2 will belong to the first class A or belong to the class B

Test data 2: It is a two-dimensional clustering problems, there are four different categories and the clustering problem is only focus on one input and concerned really class, the number of total data is about 100000, the article is about 4M bytes and it is consisted of four independent bivariate normal distribution data:

$$x_1, x_2 \sim N\left(\mu = (m_i \quad 0), \sigma = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}\right) \tag{8}$$

Where $i = 1,2,3,4$ and the μ is the average, $m_1 = -3$, $m_2 = 0$, $m_3 = 3$, $m_4 = 6$, σ is the covariance.

4.1. Comparisons of Clustering Quality and Computation Time

Through the above parallel clustering algorithm, each node is to keep the particle population diversity and complete the clustering process independently. The algorithm of the communication cost will be low. Clustering results of each node are the optimized clustering results; they are just approach to global optimization results, but not the best global optimization solution, so the optimization of node clustering results can reflect the actual global clustering results and the quality of parallel clustering algorithm. And related experiments are to verify this parallel clustering algorithm.

Each node optimization clustering result should be able to reflect the actual clustering model, the clustering quality is acceptable. Parallelization implementation of algorithm will not bring negative effects; the parallel computing can obtain the more accurate solutions. So in the group environment, through the designed data parallel clustering algorithm, it not only can obtain the higher acceleration ratio, but also can get good clustering quality.

4.2. Parallel Computing Performance Analysis

In the Windows operating system, structure library of MPI is transferred through the message, It is used to solve the parallel annealing particle swarm optimization of the test data, and communications are used the language of C\C++ and MPI. The eight computers are taken as the work station, one DELL server is worked as the main node. Realization method of asynchronous parallel are adopted, each local iteration may be ignored in the other work process, so the local and global iteration rate should be set as $s \leq 3$.

From Table 1, it can be found that, in the large group computer, the idle time produced by waiting for news is in the dominated status, when the scale of computer is reduced, the proportion of working time will be increased significantly, the parallel algorithm can reduce the communication frequency of each process and the idle time is reduced and then communication time can be neglected. Accelerated ratio is equals to single execution time; machines execution time; efficiency is equals to acceleration ratio; machine number.

In the Table 2 above asynchronous mode of the parallel algorithm, the relationship between relative work, communication and the idle time are presented. Work time includes all calculation work time, communication time includes comparison of the data and sending messages. Idle time is the time besides the work time and communication time. Work time, communication time and idle time are the execution time.

Table 1. Performance Comparisons of Several Parallel Cluster Algorithms of each Node

Cluster problem	Parallel clustering algorithm	Average computation time	Clustering error	Inner Cluster distance	Inter Cluster distance
Data 1	PCM	117	0.765	1.77	3.67
	4 nodes parallel SA-PSO	60	0.58	0.94	4.86
	8 nodes parallel SA-PSO	49	0.46	0.73	5.21
Data 2	8 nodes parallel PCM	239	0.26	0.79	1.62
	4 nodes parallel SA-PSO	110	0.20	0.62	1.84
	8 nodes parallel SA-PSO	86	0.16	0.49	2.16

Table 2. Consumed Time of Parallel Computing

Cluster problem	Average computation time	P=0	P=2	P=4	P=6	P=8
	Communication time	0	3	9	13	17
	Work time	98	56	43	32	23
Test data 1	Ratio of Communication		18.67	4.67	2.46	1.35
	Idle time		3	6	7	8
	Total time	98	62	57	52	48
	Communication time	0	15	21	29	34
	Work time	189	93	72	50	33
Test data 2	Ratio of Communication		6.2	3.43	1.72	0.97
	Idle time		7	11	16	20
	Total time	189	115	104	95	87

Table 3. Performance Analysis of Parallel Algorithm of each Node

Cluster problem	Parallel performance	P=2	P=4	P=6	P=8
Test data 1	Speed-up ratio	1.56	1.70	1.86	2.02
	efficiency	0.78	0.43	0.31	0.25
Test data 2	Speed-up ratio	1.64	1.82	1.99	2.17
	efficiency	0.82	0.45	0.33	0.27

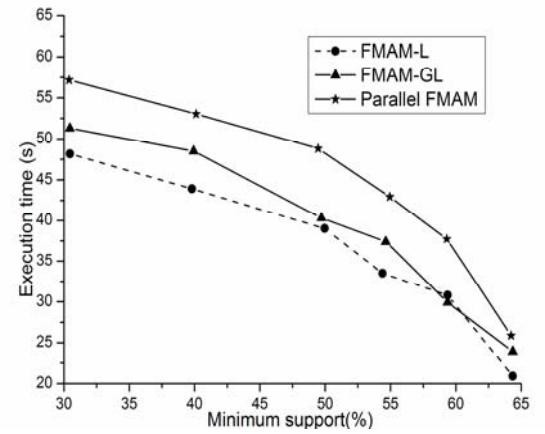

Figure 3. The Relationship of Execution Time and Min Support of Different Algorithm (in the 1st layer)

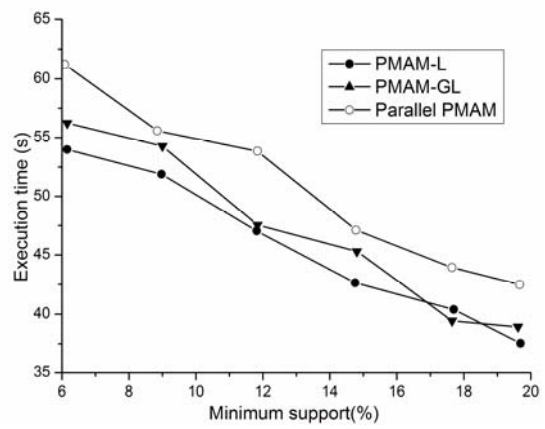

Figure 4. The Relationship of Execution Time and Min Support of Different Algorithm (in the 2nd layer)

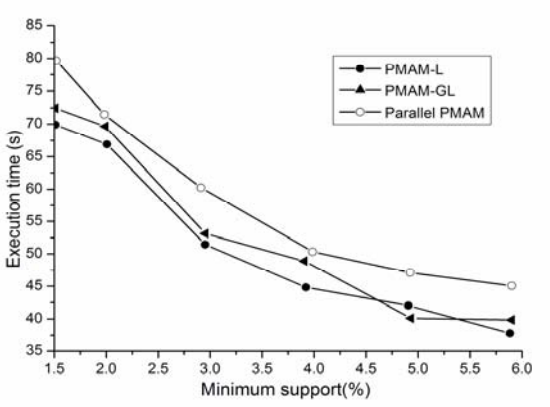

Figure 5. The Relationship of Execution Time and Min Support of Different Algorithm (in the 3rd layer)

From the Figure 3 to Figure 5, it can found that tha parallel algorithm can improve the performance of PMAM. In the Table 3, the speed ratio, efficiency and the relationship between the numbers of processor are given out. From the table, we can see that the data 1 in smaller test scale, the speed up ratio is low and with the increase of the working node, which decreases the efficiency of the algorithm. As for the data two produced by the large scale test, the speed–up ratio is near to optimized value, i.e., the processor number and its efficiency is decreased in a slow speed. In the parallel algorithm, the working time is reduced with the increasing number of the processor and the utilization factors are reduced meanwhile. So in order to find the best processor number, it needs to find the balance between increasing efficiency and reducing speed.

5. Conclusion

Information often contains mass data; clustering algorithm calculation needs a large number of I/O costs and enough memory space, which influences the scalability and response time of the algorithm. Although there are some parallel clustering algorithms, but mostly data parallel way of calculating is adopted, but how to divide the data rationally are not taken into consideration, so the quality of clustering is not high. In order to get the result of the computation quickly and improve the global search ability, we study a kind of particle swarm optimization method in the parallel clustering algorithm based on the global annealing optimization method. We utilize the particle swarm optimization algorithm and its parallel

characteristics, combined with simulated annealing algorithm to overcome the problems of the group evolution. Through the group network technology and MPI communication in the annealing parallel particle swarm optimization algorithm, the consumption of the computing time can be reduced, the clustering quality can be improved. The related experiments in the paper verify the effectiveness of the proposed algorithm.

Acknowledgements

This project is supported by the National Science Research Fund of China. This is partially supported by the Science Fund of Guangdong Province (No.60902876).

The Authors would like to thank Professor Lawson Richard for critically evaluating the manuscript and the group members of the staff room of computer engineering for their kind help at various stages of the research.

References

[1] Eberhart RC, Y Shi. *Particle swarm optimization: Developments, applications and resources*. Proc. Cong. Evol. Comput., 2001; 1: 81-86.
[2] Shi Y, RC Eberhart. *An adjusted parameter particle swarm optimizer.* Proceedings of the IEEE International Conference on Evolutionary Computation. Anchorage. 1998; 5(1): 121-125.
[3] Shi Y, RC Eberhart. *Parameter selection in particle swarm optimization*. Proceedings of the 7th Internationl Conference on Evolutionary Programming. 1998; 9: 591-600.
[4] Shi Y, RC Eberhart. *Empirical study of particle swarm optimization*. Proceeding of the IEEE International Conference on Evolutionary Computer. Piscataway, NJ., USA. 1999: 1945-1950.
[5] Suganthan PN. *Particle swarm optimiser with neighbourhood operator*. Proceedings of the Congress on Evolutionary Computation. Washington, DC., USA. 1999; 3: 958-1962.
[6] Clerc M. The swarm and the queen: *Towards a deterministic and adaptive particle swarm optimization.* Proceedings of the Congress on Evolutionary Computation, Washington, DC, USA. 1999; 1951-1957.
[7] Brits R, AP Engelbrecht, F van den Bergh. Scalability of Niche PSO. Proceedings of the International Conference on Evolutionary Computation, Seoul. 2001; 12: 125-130.
[8] *A*ngeline PJ. *Using selection to improve particle swarm optimization.* Proceedings of the International Conference Evolutionary Computation Intelligence, Anchorage, AK, USA. 1998: 84-89.
[9] *Ozc*an E, C Mohan. *Particle swarm optimization:* Surfing the waves. Proceedings of the Congress on Evolutionary Computation, Washington, DC, USA. 1999: 1939-1944.
[10] Kennedy J. *The behavior of particles.* Proceedings of the 7th International Conference on Evolutionary Programming, San Diego, CA, USA. 1998: 579-589.
[11] Shi Y, RC Eberhart. Fuzzy adaptive particle swarm optimization. *Evolut. Comput*., 2001; 1: 101-106.
[12] Van den Bergh F, AP Engelbrecht. Cooperative learning in neural networks using particle swarm optimizers. *South Afr. Comput. J.,* 2000; 26: 84 -90.
[13] Wu Xiao-chao, Wang Lian-dong, Yan Liao-liao, Xue Fang-xia. Simulation of Radar Track Based on Data Mining Techniques. *TELKOMNIKA Indonesian Journal of Electrical Engineering.* 2013; 11(7): 3780-3788.
[14] Mao Yimin, Xue Xiaofang, Chen Jinqing. An Efficient Algorithm for Mining TopK Closed Frequent Itemsets over Data Streams over Data Streams. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2013; 11(7): 3759-3766.
[15] Yanrong Guo, Baoguo Wu, Yang Liu. Multidimensional data mining using a Kmean algorithm based on the forest management inventory of Fujian Province China. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2013; 11(12): 7290-7294.