

Cooperative Approaches to Bacterial Foraging Algorithm for Clustering

Zhao Hongwei^{1,2} and Tian Liwei*²

¹*School of information science and engineering, Northeastern University,*

²*School of Information Engineering, Shenyang University*
zhw30@163.com, tianliwei@163.com

Abstract

Bacterial Foraging Optimization (BFO) is a novel optimization algorithm based on the social foraging behavior of E. coli bacteria, but it is difficult to optimize to get a high precision due to the randomness of the bacterial behavior, which belongs to intelligence algorithm. This paper presents an extended BFO algorithm, namely the Cooperative Bacterial Foraging Optimization (CBFO), which significantly improves the original BFO in solving clustering problems. A novel clustering method based on the CBFO could be used for solving clustering problems. In this work, firstly, The efficiency and performance of the CBFO algorithm was evaluated using six widely-used benchmark functions, coming up with comparative results produced by BFO, then Particle Swarm Optimization (PSO) is studied. Secondly, the algorithm with CBFO algorithms is used for data clustering on several benchmark data sets. The performance of the algorithm based on CBFO is compared with BFO algorithms on clustering problem. The simulation results show that the proposed CBFO outperforms the other three algorithms in terms of accuracy, robustness and convergence speed.

Keywords: *Bacterial Foraging Optimization, Particle Swarm Optimization, Swarm Intelligence, Data clustering*

1. Introduction

Swarm Intelligence is an innovative artificial intelligence technique for solving complex optimization problems. In recent years, many SI algorithms have been proposed: such as Ant Colony Optimization (ACO), Particle Swarm Algorithm (PSO)[1], Bacterial Foraging Optimization (BFO), etc. In recent years, bacterial foraging behaviors as a rich source of potential engineering applications and computational model have attracted more and more attentions. A few models have been developed to model the bacterial foraging behaviors and been applied for solving practical problems.

Bacterial Foraging Optimization (BFO)[2][3] algorithm is a new swarm intelligence algorithm, imitating the behaviors of real bacteria on finding food source and sharing the information of it, which has been applied successfully to some engineering problems, such as constrained optimization problems, neural networks and clustering.

A novel Cooperative optimization model, Bacterial Foraging Optimization algorithm, is designed in this paper. As a generalized neighborhood search algorithm, BFO uses swarm intelligence of biosphere to solve optimization problems, by means of heuristic search strategy, whose capacity of tracking changes rapidly gives algorithm the ability of global optimization, because of the characteristics of global convergence itself, and the initial value can be set as fixed or random allowing parameters to be set in a wider scope. BFO has strong adaptability and parallelism; many behavior combinations can be selected due to its good flexibility, and it can get better optimization performance which genetic algorithm and particle swarm optimization do not possess. This artificial intelligence model, based on biological behavior, is different from the classical pattern. Firstly design

a single entity perception, behavioral mechanisms, and then place a group of entities in the environment so that they can solve the problems in environment interaction, however making the best reaction under the stimulation of the environment is the basic idea of BFO [4].

Bacterial Foraging Optimization (BFO) is a population-based numerical optimization algorithm in the literature [5-6]. BFO has been applied successfully to some engineering problems, such as machine learning and optimal control etc. However, experimentation with complex and multimodal benchmark functions reveal that the BFO algorithm possesses a poor convergence behavior compared to other SI algorithms and its performance also heavily decreases with the growth of the search space dimensionality. Tao, X.M (2010) introduced the K-means algorithm to speed up the iteration, but the performance was unstable because of many random processes in BFO which affected the practical application of the method. Xie,J.Y.,(2010) obtained clusters automatically for the amount of K and applied them to arbitrary shape of data, better parallelism, but the quality of ultimately clustering quality was affected by the number and the size of grids which led to some limitations [7].

As an important research direction of data mining, clustering algorithm is a suitable means of classifying data for different patterns based on the different characteristics of different objects [8]. But the traditional clustering has greater ability of local search, for it is very sensitive to the initial cluster centers and easily falls into local optimum. If outliers are randomly selected as the initial centers, the whole quality of classification will decline [9]. BFO is less sensitive to initial values, even if to global optimization, which has bad convergence and slower iteration rate in late period. Aiming at the advantages and disadvantages of both algorithms, this paper presents a global optimization idea to improve clustering algorithm based on BFO, the result of which on a small data set shows that the improved algorithm obtains clear classifications and better performance [10-11].

This paper applies CBFO algorithm to solve clustering problems, which have been tested on a series of datasets, then compares the performance of CBFO on clustering with results of BFO, PSO and CBFO on the same data sets. The above data sets are provided from the UCI database [12].

2. Standard BFO Algorithm

2.1. Bacterial Chemotactic Behavior

The E. coli bacterium has a control system that enables it to achieve a complex type of search and avoidance behavior. To move forward, the flagella counterclockwise rotate and the organism swims [13]. While a clockwise rotation of the flagellum causes the bacterium to randomly tumble itself in a new direction and then swim again. Swimming is more frequent as the bacterium approaches a nutrient gradient. Tumbling, hence direction changes, is more frequent as the bacterium moves away from some food to search for more. Basically, bacterial chemotactic behavior is a complex combination of swimming and tumbling that keeps bacteria in places of higher concentration of nutrients. So, the bacterial foraging process can be subdivided into four motile behaviors namely chemotaxis, swarming, reproduction, and elimination and dispersal [14].

We can understand the bacterial chemotactic behavior. Assume that there are one discovered food sources. At the very beginning, a bacterial chemotactic behavior of forager will start. That bacterial will have no knowledge about the food sources. The bacterial utilizes its own capability to starts exploiting it. Hence, the bacterial will become an forager. The forager alternate between “swim” and “tumble” enable the bacterium search for nutrients in random directions.

2.2. The Original BFO Algorithm Steps

The original BFO algorithm steps are as follows:

[Step 1] Initialize parameters $n, S, N_c, N_s, N_{re}, N_{ed}, P_{ed}, D(i)$ ($i=1,2,\dots,S$), θ^i . Where,

$D(i)$: the run-length unit,

n : Dimension number in the search space,

S : The bacteria's number in the colony,

N_c : chemotactic steps,

N_s : swim steps,

N_{re} : reproductive steps,

N_{ed} : elimination and dispersal steps,

P_{ed} : probability of elimination

[Step 2] Elimination-dispersal loop: $e = e + 1$.

[Step 3] Reproduction loop: $d = d + 1$.

[Step 4] Chemotaxis loop: $j = j + 1$.

[4.1] For $i = 1, 2, \dots, S$, take a chemotactic step for bacteria i as follows.

[4.2] Compute fitness function, $J(i, j, d, e)$.

[4.3] Let $J_{last} = J(i, j, d, e)$ to save this value since we may find better value via a run.

[4.4] Tumble: Generate a random vector $\Delta(i) \in R^n$ with each element $\Delta_m(i)$, $m = 1, 2, \dots, S$, a random number on $[-1, 1]$.

[4.5] Move:

[4.6] Compute $J(i, j+1, d, e)$ with $\theta^{j+1, d, e}$.

[4.7] Swim:

(i) Let $m = 0$ (ii) While $m < N_s$

• Let $m = m + 1$.

• If $J(i, j+1, d, e) < J_{last}$, let $J_{last} = J(i, j+1, d, e)$. then another step of size $D(i)$ in this same direction will be use the new generated $\theta^{j+1, d, e}$ to compute the new $J(i, j+1, d, e)$.

• Else let $m = N_s$.

[4.8] Go to next bacterium ($i+1$): if $i \neq S$ go to 4.2 to process the next bacteria.

[Step 5] If $j < N_c$, go to step 3.

[Step 6] Reproduction:

[6.1] For the given d and e , and for each $i = 1, 2, \dots, S$, let J_{heal} be the healthiness of the bacteria. Sort bacterium in order of ascending values.

[6.2] The S_r bacteria with the highest J_{heal} values die and the other S_r bacteria with the best values split and the copies that are made are placed at the same location as their parent.

[Step 7] If $k < N_{re}$ go to step 2.

[Step 8] Elimination-dispersal: For $i = 1, 2, \dots, S$, with probability p_{ed} , eliminate and disperse each bacteria. If a bacterium is eliminated, simply disperse one to a random location on the optimization domain. If $e < N_{ed}$, then go to step 2; otherwise end.

3. The Cooperative Bacterial Foraging Optimization (CBFO) Algorithm

In the previous study, the Bacterial Foraging Algorithm shows that a small operation run-length unit of bacteria has the exploration ability while the bacterium with a relatively large run-length unit has the exploiting skill. The bacterium with a large run-length unit parameter has the exploring ability, while the bacterium with a relatively small run-length unit parameter has the exploiting skill. This inspired us to divide the foraging procedure of artificial bacteria colony into multiple phases, and each phase occupies a portion of generations and characterized by the different value of run-length unit parameter D .

This inspired us to divide the foraging procedure of artificial bacteria foraging into multiple phases; each characterized by the different value of run-length unit and occupies a portion of generations. This approach produces four classes of bacterial individual's chemotaxi, swarming, reproduction, elimination and dispersal, which depending on the

particular run-length unit that they used. The bacterial of chemotaxis search space and has the responsibility to find the promising domains and to leave the local optima that have visited, while the bacterial swarming focuses on the precision of the found solutions, i.e. the bacteria perform exploitation of the neighborhood of the best-so-far solutions found by the producers.

The CBFO algorithm adapts different run-length unit parameters in sequential phases. The bacterial individuals search the best positions found by each bacterium in each phase, supplies as an input to the algorithm in the next phase. In the first phase, the bacteria colony searches the whole solution space, then the bacteria to locate promising regions in local. In this first phase, Each bacterium records all its visited positions and the position with the highest fitness value is considered as a promising solution candidate.

The bacteria colony is reinitialized with relatively small D_{initia} from these promising solution candidates and the neighborhoods of these candidates. The bacteria colony is renewed again with even smaller D_{initia} to fine-tune the best-so-far solutions found in the foregoing phase. So, the CBFO algorithm can be classified into the serial heterogeneous cooperation on the implicit space decomposition level.

The pseudocode of CBFO is described in Table 1. We also embeds the chemotaxis, swarming, reproduction, elimination and dispersal processes into each tumble step. The CBFO can improve the algorithm convergence rate significantly.

Table 1. The Improved CBFO Algorithm

Algorithm1: The Improved CBFO algorithm

1: Initialization.
 Initialize the position and the associated run-length unit D_{initia} of the bacteria colony;
 Iteration=0;

2: Do while(each phase $k = 1 : N$)
 /*CHEMOTAXIS Phase*/
for (each chemotactic)
for (each bacterium)
 Generate a random vector Δ , where each element belongs to $[-1, 1]$. Move the bacterium i in the direction of $\Delta / \sqrt{\Delta^T \Delta}$ by a unit walk of size $D(k)$. Then Evaluate the fitness of the bacterium , Apply greedy selection;
end for
end for

3: Calculate Set flag = 0 for each food source;

4: /*SWARMING Phase*/
for (each unit)
 Take another unit walk in the same direction;
Calculate the new fitness ,to find a new food source in its neighborhood;
 Evaluate the fitness of the new food source;
end for

5 : /*REPRODUCTION Phase*/
 The half of bacteria with the worst fitness die and the other $S/2$ bacteria with the best fitness split;

6: /*ELIMINATION & DISPERSAL*/
 With probability p_{ed} , eliminate and disperse each bacterium;
 Iteration= Iteration +1;
end while

7: Output the best solution achieved

4. Experimental Result

4.1 Benchmark Functions

Ten well-known benchmark functions are used in the test. These functions contain three unimodal functions, four multimodal functions and three rotated functions.

The first function is Sphere function whose global minimum value is 0 at $(0, 0, \dots, 0)$. Initialization range for the function is $[-5.12, 5.12]$. It is a unimodal function with non-separable variables.

$$f_1(x) = \sum_{i=1}^n x_i^2 \quad x \in [-5.12, 5.12]^D \quad (1)$$

The second function is Rosenbrock function whose global minimum value is 0 at $(1, 1, \dots, 1)$. Initialization range for the function is $[-15, 15]$. It is a unimodal function with non-separable variables. Its global optimum is inside a long, narrow, parabolic shaped flat valley. So it is difficult to converge to the global optimum.

$$f_2(x) = \sum_{i=1}^n 100 \times (x_{i+1} - x_i^2)^2 + (1 - x_i)^2 \quad x \in [-3, 3]^D \quad (2)$$

The fourth function is Rastrigin function whose global minimum value is 0 at $(0, 0, \dots, 0)$. Initialization range for the function is $[-15, 15]$. It is a multimodal function with separable variables.

$$f_3(x) = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10) \quad (3)$$

The third function is Quadric function whose global minimum value is 0 at $(0, 0, \dots, 0)$. Initialization range for the function is $[-10, 10]$. It is a unimodal function with non-separable variables.

$$f_4(x) = \sum_{i=1}^D \left(\sum_{j=1}^i x_j \right)^2 \quad (4)$$

4.2 Results for the 20-D Problems

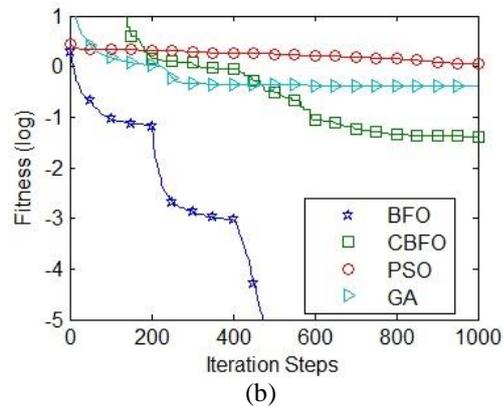
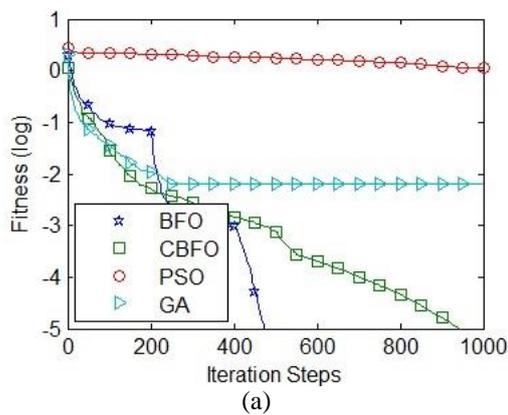
From the results, we observe that CBFO achieved better results on all test problems than the original BFO. As we can see in Figure 1, under the influence of the serial heterogeneous cooperative approach: the bacteria colony starts exploring the search space at the primordial phase ; in the succeeding phases, by refining the parameters C, the bacteria slow down near the optima to pursue the more and more precise solutions.

This experiment conducted on 20-D problems to compare the proposed CBFO algorithm with the original BFO, from the results, we observe that CBFO achieved better results on all test problems than the original BFO, GA and the Particle Swarm Optimization. This experiment runs 30 times respectively for each algorithm on each benchmark function. Table 2 lists the experimental results for each algorithm on functions $f_1 \sim f_4$. Figure 1 shows the search progresses of the average values found by all algorithms over 30 runs for functions $f_1 \sim f_4$. From Figure 1, the CBFO algorithm

surpasses all other algorithms on functions $f_1 \sim f_4$. This significant improvement is due to the ability of BFO to locate promising regions, as well as the fine-grained search property of explicit space decomposition in every stage.

Table 2. Comparison among PSO, BFO, CBFO and GA on 20D Problems

Function		PSO	BFO	CBFO	GA
f_1	Mean	1.67262e-010	1.24309e-006	1.43727e-018	2.46335e-003
	Std	2.38437e-020	1.44400e-006	3.34567e-018	4.06650e-003
	Min	5.21328e-014	2.35204e-007	3.87653e-023	8.27501e-005
	Max	1.30103e-009	5.53748e-005	1.19876e-017	1.22146e-002
f_2	Mean	4.4527e-008	4.94397e-003	4.25062e-015	1.87544e+000
	Std	5.9876e-008	5.08761e-003	9.57608e-015	2.11319e+000
	Min	1.3456e-009	5.33807e-004	5.60991e-020	6.93828e-003
	Max	2.8765e-007	2.16419e-002	4.47663e-024	7.53221e+000
f_3	Mean	1.76185e-007	6.55588e-001	3.32132e-015	1.59279e-001
	Std	1.65432e-007	3.62200e-001	8.76543e-015	2.88468e-001
	Min	7.54651e-010	8.05214e-002	0	1.88327e-003
	Max	6.34721e-007	1.16555e+000	4.26326e-014	7.93133e-001
f_4	Mean	1.90708e-004	1.34681e-001	1.06646e-008	3.62955e-001
	Std	1.11928e-004	6.32151e-002	1.67892e-008	2.58463e-001
	Min	3.34561e-005	3.21457e-002	2.43901e-010	5.12611e-002
	Max	5.87654e-004	2.09876e-001	6.54138e-008	9.27131e-001



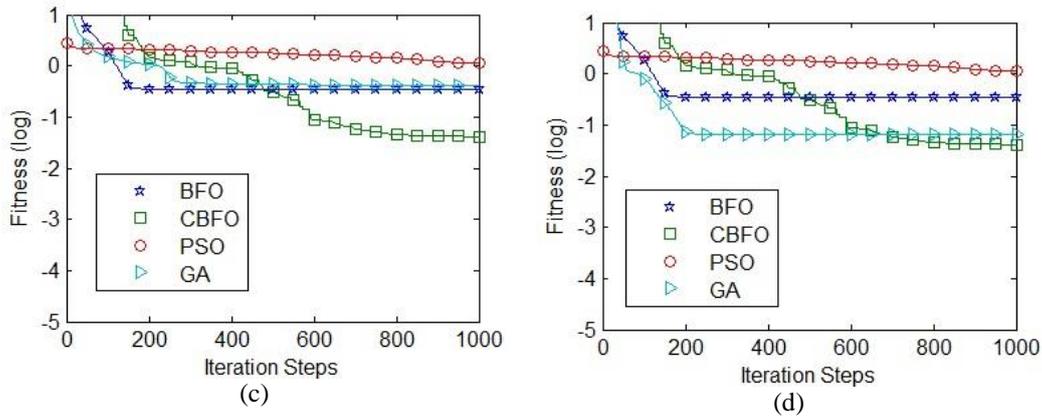


Figure 1. Convergence Results of CBFO, BFO, PSO and GA on 20-D Benchmark Functions. (a) Sphere;(b) Rosenbrock; (c) Rastrigrin; (d) Quadric

5. Second a Data Clustering Experimental Results

To analyse the performance of the proposed CBFO approach for clustering algorithm, the results of PSO,GA and BFO with different data sets have been compared in this paper, which are selected from the UCI machine learning repository.

The algorithm base on CBFO algorithms is used for data clustering on Iris data sets, which is able to provide the same partition of the data points in all runs. Iris data is thus selected from the UCI machine learning repository, cluttering result of which sets by BFO and the

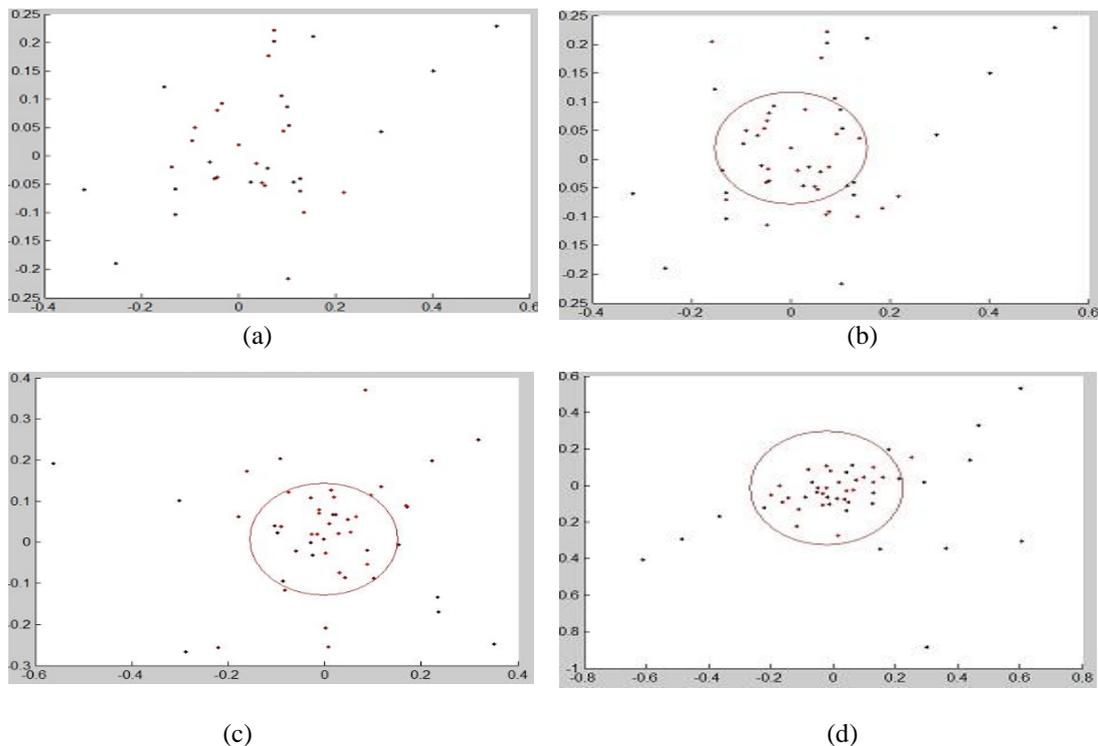


Figure 2. The Data Distribution of Iris Data Sets and the Clustering Result by CBFO and other Algorithm: (a) Iris Distribution. (b) Iris Clustering Result Base on BFO (c)Iris Clustering Result Base GA Algorithm (d) Iris Clustering Result Base on CBFO

CBFO clustering algorithm is presented in Figure 2. From the result Figure 2, for all real data sets, the basic clustering algorithm with CBFO outperforms the other methods.

6. Conclusion

In this paper, based on the cooperative approaches, a novel Bacterial Foraging Optimization (BFO) algorithm is presented, namely Cooperative Bacterial Foraging Optimization (CBFO). In order to demonstrate the performance of the CBFO algorithm, we compared it with those of BFO, PSO, GA optimization algorithms on several benchmark functions. Comparison of experimental results show, that firstly, the clustering algorithm based on CBFO makes similar data gather obviously; secondly, the model is more stable and accurate than the old one; thirdly, it distinguishes samples precisely while also improving the cluster quality and obtaining better centers with clear division which represents reducing computation amount. However, the convergence speed issue remains to be improved and researched.

Acknowledgements

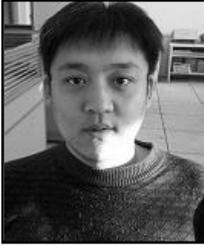
The authors are supported financially by International cooperation project (Project No.S2012ZR0191) and the Natural Science Foundation of Liaoning Province (Project No.2013020011) and the Social Science Foundation of Liaoning Province (Project No.L14ASH001) and this work is supported in part by the International S&T Cooperation Program of China (ISTCP) under Grant 2011DFA91810-5 and Program for New Century Excellent Talents in University of Ministry of Education of China under Grant NCET-12-1012.

References

- [1] M. A. Badamchizade, A. Nikdel, and M. Kouzehgar, "Comparison of Genetic Algorithm and Particle Swarm Optimization for Data Fusion Method Based on Kalman Filter", *International Journal of Artificial Intelligence*, vol.5, no.10, (2010).
- [2] H. N. Chen, Y. L. Zhu and K. Y. Hu, "Multi-colony bacteria foraging optimization with cell-to-cell communication for RFID network planning", *Applied Soft Computing*, vol.10, no.2 (2010).
- [3] D. H. Kim and J. H. Cho, "Adaptive Tuning of PID Controller for Multivariable System Using Bacterial Foraging Based Optimization", *Proceedings of 3rd Web Intelligence Conference*, (2005); Lodz, Poland..
- [4] S. Mishra, "A hybrid least square-fuzzy bacterial foraging strategy for harmonic estimation", *Evolutionary Computation*, vol.9, no.1, (2005).
- [5] M. Tripathy, S. Mishra, L. L. Lai and Q. P. Zhang, "Transmission Loss Reduction Based on FACTS and Bacteria Foraging Algorithm", *Proceedings of 9th International Conference on Parallel Problem Solving from Nature*, (2006); Reykjavik, Iceland.
- [6] M. B. Miller and B. L. Bassler, "Quorum sensing in bacteria, *Annual Review of Microbiology*", vol.55, no.1, (2010).
- [7] X. D. He and L. D. Qu, "Bacterial Foraging Optimization Clustering Algorithm", *Application Research of Computers*, vol.26, no.10, (2009).
- [8] D. Karaboga and C. Ozturk, "A novel clustering approach: Artificial Bee Colony (ABC) algorithm", *Applied Soft Computing*, vol.11, no.1, (2011).
- [9] H. Li and Q. F. Zhang, "Multi objective optimization problems with complicated Pareto sets", *IEEE Transactions on Evolutionary Computation*, vol.13, no.2, (2009).
- [10] B. Liu and Y. Q. Zhou, "A Hybrid Clustering Algorithm Based on Bacterial Foraging Optimization Algorithm", *Journal of Computer Engineering and Applications*, vol.44, no.18, (2008).
- [11] I. E. Evangelou, D. G. Hadjimitsis, A. A. Lazakidou and C. Clayton, "Data mining and knowledge discovery in complex image data using artificial neural networks", in *Proceedings of the Workshop on Complex Reasoning and Geographical Data*, (2001); Paphos, Cyprus.
- [12] M. S. Kamel and S. Z. Selim, "New algorithms for solving the fuzzy clustering problem", *Pattern Recognition*, vol.27, no.3, (1994).
- [13] T. Sato and M. Hagiwara, "Bee system: finding solution by a concentrated search; Systems, man, and cybernetics", *IEEE international conference on computational cybernetics and simulation*, (1997); Orlando, FL, USA.

- [14] X. M. Tao, J. Xu and L. B. Yang, "An Improved Hybrid Algorithm Based on Particle Swarm Optimization and K-means Algorithm", Journal of Electronics & Information Technology, vol.32, no.1, (2010).

Authors



ZhaoHongwei

male, M.Sc. ,associate professor, Shenyang University, supervisor for postgraduates, School of Information Engineering, His main research areas are Swarm intelligence theory and Optimization decision method.



TianLiwei,

male, PhD, professor, Shenyang University, supervisor for postgraduates, School of Information Engineering, His main research areas are Computational Biology and Swarm intelligence theory.

