A superior attraction bacterial foraging optimizer for global optimization

XIANGHUA CHU
Shenzhen Graduate School
Harbin Institute of Technology
Shenzhen 518055
CHINA
xianghua.chu@gmail.com

BEN NIU
College of Management
Shenzhen University
Shenzhen 518060
CHINA
drniuben@gmail.com

QIANG LU
Shenzhen Graduate School
Harbin Institute of Technology
Shenzhen 518055
CHINA
qiang.lu.home@gmail.com,

JUN DING
Shenzhen Graduate School
Harbin Institute of Technology
Shenzhen 518055
CHINA
haveding@163.com

Abstract: In order to improve the performance of basic bacterial foraging optimization (BFO) for various global optimization problems, a superior attraction bacterial foraging optimizer (SABFO) is proposed in this paper. In SABFO, a novel movement guiding technique termed as superior attraction strategy is introduced to make use of all bacteria historical experience as potential exemplars to lead individuals direction. This strategy enables the bacteria in population to exchange information and collaborate with the superior individuals to search better solutions for different dimensions. Two variants of SABFO are studied and tested on a set of sixteen benchmark functions including various properties, such as unimodal, multimodal, shifted and inseparable characteristics. Four state-of-the-art evolutionary algorithms are adopted for comparison. Experimental study demonstrates remarkable improvement of the proposed algorithm for global optimization problems in terms of solution accuracy and convergence speed.

Key–Words: Global optimization; Bacterial foraging optimization; Swarm intelligence; Engineering optimization; Movement updating; Meta-heuristic; Evolutionary algorithms.

1 Introduction

Global optimization is a challenging and practical problem which exists in many real world fields, such as engineering, economics and management. Generally, global optimization problems are characterized by unimodal, multimodal, nonlinear and irregular properties. Without loss of generality, we consider a minimization problem as the maximization model could be transformed into a minimization one and vice-versa. An unconstrained global optimization can be defined as the following:

$$\min\ f(X),\ X = [x_1, x_2, \cdots, x_D] \tag{1}$$

where $f : \mathbb{R}^D \rightarrow \mathbb{R}$ denotes a real valued objective function, D represents the number of the dimensions to be solved.

As the real world optimization problems become increasingly large-scale and complicate, it is difficult for conventional and mathematical programming methods to guarantee a global optimum or even a satisfactory solution. Thus, more efficient and effective optimization techniques are always demanded.

Inspired by swarm behaviors and biological activities in natural ecosystem, swarm intelligence have been developed to address complex problems. For instance, particle swarm optimization (PSO), originally introduced by Kennedy and Eberchart [1], mimics birds flocking and fish schooling in which the individuals search for food by collaboration; Genetic algorithm (GA), first proposed by Holland [2], emulates Darwins evolutionary theory C survival of the best by performing simple operations on gene-string codes; Ant colony optimization (ACO), developed by Dorigo et al. [3], imitates ant colonies finding the shortest route between food and home. These algorithms have shown better performance on solving complex problems and been applied to many practical areas [4, 5, 6, 7].

Most recently, the chemotactic behavior of Escherichia Coli (E. Coli) bacteria living in human intestines attracts great interests due to its simple foraging pattern [8, 9], hereby presenting bacterial foraging optimization (BFO). BFO is a population-based stochastic search algorithm that mimics the chemotaxis activity of bacterial foraging behaviors. Since its introduction, BFO has been successfully applied to some engineering fields, such as power system [10], multiobjective optimization [11], optimal feeder routing [12] and so on. However, two common criticisms still exist in BFO. First, canonical BFO suffers poor performance and slow convergence rate compared to other population-based algorithms (e.g., PSO and GA) [13, 14]. BFOs performance deteriorates with the growth of dimensionality and complexity of problem.

Some BFO variants have been designed to improve its performance and tune its parameters. Pandit et al. [15] developed an improved BFO to improve computational efficiency by employing a parameter automation strategy and crossover operation. The original BFOs performance also has a definite link with chemotactic step size , but so far it is hard to determine an appropriate chemotactic step size for various problems. Chen et al. [23] introduced an adaptive foraging strategies that enable the algorithm to dynamically modify the run-length unit parameter during optimization process. Experiments on four functions indicate its improvement. A BFO variant with adaptive step size to balance exploration and exploitation is also found in [16]. Chemotaxis size with linear and nonlinear variations was discussed by Niu [17]. The novel chemotaxis strategy with communication scheme is proposed to increase information exchange among bacteria [18]. Besides, other behaviors, i.e., elimination, reproduction and migration, would be conducted only when certain given conditions are satisfied. Sarasiri et al. [19] hybridized BFO and Tabu search to enhance algorithms exploration and exploitation capabilities. Similarly, BFO is combined with PSO to form a velocity modulated BFO which has fast convergence speed, and the method is tested on five functions to justify its enhancement [20]. Genetic algorithm was integrated into BFO and the performance of the algorithm was studied with an emphasis on mutation, crossover, variation of step sizes, chemotactic steps, and the lifetime of the bacteria [21]. Verma et al. [22] incorporated probabilistic derivative approach into BFO where the direction of bacteria is determined by a probability matrix which is calculated using derivatives along the potential orientations. Two cooperative methods were applied to BFO, i.e., the serial heterogeneous cooperation on the decomposition levels of implicit space and hybrid space, respectively [23]. Although many variants have been proposed to overcome BFOs demerits, these variants are either designed for specific problems [15, 16, 22] or lack of comprehensive experiments over more benchmark functions [13, 20, 23], thus poor convergence when addressing various and complex problems is still the shortcoming of BFO. Besides, the introduction of additional strategies increase the complexity of implementing BFO [13, 23].

In order to address the aforementioned issues, i.e., poor convergence, a novel bacterial foraging optimizer, termed as superior attraction bacterial foraging optimizer (SABFO), is proposed in this paper. A superior attraction strategy is developed to enhance the search capabilities of the original BFO, which derives two SABFO variants, namely SA-ws and SA-ns. It is envisioned that SABFOs are more robust than BFO as they are applied to a diverse set of problems with various dimensions. Moreover, the proposed technique is easy to implement and even simplifying the computational complexity of original BFO. Thirdly, the overall performance, including solution accuracy and convergence speed, is improved and verified through comprehensive comparisons against several state-of-the-art evolutionary algorithms on global optimization problems.

The rest of the paper is organized as follows. Section 2 presents the framework of basic BFO and the proposed method. Experiment settings are introduced in Section 3. Experimental results and analysis are shown in Section 4, followed by the conclusion in Section 5.

2 The Proposed Algorithm

2.1 Basic Bacterial Foraging Optimizer

The basic bacterial foraging optimizer comprises three constituent mechanisms, namely, chemotaxis, reproduction and elimination-dispersal [8]. A concise description of each mechanism is given as follows.

2.1.1 Chemotaxis

In the basic BFO, the chemotaxis process is to imitate the activity of E.coli bacterium as two types of movements, i.e., tumbling and swimming. Tumbling represents an E.coli bacterium moves with random direction, while swimming means an E.coli bacterium walks in the same direction after locating a fertile area. The position of the ith bacterium is defined as follows:

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + \frac{C(i) \times \Delta(i)}{\sqrt{\Delta^T(i) \times \Delta(i)}} \quad (2)$$

where $\theta^i(j,k,l)$ denotes the position of the *i*th bacterium at *j*th chemotaxis, *k*th reproduction and *l*th elimination-dispersal step. $C(i)$ is the run length unit which determines the step size adopted during each swimming or tumbling. $\Delta(i)$ represents a random direction vector whose values fall belong to [-1,1].

$F(\theta^i(j,k,l))$ indicates the fitness value of the *i*th bacterium corresponding to the position $\theta^i(j,k,l)$.

2.1.2 Reproduction

After every N_c chemotaxis steps the health statuses of the bacteria are evaluated and those with inferior healthy situation will die. N_c is the maximum steps in a chemotaxis process and the health status of each bacterium is defined as equation (3):

$$H(i) = \sum_{j=1}^{N_c} F(i,j,k,l) \qquad (3)$$

where $H(i)$ means the health status of the *i*th bacterium.

In the reproduction scheme, only the bacteria with the ranking of top fifty percent can survive. Then a survival bacterium is reproduced to two identical ones that are used to replace the dead individuals at the same locations. Thus, the population size of bacteria is unaltered through the reproduction.

2.1.3 Elimination-dispersal

This process is to mimic the phenomenon of bacteria renewal or migration due to environmental change. Specifically, a number of individuals are selected to be re-born or migrated to other locations based on random probability.

2.2 Superior Attraction Strategy

In the chemotaxis process of the basic BFO, the next movement of a bacterium is determined by random tumbling. Generating a random direction makes the population maintain better diversity. Nevertheless, because all individuals randomly walk around the landscape without intra-population information exchange, the search capability of BFO cannot ensure a satisfactory solution especially when there exist plenty of local optima or the optimization problem is complex. Due to the fact that all bacteria stochastically wander in the space with no direct communication with others, a bacterium that finds a nutrient area could not inform and attract other bacteria in time, which result in slow convergence of the population. With the purpose of improving BFOs search ability and convergence rate, a novel movement technique, namely, superior attraction strategy, is proposed in this study.

Table 1: Procedure for construction of BE^i

Initialization

For $d = 1 : D$
  If $rand < Pro$
    $A_1 = B(randi_1(p))$
    $A_2 = B(randi_2(p))$
    If $F(A_1) < F(A_2)$
      $BE^i_d = A_{1,d}$
    Else $BE^i_d = A_{2,d}$
    End
  Else $BE^i_d = B(i)_d$
  End
End

Output BE^i

In the proposed technique, the direction of the tumble for the *i*th bacterium is update as the following:

$$\begin{aligned}\theta^i(j+1,k,l) &= \theta^i(j,k,l) \\ &+ C(i) \times R^i \times \left(BE^i - \theta^i(j,k,l)\right)\end{aligned} \qquad (4)$$

where R^i is a D-dimensional random vector lies in [0,1]. BE^i represents a *D*-dimensional leading exemplar whose elements are made of superior historical information of itself and other bacteria. The detailed pseudo code for construction of BE^i is shown in Table 1.

In Table 1, *Pro* is a given probability that determines whether a bacterium shall exploit its past information or move close to other bacteria in the dth dimension. *B(i)* denotes the historical best of the *i*th bacterium. *p* means the population size. *randi(p)* indicates the bacterium selected from the population randomly.

All these bacteria can share their past experience with others by moving close to the superior individuals with the influence of superior attraction strategy. To ensure the latest information are exchanged timely, the leading exemplar is used to guide the bacterium just for once and will be rebuilt for every movement updating process. Compared with the original BFO, there are two main differences can be observed:

(1) Instead of evolution via random chemotaxis, the bacteria can walk along a more accurate direction towards the leading ones.

(2) Instead of separate tumbling in the search space, all bacteria except the worst one can potentially work as an exemplar to collaborate with others for various different dimensions. Apparently, the correlations of

cooperation are built among the individuals by sharing their information with each other.

2.3 Superior Attraction Bacterial Foraging Optimizer

The superior attraction bacterial foraging optimizer is proposed by replacing BFOs movement updating formula (Eq. (1)) with the superior attraction strategy.

Moreover, to further investigate the efficiency of the proposed technique and its interaction with the classical updating formula of direction, we derive another SABFO variant for comparison, where all procedures are the same as SABFO except the swimming operation. In this variant, the swimming operation (Step 4 (6)) is removed and the bacteria searching for optima completely depends on the proposed strategy. The main motivation for the removal of the swimming process is to test how the superior attraction strategy impacts the search capability and performance of BFO. To distinguish the two SABFO variants, the SABFO with and without swimming are termed as SA-ws and SA-ns, respectively.

The detailed procedures of SA-ws is shown as follows:

Step 1: Initialization of parameters:
p : population size,
N_c: chemotactic steps,
N_s: swimming steps,
N_{re}: reproduction steps,
N_{ed}: elimination-dispersal steps,
P_{ed}: possibility of elimination-dispersal,
$C(i)$: chemotactic step size (run length unit),
$\theta^i(i,j,k)$: initialized position.

Step 2: Elimination-dispersal loop: $l = l + 1$.

Step 3: Reproduction loop: $k = k + 1$.

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

(1) For $i = 1, 2, \cdots, p$, take a chemotactic step for the *i*th bacterium.
(2) Calculate $\theta(^i(j,k,l))$.
(3) Let $F' = F(\theta(^i(j,k,l)))$.
(4) Compute $\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i) \times R^i \times \left(BE^i - \theta^i(j,k,l)\right)$
(5) Calculate $\theta(^i(j+1,k,l))$.
(6) Swimming:
   (a) Initialize $m = 0$.
   (b) While $m < N_s$,
      - If $F(\theta^i(j+1,k,l)) < F'$ let $F' = F(\theta^i(j+1,k,l))$ and $\theta^i(j+1,k,l) = \theta^i(j+1,k,l) + \frac{C(i)\times\Delta(i)}{\sqrt{\Delta^T(i)\times\Delta(i)}}$, and calculate $F(\theta^i(j+1,k,l))$.
      - let $m = N_s$.
(7) If $i \neq p$ let $i = i + 1$, and go to (2)

Step 5: If $j < N_c$ then go to Step 4.

Step 6: Reproduction:

(1) Calculate health value of the *i*th bacterium, $H(i) = \sum_{j=1}^{N_c} F(i,j,k,l)$.
(2) The S_r individuals with the worst fitness values are eliminated and the other S_r bacteria with the best fitness values are split to replace the vacancies.

Step 7: If $k < N_{re}$ then go to Step 3.

Step 8: Elimination-dispersal: For $i = 1, 2, \cdots, p$, with possibility P_{ed} , eliminate and disperse each bacterium.

Step 9: If $l < N_{ed}$, then go to Step 2; Else go to the termination.

As the procedures of SA-ns are similar to that of SA-ws, they are not presented here. The flowchart of SA-ws and SA-ns are depicted in Figure 1 on the next page and Figure 2 on the following page, respectively.

3 Experiment Settings

3.1 Benchmark Functions

As we propose to fully test the performance of SABFO algorithms, sixteen benchmark functions collected from [24, 25] are adopted for the comprehensive comparison. To increase difficulty and complexity of the experiments, all test functions are implemented with shift or rotation, thus including various characteristics, such as unimodal, multimodal, separable and rotated problems. For the shift operation, the global optimal is shifted to different locations for different dimensions ($z = \theta - SV$), SV is the position of the new global optimal. As to rotation, the original variable is left multiplied by an randomly orthogonal matrix, i.e., $z = M \times \theta$.

The formulas of these functions with and without rotation are shown in Table 2 and Table 3, respectively.

3.2 Compared algorithms and relevant settings

The experiments were carried out to compare six stochastic algorithms including the proposed SABFO algorithms on sixteen test problems with dimensions of two, ten and thirty. The algorithms adopted for comparison are listed as follows:

(1) GA: Genetic algorithm [2];
(2) BFO: Canonical BFO [8];

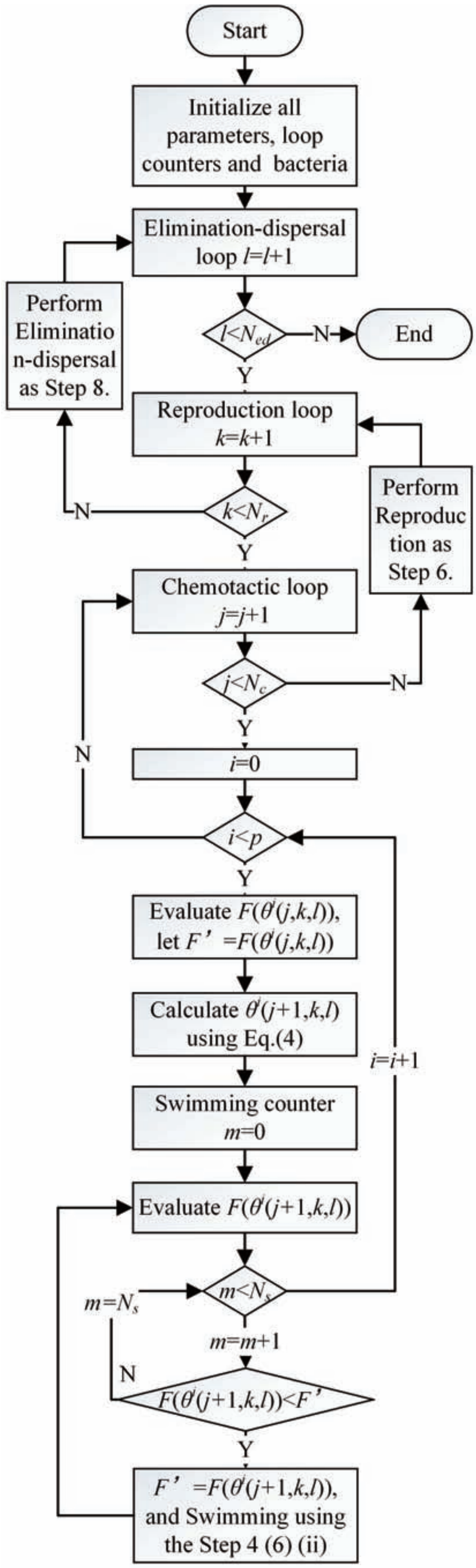

Figure 1: Flowchart of SA-ws

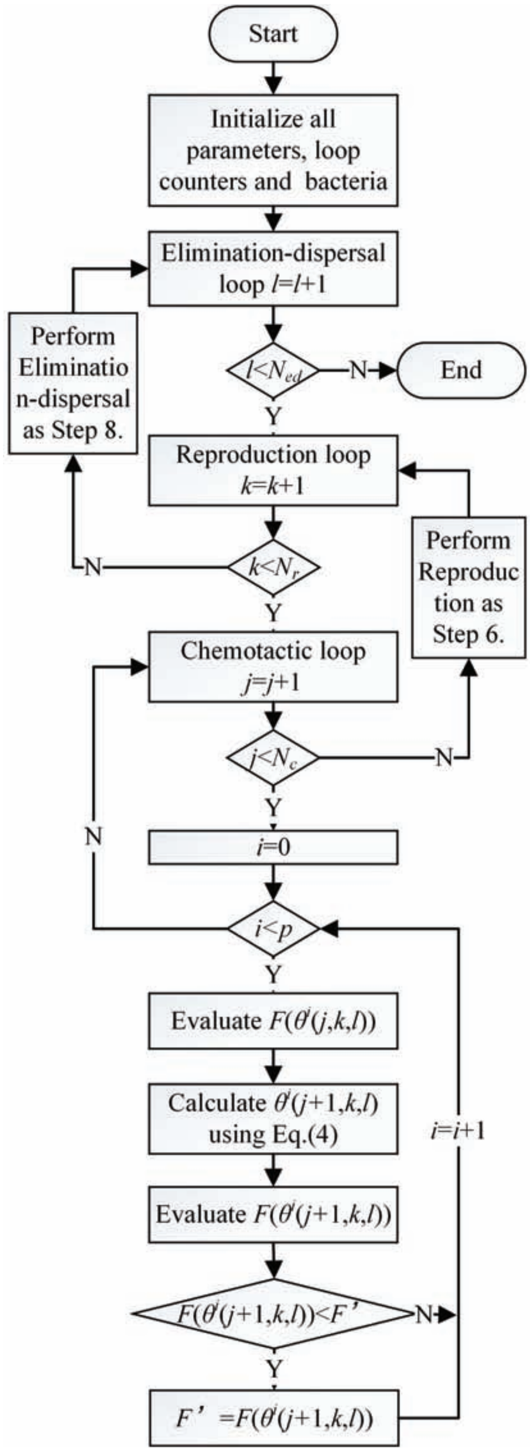

Figure 2: Flowchart of SA-ns

Table 2: Test functions without rotation

Fun	Name	Formula
f_1	Sphere	$f_1(x)=\sum_{i=1}^{D} z_i^2$
f_2	Step	$f_2(x)=\sum_{i=1}^{D}\left(\lfloor z_i+\frac{1}{2}\rfloor\right)^2$
f_3	Schwefel	$f_3(x)=418.982887273$ $\times D-\sum_{i=1}^{D} z_i\sin(\sqrt{\lvert z_i\rvert})$
f_4	2^D minima	$f_4(x)=78.332331408+$ $\frac{\sum_{i=1}^{D}(z_i^4-16z_i^2+5z_i)}{D}$
f_5	Rastrigin	$f_5(x)=\sum_{i=1}^{D}$ $(z_i^2-10\cos(2\pi z_i)+10)$
f_6	Non-Rastrigin	$f_6(x)=\sum_{i=1}^{D}(y^2$ $-10cos(2\pi y_i)+10);$ if $\lvert z_i\rvert<0.5, y_i=z_i$ else $y_i=\frac{round(2z_i)}{2}$
f_7	Ackley	$f_7(x)=20+e+f_{bias}-$ $20\exp(-\frac{1}{5}\sqrt{\frac{1}{D}\sum_{i=1}^{D} z_i^2}\ )$ $-\exp(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi z_i))$
f_8	Griewank	$f_8(x)=\sum_{i=1}^{D}\frac{z_i^2}{4000}+1$ $-\prod_{d}^{i=1} cos(\frac{z_i}{\sqrt{i}})+f_{bias}$

Table 3: Test functions without rotation

Fun.	Name	Formula
f_9	Sphere	$f_9(x)=\sum_{i=1}^{D} z_i^2$
f_{10}	Schwefel	$f_{10}(x)=\max_{i=1,\ldots,D}\lvert z_i\rvert$
f_{11}	Rosenbrock	$f_{11}(x)=\sum_{i=1}^{D-1}((z_i$ $-1)^2)+100(z_i^2-z_{i+1})^2$
f_{12}	Tablet	$f_{12}(x)=(1000z_1)^2$ $+\sum_{i=2}^{D} z_i^2$
f_{13}	Ellipset	$f_{13}(x)=\sum_{i=1}^{D}(a_i z_i)^2$ where $a_i=20^{\frac{i-1}{n-1}}$ $,i=1,\cdots,D$
f_{14}	2^D minima	$f_{14}(x)=78.332331408+$ $\frac{\sum_{i=1}^{D}(z_i^4-16z_i^2+5z_i)}{D}$
f_{15}	Griewank	$f_{15}(x)=\sum_{i=1}^{D}\frac{z_i^2}{4000}+1$ $-\prod_{d}^{i=1} cos(\frac{z_i}{\sqrt{i}})+f_{bias}$
f_{16}	Salomon	$f_{16}(x)=1+20+e-$ $cos(2\pi\sqrt{\frac{\sum_{i=1}^{D} z_i^2}{D}})$ $-\exp(\frac{\sum_{i=1}^{D} cos(2\pi z_i)}{D})$

(3) PSO-g: PSO with global topology [26];
(4) PSO-l: PSO with local topology [26];
(5) SA-ws: SABFO with swimming;
(6) SA-ns: SABFO without swimming;

For fair comparison, the maximum function evaluations (FEs) for all involved approaches is set to be proportional to the optimized dimensions, namely, $5000\times D$. Thus, when solving the 2-D problems, the maximum FEs is set at 10000. When solving 10-D problems, the maximum FEs is set as 50000, and the number is set to be 150000 for 30-D problems. The population size is set at 100 for the compared algorithms, and the specific parameter settings recommended in the original references [1, 2, 8, 26] are adopted in the study. $C(i)=1.5$ for $i=1,2,\cdots,p$. With regard to the construction probability in SABFOs, we employ the method introduced in [24] to determine the values since different Pro values generate better performance for various problems. All experiments are separately run 30 times for each algorithm on each test function. The mean values of the final solutions are given in Section 4.

4 Experimental results

4.1 Results for 2-D problems

Table 4 presents the means of the 30 runs of the six methods on the eight 2-D benchmark functions without rotation. The best results are shown in bold. It can be observed that SA-ns outperforms the other algorithms on 6 out of 8 problems in terms of means, while SA-ws obtains the best result for only one function.

Table 5 shows the results on the benchmark functions with rotation, which make problems inseparable for D one-dimensional searches. In this case, it can be seen that SA-ns performs worse than on the unrotated category in terms of means, that is, SA-ns achieves the best outcomes on 3 out of eight problems. Similarly, SA-ws outperforms the other methods on one functions.

Furthermore, comparing SA-ws with BFO, SA-ws outperforms BFO on 10 out of 16 functions.

The convergence characteristics of the compared algorithms on several representative 2-D functions (e.g., unimodal and multimodal problems) are shown in Figures 3 - 5. It can be observed that SA-ns converges fast on 2-D functions without rotation and finds the best results at early stage of evolution. For the 2-D rotated functions, PSO-l has good convergence speed.

To sum up, SA-ns performs better than the compared algorithms on 9 out of sixteen problems, and SA-ws gains best results on 2 out of sixteen functions.

Table 4: Results for 2-D unrotated problems

	f_1	f_2	f_3
GA	8.75E-03	**0.00E+00**	4.81E-03
BFO	1.23E+00	9.10E+01	6.36E+01
PSO-g	3.89E-10	**0.00E+00**	**3.78E-09**
PSO-l	5.60E-06	**0.00E+00**	1.23E-01
SA-ws	4.06E-07	**0.00E+00**	1.10E-07
SA-ns	**0.00E+00**	**0.00E+00**	1.78E+01
	f_4	f_5	f_6
GA	1.02E-03	3.20E-03	4.78E-03
BFO	1.53E-04	2.27E-02	6.52E-03
PSO-g	4.65E-10	7.72E-10	**4.63E-10**
PSO-l	2.68E-09	1.74E-05	6.04E-02
SA-ws	6.05E-06	4.98E-02	1.19E-01
SA-ns	**4.57E-10**	**0.00E+00**	2.00E-01
	f_7	f_8	
GA	7.74E-03	4.49E-02	
BFO	1.77E-02	9.62E-01	
PSO-g	1.34E-05	5.01E-03	
PSO-l	2.28E-04	7.99E-03	
SA-ws	1.46E-03	2.45E-03	
SA-ns	**0.00E+00**	**2.05E-03**	

Table 5: Results for 2-D rotated problems

	f_9	f_{10}	f_{11}
GA	7.34E-03	4.95E-02	5.73E+00
BFO	2.51E+00	1.36E-01	1.30E+01
PSO-g	3.82E-10	1.40E-05	4.43E+00
PSO-l	8.59E-08	1.90E-04	**8.71E-02**
SA-ws	3.43E-07	4.72E-04	2.65E+01
SA-ns	0.00E+00	9.46E-14	1.88E+01
	f_{12}	f_{13}	f_{14}
GA	1.06E+02	1.66E+00	9.40E-04
BFO	1.21E+02	4.00E+01	2.67E-04
PSO-g	5.58E+02	**9.01E-05**	4.59E-10
PSO-l	**9.29E+01**	1.19E-02	**1.20E-08**
SA-ws	3.55E+02	1.18E+02	7.07E-01
SA-ns	3.46E+02	3.85E+01	7.07E-01
	f_{15}	f_{16}	
GA	4.22E-02	8.56E-02	
BFO	7.54E-01	9.08E-01	
PSO-g	3.60E-03	5.13E-03	
PSO-l	7.03E-03	3.98E-02	
SA-ws	4.61E-03	**4.65E-03**	
SA-ns	**2.35E-03**	1.41E-02	

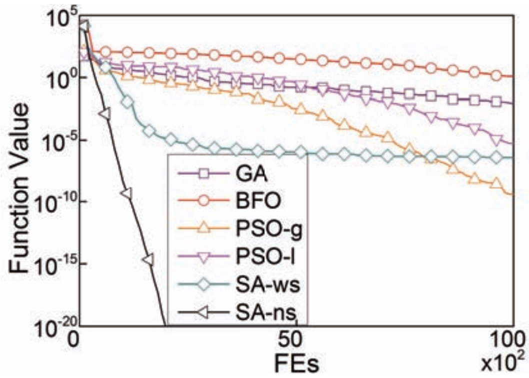

Figure 3: Convergence curves on 2-D f_1

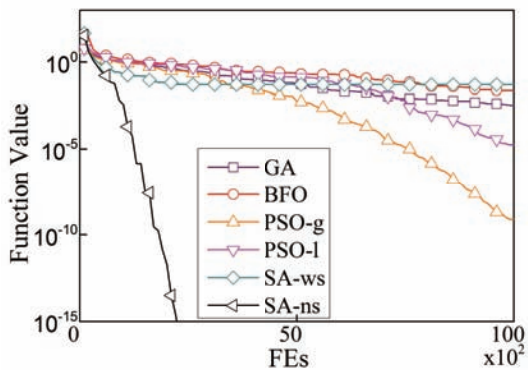

Figure 4: Convergence curves on 2-D f_5

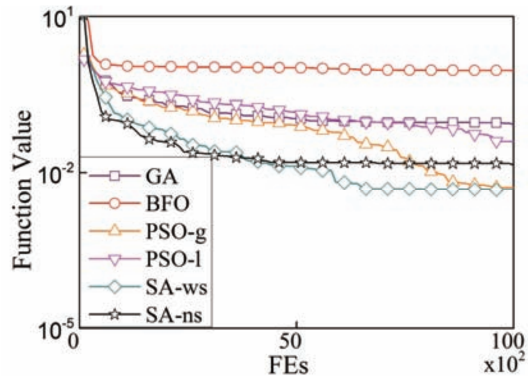

Figure 5: Convergence curves on 2-D f_{16}

Table 6: Results for 10-D unrotated problems

	f_1	f_2	f_3
GA	4.62E+01	8.85E+00	8.79E+02
BFO	8.22E+03	1.25E+04	2.15E+03
PSO-g	1.09E-13	**0.00E+00**	2.78E+02
PSO-l	4.81E-05	**0.00E+00**	1.07E+03
SA-ws	2.42E-04	**0.00E+00**	5.65E-01
SA-ns	**0.00E+00**	**0.00E+00**	**5.66E-09**
	f_4	f_5	f_6
GA	3.85E+00	1.07E+01	7.80E+00
BFO	3.54E+00	3.05E+01	1.95E+01
PSO-g	4.24E-01	2.50E+00	4.16E+00
PSO-l	8.77E-08	4.75E+00	6.17E+00
SA-ws	8.72E-04	9.89E-02	1.86E-01
SA-ns	**4.57E-10**	**0.00E+00**	**0.00E+00**
	f_7	f_8	
GA	7.75E-01	9.56E-01	
BFO	1.81E+01	1.13E+02	
PSO-g	5.44E-08	1.01E-01	
PSO-l	1.55E-02	7.36E-02	
SA-ws	2.15E-02	1.65E-03	
SA-ns	**1.78E-15**	**1.44E-05**	

4.2 Results for 10-D problems

The average results of the 30 runs of the sixteen algorithms on the 10-D problems are given in Table 6 and Table 7. As can be seen from Table 6, SA-ns outperforms the compared methods on all unrotated functions, while SA-ws performs best on function 2. For the 10-D functions with rotation, there is a obvious improvement for SA-ns as it achieves the best results on 6 out of eight rotated problems in terms of means. Comparing with its performance on 2-D problems, SA-ns improves its results for functions 12, 14 and 16. Meanwhile, SA-ws cannot obtain any best rankings for this case. However, SA-ws outperforms BFO for all the test function in terms of means.

To investigate the correlation between SABFOs convergence speed and problem dimensionality, the figures depicted on 2-D problems are repeated on the 10-D problems, as shown in Figures 6 - 8. Although the dimensions of test functions are increased, SA-ns exhibits good search capabilities and convergence faster than other algorithms on most of the test functions.

4.3 Results for 30-D problems

In addition to low-dimension problems, the experiments are repeated on the relatively complex problems with 30 dimensions to verify SABFOs performance.

Table 7: Results for 10-D rotated problems

	f_9	f_{10}	f_{11}
GA	4.70E+01	6.00E+00	1.29E+05
BFO	7.63E+03	4.55E+01	8.47E+08
PSO-g	3.09E-14	1.80E-05	1.02E+02
PSO-l	1.37E-05	1.69E-01	**7.45E+00**
SA-ws	2.61E-04	1.77E-02	4.19E+01
SA-ns	**0.00E+00**	**2.87E-06**	3.43E+01
	f_{12}	f_{13}	f_{14}
GA	2.76E+03	8.03E+03	7.39E+00
BFO	1.83E+04	2.43E+05	2.70E+00
PSO-g	1.64E-13	**1.11E-06**	2.25E+00
PSO-l	2.88E-05	3.10E+03	2.93E-02
SA-ws	2.91E-01	4.89E+00	4.45E-03
SA-ns	**0.00E+00**	1.67E-01	**5.19E-07**
	f_{15}	f_{16}	
GA	7.71E-01	7.15E-01	
BFO	1.12E+02	1.14E+01	
PSO-g	1.24E-01	1.15E-01	
PSO-l	5.45E-02	2.42E-01	
SA-ws	1.26E-02	1.23E-01	
SA-ns	**1.05E-03**	**9.99E-02**	

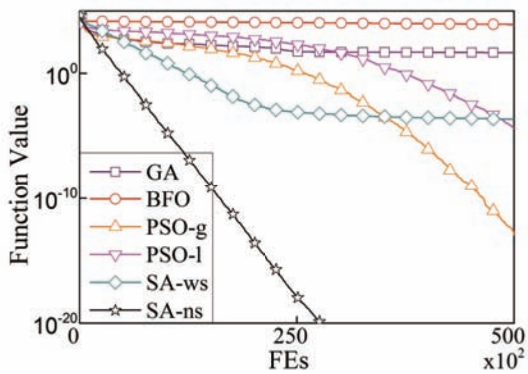

Figure 6: Convergence curves on 10-D f_1

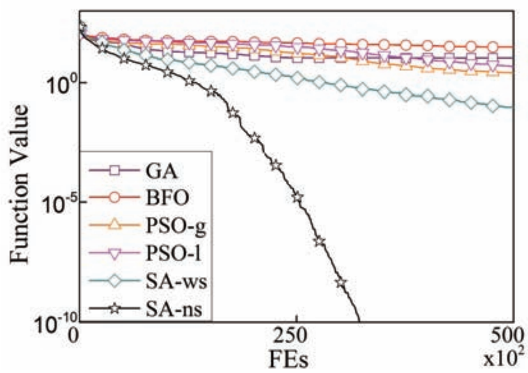

Figure 7: Convergence curves on 10-D f_5

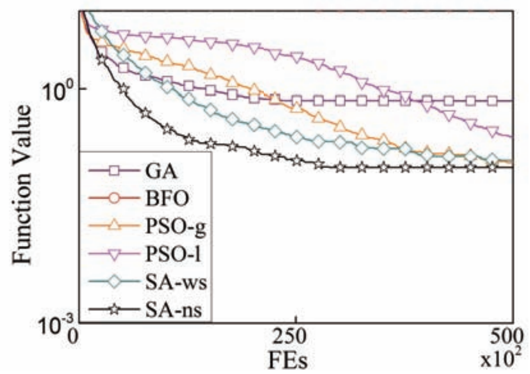

Figure 8: Convergence curves on 10-D f_{16}

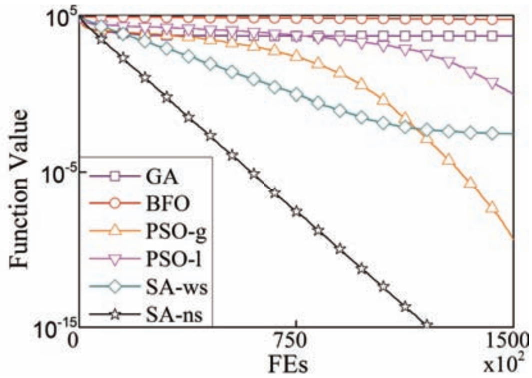

Figure 9: Convergence curves on 30-D f_1

Table 8 and Table 9 present the mean experiment results of 30 trials of the sixteen algorithms on 30-D benchmark problems. It can be observed that SA-ns gains the best results on 15 out of sixteen test problems. Moreover, SA-ws performs better than the other algorithms except SA-ns on 11 out of sixteen functions which is significantly improved than in the 2-D and 10-D categories of test. SABFOs outperforms the original BFO on all test functions.

Figures 9 - 11 present the convergence characteristics of the algorithms on 30-D problems as conducted on 2-D and 10-D problems. As can be seen from the figures, SABFOs exhibits better search abilities and convergence speed than the compared swarm intelligence algorithms on almost all test functions with 30 dimensions.

4.4 Discussion on the superior attraction strategy

By analyzing the results of the SABFO variants on 2-D, 10-D and 30-D functions, it can be concluded that SABFOs perform better with the increase of dimensionality of the test problems.

With regard to the easiest category C 2-D problems, although SABFOs obtains 10 best rankings out

Table 8: Results for 30-D unrotated problems

	f_1	f_2	f_3
GA	4.77E+03	6.70E+03	5.51E+03
BFO	5.54E+04	8.46E+04	8.42E+03
PSO-g	4.01E-10	5.00E-02	1.04E+03
PSO-l	8.54E-01	9.00E-01	5.95E+03
SA-ws	2.77E-03	**0.00E+00**	1.67E+03
SA-ns	**1.49E-20**	**0.00E+00**	**1.37E+02**
	f_4	f_5	f_6
GA	2.18E+01	1.31E+02	8.67E+01
BFO	7.73E+00	2.22E+02	2.24E+02
PSO-g	4.05E+00	2.14E+01	2.88E+01
PSO-l	6.79E-03	4.38E+01	4.28E+01
SA-ws	2.48E-03	1.50E+01	1.72E+01
SA-ns	**4.57E-10**	**1.03E+00**	**3.04E+00**
	f_7	f_8	
GA	1.33E+01	5.25E+00	
BFO	2.00E+01	5.19E+02	
PSO-g	5.24E-06	1.40E-02	
PSO-l	8.28E-01	6.32E-02	
SA-ws	4.11E-02	5.62E-04	
SA-ns	**1.10E-10**	**2.84E-12**	

Table 9: Results for 30-D rotated problems

	f_9	f_{10}	f_{11}
GA	3.42E+03	9.85E+00	2.98E+06
BFO	5.48E+04	6.72E+01	9.51E+09
PSO-g	7.06E-10	1.32E+00	7.39E+02
PSO-l	8.30E-01	6.60E+00	4.55E+02
SA-ws	2.55E-03	9.92E-01	1.11E+02
SA-ns	**1.74E-20**	**8.43E-02**	**4.34E+01**
	f_{12}	f_{13}	f_{14}
GA	4.05E+04	2.86E+05	2.49E+01
BFO	9.98E+04	2.32E+06	8.07E+00
PSO-g	3.92E+03	**1.14E+02**	7.96E+00
PSO-l	1.40E+04	6.81E+04	5.33E+00
SA-ws	3.19E+03	9.86E+03	2.80E+00
SA-ns	**1.27E+03**	1.64E+03	**1.78E+00**
	f_{15}	f_{16}	
GA	5.21E+00	8.46E+00	
BFO	5.27E+02	2.96E+01	
PSO-g	7.15E-03	4.10E-01	
PSO-l	6.21E-02	1.96E+00	
SA-ws	8.60E-04	5.71E-01	
SA-ns	**2.93E-09**	**3.00E-01**	

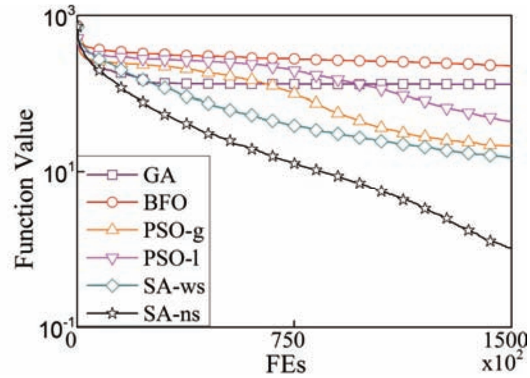

Figure 10: Convergence curves on 30-D f_5

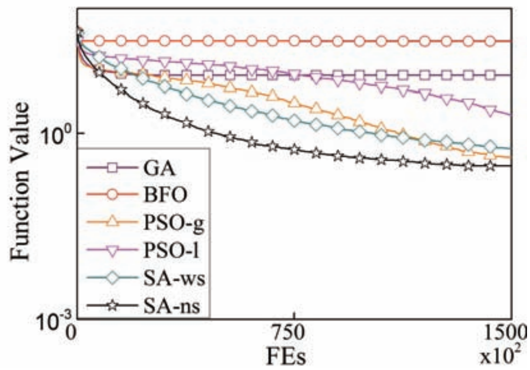

Figure 11: Convergence curves on 30-D f_{16}

of sixteen functions, SA-ns and SA-ws relatively underperforms other algorithms on the test with rotation. It indicates that all the evolutionary algorithms could solve the easy and simple problems efficiently. The performance of SABFOs need to be justified on more complex problems. With the growth of complexity of problems, namely, 10-D and 30-D problems, the performance of SABFOs improves and achieves the best results on most functions in terms of means, especially for 30-D category. This proves that the proposed strategy is more effective in addressing relatively complex and high-dimensional problems. This advantage is mainly due to SABFOs more accurate and dimensional position rule, as well as SABFOs making use of historical information.

The efficiency of the superior attraction strategy is further verified by comparing SA-ws with SA-ns. SABFO without swimming outperforms SABFO with swimming on almost all functions with various dimensionalities. As the only difference between SA-ws and SA-ns is implementing the swimming or not, it is safe to conclude that the proposed technique exhibits superior search capabilities than the original tumbling operation and improves the efficiency and effectiveness of BFO.

In addition, SA-ns simplifies the procedure of SA-ws by removing the swimming component, which leads to an easy implementation of the original BFO and fulfills the potential of the proposed strategy.

5 Conclusion

In this paper, a novel movement updating technique, i.e., superior attraction strategy, is proposed to overcome the following demerits of original BFO: (1) BFO cannot perform well on various problems with different dimensions. (2) BFO suffers poor convergence. By making use of historical information of bacteria to enhance BFO, the introduced method is effective in guiding bacteria direction and generating better quality results. The convergence speed of BFO is accelerated via information exchange among bacteria. The novel strategy lets the individuals potentially works as an exemplar to collaborate with others to search for the better solutions for different dimensions.

Experimental comparisons have been conducted on sixteen global optimization functions including unimodal and multimodal problems with various dimensions. The results indicate that the SABFOs significantly improve the performance of BFO on the diverse of functions with different complexity when compared with several existing evolutionary algorithms. The convergence curves have shown a fast convergence speed of the SA-ns on most of test problems that overcomes the poor convergence of BFO.

In addition, the implementation of the superior attraction strategy is simple and does not introduce any complex operation to basic BFO. The SA-ns even simplifies computational complexity of basic BFO.

For the future interests, the convergence and search behavior of SABFOs should be mathematically proofed. Besides, the application of SABFOs to real-world problems is also a promising topic.

Acknowledgements: This work was supported by the National Natural Science foundation of China (Grants No. 71171064, 71001072 and 71271140).

References:

[1] J. Kennedy, and R. Eberhart, Particle swarm optimization, *Proceedings of IEEE International Conference on Neural Networks Proceedings*, pp. 1942–1948, Piscataway, NJ, USA, 1995.

[2] J. H. Holland, Adaptation in natural and artificial systems: an introductory analysis with applications to biology control and artificial intelligence, Ann Arbor, USA: University of Michigan Press, 1975.

[3] M. Dorigo, V. Maniezzo, and A. Colorni, Ant system: Optimization by a colony of cooperating agents, *IEEE Transactions on Systems Man and Cybernetics Part B-Cybernetics*, Vol. 26, No. 1, 1996, pp. 29-41.

[4] B. Qiao, X. Chang, M. Cui et al., Hybrid particle swarm algorithm for solving nonlinear constraint optimization problems,*WSEAS Transactions on Mathematics*, Vol. 12, No. 1, 2013, pp. 76-84.

[5] Z. M. Nopiah, M. I. Khairir, S. Abdullah et al., Time complexity estimation and optimisation of the genetic algorithm clustering method, *WSEAS Transactions on Mathematics*, Vol. 9, No. 5, 2010, pp. 334-344.

[6] G. Fuellerer, K. F. Doerner, R. F. Hardl et al., Ant colony optimization for the two-dimensional loading vehicle routing problem, *Computers & Operations Research*, Vol. 36, No. 3, 2009, pp. 655-673.

[7] X. Chu, Q. Lu, B. Niu et al., Solving the distribution center location problem based on multi-swarm cooperative particle swarm optimizer, *8th International Conference on Intelligent Computing Technology*, pp. 626-633, Huangshan, China, 2012.

[8] K. M. Passino, Biomimicry of bacterial foraging for distributed optimization and control,*IEEE Control Systems Magazine*, Vol. 22, No. 3, 2002, pp. 52-67.

[9] S. D. Muller, J. Marchetto, S. Airaghi et al., Optimization based on bacterial chemotaxis,*IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 1, 2002, pp. 16-29.

[10] S. Mishra, and C. N. Bhende, Bacterial foraging technique-based optimized active power filter for load compensation, *IEEE Transactions on Power Delivery*, Vol. 22, No. 1, 2007, pp. 457-465.

[11] B. Niu, H. Wang, J. Wang et al., Multi-objective bacterial foraging optimization, *Neurocomputing* , Vol. 116, No. 0, 2013, pp. 336-345.

[12] S. Singh, T. Ghose, and S. K. Goswami, Optimal Feeder Routing Based on the Bacterial Foraging Technique, *IEEE Transactions on Power Delivery*, Vol. 27, No. 1, 2012, pp. 70-78.

[13] H. N. Chen, Y. L. Zhu, and K. Y. Hu, Adaptive Bacterial Foraging Optimization, *Abstract and Applied Analysis*, Vol. 2011, Article ID 108269, 2011, pp. 1-27.

[14] M. El-Abd, Performance assessment of foraging algorithms vs. evolutionary algorithms, *Information Sciences*, Vol. 182, No. 1, 2012, pp. 243-263.

[15] N. Pandit, A. Tripathi, S. Tapaswi et al., An improved bacterial foraging algorithm for combined static/dynamic environmental economic dispatch, *Applied Soft Computing*, Vol. 12, No. 11, 2012, pp. 3500-3513.

[16] P. D. Sathya, and R. Kayalvizhi, Optimal segmentation of brain MRI based on adaptive bacterial foraging algorithm, *Neurocomputing*, Vol. 74, No. 14-15, 2011, pp. 2299-2313.

[17] B. Niu, Y. Fan, H. Wang et al., Novel bacterial foraging optimization with time-varying chemotaxis step, *International Journal of Artificial Intelligence*, Vol. 7, No. 11, 2011, pp. 257-273.

[18] B. Niu, and H. Wang, Bacterial Colony Optimization, *Discrete Dynamics in Nature and Society*,Vol. 2012, Article ID 698057, 2012, pp. 1-28.

[19] N. Sarasiri, K. Suthamno, and S. Sujitjorn, Bacterial Foraging-Tabu Search Metaheuristics for Identification of Nonlinear Friction Model, *Journal of Applied Mathematics*,Vol. 2012, Article ID 238563, 2012, pp. 1-23.

[20] S. Gollapudi, S. S. Pattnaik, O. P. Bajpai et al., Velocity Modulated Bacterial Foraging Optimization Technique (VMBFO), *Applied Soft Computing*, Vol. 11, No. 1, 2011, pp. 154-165.

[21] D. H. Kim, A. Abraham, and J. H. Cho, A hybrid genetic algorithm and bacterial foraging approach for global optimization, *Information Sciences*, Vol. 177, No. 18, 2007, pp. 3918-3937.

[22] O. P. Verma, M. Hanmandlu, P. Kumar et al., A novel bacterial foraging technique for edge detection, *Pattern Recognition Letters*, Vol. 32, No. 8, 2011, pp. 1187-1196.

[23] H. N. Chen, Y. L. Zhu, and K. Y. Hu, Cooperative Bacterial Foraging Optimization,*Discrete Dynamics in Nature and Society*, Vol. 2012, Article ID 238563, 2009, pp. 1-23.

[24] J. J. Liang, A. K. Qin, P. N. Suganthan et al., Comprehensive learning particle swarm optimizer for global optimization of multimodal functions, *IEEE Transactions on Evolutionary Computation*, Vol. 10, No. 3, 2006, pp. 281-295.

[25] M. Hu, T. Wu, and J. D. Weir, An intelligent augmentation of particle swarm optimization with multiple adaptive methods, *Information Sciences*, Vol. 213, 2012, pp. 68-83.

[26] J. Kennedy, and R. Mendes, Population structure and particle swarm performance,*Proceedings of IEEE congress on Evolutionary Computation*, pp. 1671-1676, Hawaii, USA, 2002.