Adaptive Bacterial Foraging Optimization Algorithm Based on Social Foraging Strategy

Hai Shen
College of Physics Science and Technology, Shenyang Normal University, Shenyang, China

Yunlong Zhu
3Laboratory of Information Service and Intelligent Control, Shenyang Institute of Automation, Chinese Academy of Sciences Shenyang, Shenyang, China

***Abstract*—In 2002, K. M. Passino proposed Bacterial Foraging Optimization Algorithm (BFOA) for distributed optimization and control. Biologic foraging strategies are diverse. Based on social and intelligent foraging theory, this paper proposed an adaptive bacterial foraging optimization algorithm, and introduced six foraging operators: chaos run operator, assimilation run operator, tumble operator, swimming operator, reproduction operator and elimination-dispersal operator. Among those operators, chaos run operator, assimilation run operator and reproduction operator were redefined in accordance with social foraging strategy. And others were same with the original algorithm. Experiments were conducted on 10 multimodal unconstrained benchmark optimization problems for demonstration the effectiveness and stability. The results demonstrate remarkable performance of the proposed algorithm on all chosen benchmark functions when compared to several successful optimization techniques.**

***Index Terms*—Bacterial Foraging Optimization Algorithm; Social Foraging; Multimodal Numerical Optimization**

I. INTRODUCTION

Once an organism begins its life, it immediately faces survival needs. For this purpose, most biology requires food, water, sunlight, minerals, and oxygen to survive and grow. They get these resources in many different ways. The behavior of get resources is called foraging [1, 2]. In the process of foraging, the choice of foraging strategy is essential. A better foraging strategy makes forager get more resources in the shortest time, then forager would has enough nutrition to survive or breed the next generation. Conversely, if a forager selects the failing strategy continuously, and don't gain nutrition resources, meanwhile its energy was also consumed, then it will be eliminated gradually according to the natural selection theory. Biologic foraging strategies are diverse. From the sociability point of view, foraging strategies were divided into two categories: non-social foraging (namely independent foraging or individual foraging, the later statement was adopted in this paper) and social foraging [3].

A. Individual Foraging

In the process of foraging, forager according to its own energy and its own foraging way to foraging, and needn't the help come from other members within the same population. Biology have strong survival instincts, which would indicate how organism foraging. Some animals are "cruise" searchers, some animals are "salutatory" searchers, and others are "ambush" searchers and so on. To envision this strategy, consider Figure 1.

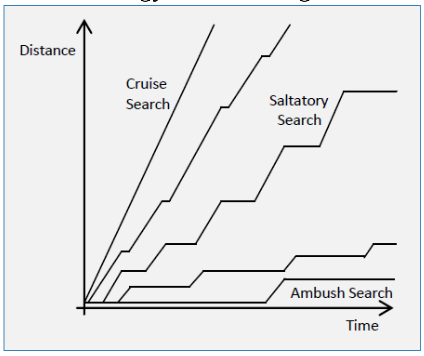


Figure 1. Animal foraging strategies

B. Social Foraging

In the process of social foraging, individual forage is based on not only the resource availability but also the other forager's behavior. Much biology lives together in relationships where one depends upon the other. Individuals of population would find a way to share the natural resources, allowing both to survive. This interdependent relationship is known as mutualism. For example, some sparrows living together form a small group, when the member A know the best foraging position, which was found by member B, A would follow the foraging route of member B to foraging.

In 2002, inspired by the researches on the foraging behavior of E. coli bacteria, Prof. K. M. Passino proposed Bacterial Foraging Optimization Algorithm (BFOA) [4], which also has been applied to many engineering problems [5-9]. In the foraging process, if bacteria find no better food in the original direction, it will turn to a new direction. This process is defined as tumble action. After a tumble behavior, if the fitness value is improved,

doi:10.4304/jnw.9.3.799-806

the bacteria will continue to move in the same direction a number of steps until no improvement in fitness value, or reaches a predetermined threshold number of moving steps. This process is defined as the run.

In fact, the foraging behavior of bacterial has social characteristic. Bacterial can continual to feel the changes in the surrounding environment, and can use past experience to find the most advantages [10-13]. So, this paper proposed an adaptive bacterial foraging optimization algorithm with social foraging strategy. The rest of this paper is organized as follows. Section 2 describes the classical BFOA. Section 3 describes the proposed adaptive bacterial foraging optimization algorithm technique. Sections 4 and 5 present and discuss computational results. The last section draws conclusions and gives directions of future work.

II. THE BACTERIAL FORAGING OPTIMIZATION ALGORITHM

A. Chemotaxis

The motion patterns that the bacteria will generate in the presence of chemical attractants and repellents are called chemotaxes. For E. coli, this process was simulated by two different moving ways: run or tumble. A Bacterium alternates between these two modes of operation its entire lifetime. The bacterium sometimes tumbles after a tumble or tumbles after a run. This alternation between the two modes will move the bacterium, and this enables it to "search" for nutrients. Suppose $\theta^i(j,k,l)$ represent the position of the each member in the population of S bacterial at the *j*th chemotactic step, and *k*th reproduction step, and *l*th elimination, then the movement of bacterium may be presented by:

$$\theta^i(j+1,k,l)=\theta^i(j,k,l)+C(i)\varphi(j)$$

$$\varphi(j)=\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}$$

where $C(i)(i=1,2,\cdots,S)$ is the size of the step taken in the random direction specified by the tumble. $\varphi(j)$ was used to define the random direction of movement after a tumble. $J(i,j,k,l)$ is the fitness, which also denote the cost at the location of the *i*th bacterium $\theta^i(j,k,l)\in R^n$. If at $\theta^i(j+1,k,l)$ the cost $J(i,j+1,k,l)$ is better (lower) than at $\theta^i(j,k,l)$, then another step of size $C(i)$ in this same direction will be taken. Otherwise, bacteria will tumble via taking another step of size $C(i)$ in random direction $\varphi(j)$ in order to seek better nutrient environment.

B. Swarming

An interesting group behavior has been observed for several motile species of bacteria including E.coli and S. typhimurium. When a group of E. coli cells is placed in the center of a semisolid agar with a single nutrient chemo-effector, they move out from the center in a traveling ring of cells by moving up the nutrient gradient created by consumption of the nutrient by the group. To achieve this, function to model the cell-to-cell signaling via an attractant and a repellant. The mathematical representation for E.coli swarming can be represented by:

$$J_{cc}(\theta,P(j,k,l))=\sum_{i=1}^{S}J_{cc}(\theta,\theta^i(j,k,l))=$$
$$\sum_{i=1}^{S}\left[-d_{attract}\exp(-w_{attract}\sum_{m=1}^{D}(\theta_m-\theta_m^i)^2)\right]$$
$$+\sum_{i=1}^{S}\left[-h_{repellant}\exp(-w_{repellant}\sum_{m=1}^{D}(\theta_m-\theta_m^i)^2)\right]$$

where $J_{cc}(\theta,P(j,k,l))$ is the cost function value to be added to the actual cost function. S is the total number of bacteria and p is the number of parameters to be optimized which are present in each bacterium. $d_{attract}$ is the depth of the attractant released by the cell and $w_{attract}$ is a measure of the width of the attractant signal. $h_{repellant}=d_{attract}$ is the height of the repellant effect and $w_{repellant}$ is a measure of the width of the repellant. After a chemotaxes behavior, the new fitness function value is as follows:

$$J(i,j+1,k,l)=J(i,j+1,k,l)$$
$$+J_{cc}(\theta^i(j+1,k,l),P(j+1,k,l)).$$

C. Reproduction

According to the rules of evolution, individual will reproduce themselves in appropriate conditions in a certain way. For bacterial, a reproduction step takes place after all chemotactic steps.

$$J_{health}^i=\sum_{j=1}^{Nc+1}J(i,j,k,l)$$

where J_{health}^i is the health of bacterium i. Sort bacteria and chemotactic parameters $C(i)$. For keep a constant population size, bacteria with the highest J_{health} values die. The remaining bacteria are allowed to split into two bacteria in the same place.

D. Elimination-Dispersal

In the evolutionary process, elimination and dispersal events can occur such that bacteria in a region are killed or a group is dispersed into a new part of the environment due to some influence. They have the effect of possibly destroying chemotactic progress, but they also have the effect of assisting in chemotaxis, since dispersal may place bacteria near good food sources. From the evolutionary point of view, elimination and dispersal was used to guarantees diversity of individuals and to strengthen the ability of global optimization. In BFOA, bacteria are eliminated with a probability of P_{ed}. In order to keeping the number of bacteria in the population constant, if a bacterium is eliminated, simply disperse one to a random location on the optimization domain.

III. ADAPTIVE FORAGING BACTERIAL FORAGING OPTIMIZATION ALGORITHM

In E. coli bacterium chemotaxis behavior, it can move in two different ways; it can run or it can tumble. First, tumble behavior is equal to individual foraging strategy. So, this proposed algorithm defined a tumble operator, which performs the same operation with the original algorithm. However running behavior becomes more complicated. So, based on social foraging strategy, this paper redefined run behavior, and introduced two run operators, named chaos run operator and assimilation run operator, respectively. For reproduction behavior, proposed algorithm redesigned this behavior and defined reproduction operator. Beyond that, proposed algorithm defined swimming operator and elimination-dispersal operator, which also performs the same operation (swimming and elimination-dispersal) with the original algorithm.

Next, the paper will introduce the redesigned three operators: chaos run operator, assimilation run operator and reproduction operator.

A. Chaos Run Operator

The optimal individual of population selects chaos foraging strategy. Since the optimal forager in the current iteration possess the greatest energy, so he has the ability of seeking the better location which with more nutrient resources in global scope. And the seeking mode taken by optimal foraging individual is not the same as the migration method of non-optimal individual, and also is not a simple migration or position moving, but a rather powerful foraging strategy, such as chaos search. Since 1970's, a large number of biologic model simulation explained that the chaos is widespread exist in biologic systems. Chaos process seems confusion, in fact, it is not completely disorder, but exist subtle regularity inherent. It has ergodicity, randomness and regularity and the others feature [14]. Chaos motion should traversal all states which were not repeated in the way of its own "law" in a certain area.

Chaos run operator which was employed by the optimal individual performs chaos search strategy. The basic idea is introducing logistic map to optimization variables using a similar approach to carrier, and generate s set of chaotic variables, which can show chaotic state [15, 16]. Simultaneously, enlarge the traversal range of chaotic motion to the value range of the optimization variables. Finally, the better solution was found directly using chaos variable.

(1) The current optimization variable is denoted by X_0, and its fitness value is $f(X_0)$.

(2) Generating n chaotic variables $(X_1, X_2, \cdots, X_n)$ by logistic mapping:

$$X_{i+1} = 4X_i(1-X_i)\ ,\quad i = 0, 1, 2, \cdots, n-1$$

(3) Transform the chaotic motion traverse range to optimize variable domain.

$$X_i = B_{lo} + (B_{up} - B_{lo})X_i\ ,\quad i = 1, 2, \cdots, n$$

where B_{up} and B_{lo} is the upper and lower boundary of the search space.

(4) Computing fitness values of n chaotic variables.

(5) If $f(X_i) > f(X_0)$, then $X_0 \Leftarrow X_i$, $f(X_0) \Leftarrow f(X_i)$

B. Assimilation Run Operator

Individuals of selecting assimilation run foraging strategy will gain resource directly from the optimal individual in the way of using a random step towards the optimal individual. Assimilation run operator performs the following operation.

$$X_{i+1} = X_i + r_1(X_p - X_i)$$

where $r_1 \in \mathrm{R}^n$ is a uniform random sequence in the range (0,1). X_p is the best individual of the current population. X_i is the position of an individual who perform assimilation operator and X_{i+1} is the next position of this individual.

C. Reproduction Operator

Reproduction operator selects single-point crossover method. One crossover point is selected, string from beginning of individual to the crossover point is copied from one parent, and the rest is copied from the second parent. According to "the survival of the fittest" theory, and for ensuring a fixed population size, proposed algorithm performs elitist selection strategy. A number of individuals with the best fitness values are chosen to pass to the next generation.

Figure 2 is the adaptive foraging bacterial foraging optimization algorithm flowchart.

IV. EXPERIMENTS SETTING

A. Illustrative Examples

To fully evaluate the performance of the proposed algorithm without a biased, we employed 10 multimodal benchmark functions which were tested widely in evolutionary computation domain to show the quality solution and the convergence rate [17]. In those functions, functions f_1 to f_7 are uni-modal functions; functions f_8 to f_{13} are multi-modal functions with many local minima; functions f_{14} to f_{23} are multimodal functions with few local minima. In this paper, functions f_{14} to f_{23} was used to test proposed algorithm performance. And these benchmark functions and related parameters were given in Table 1 and Appendix A, respectively.

B. Settings for Involved Algorithms

We compared the optimization performance of proposed algorithm with the well-known algorithm: the standard PSO, and the standard GA and Group Search Optimizer (GSO) [18]. The parameter settings of every algorithm were manually tuned. Each of the experiments was repeated 30 times, and the max iterations in a run $T_{\max}$ =3000. In every run, with the purpose of making the comparison fairly, the initialization populations for all the considered algorithms were generated using the same population which satisfied the normal distribution. The same population size S=50. The other specific settings for each of the algorithms are described below.

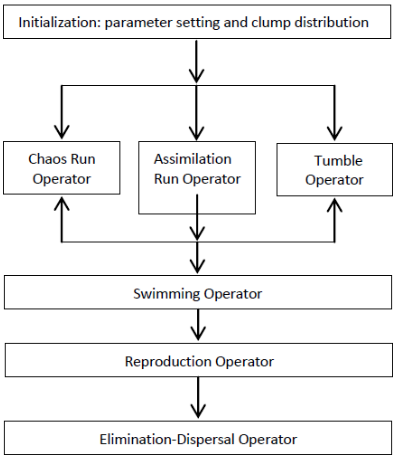


Figure 2. Adaptive bacterial foraging optimization algorithm flowchart

1) Adaptive BFOA Settings:

In this algorithm, the probability using for decision the individual's foraging strategy $P_f = 0.1$; the number of chaos variables $S_c = 100$; crossover probability $P_c = 0.7$; mutation probability $P_m = 0.02$.

2) PSO Settings:

In PSO, the acceleration factors c_1=1, and c_2=1.49; and a decaying inertia weight *w* starting at 0.9 and ending at 0.4 was used.

3) GA Settings:

In GA, crossover probability P_c and mutation probability P_m is respectively 0.7 and 0.05; selection operation is roulette wheel method.

4) GSO Settings:

The parameter setting of the GSO algorithm can refer to references 18. The initial head angle φ^0 of each individual is set to be $\pi/4$. The constant a is given by $round(\sqrt{n+1})$ where n is the dimension of the search space. The maximum pursuit angle $\theta_{\max}$ is π/a^2. The maximum turning angle a is set to be $\pi/2a^2$. The maximum pursuit distance $l_{\max}$ is calculated from:

$$l_{\max} = \|U_i - L_i\| = \sqrt{\sum_{i=1}^{n}(U_i - L_i)^2}$$

where L_i and U_i are the lower and upper bounds for the i_{th} dimension. The percentage of dispersed members is 20%.

V. Results and Discussion

The experimental results were recorded in two categories: the "Mean Best" and the "Std", which were widely used for testing and comparing the performance of optimization algorithm owing to their meaning, and they will be discussed detailed in this section. Moreover, in order to be more intuitive compare and analysis of performance of four algorithms, the convergence results or performance analysis for selected benchmark problems were attached in this paper.

A number of experimental data come from printed research papers have shown that PSO and GA can find the optimum of some functions. But in this paper, it becomes powerless. The cause is that the way of generating initialization population is changed, from random distribution method into clumped distribution method. In a sense, the clumped distribution is the special form of random distribution. But as stated before, random distribution is rare in reality, clumped distribution is the commonest. So the finally optimum solution generated via initialization population of random distribution can't be applied to illustrate the algorithms perform for solving reality and complex optimization problems.

TABLE I. Benchmark functions.

Test Functions	N	SD	$f_{\min}$
$f_1(x)=\left[\frac{1}{500}+\sum_{j=1}^{25}\frac{1}{j+\sum_{j=1}^{2}(x_i-a_{ij})^6}\right]^{-1}$	2	[-65.536, 65.536]	0
$f_2(x)=\sum_{i=1}^{11}\left[a_i-\frac{x_1(b_i^2+b_ix_2)}{b_i^2+b_ix_3+x_4}\right]^2$	4	[-5, 5]	0.0003075
$f_3(x)=4x_1^2-2.1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_2^4$	2	[-5, 5]	-1.0316
$f_4(x)=\left(x_2-\frac{5.1}{4\pi^2}x_1^2+\frac{5}{\pi}x_1-6\right)^2+10\left(1-\frac{1}{8\pi}\right)\cos x_1+10$	2	[-5, 10] [0, 15]	0.398
$f_5(x)=[1+(x_1+x_2+1)^2(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2)]$ $\times[30+(2x_1+1-3x_2)^2(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2)]$	2	[-2, 2]	3
$f_6(x)=-\sum_{i=1}^{4}\exp\left[-\sum_{j=1}^{3}a_{ij}(x_j-p_{ij})^2\right]$	3	[0, 1]	-3.86
$f_7(x)=-\sum_{i=1}^{4}\exp\left[-\sum_{j=1}^{6}a_{ij}(x_j-p_{ij})^2\right]$	6	[0, 1]	-3.32
$f_8(x)=-\sum_{i=1}^{5}\left[(x_i-a_i)(x_i-a_i)^T+c_i\right]^{-1}$	4	[0, 10]	-10
$f_9(x)=-\sum_{i=1}^{7}\left[(x_i-a_i)(x_i-a_i)^T+c_i\right]^{-1}$	4	[0, 10]	-10
$f_{10}(x)=-\sum_{i=1}^{10}\left[(x_i-a_i)(x_i-a_i)^T+c_i\right]^{-1}$	4	[0, 10]	-10

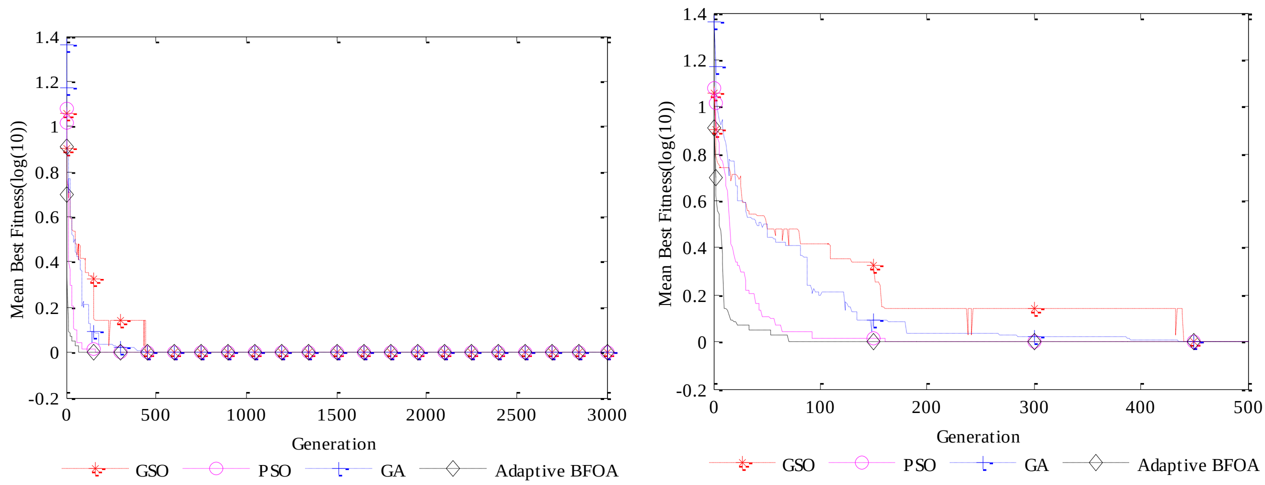
These functions f_1 to f_{10} is multimodal functions with few local minima, and possess rather unique features, which can verify the adaptation of algorithms to the different optimization environment. Tables 2 present all optimization results. For these ten functions, in terms of testing the indicators, adaptive foraging BFOA ranked first on all functions. It can be concluded that the order of the search performance of these four algorithms is adaptive BFOA >GSO > PSO > GA.

Function f_1 and f_3 are all easy problems, and all algorithms can find the exact optimum solutions. On functions f_5 and f_6, adaptive BFOA, GSO and PSO yield the exact optimum while GA yielded the approximate optimum. All algorithms come very close to the global optimum on f_7. Figure 3 (a) and Figure 4 (a) shows the convergence results on function f_1 and f_6. Moreover, we can see that adaptive foraging BFOA has the fastest convergence speed from Figure 3 (b) and Figure 4 (b). Figure 5 and Figure 6 also shows the best convergence results on function f_2 and f_5.

For example, the problem f_8 shown in Figure 7 (a), have five extreme, the bottom point at the deepest hole is the global optimal position and the other holes are deceptive. Figure 7 (b) shows convergence results of four algorithms. All algorithms have quickly in the early iterations. But the GSO, PSO and GA stagnate before finding the global optimum, and proposed algorithm stagnates until finding it. At the beginning of the searching, there is a number of promising "fox holes", so the convergence rate of these algorithms is fast. But after a short period, owing to lacking the ability of jumping out of the local extreme, the solutions obtain by GSO, PSO and GA fall into the "fox holes" deeply, the evolutionary curve tend to stop. The optimization tactics make proposed algorithm can escape from the deceptive region, and migrate towards the global one. The properties of function f_9 and f_{10} are similar to function f_8. Figure 8 shows the same convergence results on function f_9 and f_{10}.

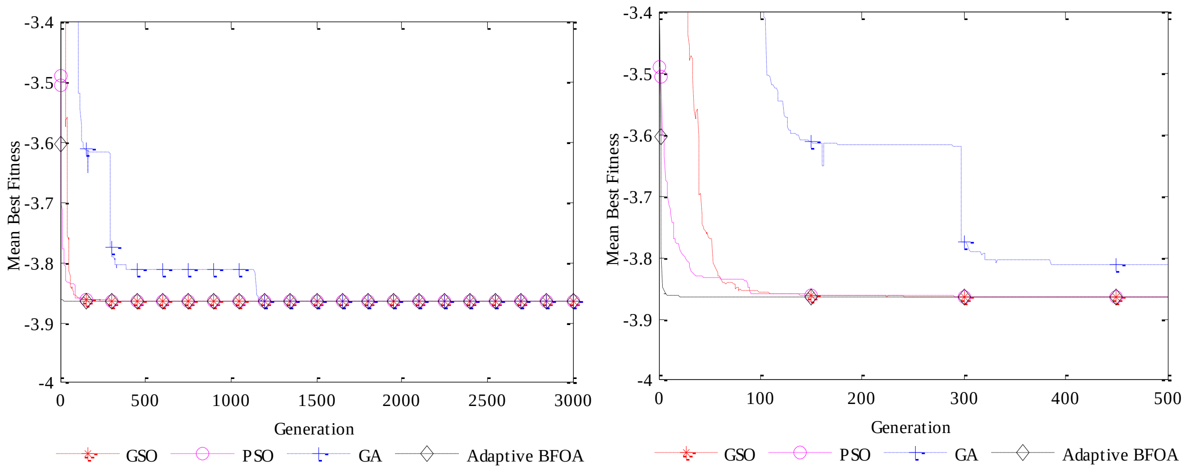
TABLE II. RESULTS FOR ALL ALGORITHMS ON MULTIMODAL FUNCTIONS WITH FEW LOCAL MINIMA.

	Adaptive BFOA		GSO		PSO		GA	
Fun. No (f_{min})	Mean Best (rank)	Std (rank)	Mean Best (rank)	Std (rank)	Mean Best (rank)	Std (rank)	Mean Best (rank)	Std (rank)
1	9.98E-01	8.56E-13	9.98E-01	2.26E-16	9.98E-01	1.13E-16	9.98E-01	2.80E-07
(1)	(1)	(2)	(1)	(1)	(1)	(1)	(1)	(3)
2	5.61E-03	3.37E-03	5.72E-03	6.12E-03	1.09E-02	1.08E-02	2.16E-02	1.40E-02
(3.075E-4)	(1)	(1)	(2)	(2)	(3)	(3)	(4)	(4)
3	-1.03E+00	1.04E-08	-1.03E+00	5.07E-16	-1.03E+00	6.78E-16	-1.03E+00	1.19E-04
(-1.0316)	(1)	(2)	(1)	(1)	(1)	(1)	(1)	(3)
4	0.3984	1.20E-02	0.4012	1.90E-02	0.4396	3.73E-02	0.4001	6.80E-02
(0.398)	(1)	(1)	(3)	(2)	(4)	(3)	(2)	(4)
5	3.00E+00	1.42E-07	3.00E+00	6.35E-14	3.00E+00	1.33E-15	3.91E+00	4.94E+00
(3)	(1)	(3)	(1)	(1)	(1)	(2)	(2)	(4)
6	-3.86E+00	2.75E-09	-3.86E+00	1.71E-15	-3.86E+00	1.44E-03	-3.84E+00	1.41E-01
(-3.86)	(1)	(2)	(1)	(1)	(1)	(3)	(2)	(4)
7	-3.27E+00	5.94E-02	-3.29E+00	5.56E-02	-3.20E+00	1.61E-01	-3.29E+00	5.56E-02
(-3.32)	(2)	(2)	(1)	(1)	(3)	(3)	(1)	(1)
8	-1.02E+01	2.90E-07	-6.45E+00	3.01E+00	-7.11E+00	2.99E+00	-6.95E+00	3.17E+00
(-10)	(1)	(1)	(4)	(2)	(2)	(3)	(3)	(4)
9	-9.96E+00	1.69E+00	-7.24E+00	3.09E+00	-8.45E+00	2.61E+00	-6.05E+00	3.00E+00
(-10)	(1)	(1)	(3)	(4)	(2)	(2)	(4)	(3)
10	-9.64E+00	2.32E+00	-6.70E+00	3.29E+00	-8.39E+00	2.93E+00	-6.23E+00	3.21E+00
(-10)	(1)	(1)	(3)	(4)	(2)	(2)	(4)	(3)
Total rank	1	1	2	2	2	3	3	4



(a) (b)

Figure 3. Convergence results of function f_1



(a) (b)

Figure 4. Convergence results of function f6

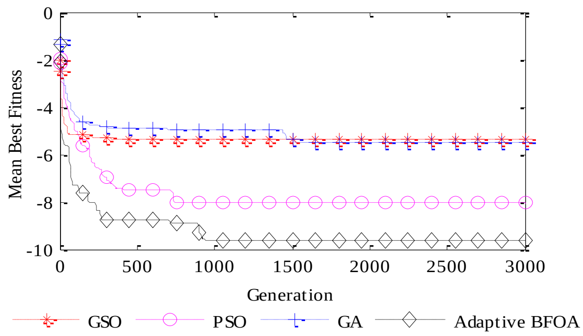


Figure 5. Convergence results of function f_2

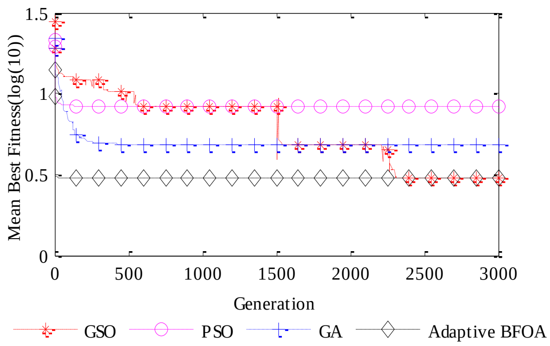
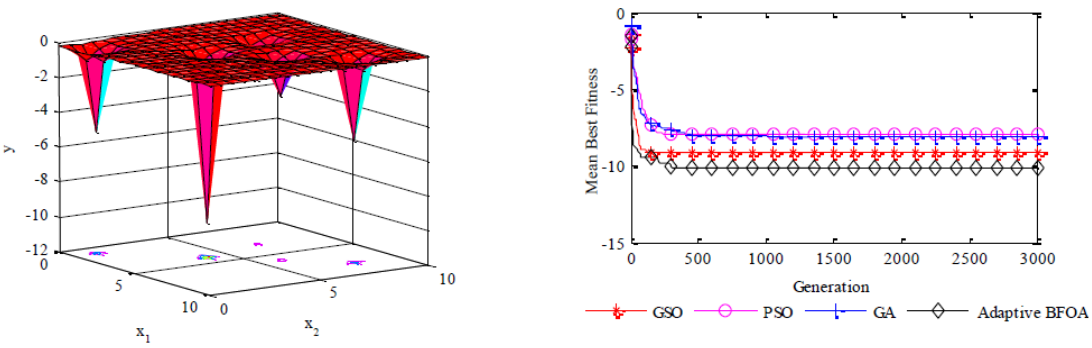
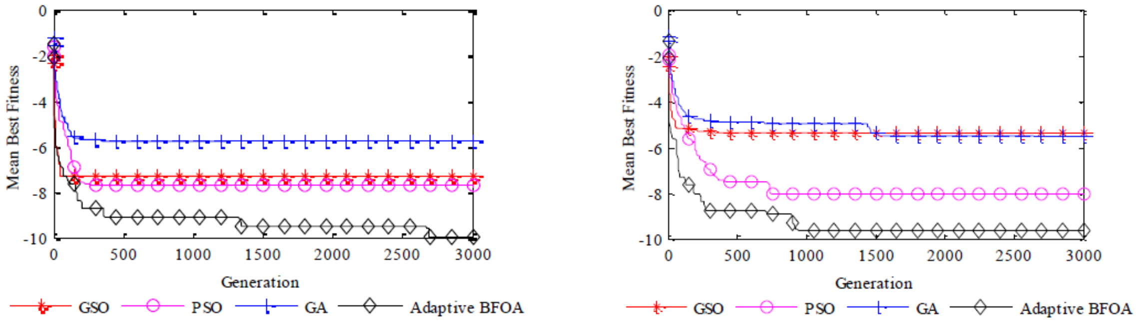


Figure 6. Convergence results of function f_5



(a) (b)

Figure 7. Function f_8. (a) Function graphs with a dimension of 2, (b) Convergence results of all algorithms



(a) f_9 (b) f_{10}

Figure 8. Convergence results of function f_9 and f_{10}

VI. CONCLUSIONS

In this paper, we explain the biology foraging behavior and a variety of individual foraging behavior and social foraging behavior. Next, bacterial adaptive foraging strategy is presented. Based on classical bacterial foraging optimization algorithm, this algorithm introduced three redesigned behaviors, which also were redefined optimization operators, including chaos run operator, assimilation run operator and reproduction operator. To illustrate its operation, 10 multimodal benchmark functions was used to test proposed algorithm, and two performance metrics which include solution quality and speed of convergence were used. The adaptive BFOA with social foraging strategy was shown to provide better results than standard PSO, GA and GSO for all of the tested problems. As part of our future work, adaptive BFOA also could be studied and tested on real-world problems, such as mechanical design problem of mechanical engineering, location problem of manufacturing systems, network routing problem of computer engineering, parameter identification problem of industrial engineering, electrical engineering problem, aerospace engineering problem, and bioengineering problem and so on.

APPENDIX A

f_1: $a_{ij} = \begin{pmatrix} -32, & -16,0,16,32,\ldots,-32,-16,0,16,32 \\ & -32,\ldots,-16,\ldots,0,\ldots16,\ldots32,\ldots \end{pmatrix}$

f_2:

$a = (0.1957.0.1947, 0.1735, 0.1600, 0.0844, 0.6027, 0.0456, 0.0342, 0.0323, 0.0235, 0.0246)$

$b = (\frac{1}{0.25}, \frac{1}{0.6}, \frac{1}{1}, \frac{1}{2}, \frac{1}{4}, \frac{1}{6}, \frac{1}{8}, \frac{1}{10}, \frac{1}{12}, \frac{1}{14}, \frac{1}{16})$

f_6:

i	a_{ij} , $j=1,\cdots,6$						c_i
1	10	3	17	3.5	1.7	8	1
2	0.05	10	17	0.1	8	14	1.2
3	3	3.5	1.7	10	17	8	3
4	17	8	0.05	10	0.1	14	3.2

i	p_{ij} , $j=1,\cdots,6$					
1	0.1312	0.1696	0.5569	0.0124	0.8283	0.5886
2	0.2329	0.4135	0.8307	0.3736	0.1004	0.9991
3	0.2348	0.1415	0.3522	0.2883	0.3047	0.6650
4	0.4047	0.8828	0.8732	0.5743	0.1091	0.0381

f_7:

i	a_{ij} , $j=1,2,3$			c_i	p_{ij} , $j=1,2,3$		
1	3	10	30	1	0.3689	0.1170	0.2673
2	0.1	10	35	1.2	0.4699	0.4387	0.7470
3	3	10	30	3	0.1091	0.8732	0.5547
4	0.1	10	35	3.2	0.03815	0.5743	0.8828

$f_{8\sim10}$:

i	a_{ij} , $j=1,\cdots,4$				c_i
1	4.0	4.0	4.0	4.0	0.1
2	1.0	1.0	1.0	1.0	0.2
3	8.0	8.0	8.0	8.0	0.2
4	6.0	6.0	6.0	6.0	0.4
5	3.0	7.0	3.0	7.0	0.4
6	2.0	9.0	2.0	9.0	0.6
7	5.0	5.0	3.0	3.0	0.3
8	8.0	1.0	8.0	1.0	0.7
9	6.0	2.0	6.0	2.0	0.5
10	7.0	3.6	7.0	3.6	0.5

ACKNOWLEDGMENT

This project is supported by the National Natural Science Foundation of China (Grant Nos. 61174164, 51205389, 61105067, 71001072, 71271140, 71240015), and the National Natural Science Foundation of Liaoning province of China (Grant No. 201102200), and the Natural Science Foundation of Guangdong Province (Grant Nos. S2012010008668, 9451806001002294).

REFERENCES

[1] D. Stephens and J. Krebs. Foraging theory. Princeton, NJ: Princeton Univ. Press, 1986.

[2] W. O'Brien, H. Browman, and B. Evans. Search strategies of foraging animals. *Amer. Scientist*, 1990, 78:152-160.

[3] K. M. Passino. Biomimicry for optimization, control, and automation, *Springer-Verlag*, 2005.

[4] K. M. Passino. Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Systems Magazine*, 2002, 22: 52–67.

[5] R. Majhi, G. Panda, and G. Sahoo et al. Stock market prediction of S & P 500 and DJIA using bacterial foraging optimization technique. *In the Proceedings of IEEE Congress on Evolutionary Computation*, 2007: 2569–2575.

[6] S. Mishra and C. N. Bhende. Bacterial foraging technique-based optimized active power filters for load compensation. *IEEE Transactions on Power Delivery*, 2007, 22(2): 457–465.

[7] L. Ulagammai, P. Vankatesh, and P. S. Kannan et al. Application of bacteria foraging technique trained and

artificial and wavelet neural networks in load forecasting. Neurocomputing, 2007, 70(16-18): 2659–2667.
[8] D. H. Kim and J. H. Cho. Adaptive tuning of PID controller for multivariable system using bacterial foraging based optimization. *Lecture Notes in Computer Science*, 2005, 3528:231–235..
[9] B. Niu, Y. l. Zhu, and X. X. He et al. Optimum design of PID controllers using only a germ of intelligence. *In the Proceedings of the World Congress on Intelligent Control and Automation,* 2006:3584–3588.
[10] D. E. Koshland, Jr, Bacterial chemotaxis as a model behavioral system. New York: Raven, 1980.
[11] J. M. Lackie and P. C. Wilkinson. Biology of the Chemotactic Reponse. Cambridge, U.K.: Cambridge Univ. Press, 1981.
[12] J. M. Lackie. Cell movement and cell behavior. *London, U.K.: Allen and Unwin,* 1986.
[13] D. G. Davies, M. R. Parsek and J. P. Pearson, et al. The involvement of cell-to-cell signals in the development of a bacterial biofilm. *Science,* 1998, 280: 295–298.
[14] E. N. Lorenz. Deterministic non-periodic flow. *Journal of the Atmospheric Sciences*, 1963, 20: 130–141.
[15] P. F. Verhulst. Recherches mathématiques sur la loi d'accroissement de la population. Nouv. mém. de l'Academie Royale des Sci. et Belles-Lettres de Bruxelles 18, 1845: 1-41.
[16] P. F. Verhulst. Deuxième mémoire sur la loi d'accroissement de la population. Mém. de l'Academie Royale des Sci., des Lettres et des Beaux-Arts de Belgique,1847, 20:1-32.
[17] X. Yao, Y. Liu and G. M. Lin .Evolutionary programming made faster. *IEEE Transactions on Evolutionary Computation,* 1999, 3(2): 82–102.
[18] S. He, Q. H. Wu and J. R. Saunders .A novel group search optimizer inspired by animal behavioral ecology. *In the Proceedings of IEEE congress on evolutionary computation,* 2006: 1272–1278.

Hai Shen was born in China in 1976. She has received the B.Sc. and M. Eng. Degrees in computer applications technology, Shenyang University of Technology, Shenyang, China, in 1998 and 2005, respectively. She is currently working toward the Ph.D. degree with the Shenyang Institute of Automation, Chinese Academy of Sciences, China. Now, she is an Associate Professor with the College of Physics Science and Technology, Shenyang Normal University, China. Her main research interests are in advanced computational intelligence with a focus on: nature inspired hybrid intelligent systems, swarm intelligence, evolutionary computation, distributed artificial intelligence, multi-agent systems and other heuristics swarm intelligence.

Yunlong Zhu is the director of the Key Lab. of Advanced Manufacturing Technology, Shenyang Institute of Automation of the Chinese Academy of Sciences. He received his Ph.D. in 2005 from the Chinese Academy of Sciences, China. He has research interests in various aspects of Enterprise Information Management but he has ongoing interests in Artifical Intelligence, Machine Learning, and related areas. Prof Zhu's research has led to a dozen professional a publication in these areas includes the biography here.