

Best Moments Magnetotactic Bacteria Algorithm for Optimization

Ce Yang and Zhaofeng Chen

Naval Academy of Armament
xyangcej@gmail.com, zhaofeng1981225@126.com

Abstract

Magnetotactic bacteria optimization algorithm(MBOA) is an optimization algorithm based on the characteristics of magnetotactic bacteria, which is a kind of polyphyletic group of prokaryotes with the characteristics of magnetotaxis that make them orient and swim along geomagnetic field lines. In this paper, an improved MBOA is proposed. At first, it generates moments based on the interaction energy of cells. Then, the moments of cells are regulated based on the best cell. Some cells have chance to move towards the best one. At last, some worse cells will be replaced by randomly generated solutions in order to keep diversity. Thus, it can find optimum solution. It is tested on ten standard function problems and compared with many popular optimization algorithms, including CLPSO, LeDE and ABC. Experiment results show that the improved MBOA is very effective in optimization problems and has superior performance to the compared methods on many benchmark functions.

Keywords: Magnetotactic bacteria optimization algorithm, best individual, moments

1. Introduction

Swarm Intelligence(SI), such as Ant Colony Optimization(ACO) [1] and Particle Swarm Optimization(PSO) [2], is one of important branches of natural computing. Learning from life systems, people have developed many SI methods to solve complicated optimization problems since evolutionary algorithms(EAs) [3] were inspired by evolutionary biology in 1960s. Since 2000s, Some relative new natural computing techniques, such as Artificial Bee Colony(ABC) [4], Artificial Fish Swarm(AFS) [5], Biogeography-Based Optimization Algorithm(BBO) [6] were developed based on different life systems. All of them are inspired by animals behaviors.

In nature, magnetotactic bacteria(MTB) [7] is a special kind of bacteria that can orient and swim along magnetic field lines. They are different from chemotactic bacteria in that they have mineral particles inside their bodies, and their enveloping membrane, together called magnetosomes(MTSs). With the aid of magnetosome chains, magnetotactic bacteria can orient themselves along geomagnetic field lines (magnetotaxis) in the earth magnetic field [8].

Based on the biology principle of MTB, Mo has proposed an original magnetotactic bacteria optimization algorithm [9].It has been tested on standard benchmark functions including multi-modal functions and compared with many popular and classical NIAs. MBOA shows better performance and good potential ability in solving optimization problems. It has been improved into several new variants of MBOA [10-15]. It shows the potential ability of solving optimization problems and has very fast convergence speed.

In this paper, the original MBOA is improved. The problem solutions are generated by moments mechanisms based on interaction energy among solutions in the new improved algorithm. It works by mainly three steps: MTSs generation, MTSs migration and MTSs replacement. The MTSs migration is based on the half better individual's moments.

2. Magnetotactic Bacteria Optimization Algorithm Based on Best Moments

2.1. Biological Basis

In MTB, a chain of magnetosomes makes up of compass of magnetotactic bacteria, which keeps the long axis of the bacteria parallel to the geomagnetic field direction [16]. For the MTB, each cell carries a remanent magnetic moment, the direction of which is given by the orientation of the magnetosome-chain axis and its magnetic polarity [17]. If each cell is to align its magnetosome chain parallel to the other ones, with the same polarity would yield the most efficient swimming way for living. This specific behavior is the inspiration source of the MBOA. For the MBOA, we consider the state that each cell is to align its magnetosome chain parallel to the other ones, with the same polarity would yield the most efficient swimming way for living as finding the optimal solution. The interaction energy between different chains in different cells make MTB strive for better living. In Figure 1, the compass of multi-cells in a magnetotactic bacterium is shown.

The simplest hypothesis for magnetotaxis is passive orientation of the swimming bacterium along the magnetic field lines by the torque exerted by the field on the magnetic moment. Considering a chain of magnetosomes as a cylinder of infinite length in a magnetic field B , its energy of the bacterial moment can be estimated as follows [18].

$$E_m = -M \cdot B = -MB \cos \theta \quad (1)$$

where θ is the angle between M and B .

According to [19], the interaction energy between two dipoles from different magnetosome chains is:

$$E_{n,m} = \left(\frac{D}{1 + nD + mD} \right)^3 \quad (2)$$

where $n, m = 0, 1, 2, \dots$ are the number of magnetosomes of two cells, D is the distance between neighbor centers in a chain.

Suppose the interaction energy between two cells in a MTB is as follows:

$$\frac{1}{2}(E_n + E_m) = E_{n,m} \quad (3)$$

where E_n, E_m are the energy of two cells, respectively. If two cells have the same number of magnetosomes, that is, $n = m$, and suppose $E_n = E_m$, then we have

$$E_{n,m} = E_n = E_m.$$

2.2. Procedures of MBOA

In the MBOA, a population is a magnetotactic bacterium with multi-cells. A cell is a candidate solution, that is, a vector. The attributes of a solution vector are considered as magnetosomes. The values of attributes of a solution vector is considered as the moments of a cell. The interaction between two cells is represented by the differential vector between them. In the following we describe the main procedures of MBOA.

Initialization of the population The initial population is filled with number of randomly generated-dimensional real-valued vectors (*i.e.*, cells of a MTB). Let represents the i th cell (for generation index $t=0$) initialized randomly. Then each cell is generated as follows:

$$x_{i,j} = x_{\min,j} + rand(0,1) \times (x_{\max,j} - x_{\min,j}) \quad (4)$$

where $i = 1, 2, \dots, SN$, $j = 1, 2, \dots, n$. $x_{\max,j}$ and $x_{\min,j}$ are upper and lower bounds for the dimension j , respectively. $rand(0,1)$ is a random number between 0 and 1. Thus, a $N \times n$ distance matrix $D = [D_1, D_2, \dots, D_N]$ is generated, where i and r is mutually different integer indices from $\{1, 2, \dots, N\}$, and r is randomly chosen one.

MTSs generation Based on the distances among cells, the interaction energy e_i between two cells based on (2) is defined as

$$e_{i,j}(t) = \left(\frac{d_{i,j}(t)}{1 + c_1 \times norm(D_i) + c_2 \times d_{p,q}(t)} \right)^3 \quad (6)$$

where t is the generation index, c_1 and c_2 are constants. $d_{p,q}$ is randomly selected from D_p . $norm(D_i)$ is the Euclidean length of vector D_i . p and r are mutually different and randomly chosen integer indices from $\{1, 2, \dots, N\}$. $q \in \{1, 2, \dots, n\}$ stands for one randomly chosen integer. n is the dimensions of a cell. D_i stands for the Euclidean distance between two cells X_i, X_r .

After obtaining interaction energy, for simplifying calculation, ignore where θ is the angle between M and B and direction in (1), the moments M_i are generated as follows:

$$M_i(t) = \frac{E_i(t)}{B} \quad (7)$$

where B is a constant named magnetic field strength. Then the total moments of a cell is regulated as follows:

$$x_{i,j}(t) = x_{i,j}(t) + m_{r,q}(t) \times rand \quad (8)$$

where $m_{r,q}$ is randomly chosen from m_i . $rand$ is a random number in interval (0,1). MTSs regulation After moments generation, evaluate the population according to cells' fitness, then the moments regulation is realized as follows. We set a magnetic field strength probability mp . If $rand > mp$, the moments in the cell migrate as follows:

$$X_i(t+1) = X_{best}(t) + (X_{best}(t) - X_i(t)) \times rand \quad (9)$$

Otherwise,

$$X_i(t+1) = X_r(t) + (X_{best}(t) - X_i(t)) \times rand \quad (10)$$

$rand$ is a random number in interval (0,1). r is randomly chosen from $\{1, 2, \dots, N\}$.

In this step, the best cell's moments are used to regulate the moments of the cells in next generation. Based on (9), some new cells will be generated base on the best one in the current generation. Thus it has an enhanced local search ability. Based on(10), some randomly chosen cells will receive information and regulate their moments based on the interaction between the best cell and some cells of last next generation. Thus, it will have chance to approximate to the best one. Thus, it can enhance the ability of global search. And it can also increase the solutions diversity of the algorithm. Based on this step, some cells in the population will 50% chance to be regulated based on the moments of the best cell.

MBOA obtains the optimal solution by regulating the moments of cells continually by the process of MTS generation, MTS expanding and MTS replacement. When the MBOA obtains the optimal solution, it corresponds to the state that when the moments of all cells are oriented in the geomagnetic field.

MTSs replacement after the moments migration, evaluate the population according to cells' fitness, then some worse moments are replaced by the following way:

$$X_i(t+1) = m_{r,q}(t) \times ((rand(1,n) - 1) \times rand(1,n)) \quad (11)$$

where $m_{r,q}$ is randomly chosen from m_r . r is randomly chosen from $\{1, 2, \dots, N\}$. $q \in \{1, 2, \dots, n\}$ stands for one randomly chosen integer. $rand(1,n)$ is a random vector with dimensions.

At last, evaluate the population according to cells' fitness after replacement. In general, one fifth of the cells with worse fitness will be replaced by randomly generated vectors based on(11).

Generally, MBOA includes the following steps:

I. Define the simple bounds, determination of algorithm parameters.

II. Initialization: Randomly create the initial population in the search space.

III. while stop criteria is not met

 for =1: N

 Calculate interaction distance according to(5)

 end for

 for =1: N

 for =1: N

 Calculate interaction energy according to (6)

 end for

 Obtain moments according to (7)

 for =1: N

 MTSs generation according to (8)

 end for

 end for

 Evaluate the population according to fitness

 for =1: N

 if rand > 0.5

 MTSs regulation according to (9)

 else

 MTSs regulation according to (10)

 end if

 end for

 Evaluate the population according to fitness

 for = N/5:N

 MTSs replacement to (11)

 end for

 Memorize the best solution achieved so far

VI. end while

3. Simulation Results

To test the performance of the proposed algorithm, we demonstrate the effectiveness of MBOA by comparing the performance of MBOA with the state-of-the-art evolutionary algorithms on 10 benchmark functions. These benchmark functions are widely used in evaluating global numerical optimization algorithms.

3.1. Benchmark Functions and Experiments Settings

A short description of 10 benchmark functions[20] is shown in Tables 1. These functions can be classified into two groups. f1– f6 are unimodal; f7 is a noisy quartic function; The unimodal functions here are used to test if MBOA can still maintain the fast-converging feature. The next seven functions f8–f10 are multimodal functions and the number of local minima increases exponentially with the problem dimension. These

functions can be used to test the global search ability of the algorithm in avoiding premature convergence.

To validate the effectiveness of the proposed algorithm, we compare MBOA with the CLPSO [21], Learning -enhanced differential evolution (LeDE) [22], standard ABC [23]. And for the compared methods, we follow the parameter settings in the original paper. In the MBOA, there are three parameters settings, including the magnetic field strength $B = 3$, $c_1 = 50$, $c_2 = 0.003$, magnetic field strength probability $mp = 0.5$.

Table 1. Benchmark Functions used in Experiments

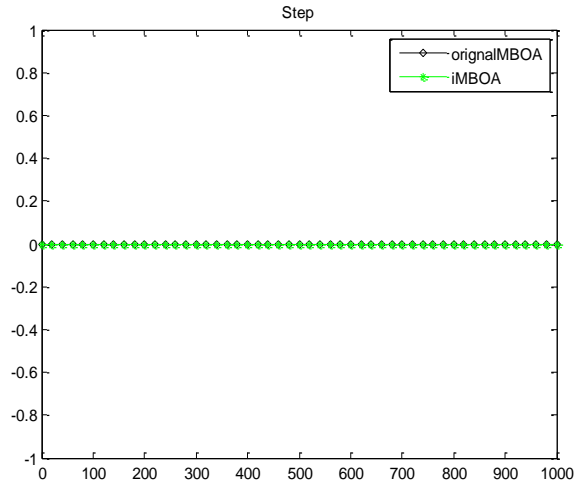
Function	Range	D	Formulation
f_1 : Sphere	[-100, 100]	30	$f(x) = \sum_{i=1}^n x_i^2$
f_2 : Schwefel2.22	[-10, 10]	30	$f(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $
f_3 : Schwefel1.2	[-100, 100]	30	$f(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$
f_4 : Zakharov	[-5, 10]	30	$f(x) = \sum_{i=1}^n x_i^2 + (\sum_{i=1}^n 0.5ix_i)^2 + (\sum_{i=1}^n 0.5ix_i)^4$
f_5 : Step	[-100, 100]	30	$f(x) = \sum_{i=1}^n (\lfloor x_i + 0.5 \rfloor)^2$
f_6 : Schwefel2.21	[-100, 100]	30	$f(x) = \max_i \{ x_i , 1 \leq i \leq n\}$
f_7 : Noisy Quartic	[-1.28, 1.28]	30	$f(x) = \sum_{i=1}^n ix_i^4 + random[0,1)$
f_8 : Rosenbrock	[-30, 30]	30	$f(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$
f_9 : Schwefel	[-500, 500]	30	$f(x) = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$
f_{10} : Rastrigin	[-5.12, 5.12]	30	$f(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$

3.2. Comparison Results

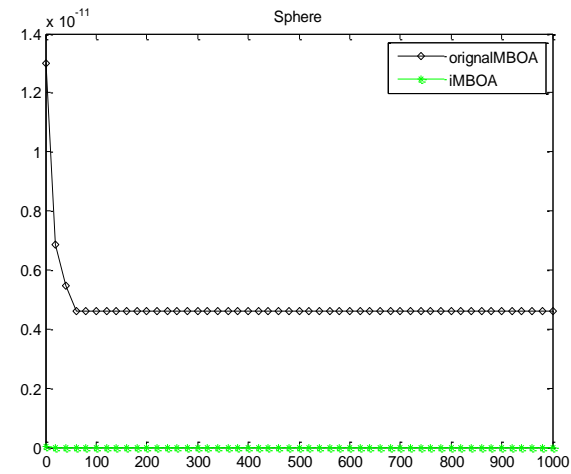
In this section, we compare the improved MBOA(iMBOA) with the original MBOA. And it is also compared with CLPSO, LeDE and ABC on ten benchmark functions, where LeDE uses the number of FEs as described in [22] and the other algorithms use the number of 3.0×10^5 FEs. The results of LeDE are gained from [22] directly. The convergence comparison of original MBOA and iMBOA is shown in Figure1 for four examples.

Table 2. Comparison between Original MBOA and iMBOA

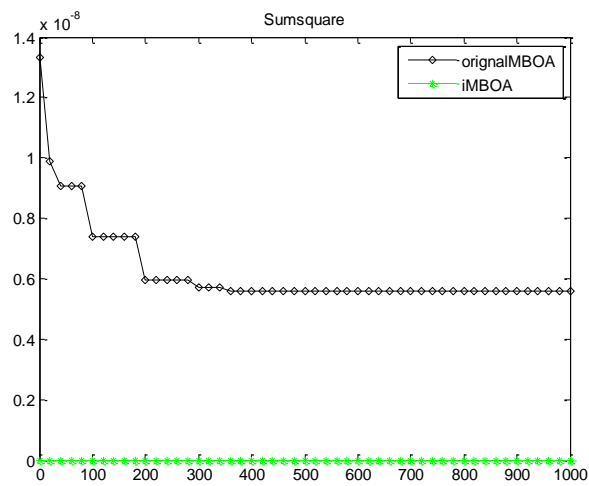
Fun	NFES	Original MBOA	iMBOA
f_1 : Step	1500×100	0(0)	0(0)
f_2 : Sphere	2000×100	4.04e-12(4.42e-13)	0(0)
f_3 : Sumsquare	5000×100	4.89e-13(7.68e-10)	0(0)
f_4 : Noisy Quartic	1500×100	8.85e-06(8.14e-06)	1.36e-05(1.33e-05)
f_5 : Easom	1500×100	-2.67e-09(1.81e-17)	-1(1.16e-08)
f_6 : Schwefel1.2	3000×100	5.68e-10(7.22e-11)	0(0)
f_7 : Zakharov	2000×100	5.59e-10(1.56e-10)	0(0)
f_8 : Powell	1500×100	6.21e-09(1.06e-09)	0(0)
f_9 : Rotatedhyper	1500×100	1.17e-10(1.44e-11)	0(0)
f_{10} : Rastrigin	4000×100	2.98e-07(4.03e-08)	0(0)
f_{11} : Branin	100×100	-1.23 (0.57)	0.3979(1.30-05)
f_{12} : Dropwave	200×100	-1 (3.96e-14)	-1 (0)
f_{13} : Schaffer		0(0)	0(0)
f_{14} : Griewank		2.67e-17(3.54e-18)	0(0)



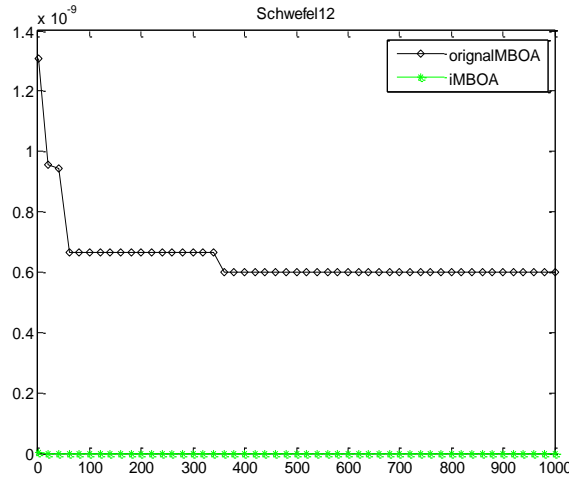
(a) Step



(b) Sphere



(c) Sumsquare



(d) Schwefel1.2

Figure 1. Convergence Comparison of Original MBOA and iMBOA for Functions (a)Step,(b)Sphere, (c) Sumsquare and (d) Schwefel1.2

In Figure 1, it can be seen that the improved MBOA is better than original MBOA on the performance of coverage. The results are shown in Table 3 in terms of the mean, st.d, mid, best, and worst of the solutions obtained in the 30 independent runs by each algorithm. And the best results of each function are highlighted in boldface.

Table 3. Performance Comparisons of MBOA, CLPSO, ABC and LeDE

Func.	Min		CLPSO	LeDE[23]	ABC	iMBOA
f_1	0	Mean	1.6107e+03	2.19e-34	3.7410e-04	0
		St.d	8.8223e+03	1.18e-34	5.2756e-04	0
		Mid	1.8656e-05	/	1.8222e-04	0
		best	1.4310e-07	/	1.0277e-05	0
		worst	4.8322e+04	/	0.0021	0
f_2	0	Mean	7.0509	1.09e-24	0.0185	0
		St.d	26.8787	4.09e-25	0.0139	0
		Mid	5.5581e-05	/	0.0126	0
		best	5.0224e-06	/	0.0060	0
		worst	111.8162	/	0.0691	0
f_3	0	Mean	2.4509e+04	1.16e-38	1.8136e+04	0
		St.d	2.2241e+04	2.28e-38	3.0732e+03	0
		Mid	1.6661e+04	/	1.8838e+04	0
		best	1.9675e+03	/	1.0380e+04	0
		worst	9.7116e+04	/	2.3634e+04	0
f_4	0	Mean	1.2006	/	6.1615	0
		St.d	6.3836	/	4.5043	0
		Mid	0.0118	/	4.3177	0
		best	9.7792e-04	/	0.6140	0
		worst	34.9975	/	15.0324	0
f_5	0	Mean	0.0667	0	0.1333	0
		St.d	0.2537	0	0.3457	0
		Mid	0	/	0	0
		best	0	/	0	0
		worst	1	/	1	0
		Mean	5.0312	4.66e-27	53.4590	0

f_6	0	St.d	3.6695	2.31e-26	7.0503	0
		Mid	3.5064	/	55.1662	0
		best	1.4613	/	34.9710	0
		worst	13.8917	/	67.9291	0
		Mean	4.3756	1.10e-03	0.2888	7.5250e-06
f_7	0	St.d	16.7644	5.58e-04	0.07524	6.8837e-06
		Mid	0.0506	/	0.2859	5.1280e-06
		best	0.0152	/	0.16793	2.8292e-08
		worst	78.0337	/	0.4516	2.9347e-05
		Mean	85.3100	0	29.5949	25.0222
f_8	0	St.d	65.4288	0	19.5743	2.5887
		Mid	87.4317	/	25.6863	24.4530
		best	0.3284	/	7.5442	21.8180
		worst	310.5533	/	85.1434	28.7978
		Mean	-11850.1977	-12569.5	-11478.017	-12411.5675
f_9	-12569.5	St.d	1160.4331	0	203.50099	507.60398
		Mid	-12569.4338	/	-	-12569.4866
		best	-12569.4866	/	-	-12569.4866
		worst	-8270.7658	/	-	-10437.5966
		Mean	4.6539	0	7.0311	0
f_{10}	0	St.d	7.3767	0	2.5665	0
		Mid	0.4634	/	7.2404	0
		best	5.5516e-06	/	2.4916	0
		worst	25.4025	/	12.2860	0

As is shown in Table 3, we can see that MBOA outperforms CLPSO, LeDE and ABC on most of the test functions, especially when solving the unimodal problems, the MBOA algorithm offers the best performance on all the test functions. In particular, it offers the highest accuracy on functions f1 to f7. And MBOA and LeDE show the same performance on function f5. For multimodal and high-dimension functions f8 and f9, MBOA performs better than CLPSO and ABC but has worse performance compared with LeDE. The MBOA algorithm and LeDE can also find the optimal solutions on the complex multimodal functions f10.

4. Conclusion

MBOA is a new optimization algorithm inspired by the biology behaviors of magnetotactic bacteria. It shows potential ability in solving optimization problems. In this paper, MBOA is improved based on the interaction between the moments of the best individual and those of some randomly chosen individual. It is compared with CLPSO, LeDE and ABC. The comparison results show that the proposed algorithm has superior performance on many benchmark functions and has the fastest convergence speed.

In future, it will be analyzed in theory and improved its performance for solving those problems which are not solved well in this paper. In addition, MBOA will be improved to solve more complex problems including constrained optimization, multi-objective optimization and some real engineering problems.

References

- [1] M. Dorigo, V. Manianiezzo and A. Colomi, "The ant system: optimization by a colony of cooperating agents IEEE Trans on System", Man and Cybernetics Part B: Cybernetics, vol. 26, no. 1, (1996), pp. 29-41.
- [2] J. Kennedy and R. Eberhart, "Particle swarm optimization", IEEE International Conference on Neural Networks Piscataway, NJ, (1995), pp. 1942-1948.

- [3] T. Back, "Evolutionary Algorithms in Theory and Practice", Oxford, U.K: Oxford Univ Press, (1996).
- [4] D. Karaboga and B. Akay, "A comparative study of Artificial Bee Colony algorithm", Applied Mathematics and Computation, vol. 214, (2009), pp. 108-132.
- [5] F. Bastos. Carmelo, J. A. Lima and B. De Fernando, "A novel search algorithm based on fish school behavior", IEEE International Conference on Systems, Man and Cybernetics, Cingapura, Singapore, (2002).
- [6] D. Simon, "Biogeography-based optimization", IEEE Trans on Evol. Comput., vol. 12, (2006), pp. 702-713.
- [7] D. Faivre and D. Schuler, "Magnetotactic bacteria and magnetosomes", Chem. Rev., vol. 108, (2008), pp. 4875-4898.
- [8] J. G. Mitchell and K. Kogure, "Bacterial motility: links to the environment and a driving force for microbial physics", FEMS Microbiol. Ecol., vol. 55, (2006), pp. 3-16.
- [9] H. W. Mo, "Research on Magnetotactic Bacteria Optimization Algorithm", Proceedings of the Fifth International Conference on Advanced Computational Intelligence, Nanjing, China, (2012) August 15-18.
- [10] H. W. Mo and L. F. Xu, "Magnetotactic Bacteria Optimization Algorithm for Multimodal Optimization", Proceedings of IEEE Symposium on Swarm Intelligence (SIS), Singapore, (2013) November 22-25.
- [11] H. W. Mo, L.L. Liu, L. F. Xu and Y.Y. Zhao, "Performance Research on Magnetotactic Bacteria Optimization Algorithm Based on the Best Individual", Proceeding of the sixth International Conference on Bio-inspired Computing, Wuhan, China, (2014) October 16-18.
- [12] H. W. Mo and M. J. Geng, "Magnetotactic Bacteria Optimization Algorithm Based on Best-Rand Scheme", Proceedings of 6th Naturei and Biologically Inspired Computing, Porto Portugal, (2014) June 22-25.
- [13] H. W. Mo and L. L. Liu, "Magnetotactic Bacteria Optimization Algorithm Based on Best-Target Scheme", International Conference On Nature Computing and Fuzzy Knowledge, Xiamen, China, (2014) August 22-25.
- [14] H. W. Mo, L. L. Liu and L. F. Xu, "A power spectrum optimization algorithm inspired by magnetotactic bacteria", Neural Computing and Applications, vol. 25, no. 7-8, (2014), pp. 1823-1844.
- [15] H. W. Mo, L. L. Liu and M. J. Geng, "A New Magnetotactic Bacteria Optimization Algorithm Based on Moment Migration", 2014 International Conference on Swarm Intelligence, Hefei, China, (2014) October 16-18, pp. 103-114.
- [16] Y. X. Pan, W. Lin and J. H. Li, "Reduced efficiency of magnetotaxis in magnetotactic coccoid bacteria in higher than geomagnetic fields", Bbiophysical Journal, vol. 97, (2009), pp. 986-991.
- [17] W. Michael, G. A. Leida and F. D. Alfonso, "Barros magnetic optimization in a multicellular magnetotactic organism", Biophysical Journal, vol. 92, (2007), pp. 661-670.
- [18] R. B. Frankel, "Magnetic guidance of organisms", Ann. Reo. Biophys. Bioeng., vol. 13, (1984), pp. 85-10.
- [19] A. P. Philipse and D. Maas, "Magnetic colloids from magnetotactic bacteria: chain formation and colloidal stability Langmuir", vol. 18, (2002), pp. 9977-9984.
- [20] W. F. Gao and S. Y. Liu, "Improved artificial bee colony algorithm for global optimization", Information Processing Letters, vol. 111, (2011), pp. 871-882.
- [21] J. J. Liang, A. K. Qin, P. N. Suganthan and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions", IEEE Trans. Evol. Comput., vol. 10, (2006), pp. 281-295.
- [22] Y. Q. Cai, W. J. H and Y. J., "Learning-enhanced differential evolution for numerical optimization", Soft Comput., vol. 16, (2012), pp. 303-330.
- [23] D. Karaboga and B. Basturk, "A comparative study of artificial bee colony algorithm", Appl. Math. Comput., vol. 214, (2009), pp. 108-132.

