A Journal of Software Engineering and Applications, **2012, 5, 66-71**
doi:10.4236/jsea.2012.512b014 Published Online December 2012 (http://www.scirp.org/journal/jsea)

Magnetotactic Bacteria Algorithm for Function Optimization

Hongwei Mo[1], Lifang Xu[2]

[1]Automation College, Harbin Engineering University, Harbin, China; [2]Engineering Training Center, Harbin Engineering University, Harbin, China
Email: honwei2004@126.com

ABSTRACT

Magnetotactic bacteria is a kind of polyphyletic group of prokaryotes with the characteristics of magnetotaxis that make them orient and swim along geomagnetic field lines. A magnetotactic bacteria optimization algorithm(MBOA) inspired by the characteristics of magnetotactic bacteria is researched in the paper. Experiment results show that the MBOA is effective in function optimization problems and has good and competitive performance compared with the other classical optimization algorithms.

Keywords: Magnetotactic bacteria optimization algorithm; Function optimization; Nature inspired computing

1. Introduction

Learning from life system, people have developed many nature inspired computing(NIC) methods to solve complicated optimization computation problems in recent decades. There has been a considerable attention paid for employing algorithms inspired from natural processes and/or events in order to solve optimization problems. For example, genetic algorithms(GAs) which was first introduced by Holland are now a standard optimization tool in engineering. Since 1980s, more and more NIC algorithms were developed following GAs, including Ant Colony Optimization(ACO)[1] and Particle Swarm Optimization(PSO)[2], Immune Algorithm(IA)[3], Artificial Bee Colony(ABC)[4], Bacterial Chemotaxis Algorithm[5], Biogeography based Optimization[6] and so on.

Since many studies were carried out with inspirations from ecological phenomena for developing optimization techniques, we still pay attention to find new inspiration source. ‘No free lunch theorem’ had told us that there is no universal algorithm which can be better over all possible problems. So it is necessary for us to develop new algorithms for problem solving. Tayarani proposed a magnetic optimization algorithm which is based on the principle of particles’ interaction in magnetic optimization [7]. In nature, there is a kind of polyphyletic group of prokaryotes that can orient and swim along magnetic field lines. They are called magnetotactic bacteria (MTBs) [8]. A striking property of MTBs is their ability to orient and propel themselves along geomagnetic field lines (magnetotaxis) in the earth magnetic field. Propelled by their flagella, the bacteria migrate in a net downward direction, following the declination of the field lines of the earth, toward oxygen-poor regions with advantages for survival.

In[9], we have proposed a new algorithm called magnetotactic bacteria optimization algorithm (MBOA) inspired by the distinct behavior of MTBs. It had been tested on some standard benchmarks and compared with some optimization algorithms, and it shows good performance in solving some optimization problems of standard functions. In this paper, MBOA is researched in further.

The remainder of this paper is organized as follows: Section 2 describes the basic procedure of MBOA. In Section 3, experiments on 14 standard functions optimization and analysis are provided. Finally, the conclusions are drawn in Section 4 .

2. MBOA

2.1. Principles of MBOA

MTBs occur widely in natural sediments from both marine and freshwater habitats. They produce intracellular, membrane-bounded magnetite particles and synthesize a kind of magnetite colloids with enveloping membrane. It is called magnetosomes, which are typically arranged in the form of one or several chains and impart a permanent magnetic dipole moment to the bacterium[10].

Copyright © 2012 SciRes.

In magnetotactic bacteria, magnetosomes play important role in regulating the movement of MTBs. The magnetic field lines bend in some of the magnetosomes to minimize their magnetostatic energy[11], whereas in others their direction differs slightly from that of the chain axis. In fact, the MTBs have evolved to be adaptive to the magnetic field. Based on the biology knowledge, we know that one kind of MTBs has multiple cells with chains of magnetosomes. Only those MTBs with magnetosomes in their cells which can make magnetic field lines bend in some of the magnetosomes to minimize their magnetostatic energy can survive in nature. Each magnetosome can produce moment[11]. The MTBs with multi-cell need to produce magnetosome moments with which can minimize their magnetostatic energy. We can consider such a process as an optimization one.

When the MBA runs to solve a problem, it corresponds to the process of producing magnetosomes to be adaptive to magnetic field. It needs to regulate the moments of each magnetosome, just like producing feasible solutions. MBOA obtains the optimal solution by regulating the moments of cells continually.

Consider a problem solving inspired by MTBs, the minimal magnetostatic energy is looked as optimal solution. The multiple cells are looked as feasible solutions. The magnetosomes in each cell can be looked as the features of a candidate solution. The moment of a magnetosome corresponds to feature value.

2.2. Procedures of MBOA

Considering a chain of magnetosomes as a cylinder of infinite length in a magnetic field B, its energy E_a of the bacterial, moment can be estimated as follows.

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

where θ is the angle between M and B.

According to[9], the interaction energy between two dipoles from different magnetosome chains in a MTB with multi cells is:

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

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

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

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

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

$$E_{n,m} = E_{n,m} \qquad (4)$$

The total procedure of MBOA is described as follows:

1: Generate initial cells population C^n and set constant λ, ρ, a .
2: While ($t < MaxGeneration$)
3: calculate cost of each cell
4: normalize cost to calculate magnetic field B
5: for $i = 1:n$
6: for $j = 1:n$
7: If $i \neq j$
8: calculate the distance D
9: If $D > r$
10: calculate interaction energy E of two cells
11: else
12: $E = rand(1, p) * R$
13: end
14: end
15: for $i = 1:n$
16: calculate moment M of each cell
17 regulate the moment of each cell by M
18: end
19: calculate the cost J of each cell, rank the cells, replace some proportional cells by randomly produced moments.
20: rank the cells and find the optimal solution
21: end while

The procedure of MBA is described as follows in detail:

Step 1: All of the moments of magnetosomes in cell population (for t=0) are initialized randomly between upper and lower limit of feature value.

In step 4: The i th magnetic field value f_i is normalized as follows:

$$f_i = \frac{f_i - \min(f_i)}{\max(f_i) - \min(f_i)} \qquad (5)$$

$i = 1,2...,n$, n is the size of population.

The magnetic field of a cell is defined as [9].

Copyright © 2012 SciRes.

$$B_i = \lambda f_i + \rho \quad (6)$$

where λ and ρ are constant.

In step 8: we define the distance between two cells as:

$$D = \sum_{i,j=1}^{n} d_{i,j} / \mathrm{U} \quad (7)$$

Suppose that $x_{i,k}, x_{j,k} \in [-L, U]$ are the k th feature value (moment) of $\vec{x}_i, \vec{x}_j$, respectively. n is the population size. $-L$ and U are the lower limit and upper limit of feature value. Then D can be defined as the following function.

$$D(\vec{x}_i, \vec{x}_j) = \frac{1}{p} \sum_{k=1}^{p} \left| \frac{x_{i,k} - x_{j,k}}{U} \right| \quad (8)$$

In step 9, $r = a * (U/2) * p$, where a is a distance threshold constant.

In step 10, suppose that $M_i = (m_1, m_2, ..., m_p)$, $i = 1, ..., n$ is the i th moment of magnetosome of a cell. Assume the interaction energy E between two cells as Equation (9).

$$E_{i,j} = \left(\frac{D}{1 + 2mD} \right)^3 \quad (9)$$

According to Equation (1), and for simplification, suppose $\cos\theta = 1$, then we get

$$M_i = \frac{E_{i,j}}{B_i} \quad (10)$$

So we have the ways of regulating moments of magnetosomes in a cell (individual) as follows:

$$x_i^t = x_i^{t-1} + M_i^t \quad (11)$$

where x_i^t, x_i^{t-1} are the i th moment of the i th individual(cell) in t generation. M_i^t is the moment of corresponding individual in t generation.

In step 19: after the regulation, the solutions are sorted according to their costs in ascending. The last half of cells is replaced by the following way:

$$x_i = \lambda * ((rand(1, m) - 1) * rand(1, m)) / R \quad (12)$$

In general, the generation number is set as the stopping condition. At last, find the optimal result and output the result.

3. Experiment Results and Analysis

3.1. Problem Definition

Global numerical optimization problems are frequently arisen in almost every field of engineering design, applied sciences, molecular biology and other scientific applications. Without loss of generality, the global minimization problem can be formalized as a pair (S, f), where $S \subseteq R^D$ is a bounded set on R^D and $f : S \to R$ is a D-dimensional real value function. The problem is to find a point $X^* \in S$ such that $f(X^*)$ is the global minimum on S. More specifically, it is required to find an $X^* \in S$ such that

$$\forall X \in S : f(X^*) \le f(X) \quad (13)$$

where f does not need to be continuous but it must be bounded.

3.2. Parameter Settings

In all experiments in this section, all algorithms are the basic ones without any improvement. The values of the common parameters used in each algorithm such as population size and total evaluation number were chosen to be the same. Population size was 50 and the maximum evaluation number was 500 for all functions. The other specific parameters of algorithms are given below[12]:

GA Settings: Single point crossover operation with the rate of 0.8 was employed. Mutation rate was 0.01. Stochastic uniform sampling technique was our selection method.

DE Settings: F is a real constant which affects the differential variation between two solutions and set to 0.5 in our experiments. Value of crossover rate was chosen to be 0.9.

PSO Settings: Cognitive and social components are constants that can be used to change the weighting between personal and population experience, respectively. In our experiments cognitive and social components were both set to 1.8. Inertia weight, which determines how the previous velocity of the particle influences the velocity in the next iteration, was 0.6.

Copyright © 2012 SciRes.

MBOA: For MBOA, we only need to set λ and ρ to decide magnetic field. In our experiments, $\lambda = 0.5$, $\rho = 0.0001$.

3.3. Experiment Results

In order to characterize the type of problems for which the algorithm is suitable and test the performance of MBOA, we used 14 benchmark problems in order to comparison the performance of these algorithms. This set is large enough to include many different kinds of problems such as unimodal(U), multimodal(M), regular, irregular, separable(S), non-separable(N) and multidimensional. Initial range, formulation, the dimensions(D), parameters setting and characteristics(C) of these problems are listed in **Table 1**. The minimal values of Easom and Dropwave are -1. The minimal value of all the other functions is 0. The formulations of benchmark functions are shown in **Table 2**.

The compared results of the MAB with GA, PSO, DE on a large set of functions are listed in **Table 3** and **Table 4**. Each of the experiments in this section was repeated 30 times with different random seeds and the mean best values produced by the algorithms have been recorded. In order to make comparison clear, the values below 10^{-12} are assumed to be 0.

Table 1. Characteristic of benchmark functions

No.	Function	Range	D	C
1	Step	[-100, 100]	30	US
2	Sphere	[-100, 100]	30	US
3	SumSquares	[-10, 10]	30	US
4	Quartic	[-1.28, 1.28]	30	US
5	Easom	[-100, 100]	2	UN
6	Schwefel1.2	[-100, 100]	30	UN
7	Zakharov	[-5, 10]	10	UN
8	Powell	[-4, 5]	24	UN
9	Rotatedhyper	[-65.536,65.536]	30	UN
10	Rastrigin	[-5.12, 5,12]	30	MS
11	Branin	$[-5,10]\times[0,15]$	2	MS
12	Dropwave	[-5.12,5.12 1]	2	MS
13	Schaffer	[-100, 100]	2	MN
14	Griewank	[-600, 600]	30	MN

Table 2. Benchmark function formulations

No.	Formulations
1	$f(x)=\sum_{i=1}^{n}\left(\lfloor x_i+0.5\rfloor\right)^2$
2	$f(x)=\sum_{i=1}^{n}x_i^2$
3	$f(x)=\sum_{i=1}^{n}ix_i^2$
4	$f(x)=\sum_{i=1}^{n}ix_i^4+random[0,1)$
5	$f(x)=-\cos(x_1)\cos(x_2)\exp(-(x_1-\pi)^2-(x_2-\pi)^2)$
6	$f(x)=\sum_{i=1}^{n}(\sum_{j=1}^{i}x_j)^2$
7	$f(x)=\sum_{i=1}^{n}x_i^2+(\sum_{i=1}^{n}0.5ix_i)^2+(\sum_{i=1}^{n}0.5ix_i)^4$
8	$f(x)=\sum_{i=1}^{n/k}(x_{4i-3}+10x_{4i-2})^2+5(x_{4i-1}-x_{4i})^2+(x_{4i-2}-x_{4i-1})^4$ $+10(x_{4i-3}-x_{4i})^4$
9	$f(x)=\sum_{i=1}^{n}\left(\sum_{j=1}^{i}x_j\right)^2$
10	$f(x)=\sum_{i=1}^{n}[x_i^2-10\cos(2\pi x_i)+10$
11	$f(x)=(x_2-\frac{5.1}{4\pi^2}x_1^2+\frac{5}{\pi}x_1-6)^2+10(1-\frac{1}{8\pi})\cos x_1+10$
12	$f(x_1,x_2)=-\frac{1+\cos(12\sqrt{x^2+x_2^2})}{1/2(x_1^2+x_2^2)+2}$
13	$f(x)=0.5+\frac{\sin^2(\sqrt{x_1^2+x_2^2})-0.5}{(1+0.001(x_1^2+x_2^2))^2}$
14	$f(x)=\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$

Because of space limit, we separate experiment results to two sets as shown in **Table 3** and **Table 4**, respectively. Statistical results of 30 runs obtained by GA, DE MBOA are shown in **Table 3** and those of PSO and MBOA are shown in **Table 4**, where Mean: Mean of the Best Values, Std: Standard Deviation of the Best Values.

Table 3. Comparison results of GA, DE and MBOA

No.		GA	DE	MBOA
1	Mean	78.2333	0	0
	Std	41.1563	0	0
2	Mean	91.1986	0	0
	Std	36.4409	0	0
3	Mean	0.4881	0	0
	Std	1.1443	0	0
4	Mean	0.2670	0.006891	2.7511e-05
	Std	0.2237	0.00148	3.8863e-05
5	Mean	-0.6413	-1	-0.9966
	Std	0.4614	0	0.0037
6	Mean	2.4527e+04	18324.1	0
	Std	6.4361e+03	3022.165	0
7	Mean	0.2301	0	0
	Std	0.6375	0	0
8	Mean	2.5631	0.059135	0
	Std	1.6810	0.02199	0
9	Mean	395.4911	1.9523e-09	0
	Std	182.5482	2.3682e-09	0
10	Mean	0	98.21781	1.3281e-10
	Std	0	7.976648	1.5040e-11
11	Mean	0.4075	0.3979	0.3984

Copyright © 2012 SciRes.

	Std	0.0127	0	4.9229e-04
12	Mean	-0.9793	-1	-1
	Std	0.0561	0	0
13	Mean	0.0010	0	0
	Std	0.0031	0	0
14	Mean	0.7959	0	0
	Std	0.4205	0	0

As seen from **Tables 3** and **Tables 4**, there are 8 functions with 30 variables. MBOA outperforms all the other algorithms on 4(Quartic, Powell, Rotatedhyper, Rastrigin) and has the same performance on 1(Matyas) with GA, DE, PSO. It has the same performance on 5 (Step, Sphere, Sumsquares, Schaffer, Griewank) with DE, PSO, GA has the worst performance on these functions. It is better than GA on Step, Sphere, Schaffer, Griewank, Easom, Schwefel1.2, but is worse than PSO, DE on Easom, Branin. It is better than GA, PSO on Zakhavov. And it has the same performance as DE on Dropwave. It is better than DE, PSO on Rastrigin and worse than GA. It is better than GA but worse than PSO, DE on Branin. In total, it is better than PSO on 6, GA on 12, DE on 5 of these 14 functions. So, MBOA has better performance than GA on these functions and is competitive with PSO, DE on these functions.

Table 4. Comparison results of PSO and MBOA

No.		*PSO*	*MBOA*
1	Mean	0	0
	Std	0	0
2	Mean	0	0
	Std	0	0
3	Mean	0	0
	Std	0	0
4	Mean	0.00115659	2.7511e-05
	Std	0.000276	3.8863e-05
5	Mean	-1	-0.9966
	Std	0	0.0037
7	Mean	0	0
	Std	0	0
8	Mean	91.8336	0
	Std	50.8704	0
9	Mean	403.7601	0
	Std	886.0659	0
10	Mean	2.3695e+06	0
	Std	8.6174e+06	0
11	Mean	43.9771369	1.3281e-10
	Std	11.728676	1.5040e-11
12	Mean	0.3978	0.3984
	Std	0	4.9229e-04
13	Mean	-0.8473	-1
	Std	0.1643	0
14	Mean	0	0
	Std	0	0

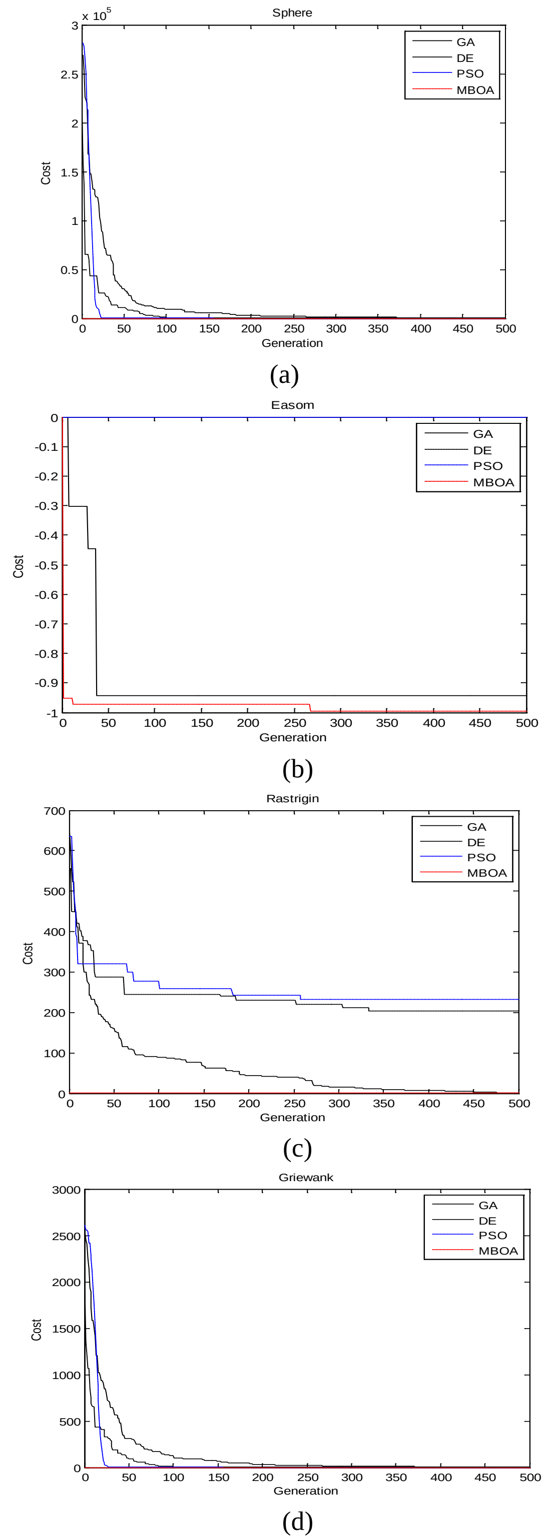

Figure 1. Comparison on convergence of the four algorithms.

In **Figure 1**, the performance of convergence of the four algorithms on four functions is shown as examples.

Copyright © 2012 SciRes.

The four functions have difference characteristics as shown in Table1. We can see that MBOA converges much faster than PSO, DE and GA.

4. Conclusions

In this paper, a new nature inspired computing method-Magnetotactic Bacteria Optimization Algorithm is researched. It adopts the principles of energy and moment of magnetosomes in magnetotactic bacteria to produce optimal solution for engineering problems. It has simple procedure and is easy to implement. The experimental results show that it is effective in solving optimization problems and is competitive with the compared classical algorithms PSO and DE. And it converges faster than PSO, DE and GA. It shows competitive performance with some classical algorithms, such as GA, DE, PSO. In future, it needs to be analyzed in theory and improved its performance for solving more complex problems.

5. Acknowledgements

This work is supported by the National Natural Science Foundation of China under Grant No.61075113, the Excellent Youth Foundation of Heilongjiang Province of ChinaNo.JC201212 and the Fundamental Research Funds for the Central Universities No.HEUCFZ1209.

REFERENCES

[1] L. N. De Castro, F. J. Von Zuben. "Learning and Optimization Using the Clonal Selection Principle," *IEEE Trans. on Evolutionary Computation*, Vol.6,No.3, pp.239–251, 2002. doi: 10.1109/TEVC.2002.1011539

[2] M. Dorigo, V. Manianiezzo, A. Colorni. "The Ant System: Optimization by a Colony of Cooperating Agents," *IEEE Trans.Sys. Man and Cybernetics*, Vol.26, No.1,pp. 1-13. doi:10.1109/3477.484436

[3] R. E. Dunin-Borkowski, M. R. McCartney, R. B. Frankel, D. A. Bazylinski, M. Posfai, P. R Buseck. "Magnetic Microstructure of Magnetotactic Bacteria by Electron Holography," *Science*, Vol. 282, 1998, pp.1868-1870. doi:10.1126/science.282.5395.1868

[4] D. Karaboga, B. Akay. "A Comparative Study of Artificial Bee Colony Algorithm," *Applied Mathematics and Computation*, Vol 214, No.1,2009, pp.108–132. doi: 10.1016/j.amc.2009.03.090

[5] J. Kennedy, R. Eberhart. Particle swarm optimization. *IEEE Int Conf on Neural Networks*. Piscataway, NJ, 1995,pp.1942-1948. doi:10.1109/ICNN.1995.488968

[6] S. Müeller, J. Marchetto, S. Airaghi, P. Koumoutsakos. "Optimization Based on Bacterial Chemotaxis," *IEEE Trans on Evolutionary Computation*, Vol.6, No.1,2002, pp.16-29. doi:10.1109/4235.985689

[7] A. P. Philipse, D. Maas. "Magnetic Colloids from Magnetotactic Bacteria: Chain Formation and Colloidal Stability," Langmuir, Vol.18, 2002,pp.9977-9984. doi:10.1021/la0205811

[8] D. Simon. "Biogeography-based Optimization," *IEEE Trans on Evolutionary Computation*, Vol. 12, 2008, pp.702-713.doi:10.1109/TEVC.2008.919004

[9] Mo Hongwei, "Research on Magnetotactic Bacteria Optimization Algorithm," *The Fifth International Conference on Advanced Computational Intelligence*,Nanjing,Oct,2012,pp.417-422.

[10] M. H. N. Tayarani, T. Akbarzadeh. "Magnetic Optimization Algorithms A New Synthesis," *in Proc. of IEEE Congress on Evolutionary Computation*. Hong Kong, China, 1-6,June,2008,pp. 2659-2665. doi:10.1109/CEC.2008.4631155

[11] M. Winklhofer, L. G. Abraçado, A. F. Davila, C. N. Keim, H. G. Lins de Barros. P. "Magnetic Optimization in A Multicellular Magnetotactic Organism," *Biophysical Journal*, Vol 92, 2007,pp. 661-670. doi:10.1529/biophysj.106.093823

[12] V. Tereshko. "Reaction–diffusion Model of A Honeybee Colony's Foraging Behaviour," *in Parallel Problem Solving from Nature VI, Lecture Notes in Computer Science*, Vol. 1917, Springer–Verlag, Berlin, 2000,pp. 807–816. doi:10.1007/3-540-45356-3_79

Copyright © 2012 SciRes.