



Bacterial foraging optimization based on improved chemotaxis process and novel swarming strategy

Bao Pang¹ · Yong Song² · Chengjin Zhang² · Hongling Wang¹ · Runtao Yang²

Published online: 5 November 2018
© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract

Bacterial foraging optimization (BFO), a biological-inspired optimization algorithm, has been applied in various fields, such as complex function optimization, robot path planning. However, there still exist several insufficiencies in BFO algorithm due to the fixed chemotaxis step-size, the less-efficient search direction for tumbling and the swarming strategy with lower convergence rate. In order to deal with these issues, based on the Lévy flight step-size and particle swarm optimization (PSO) operator, this paper proposes the improved BFO algorithm (LPBFO). To reduce the mutual interference among different dimensions, each bacterium selects one dimension for tumbling randomly during the chemotactic process in LPBFO. The step-size of each bacterium is determined by the stochastic flight lengths of the improved Lévy flight which can generate small step-size with high frequency and big step-size occasionally; moreover, the stochastic step-size is also reduced adaptively based on the evolutionary generations, which makes the bacteria transform from global search to local search. Furthermore, inspired by the social information term in PSO, this paper employed the global best solution to improve the swarming performance. Several experiments on benchmark functions are carried out with the purpose of evaluating the performance of the proposed method. Experimental results show that the proposed algorithm achieves noticeable improvement compared with other competitive algorithms.

Keywords Bacterial foraging optimization · Lévy flight · Particle swarm optimization · Chemotaxis step-size · Self-regulation

1 Introduction

The BFO algorithm, which was proposed by Passino [21], is a novel category of bionic intelligent optimization algorithm.

In the process of foraging, the bacteria takes necessary actions (such as chemotaxis, swarming, reproduction, elimination and dispersal) to search for nutrient-rich areas so as to maximize the energy intake. The position of one bacterium represents a solution of the solving problem. According to tumbling to change the present direction and keeping swimming in the selected direction to update its position, the bacteria can seek for the optimal position by continuous movement and the position of best bacterium is the best solution of the solving problem. Up to now, the BFO algorithm has received widespread attention and has been applied successfully to a variety of fields, such as feature selection [7, 20], image processing [2, 27], scheduling problem [34] and continuous optimization problems [18, 22, 25].

However, some researchers have shown that the original BFO algorithm suffers from low solution accuracy and poor convergence performance. Meanwhile, many strategies and methods have been proposed to improve the performance of BFO algorithm. In the BFO algorithm, the constant chemotaxis step-size can not only lead to bad convergence performance but also result in sustained oscillation of the optimization procedure. To improve convergence performance,

✉ Yong Song
songyong@sdu.edu.cn

Bao Pang
pang_bao11@163.com

Chengjin Zhang
cjzhang@sdu.edu.cn

Hongling Wang
wanghongling720409@163.com

Runtao Yang
runtao-sd@163.com

¹ School of Control Science and Engineering, Shandong University, Jinan 250061, China

² School of Mechanical, Electrical and Information Engineering, Shandong University at Weihai, Weihai 264209, China

Mishra proposed a least square-fuzzy scheme to generate the changing chemotactic step-size adaptively based on the minimum value of cost function [17]. Moreover, the authors proposed two simple schemes to adaptively change the chemotactic step-size to avoid the oscillatory behavior and to speed up the convergence speed [8]. In optimization problem, the algorithm should combine global search with local search to balance exploration and exploitation process. Niu et al. proposed a non-linearly decreasing exponential modulation model to optimize the step-size for global optimization [19]. In [26], the authors proposed a new approach involving adaptable chemotactic step-size which changes depending on the fitness value to balance the local search and global search in the optimization process. More recently, Xu et al. proposed a novel adaptive scheme, which based on the function period, predefined maximum and minimum values, to adapt the chemotactic step-size and introduced a novel operation of micro-swim with the aim of improving its convergence behavior [31]. Yang et al. adjusted the step-size of each bacterium adaptively based on the evolutionary generations and the information of the globally best individual [33]. To enhance the global search ability and extend the diversity of bacterial population, an improved quantum BFO algorithm was proposed which utilizes a continuously varying rotation angle to update the rotation gate [13, 14].

These proposed algorithms were all shown to outperform the classical BFO algorithm over a few numerical benchmark functions. However, little information being communicated and no cooperation among bacteria existing in these algorithms affect the performance of the BFO algorithm. Inspired by the social information term of PSO [10], Biswas et al. proposed a hybrid algorithm involving PSO and BFO to balance between local search and global search [3]. The proposed algorithm executes local search by means of the chemotactic movement operation of BFO and performs global search over the entire search space through a PSO operator. Chen et al. introduced the communication strategies among multi-colony bacterial community, which can maintain the diversity in the whole bacterial community [5]. The proposed multi-colony cooperative approach speeds up the bacterial community to converge to the global optimum significantly. Chen et al. extended the basic BFO to a self-adaptive and cooperative bacterial colony foraging (BCF) by using the optimal location information and analyzing the current status to adjust the chemotactic step-size adaptively [4]. By combining cell-to-cell communication and self-adaptive search strategies, the authors proposed a modified BFO algorithm (BCFO) that the bacterial colony can maintain a balance between global search and local search [6]. Inspired by the differential evolution algorithm, a chemotaxis-enhanced bacteria foraging optimisation (CEBFO) algorithm was proposed to solve the

tumble failure problem and enhance the convergence speed of the original algorithm [34]. Yang et al. proposed a new bacterial foraging optimization algorithm (BFO-CC) based on the new designed chemotaxis standard and conjugation strategies, which makes the algorithm keep a better balance between exploration and exploitation [33]. Tang et al. proposed an improved bacterial foraging (MBFO) algorithm by incorporating the PSO operator into chemotactic process to quicken the convergence rate and to improve the quality of the final solution [27]. As different swarm intelligence algorithms have different features, to improve the convergence performance, Turanoglu proposed a new hybrid heuristic algorithm, which combined the simulated annealing and BFO algorithm [28].

The above methods have achieved some progress, but there still exists some problems. On one hand, when solving optimization problem, the solutions are easy to trap in local optimum, but the above algorithms cannot escape from the local optimum effectively. To solve such problem, this paper uses the Lévy flight length as the chemotaxis step-size which can generate small step-size with high frequency and big step-size occasionally to balance local search and global search. Moreover, the solutions using the big step-size can escape from the local optimum effectively. For population-based optimization methods, it is desirable to encourage the individuals to wander through the entire search space at the initial phase of the optimization; moreover, it is very important to fine-tune the candidate solutions in the succeeding phases of the optimization [23, 33]. Therefore, to make the bacteria transform from global search to local search gradually, this paper constructs an adaptive Lévy flight step-size, which is reduced adaptively along with the increase of the number of iterations. On the other hand, in the original BFO algorithm, the bacteria choose one direction from an infinite number of directions randomly in the tumbling process, which not only possesses smaller probability of choosing the right direction but also causes mutual interference among different dimensions; such mechanism reduces the search efficiency greatly. In order to improve the search efficiency, this paper proposes an improved strategy to determine the search direction, where each bacterium selects one dimension randomly from the search space to tumble. As the selected dimension possess two directions, the selected direction is toward the better solution with the probability of fifty percent. Moreover, the bacterium swimming along one dimension can avoid the interference among different dimensions effectively and can maximize the search ability of the algorithm. Last, to improve swarming mechanism, this paper uses the global best solution of PSO algorithm to quicken the convergence rate and improve the swarming performance.

The remainder of the paper is organized as follows. Section 2 outlines the reviews of the classical BFO algorithm

briefly. Section 3 introduces the proposed LPBFO algorithm. Section 4 comparatively analyze the LPBFO and other bionic algorithm based on eight benchmark functions and the experimental result verifies that LPBFO algorithm significantly outperforms other bionic algorithm in the literature in terms of t-test. Finally, the conclusion is drawn in Section 5.

2 Classical BFO algorithm

The classical BFO algorithm includes four principal mechanisms, namely chemotaxis, swarming, reproduction, and elimination-dispersal [21]. In the following, we describe the contents of these four processes briefly.

2.1 Chemotaxis

The chemotaxis process simulates the movement of the E. coli bacteria through tumbling and swimming. The E. coli bacteria can tumble for another direction through selecting a direction randomly from the search space. After that, bacteria are going to move a step size. If the fitness becomes better, then they keep on swimming along this direction. Suppose $\theta^i(j, k, l)$ is the position of the i th bacterium at j th chemotactic, k th reproductive, and l th elimination-dispersal step. The chemotactic movement of the i th bacterium can be represented as

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

where $C(i)$ is the fixed step-size in each swimming. $\Delta(i)$ is the selected direction vector by tumbling whose elements lie in the range $[-1, 1]$.

2.2 Swarming

A cell-to-cell signaling via an attraction and repellent is simulated in swarming stage. The cell releases attractants to signal other cells that they should swarm together. The cell also releases repellent substance to repel a nearby cell. The combined cell-to-cell attraction and repelling effects is denoted as follows:

$$\begin{aligned} Jcc(\theta, P(j, k, l)) &= \sum_{i=1}^S Jcc(\theta, \theta^i(j, k, l)) \\ &= \sum_{i=1}^S \left[-d_{attract} \exp \left(-\omega_{attract} \sum_{m=1}^D (\theta_m - \theta_m^i)^2 \right) \right] \\ &\quad + \sum_{i=1}^S \left[h_{repellant} \exp \left(-\omega_{repellant} \sum_{m=1}^D (\theta_m - \theta_m^i)^2 \right) \right] \end{aligned} \quad (2)$$

where $Jcc(\theta, P(j, k, l))$ is the cell-to-cell communication value to be added to the actual fitness function to present

a time-varying fitness function. S is the total number of bacteria; D is the number of dimension to be optimized; $\theta = [\theta_1, \dots, \theta_D]^T$ is a point in the D -dimensional search space; $d_{attract}$, $\omega_{attract}$, $h_{repellant}$, $\omega_{repellant}$ are different coefficients that should be chosen properly. The fitness value of bacterium i can be represented by

$$J(i, j, k, l) = J(i, j, k, l) + Jcc(\theta, P(j, k, l)) \quad (3)$$

2.3 Reproduction

N_c is the lifetime of a bacterium and after N_c chemotactic steps, the bacterium performs reproduction. The health value of bacterium i ($J_{i,health}$) equals to the sum of the fitness value during its life.

$$J_{i,health} = \sum_{i=1}^{N_c} J(i, j, k, l) \quad (4)$$

Based on the health value, all bacteria are sorted in descending order. In the reproduction stage, the S_r ($S_r = S/2$) healthiest bacteria survive and each split into two bacteria, which are placed in the same locations, thereby maintaining a constant swarm size; the other S_r least healthy bacteria die.

2.4 Elimination and dispersal

Due to the adverse and unexpected changes in the local environment, some bacteria in this region may be killed or dispersed into new location. This phenomenon is simulated by the elimination in BFO and the bacteria are killed with a small probability P_{ed} . Simultaneously, some new bacteria are randomly generated within the environment for replacement.

3 LPBFO algorithm

In order to improve the performance, three improvements of LPBFO are introduced. First, the adaptive Lévy flight step-size is used as chemotaxis step-size to keep the balance between global search and local search. Moreover, the adaptive step-size can also make the solutions escape from the local optimum effectively. Second, instead of choosing one direction randomly from the search space, the bacterium selects a dimension randomly as the search direction to update the position of the solution. Third, in order to accelerate convergence speed, the global best location information is used to improve the swarming performance.

3.1 Adaptive Lévy flight step-size

A Lévy flight is a random walk in which the step-size have a probability distribution that is heavy-tailed. The probability distribution is called Levy distribution, satisfying the form of a power-law distribution, and can be expressed as follows [30]:

$$P(s) = s^{-\lambda} \tag{5}$$

where s is the step-size with $1 < \lambda \leq 3$. Lévy flight can generate smaller step-size with high frequency and occasionally generate larger step-size. In the search process, the bacteria with a larger step-size can reach the full range of the search space to locate the potential optimal regions; the bacteria with a smaller step-size tend to fine-tune the current solution to obtain the global optimal solution. Therefore, Lévy flight can well balance local search and global search. When solving optimization problem, the bigger step-size can also make the solutions escape from the local optimum effectively. The foraging behaviors of many creatures in nature satisfy Lévy flight, such as albatrosses' foraging flight trajectory [15, 29] and drosophilas' intermittent foraging flight trajectory [24]. Many studies suggest that Lévy flight is an optimal search strategy when the target sites are sparse and distributed randomly.

This paper uses the method proposed by [16] to calculate Lévy flight step-size:

$$s = \frac{u}{|v|^{1/\beta}} \tag{6}$$

where $\beta \in [0.3, 1.99]$, u and v are two normal stochastic variables with standard deviation σ_u and σ_v .

$$u \sim N(0, \sigma_u^2), \quad v \sim N(0, \sigma_v^2) \tag{7}$$

$$\sigma_u = \left\{ \frac{\Gamma(1 + \beta) \sin(\pi\beta/2)}{\Gamma[(1 + \beta)/2] 2^{(\beta-1)/2} \beta} \right\}^{1/\beta}, \quad \sigma_v = 1 \tag{8}$$

where $\Gamma(z)$ is gamma function.

As shown in Fig. 1, Lévy flight with a mix of large step-size and small step-size can balance local search and global search. Therefore, the Lévy flight step-size s can be used to replace chemotaxis step-size to improve search efficiency. However, at the early stage of the optimization process, the bacteria are inclined to locate the potential optimal regions by wandering through the entire search space; conversely, most of the bees are apt to fine-tune the present solutions to obtain the optimal solution at the latter stage of the optimization. Therefore, this paper constructs an adaptive Lévy flight step-size, upon which the step-size dwindle down gradually with the increase of the number of iterations. The equation for the adaptive nonuniform step-size is given below.

$$C'(i) = \frac{\alpha}{t(i)} * |s| \tag{9}$$

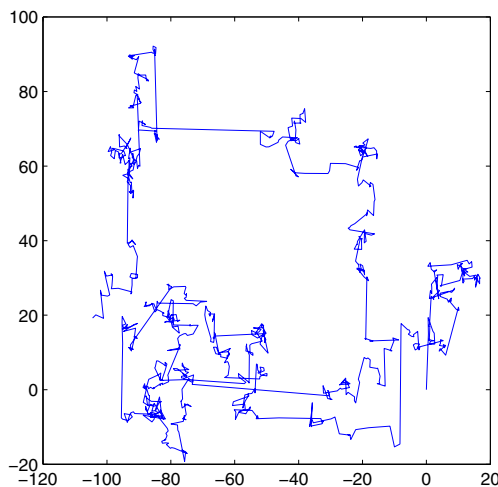


Fig. 1 An example of 1000 steps of a Lévy flight in two dimensions. The origin of the motion is at [0,0], the step-size is generated according to (6) with $\beta = 1.5$ and the angular direction is uniformly distributed

where α is a control parameter determining the change strength of step-size. Parameter $t(i)$ denotes the number of chemotaxes that the i th bacterium has experienced and $t(i)$ increases in the course of the optimization. At the early stage of the optimization, the bacteria with larger step-size can achieve global search by wandering through the entire search space. At the latter stage of the optimization, the bacteria with the adaptively reduced step-size can achieve local search by being apt to fine-tune the present solution.

3.2 Improved search direction

In the classical BFO algorithm, it is less-efficient for each bacterium to choose one direction randomly in each chemotaxis process. First, there exists an infinite number of directions can be chosen in D-dimensional search space and each bacterium should choose one direction randomly from these directions. Therefore, the selected direction is toward the better solution with a low probability. Second, each selected direction contains D dimensions in which some dimensions are possible to have the correct directions and some dimensions are possible to possess the incorrect directions. Although the new solution gets better fitness value, some dimensions may get worse. Therefore, different dimensions may interfere with each other and the strategy updating all dimensions regarding a solution at each iteration reduces search efficiency.

Therefore, to improve the search efficiency, this paper proposes an improved strategy to determine the search direction. In chemotaxis stage, each bacterium chooses one dimension randomly to tumble which possess two directions. Therefore generally speaking, the selected direction is toward the better solution with the probability of fifty percent. Moreover, the bacterium swimming along

one dimension can avoid the interference among different dimensions effectively and can maximize the search ability of the algorithm.

Based on the adaptive Lévy flight step-size and the improved search direction, the chemotactic movement of the *i*th bacterium in LPBFO can be represented as

$$\theta^i(j + 1, k, l) = \theta^i(j, k, l) + C'(i) * \Delta'(i) \tag{10}$$

where $\Delta'(i)$ is the improved search direction which is a D-dimensional vector with only one coordinate being 1 or -1 and all the others being 0; $C'(i)$ is the adaptive Lévy flight step-size taken in direction $\Delta'(i)$. As can be seen from (10), each bacterium only updates the corresponding solution in one dimension during every chemotaxis process.

In the original BFO, the bacterium will calculate its fitness value after each movement. During the chemotaxis stage, if the fitness value is better than the previous position, the bacterium will keep on moving along this direction. If the fitness value gets worse, the bacterium will not move forward. However, the bacterium has already in the worse position, which can slow down the chrematistic activity of the bacterium. If this chemotaxis behavior happens frequently, the searching efficiency will be greatly reduced. In order to overcome this problem, this paper employs a self-regulation strategy. When the bacterium arrives at the worse position, the bacteria will return to the last position automatically and the bacteria continues the next step.

3.3 Modified swarming process

In the original BFO algorithm, the bacteria complete the swarming process through the fitness function which is related to attraction-repellant and the distance among bacteria. Without directionality, the swarming process cannot guarantee that bacteria converge to the optimal solution. The optimal position information is not considered, and therefore the swarming process possess a lower convergence rate.

To accelerate the convergence rate and improve the accuracy of convergence, this paper takes advantage of the information of the global best (gbest) solution to guide the movement of the bacteria, we modify the swarming equation

$$\theta^i(j + 1, k, l) = \theta^i(j, k, l) + (\theta_{gbest} - \theta^i(j, k, l)) * C'(i) * \Delta'(i) \tag{11}$$

According to (11), the gbest term can drive the bacteria towards the global best solution in one dimension, therefore, the modified swarming equation can improve convergence rate of the swarming process. Moreover, the adaptive Lévy flight step-size is also used in the modified swarming equation to balance the global search and the local search.

Table 1 Numerical benchmark functions

Type	Test Function	Formulation	Search range	Minimum value
Unimodal	Sphere	$f_1(\vec{x}) = \sum_{i=1}^D x_i^2$	$x_i \in [-100, 100]$	$f_1(\vec{0}) = 0$
	Quartic	$f_2(\vec{x}) = \sum_{i=1}^D ix_i^4 + random[0, 1]$	$x_i \in [-1.28, 1.28]$	$f_2(\vec{0}) = 0$
	Schwefel's Problem 2.22	$f_3(\vec{x}) = \sum_{i=1}^D x_i + \prod_{i=1}^D x_i $	$x_i \in [-10, 10]$	$f_3(\vec{0}) = 0$
Multimodal	Rosenbrock	$f_4(\vec{x}) = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i)^2 + (1 - x_i)^2]$	$x_i \in [-30, 30]$	$f_4(\vec{1}) = 0$
	Rastrigin	$f_5(\vec{x}) = \sum_{i=1}^D [x_i^2 - 10\cos(2\pi x_i) + 10]$	$x_i \in [-5.12, 5.12]$	$f_5(\vec{0}) = 0$
	Griewank	$f_6(\vec{x}) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$x_i \in [-600, 600]$	$f_6(\vec{0}) = 0$
	Ackley	$f_7(\vec{x}) = -20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2 / D}\right) + 20 - \exp\left(\frac{D}{\sum_{i=1}^D \cos(2\pi x_i / D)}\right) + e$	$x_i \in [-30, 30]$	$f_7(\vec{0}) = 0$
	Schwefel's Problem 2.26	$f_8(\vec{x}) = 418.9829D - \sum_{i=1}^D (x_i \sin(\sqrt{ x_i }))$	$x_i \in [-500, 500]$	$f_8(\vec{420.9687}) = 0$

3.4 Description of the LPBFO algorithm

The pseudo code of the LPBFO algorithm can be described as follows:

step 1. Initialize parameters: $S, N_c, N_{re}, N_{ed}, N_s, S_r, P_{ed}, \beta, \alpha$.

S : number of bacteria,
 N_c : number of chemotactic steps,
 N_{re} : number of reproduction steps,
 N_{ed} : number of elimination-dispersal events,
 N_s : number of swimming and swarming length,
 S_r : number of healthiest bacteria $S_r = S/2$,
 P_{ed} : elimination-dispersal probability,
 α : control parameter determining the change strength of Levy flight step size.

step 2. Update the following parameters (initially, $i = j = k = 0$).

$\theta^i(j, k, l)$: position of i th bacterium which is distribute across the domain

of the optimization problem randomly,

$J(i, j, k, l)$: the fitness function of the i th bacterium,

θ_{gbest} : the global best position of the swarm discovered thus far,

J_{best} : the fitness of the best position found thus far.

step 3. Elimination-dispersal loop: $l = l + 1$

step 4. Reproduction loop: $k = k + 1$

step 5. Chemotactic loop: $j = j + 1$

step 6. Bacterial index loop: $i = i + 1$

6.1) Compute the value of objective function $J(i, j, k, l) = f(\theta^i(j, k, l))$

6.2) Let $\theta_{last} = \theta^i(j, k, l)$, $J_{last} = J(i, j, k, l)$ to save the position and fitness value because we may find a better position and fitness value through a run.

6.3) Tumble: generate a random D-dimensional vector $\Delta'(i)$ with only one coordinate being 1 or -1 and all the others being 0.

6.4) Move: let

$$\theta^i(j+1, k, l) = \theta^i(j, k, l) + \frac{\alpha}{t(i)} * \left| \frac{u}{|v|^{1/\beta}} \right| * \Delta'(i)$$

where $t(i) = j + (k - 1) * N_c + (l - 1) * N_c * N_{re}$, the result in a Levy flight step-size in the dimension of the tumbling for bacterium i .

6.5) Compute the value of objective function $J(i, j + 1, k, l) = f(\theta^i(j + 1, k, l))$

6.6) Swim:

i) Let $m = 0$ (counter for swim length).

ii) **While** $m < N_s$

Let $m = m + 1$

If $J(i, j + 1, k, l) < J_{last}$

$\theta_{last} = \theta^i(j + 1, k, l)$, $J_{last} = J(i, j + 1, k, l)$

Keep on moving along this dimension:

$$\theta^i(j+1, k, l) = \theta^i(j+1, k, l) + \frac{\alpha}{t(i)} * \left| \frac{u}{|v|^{1/\beta}} \right| * \Delta'(i)$$

$$J(i, j+1, k, l) = f(\theta^i(j+1, k, l))$$

Else

let $m = N_s$.

This is the end of the while statement.

iii) Self-regulation strategy: in order to avoid the worse position slow down the bacterium's activity, we let

$$\theta^i(j+1, k, l) = \theta_{last}, J(i, j+1, k, l) = J_{last}$$

6.7) Swarm:

i) Let $m = 0$ (counter for swarm length).

ii) **While** $m < N_s$

Let $m = m + 1$

The bacteria swarm to the global best solution along one dimension in a Levy flight step-size.

$$\theta^i(j+1, k, l) = \theta^i(j+1, k, l) + (\theta_{gbest} - \theta^i(j+1, k, l)) * \frac{\alpha}{t(i)} * \left| \frac{u}{|v|^{1/\beta}} \right| * e$$

where e is a random D-dimensional base vector.

$$J(i, j+1, k, l) = f(\theta^i(j+1, k, l))$$

If $J(i, j+1, k, l) < J_{last}$

$$\theta_{last} = \theta^i(j+1, k, l), J_{last} = J(i, j+1, k, l)$$

Else

Self-regulation strategy: in order to avoid the worse position slow down the bacterium's activity,

$$\theta^i(j+1, k, l) = \theta_{last}, J(i, j+1, k, l) = J_{last}$$

This is the end of the while statement.

6.8) Update the global best position and the fitness value.

If $J_{last} < J_{best}$

$$J_{best} = J_{last}, \theta_{gbest} = \theta_{last}.$$

6.9) If $i < S$, go to step 6.

step 7. if $j < N_c$, then go to step 5.

step 8. Reproduction:

8.1) As LPBFO uses self-regulation strategy, the health of bacterium i is $J_{i,health} = J(i, N_c, k, l)$

8.2) Sort the bacteria $J_{i,health}$ in descending order. A higher value means that the health of bacterium is worse.

8.3) The S_r healthier bacteria split into two exactly same bacteria and the other S_r bacteria die.

8.4) If $k < N_{re}$, go to step 4.

step 9. Elimination-dispersal:

9.1) For each bacterium $i = 1, 2, \dots, S$ with a random number r_i ,

If $r_i < P_{ed}$

The bacterium i is dispersed to a random location within the optimization domain.

Else

The bacterium i remains at its original position.

9.2) If $l < N_{ed}$, go to step 3; otherwise, output the optimal solution J_{best} .

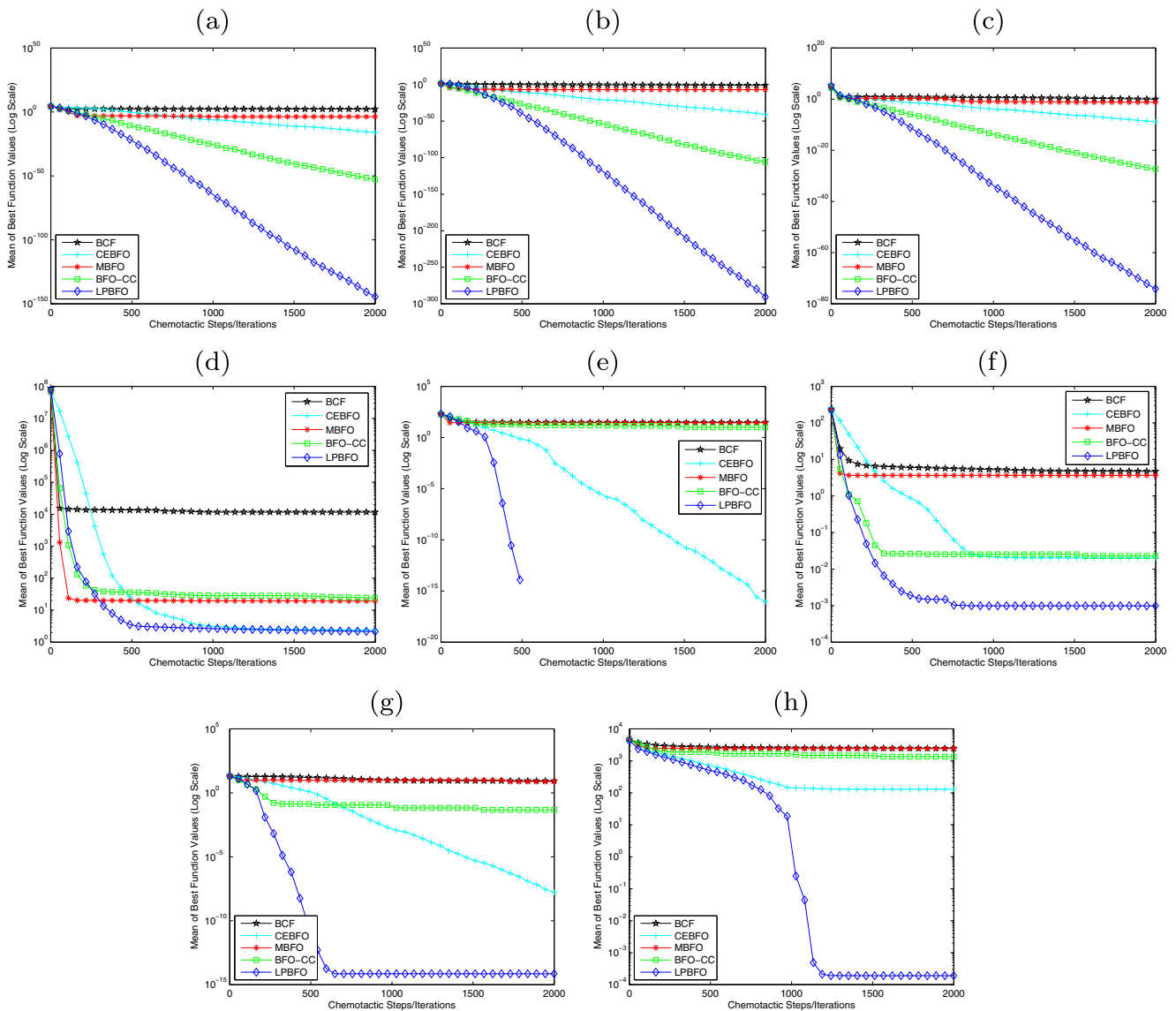


Fig. 2 Comparison of the convergence results of the average optimum value with 15 dimensions. **a** Sphere Function **b** Quartic Function **c** Schwefels Problem 2.22 **d** Rosenbrock Function **e** Rastrigin Function **f** Griewank Function **g** Ackley Function **h** Schwefel’s Problem 2.26

4 Experiments and discussion

To fully test the performance of the LPBFO algorithm, eight benchmark functions shown in Table 1 were used to conduct the experiments. The first four functions are unimodal functions with only one local optimum which is the global optimum; the others are multimodal functions with a large number of local optimum and algorithms may easily fall into a local optimum when intending to solve the functions. The unimodal functions can be used to analysis the intensification capability and convergence rate of the algorithms. The multimodal functions are commonly used to show the exploration capability of the algorithms. In

Table 1, D denotes the dimensions of the solution space and 15, 30, 45, and 60 dimensions are used in the present paper.

The effectiveness of the proposed LPBFO algorithm was evaluated by comparing its results with other related algorithms, such as BCF [6], CEBFO [34], MBFO [27], and BFO-CC [33]. Following the parameter settings in the previous studies, the values of the population size used in each algorithm were chosen to be the same as $S = 50$. To make a fair comparison, we set $N_c = 100$, $N_{re} = 5$, $N_{ed} = 4$ and then the algorithms perform totally $T = N_c \times N_{re} \times N_{ed} = 2000$ iterations times in each run; all algorithms were tested with the maximum

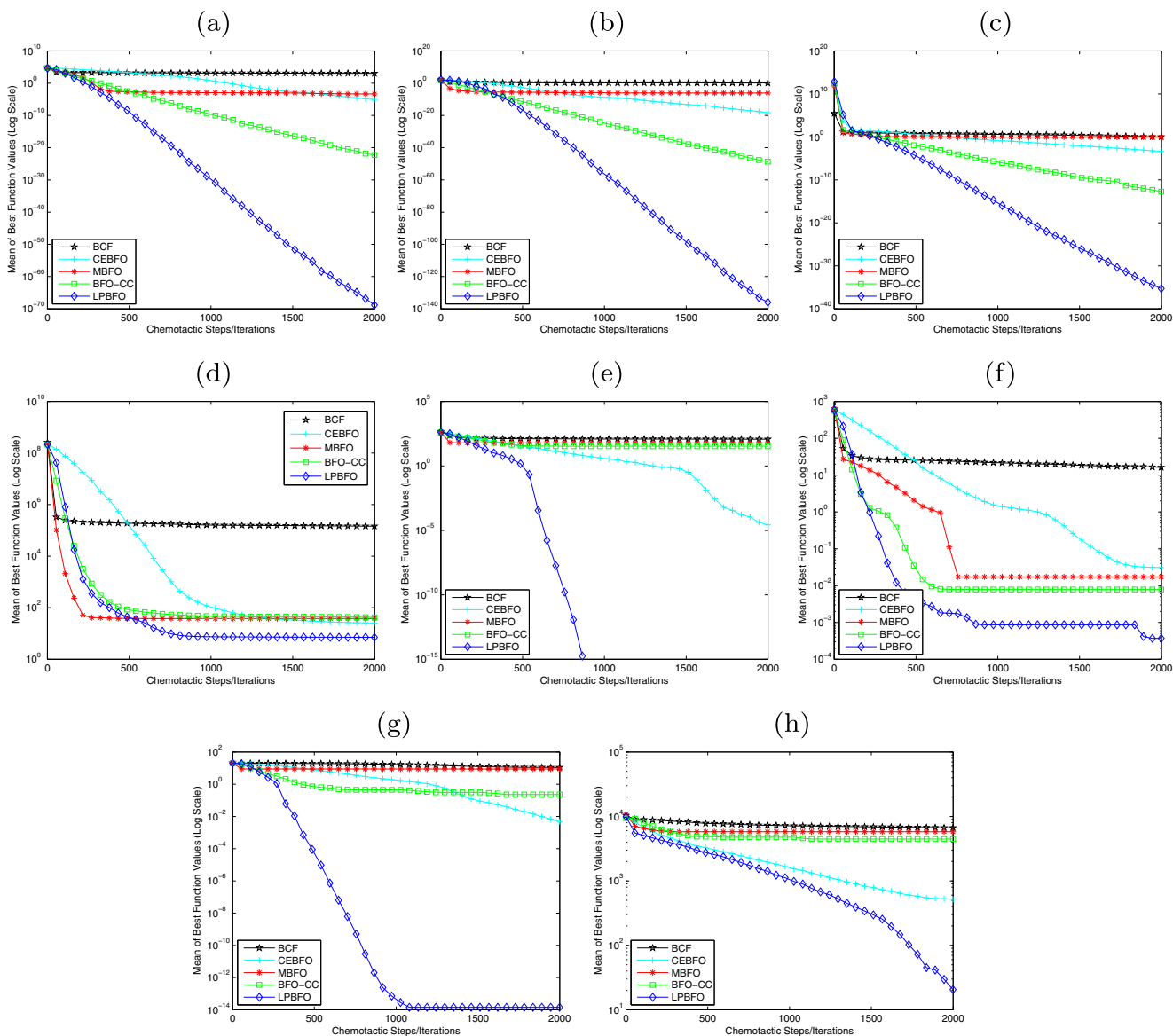


Fig. 3 Comparison of the convergence results of the average optimum value with 30 dimensions. **a** Sphere Function **b** Quartic Function **c** Schwefels Problem 2.22 **d** Rosenbrock Function **e** Rastrigin Function **f** Griewank Function **g** Ackley Function **h** Schwefel's Problem 2.26

number of swimming and swarming length $N_s = 4$. The other detailed parameters of LPBFO algorithm, we take $P_{ed} = 0.25$, $\beta = 1.5$ following the original studies and $\alpha = 1000$ using trial and error. Additionally, as for other specific parameters of each comparison algorithm, we followed the settings in a corresponding original paper.

BCF settings: For the proposed BCF algorithm, we set $k = 0.8$, $\phi_1 = 1.494$, $\phi_2 = 1.494$, $\varepsilon^{initial} = 100$, $K_u = 20$, $\lambda = 10$, and the initialized run-length unit $C^{initial}$ is set to be 1% of the search space.

CEBFO settings: For the CEBFO algorithm, we take $P_{ed} = 0.25$, $CR = 0.9$, $F = 0.5$. The step length C is set for each benchmark function as shown in Table 3 in [34].

MBFO settings: For the MBFO algorithm, we set $P_{ed} = 0.03$, $C(i) = 0.1$, $d_{attract} = 0.1$, $\omega_{attract} = 0.2$, $h_{repellant} = 0.15$, $\omega_{repellant} = 0.1$.

BFO-CC settings: For the BFO-CC algorithm, the parameters are set as $P_{ed} = 0.3$, $P_c = 0.2$, $L = 0.1 * D$, $b = 0.5$.

In order to ensure the experiment results stability, we repeated each algorithm 20 times and averaged the results.

4.1 Experimental results

To analyze the experimental data, this subsection uses five indexes introduced by [33] to evaluate the effectiveness of the proposed method: the best solution (Best), median

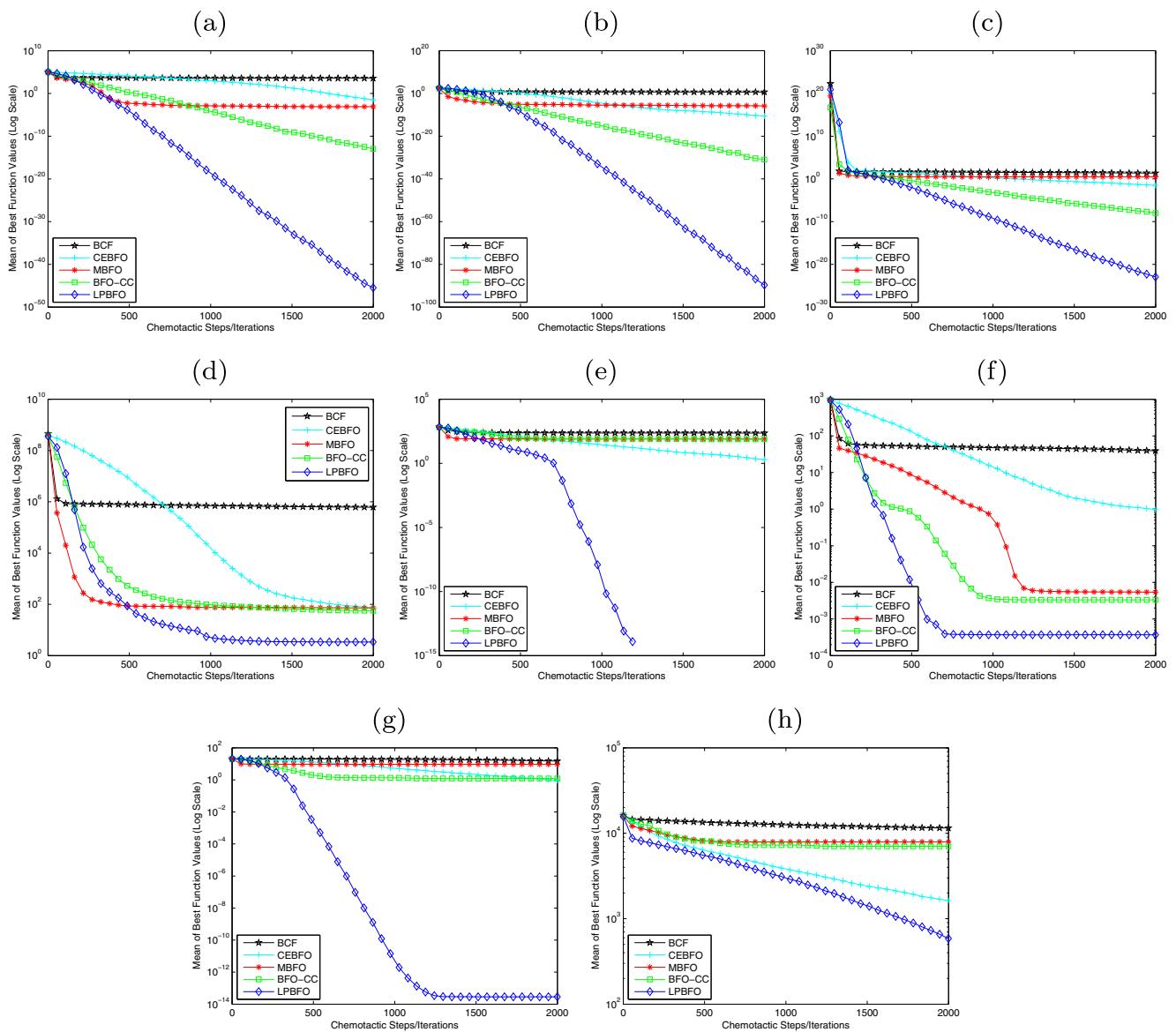


Fig. 4 Comparison of the convergence results of the average optimum value with 45 dimensions. **a** Sphere Function **b** Quartic Function **c** Schwefels Problem 2.22 **d** Rosenbrock Function **e** Rastrigin Function **f** Griewank Function **g** Ackley Function **h** Schwefel’s Problem 2.26

solution (Median), worst solution (Worst), average solution (Mean), and standard deviation (Std). Figures 2, 3, 4, and 5 show how the mean of the best solution through 20 runs for each algorithm changes with the number of iterations times based on 8 basic benchmark functions with 15, 30, 45, 60 dimensions. The lines that do not extend to the end of the experiments indicate that they have converged to 0 in the next calibration.

At the initial phase of the optimization procedure, Figs. 2–5 show that the proposed LPBFO algorithm does not possess obvious advantage on the convergence rate

compared with other algorithms. The reason is that LPBFO algorithm with large step-size realizes global search and does poorly in local search at this stage; the global search are inclined to locate the potential optimal regions and local search are apt to fine-tune the present solutions to improve the convergence rate. Further, the probability of obtaining a better solution is very high at the initial stage for all the algorithms; the search strategy in the proposed LPBFO algorithm, where each bacterium selects one dimension randomly from the search space to tumble, does not have obvious advantages. It can be seen from the Figs. 2–5 that

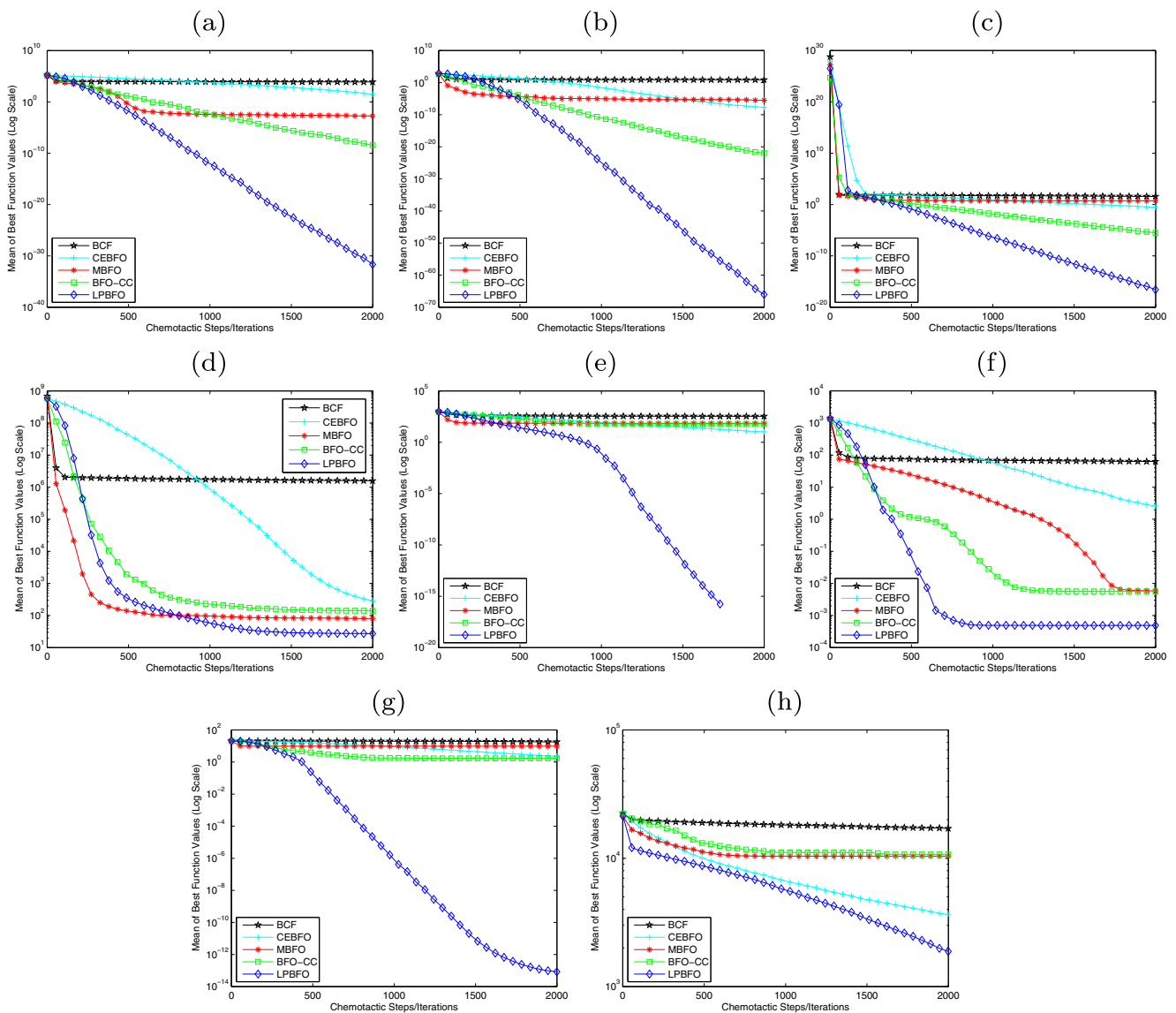


Fig. 5 Comparison of the convergence results of the average optimum value with 60 dimensions. **a** Sphere Function **b** Quartic Function **c** Schwefels Problem 2.22 **d** Rosenbrock Function **e** Rastrigin Function **f** Griewank Function **g** Ackley Function **h** Schwefel's Problem 2.26

the MBFO algorithm converges faster than other algorithms at the initial phase. By incorporating the PSO operator into chemotactic step, the MBFO algorithm quickens the convergence rate effectively and improve the quality of the solution.

As the adaptive Lévy flight step-size reduces along with the increase of the number of iterations, the LPBFO algorithm transforms from global search to local search to fine-tune the present solutions. Moreover, when it is difficult to obtain a better solution at the latter stage of the optimization, the advantages of the proposed search strategy can be well reflected. Therefore, the proposed LPBFO algorithm converges fast and achieves significantly better solution on all benchmark functions than other algorithms in the succeeding phase.

In chemotaxis stage, each bacterium randomly choosing a dimension to move forward improves the search efficiency. By taking into account the information of global best solution, the modified swarming process accelerates the convergence rate and improves the accuracy of the solutions.

Tables 2, 3, 4 and 5 shows the comparison among the experimental results based 8 basic benchmark functions with 15, 30, 45, 60 dimensions. The bold emphases in Tables 2, 3, 4, 5, 7, 8 mean that the corresponding experimental results are the best compared with other algorithms under the corresponding index. The results suggest that LPBFO algorithm offers the higher solution accuracy on all the functions, which proves that LPBFO algorithm is superior to other algorithms.

Table 2 Comparison between LPBFO and other algorithms for 20 times independent runs tested on 8 basic benchmark functions with 15 dimensions

Functions	Metrics	BCF	CEBFO	MBFO	BFO-CC	LPBFO
Sphere Function	Best	1.08e+01	8.57e−18	1.92e−04	1.25e−53	1.43e−149
	Median	1.83e+02	2.92e−17	2.71e−04	1.41e−52	7.72e−146
	Worst	4.40e+02	5.32e−16	3.65e−04	4.61e−50	4.04e−141
	Mean	1.84e+02	7.93e−17	2.69e−04	2.58e−51	2.03e−142
	Std	9.73e+01	1.23e−16	5.32e−05	1.02e−50	9.03e−142
Quartic Function	Best	9.31e−02	1.11e−44	4.49e−08	3.76e−110	2.16e−295
	Median	9.78e−02	4.03e−42	9.87e−08	1.44e−107	4.71e−290
	Worst	1.11e−01	3.46e−41	1.91e−07	1.19e−103	2.08e−282
	Mean	9.87e−02	9.44e−42	1.07e−07	1.14e−104	1.04e−283
	Std	5.20e−03	1.17e−41	3.86e−08	3.30e−104	0.00e+00
Schwefel's Problem 2.22	Best	3.32e−01	4.28e−10	3.78e−02	6.69e−30	2.09e−76
	Median	8.35e−01	1.13e−09	5.94e−02	9.72e−29	8.57e−75
	Worst	2.73e+00	4.44e−09	1.97e+00	4.38e−28	6.35e−73
	Mean	1.02e+00	1.45e−09	1.58e−01	1.43e−28	5.29e−74
	Std	6.35e−01	9.93e−10	4.26e−01	1.21e−28	1.42e−73
Rosenbrock Function	Best	1.23e+03	1.01e−02	8.85e+00	3.96e−02	9.51e−04
	Median	7.13e+03	4.60e−01	1.29e+01	5.13e+00	9.54e−02
	Worst	6.03e+04	1.48e+01	8.78e+01	2.15e+02	9.98e+00
	Mean	1.16e+04	2.41e+00	1.93e+01	2.43e+01	2.16e+00
	Std	1.43e+04	3.81e+00	2.11e+01	5.32e+01	3.51e+00
Rastrigin Function	Best	1.02e+01	0.00e+00	1.00e+01	3.98e+00	0.00e+00
	Median	2.48e+01	0.00e+00	2.64e+01	9.45e+00	0.00e+00
	Worst	6.56e+01	1.78e−15	4.78e+01	2.69e+01	0.00e+00
	Mean	2.95e+01	8.88e−17	2.64e+01	1.08e+01	0.00e+00
	Std	1.51e+01	3.97e−16	1.11e+01	5.43e+00	0.00e+00
Griewank Function	Best	8.80e−01	1.78e−15	2.51e−01	0.00e+00	0.00e+00
	Median	4.57e+00	1.97e−02	8.84e−01	1.48e−02	0.00e+00
	Worst	7.74e+00	4.17e−02	4.48e+00	1.38e−01	9.90e−03
	Mean	4.71e+00	2.02e−02	1.36e+00	2.29e−02	9.86e−03
	Std	1.84e+00	1.34e−02	1.28e+00	3.10e−02	3.00e−03
Ackley Function	Best	5.37e+00	2.63e−09	2.12e+00	7.11e−15	3.55e−15
	Median	8.69e+00	1.31e−08	5.62e+00	1.60e−14	7.11e−15
	Worst	1.12e+01	4.84e−08	9.53e+00	9.41e−01	7.11e−15
	Mean	8.64e+00	1.64e−08	5.85e+00	4.70e−02	6.75e−15
	Std	1.20e+00	1.19e−08	2.21e+00	2.10e−01	1.09e−15
Schwefel's Problem 2.26	Best	1.24e+03	1.91e−04	1.90e+03	7.70e+02	1.91e−04
	Median	2.46e+03	1.18e+02	2.69e+03	1.36e+03	1.91e−04
	Worst	3.24e+03	3.55e+02	3.82e+03	1.74e+03	1.91e−04
	Mean	2.46e+03	1.30e+02	2.80e+03	1.36e+03	1.91e−04
	Std	4.72e+02	1.08e+02	5.24e+02	2.60e+02	0.00e+00

4.2 Comparison regarding the t-test

To investigate whether the experimental results obtained by LPBFO algorithm are statistically significantly different

from results achieved by other algorithms, the F-test and t-test were performed in this section. The F-test was used to analyze the homogeneity of variances and t-test was used to determine if two sets of data are significantly

Table 3 Comparison between LPBFO and other algorithms for 20 times independent runs tested on 8 basic benchmark functions with 30 dimensions

Functions	Metrics	BCF	CEBFO	MBFO	BFO-CC	LPBFO
Sphere Function	Best	5.81e+02	5.87e−07	3.22e−04	1.09e−24	1.61e−71
	Median	9.88e+02	5.97e−06	5.67e−04	6.16e−24	1.73e−70
	Worst	3.41e+03	2.77e−05	7.55e−04	5.35e−23	2.46e−67
	Mean	1.21e+03	8.62e−06	5.62e−04	1.35e−23	1.28e−68
	Std	6.79e+02	6.61e−06	1.26e−04	1.48e−23	5.49e−68
Quartic Function	Best	5.12e−01	8.16e−20	3.69e−07	1.30e−51	3.30e−143
	Median	1.25e0+00	3.84e−19	5.96e−07	6.51e−50	4.98e−139
	Worst	2.72e+00	2.01e−18	1.42e−06	1.80e−48	1.69e−135
	Mean	1.42e+00	5.77e−19	7.17e−07	1.87e−49	9.35e−137
	Std	6.32e−01	5.31e−19	3.24e−07	4.07e−49	3.77e−136
Schwefel's Problem 2.22	Best	2.56e+00	1.42e−04	2.49e−01	4.64e−14	4.60e−37
	Median	8.31e+00	3.66e−04	2.05e+00	1.10e−13	2.22e−36
	Worst	2.83e+01	1.09e−03	6.18e+00	2.12e−13	1.75e−35
	Mean	9.16e+00	4.20e−04	2.47e+00	1.20e−13	4.81e−36
	Std	5.28e+00	2.16e−04	1.59e+00	4.52e−14	5.38e−36
Rosenbrock Function	Best	2.12e+04	8.12e−01	2.14e+01	3.20e−03	1.09e−04
	Median	1.19e+05	2.21e+01	2.85e+01	3.14e+00	9.70e−01
	Worst	7.18e+05	9.05e+01	1.10e+02	1.26e+02	7.00e+01
	Mean	1.46e+05	2.57e+01	3.80e+01	4.20e+01	7.13e+00
	Std	1.48e+05	2.26e+01	2.69e+01	4.14e+01	1.61e+01
Rastrigin Function	Best	4.21e+01	1.84e−06	2.50e+01	1.79e+01	0.00e+00
	Median	1.23e+02	2.02e−05	5.33e+01	2.89e+01	0.00e+00
	Worst	1.72e+02	9.64e−05	9.46e+01	4.88e+01	0.00e+00
	Mean	1.19e+02	2.70e−05	5.81e+01	3.09e+01	0.00e+00
	Std	2.63e+01	2.57e−05	2.20e+01	9.21e+00	0.00e+00
Griewank Function	Best	8.98e+00	2.38e−04	4.13e−05	0.00e+00	0.00e+00
	Median	1.48e+01	2.60e−02	1.73e−02	3.70e−03	0.00e+00
	Worst	3.10e+01	9.16e−02	3.94e−02	3.94e−02	7.40e−03
	Mean	1.63e+01	3.09e−02	1.73e−02	7.88e−03	3.70e−04
	Std	6.02e+00	2.65e−02	1.26e−02	1.06e−02	1.70e−03
Ackley Function	Best	7.02e+00	1.96e−03	6.95e+00	1.83e−12	1.42e−14
	Median	1.09e+01	4.16e−03	9.45e+00	8.95e−12	1.78e−14
	Worst	1.35e+01	1.04e−02	1.12e+01	1.34e+00	2.84e−14
	Mean	1.09e+01	4.77e−03	9.36e+00	2.29e−01	1.83e−14
	Std	1.64e+00	2.33e−03	1.24e+00	4.75e−01	4.04e−15
Schwefel's Problem 2.26	Best	5.67e+03	1.57e+02	4.24e+03	3.28e+03	3.82e−04
	Median	6.95e+03	4.75e+02	5.41e+03	3.88e+03	4.38e−04
	Worst	7.52e+03	9.49e+02	8.44e+03	5.61e+03	1.18e+02
	Mean	6.67e+03	5.21e+02	5.76e+03	4.03e+03	2.06e+01
	Std	5.95e+02	2.02e+02	1.28e+03	5.79e+02	4.40e+01

different from each other. The confidence level was set to 95%, which means that the probability of receiving a difference by chance is not more than 5%. In the F-test, if the probability p is less than 0.05, the corresponding

experimental results are heterogeneity of variance. If the probability p achieved in the t-test is less than 0.05, we can believe that there exists a significant difference between the corresponding experimental results. Table 6 lists the results

Table 4 Comparison between LPBFO and other algorithms for 20 times independent runs tested on 8 basic benchmark functions with 45 dimensions

Functions	Metrics	BCF	CEBFO	MBFO	BFO-CC	LPBFO
Sphere Function	Best	1.59e+03	1.45e−02	5.52e−04	5.61e−15	3.74e−46
	Median	3.40e+03	3.13e−02	1.00e−03	3.15e−14	5.97e−45
	Worst	7.30e+03	5.83e−02	1.30e−03	1.82e−13	8.48e−44
	Mean	3.68e+03	3.17e−02	9.98e−04	4.96e−14	1.17e−44
	Std	1.51e+03	1.16e−02	1.89e−04	4.88e−14	1.83e−44
Quartic Function	Best	1.29e+00	4.21e−12	9.00e−07	1.09e−32	5.93e−92
	Median	5.02e+00	2.16e−11	1.72e−06	2.20e−31	4.82e−90
	Worst	6.42e+00	1.02e−10	4.12e−06	1.44e−30	2.45e−87
	Mean	4.59e+00	2.53e−11	1.86e−06	3.11e−31	1.59e−88
	Std	1.41e+00	2.21e−11	6.88e−07	3.93e−31	5.42e−88
Schwefel's Problem 2.22	Best	1.06e+01	2.15e−02	2.04e+00	7.20e−09	2.77e−24
	Median	2.30e+01	2.82e−02	3.45e+00	1.68e−08	1.29e−23
	Worst	4.43e+01	4.33e−02	1.11e+01	4.34e−08	5.29e−23
	Mean	2.28e+01	3.07e−02	4.20e+00	1.93e−08	1.52e−23
	Std	9.54e+00	6.32e−03	2.29e+00	9.95e−09	1.23e−23
Rosenbrock Function	Best	2.34e+05	1.81e+01	3.61e+01	4.72e+00	3.00e−03
	Median	4.61e+05	7.06e+01	4.36e+01	1.72e+01	1.25e+00
	Worst	2.29e+06	1.58e+02	2.94e+02	2.29e+02	9.36e+01
	Mean	6.12e+05	7.31e+01	7.23e+01	5.61e+01	1.51e+01
	Std	4.82e+05	3.67e+01	5.96e+01	6.74e+01	2.89e+01
Rastrigin Function	Best	1.90e+02	8.38e−03	4.39e+01	3.58e+01	0.00e+00
	Median	2.29e+02	2.01e+00	8.42e+01	5.27e+01	0.00e+00
	Worst	2.97e+02	3.01e+00	1.31e+02	7.86e+01	0.00e+00
	Mean	2.29e+02	2.01e+00	8.47e+01	5.49e+01	0.00e+00
	Std	2.85e+01	8.55e−01	2.08e+01	1.02e+01	0.00e+00
Griewank Function	Best	1.63e+01	7.31e−01	9.04e−05	4.81e−14	0.00e+00
	Median	3.66e+01	1.02e+00	1.49e−04	1.67e−12	0.00e+00
	Worst	6.38e+01	1.09e+00	2.23e−02	1.23e−02	7.40e−03
	Mean	3.91e+01	9.64e−01	5.40e−03	3.30e−03	3.70e−04
	Std	1.32e+01	1.21e−01	7.30e−03	4.90e−03	1.70e−03
Ackley Function	Best	1.13e+01	2.95e−01	8.41e+00	1.66e−07	2.13e−14
	Median	1.45e+01	9.68e−01	1.03e+01	1.22e+00	3.02e−14
	Worst	1.83e+01	1.36e+00	1.31e+01	1.82e+00	4.26e−14
	Mean	1.49e+01	9.18e−01	1.03e+01	1.04e+00	2.88e−14
	Std	2.09e+00	3.20e−01	1.18e+00	5.74e−01	6.20e−15
Schwefel's Problem 2.26	Best	1.00e+04	1.26e+03	7.21e+03	5.41e+03	3.33e+02
	Median	1.16e+04	1.56e+03	8.33e+03	7.48e+03	5.61e+02
	Worst	1.27e+04	2.50e+03	1.26e+04	8.63e+03	9.49e+02
	Mean	1.15e+04	1.63e+03	8.65e+03	7.40e+03	5.84e+02
	Std	6.88e+02	3.02e+02	1.26e+03	7.95e+02	1.59e+02

of F-test and t-tests between LPBFO and the best results of the other algorithms regarding the indexes “Mean” for different benchmark functions (listed are the p(F-test), the p(t-test), and the significance of the results). In Table 6, the

sample size and number of degrees of freedom were set as 20 and 38, respectively. “Yes” means that the results of LPBFO and of other algorithms are significantly different. “No” indicates that there is not a significant difference

Table 5 Comparison between LPBFO and other algorithms for 20 times independent runs tested on 8 basic benchmark functions with 60 dimensions

Functions	Metrics	BCF	CEBFO	MBFO	BFO-CC	LPBFO
Sphere Function	Best	3.97e+03	1.10e+01	1.20e−03	3.51e−10	2.26e−33
	Median	7.79e+03	3.07e+01	1.60e−03	9.61e−10	2.37e−32
	Worst	1.15e+04	5.37e+01	2.20e−03	6.08e−09	8.71e−32
	Mean	7.50e+03	3.01e+01	1.60e−03	1.52e−09	3.04e−32
	Std	2.40e+03	1.18e+01	2.76e−04	1.55e−09	2.77e−32
Quartic Function	Best	2.12e+00	1.40e−08	2.00e−06	6.51e−24	4.28e−67
	Median	8.14e+00	1.77e−08	5.94e−06	8.37e−23	3.56e−65
	Worst	9.59e+00	2.32e−08	7.41e−06	1.28e−21	1.74e−63
	Mean	7.56e+00	1.84e−08	5.55e−06	2.07e−22	2.74e−64
	Std	1.85e+00	2.96e−09	1.67e−06	3.25e−22	5.16e−64
Schwefel's Problem 2.22	Best	1.75e+01	1.90e−01	3.61e+00	1.82e−06	1.77e−17
	Median	3.87e+01	2.88e−01	7.01e+00	3.13e−06	3.14e−17
	Worst	5.35e+01	4.25e−01	1.43e+01	4.60e−06	8.41e−17
	Mean	3.85e+01	3.04e−01	7.72e+00	3.04e−06	3.56e−17
	Std	9.24e+00	5.08e−02	2.92e+00	7.17e−07	1.82e−17
Rosenbrockf Function	Best	3.96e+05	1.97e+02	5.31e+01	9.09e+00	3.67e−01
	Median	1.66e+06	2.81e+02	5.84e+01	8.34e+01	1.17e+01
	Worst	2.78e+06	3.53e+02	2.04e+02	3.50e+02	1.43e+02
	Mean	1.57e+06	2.85e+02	8.09e+01	1.14e+02	2.74e+01
	Std	6.66e+05	4.38e+01	3.84e+01	9.29e+01	3.77e+01
Rastrigin Function	Best	2.75e+02	7.03e+00	5.31e+01	4.88e+01	0.00e+00
	Median	3.36e+02	1.10e+01	9.98e+01	7.01e+01	0.00e+00
	Worst	4.05e+02	1.30e+01	1.66e+02	1.00e+02	0.00e+00
	Mean	3.39e+02	1.05e+01	1.00e+02	7.11e+01	0.00e+00
	Std	4.12e+01	1.43e+00	2.93e+01	1.61e+01	0.00e+00
Griewank Function	Best	3.20e+01	1.67e+00	1.88e−04	1.01e−09	0.00e+00
	Median	5.96e+01	2.62e+00	4.10e−03	7.40e−03	0.00e+00
	Worst	1.08e+02	3.47e+00	1.75e−02	2.22e−02	9.90e−03
	Mean	6.31e+01	2.57e+00	5.90e−03	5.70e−03	4.93e−03
	Std	2.17e+01	4.75e−01	6.10e−03	6.20e−03	2.20e−03
Ackley Function	Best	1.37e+01	2.02e+00	8.78e+00	1.93e−05	4.97e−14
	Median	1.79e+01	2.29e+00	1.04e+01	1.71e+00	8.53e−14
	Worst	1.94e+01	2.52e+00	1.26e+01	2.17e+00	1.17e−13
	Mean	1.77e+01	2.28e+00	1.04e+01	1.50e+00	8.37e−14
	Std	1.31e+00	1.38e−01	9.07e−01	5.47e−01	1.89e−14
Schwefel's Problem	Best	1.56e+04	3.07e+03	8.73e+03	9.40e+03	1.39e+03
	Median	1.69e+04	3.58e+03	1.15e+04	1.06e+04	1.88e+03
	Worst	1.89e+04	4.41e+03	1.61e+04	1.26e+04	2.52e+03
	Mean	1.71e+04	3.65e+03	1.17e+04	1.07e+04	1.88e+03
	Std	8.94e+02	3.38e+02	1.72e+03	7.97e+02	3.23e+02

between the results of LPBFO and the best results of the other algorithms.

From Tables 2–5, LPBFO algorithm has higher solution accuracy on all the functions. Based on the results of

statistical tests in Table 6, almost all of the results have obvious differences and it is clear that the results of LPBFO algorithm are significantly better than the results of other algorithms.

Table 6 Comparisons between LPBFO's results and the best results of the other algorithms on F-test and t-test

Functions	Dim	p(F-test)	p(t-test)	significance	Two-tailed P
Sphere Function	D=15	0.042	0.273	NO	0.05
	D=30	2.00e-06	6.58e-04	YES	0.05
	D=45	1.20e-05	2.20e-04	YES	0.05
	D=60	1.50e-04	3.17e-04	YES	0.05
Quartic Function	D=15	3.00e-03	0.137	NO	0.05
	D=30	6.83e-03	0.054	YES	0.05
	D=45	7.80e-05	2.21e-03	YES	0.05
	D=60	1.82e-04	0.010	YES	0.05
Schwefel's Problem 2.22	D=15	7.63e-08	3.90e-05	YES	0.05
	D=30	3.50e-07	2.99e-10	YES	0.05
	D=45	7.00e-06	4.96e-08	YES	0.05
	D=60	6.84e-07	8.15e-14	YES	0.05
Rosenbrock Function	D=15	0.884	0.827	NO	0.05
	D=30	0.513	0.005	YES	0.05
	D=45	6.76e-04	0.019	YES	0.05
	D=60	6.01e-03	7.71e-04	YES	0.05
Rosenbrock Function	D=15	0.041	0.330	NO	0.05
	D=30	2.30e-05	1.56e-04	YES	0.05
	D=45	5.70e-05	2.40e-09	YES	0.05
	D=60	2.00e-06	3.18e-18	YES	0.05
Griewank Function	D=15	1.40e-05	3.00e-06	YES	0.05
	D=30	3.00e-05	5.22e-03	YES	0.05
	D=45	1.47e-07	0.017	YES	0.05
	D=60	8.61e-03	0.099	NO	0.05
Ackley Function	D=15	4.65e-04	6.00e-06	YES	0.05
	D=30	5.38e-08	2.05e-08	YES	0.05
	D=45	1.00e-06	8.18e-11	YES	0.05
	D=60	1.10e-05	1.77e-10	YES	0.05
Schwefel's Problem 2.26	D=15	2.39e-07	3.30e-05	YES	0.05
	D=30	5.90e-05	5.43e-10	YES	0.05
	D=45	0.024	3.85e-14	YES	0.05
	D=60	0.630	2.83e-19	YES	0.05

4.3 Comparison with other algorithms

To simulate the swarm behavior in birds flocking and fish schooling, Kennedy et al. proposed the PSO algorithm [10]. Inspired by the swarm foraging behavior in honey bees, Karaboga proposed the artificial bee colony (ABC) algorithm [9]. Both PSO and ABC, which can be used to guide the particles to search for globally optimal solutions, have received much attention and achieved better results in solving the optimization problems.

In order to effectively solve the problem that the PSO algorithm easily falls into local optima, Li et al. proposed a self-learning PSO (SLPSO) algorithm, where four search strategies are used and each particle chooses the optimal

strategy based on the adaptive learning framework [12]. To improve the performance of PSO, Lai et al. proposed a parallel PSO based on osmosis (PPBO), where the whole particle population is divided into several subpopulations and each subpopulation achieves migration based on osmosis [11]. In order to solve the stagnation behavior in ABC algorithm, Babaoglu proposed distribution-based ABC (distABC) algorithm, which uses the mean and standard deviation of the selected two solutions to obtain a new candidate solution [1]. Xue et al. proposed a self-adaptive ABC algorithm based on the global best candidate (SABC-GB), where several different candidate solution generating strategies and self-adaptive strategies are used to balance the rate of convergence and robustness of the algorithm [32].

Table 7 Comparison between LPBFO and other algorithms for 20 times independent runs tested on 8 basic benchmark functions with 30 dimensions

Functions	Metrics	SLPSO	PPBO	distABC	SABC-GB	LPBFO
Sphere Function	Best	1.79e-38	3.28e+02	2.68e-39	1.37e-53	1.61e-71
	Median	6.42e-36	8.38e+02	1.08e-36	6.32e-53	1.73e-70
	Worst	7.52e-33	1.46e+03	1.23e-29	3.56e-52	2.46e-67
	Mean	4.91e-34	8.64e+02	1.28e-30	8.14e-53	1.28e-68
	Std	1.68e-33	2.84e+02	3.52e-30	8.49e-53	5.49e-68
Quartic Function	Best	4.17e-81	4.14e-03	1.19e-82	3.37e-111	3.30e-143
	Median	2.25e-75	3.64e-02	9.12e-77	6.20e-110	4.98e-139
	Worst	2.24e-70	1.04e-01	2.85e-51	3.92e-108	1.69e-135
	Mean	2.24e-71	4.13e-02	1.43e-52	3.89e-109	9.35e-137
	Std	6.53e-71	2.53e-02	6.38e-52	8.82e-109	3.77e-136
Schwefel's Problem 2.22	Best	6.09e-20	6.33e+00	8.78e-21	2.00e-28	4.60e-37
	Median	5.33e-19	2.15e+01	1.70e-19	5.22e-28	2.22e-36
	Worst	6.63e-17	4.22e+01	1.66e-16	1.19e-27	1.75e-35
	Mean	5.77e-18	2.29e+01	2.02e-17	5.27e-28	4.81e-36
	Std	1.49e-17	9.55e+00	4.45e-17	2.53e-28	5.38e-36
Rosenbrock Function	Best	1.04e-29	3.55e+03	3.08e-03	2.70e-02	1.09e-04
	Median	9.80e-22	4.69e+04	3.46e-01	1.04e+00	9.70e-01
	Worst	8.12e-07	1.74e+05	3.59e+00	8.47e+01	7.00e+01
	Mean	4.01e-08	5.23e+04	7.61e-01	1.11e+01	7.13e+00
	Std	1.82e-07	4.47e+04	1.05e+00	2.18e+01	1.61e+01
Rastrigin Function	Best	0.00e+00	6.49e+01	0.00e+00	0.00e+00	0.00e+00
	Median	0.00e+00	1.13e+02	0.00e+00	0.00e+00	0.00e+00
	Worst	0.00e+00	1.73e+02	0.00e+00	0.00e+00	0.00e+00
	Mean	0.00e+00	1.12e+02	0.00e+00	0.00e+00	0.00e+00
	Std	0.00e+00	2.57e+01	0.00e+00	0.00e+00	0.00e+00
Griewank Function	Best	0.00e+00	3.82e+00	0.00e+00	0.00e+00	0.00e+00
	Median	1.48e-02	7.76e+00	0.00e+00	0.00e+00	0.00e+00
	Worst	8.06e-02	1.46e+01	5.55e-16	7.66e-03	7.40e-03
	Mean	1.96e-02	7.86e+00	2.78e-17	3.83e-04	3.70e-04
	Std	2.15e-02	2.68e+00	1.24e-16	1.71e-03	1.70e-03
Ackley Function	Best	3.20e-14	6.45e+00	3.20e-14	2.49e-14	1.42e-14
	Median	4.26e-14	8.86e+00	3.91e-14	2.84e-14	1.78e-14
	Worst	5.68e-14	1.04e+01	4.97e-14	3.91e-14	2.84e-14
	Mean	4.16e-14	8.76e+00	4.12e-14	3.02e-14	1.83e-14
	Std	6.93e-15	1.00e+00	4.52e-15	4.19e-15	4.04e-15
Schwefel's Problem 2.26	Best	2.11e-02	4.48e+03	3.82e-04	3.82e-04	3.82e-04
	Median	4.68e+00	6.61e+03	3.82e-04	3.82e-04	4.38e-04
	Worst	3.76e+01	8.48e+03	7.86e-04	3.82e-04	1.18e+02
	Mean	7.18e+00	6.67e+03	4.19e-04	3.82e-04	2.06e+01
	Std	8.60e+00	1.09e+03	1.03e-04	5.60e-13	4.40e+01

Therefore, to further analyze the competitiveness of LPBFO algorithm, this section compares LPBFO with the state-of-the-art bionic algorithm: SLPSO [12], PPBO [11], distABC [1], SABC-GB [32].

All the algorithms were tested on eight benchmark functions with 30, 60 dimensions and each algorithm were repeated for 20 times. To make a fair comparison, the values of the population size used in each algorithm were chosen to

Table 8 Comparison between LPBFO and other algorithms for 20 times independent runs tested on 8 basic benchmark functions with 60 dimensions

Functions	Metrics	SLPSO	PPBO	distABC	SABC-GB	LPBFO
Sphere Function	Best	1.63e−21	1.79e+03	1.54e−16	1.65e−23	2.26e−33
	Median	1.46e−19	3.39e+03	7.13e−16	6.16e−23	2.37e−32
	Worst	1.77e−17	4.70e+03	2.37e−14	2.18e−22	8.71e−32
	Mean	1.70e−18	3.30e+03	2.08e−15	8.37e−23	3.04e−32
	Std	4.10e−18	7.90e+02	5.17e−15	6.02e−23	2.77e−32
Quartic Function	Best	1.02e−49	2.11e−01	9.48e−39	3.40e−51	4.28e−67
	Median	5.69e−44	6.40e−01	1.23e−37	3.88e−50	3.56e−65
	Worst	2.33e−39	1.89e+00	1.10e−33	1.34e−49	1.74e−63
	Mean	1.33e−40	6.97e−01	5.58e−35	4.24e−50	2.74e−64
	Std	5.20e−40	3.99e−01	2.46e−34	3.58e−50	5.16e−64
Schwefel's Problem 2.22	Best	5.36e−11	3.79e+01	6.33e−09	1.04e−12	1.77e−17
	Median	3.01e−10	6.07e+01	9.95e−09	1.57e−12	3.14e−17
	Worst	2.06e−09	1.21e+02	9.72e−06	2.78e−12	8.41e−17
	Mean	5.04e−10	6.23e+01	4.98e−07	1.59e−12	3.56e−17
	Std	5.40e−10	1.90e+01	2.17e−06	4.51e−13	1.82e−17
Rosenbrock Function	Best	8.46e−12	1.89e+05	9.56e−01	1.58e−01	3.67e−01
	Median	1.31e−02	4.78e+05	3.58e+00	1.00e+01	1.17e+01
	Worst	8.23e+01	9.21e+05	3.90e+01	9.48e+01	1.43e+02
	Mean	5.27e+00	4.71e+05	5.92e+00	3.08e+01	2.74e+01
	Std	1.83e+01	1.86e+05	8.33e+00	3.42e+01	3.77e+01
Rastrigin Function	Best	0.00e+00	2.84e+02	1.72e−05	0.00e+00	0.00e+00
	Median	0.00e+00	3.25e+02	9.95e−01	0.00e+00	0.00e+00
	Worst	0.00e+00	3.78e+02	2.99e+00	1.99e+00	0.00e+00
	Mean	0.00e+00	3.33e+02	8.78e−01	1.99e−01	0.00e+00
	Std	0.00e+00	3.06e+01	7.99e−01	5.21e−01	0.00e+00
Griewank Function	Best	0.00e+00	2.28e+01	2.33e−15	0.00e+00	0.00e+00
	Median	0.00e+00	3.32e+01	3.28e−14	0.00e+00	0.00e+00
	Worst	2.70e−02	4.83e+01	1.82e−10	6.56e−10	9.90e−03
	Mean	6.77e−03	3.32e+01	1.17e−11	3.32e−11	4.93e−03
	Std	9.73e−03	6.44e+00	4.13e−11	1.47e−10	2.20e−03
Ackley Function	Best	8.65e−12	8.15e+00	1.56e−08	1.05e−11	4.97e−14
	Median	1.87e−10	1.00e+01	3.98e−08	1.71e−11	8.53e−14
	Worst	6.93e−10	1.23e+01	2.92e−07	3.94e−11	1.17e−13
	Mean	2.17e−10	1.02e+01	7.11e−08	1.92e−11	8.37e−14
	Std	1.83e−10	1.05e+00	7.56e−08	7.59e−12	1.89e−14
Schwefel's Problem 2.26	Best	2.11e+00	1.33e+04	5.85e+02	7.64e−04	1.39e+03
	Median	2.21e+01	1.52e+04	8.50e+02	7.64e−04	1.88e+03
	Worst	5.19e+01	1.91e+04	1.10e+03	1.18e+02	2.52e+03
	Mean	2.13e+01	1.58e+04	8.47e+02	5.92e+00	1.88e+03
	Std	1.67e+01	1.78e+03	1.51e+02	2.65e+01	3.23e+02

be the same as $S = 50$ and the algorithms performed totally 2000 iterations times in each run expect SLPSO algorithm. In SLPSO algorithm, the number of iterations times was

set to 400000 times. Additionally, as for other specific parameters of each comparison algorithm, we followed the settings in a corresponding original paper.

Table 9 Results of t-tests between LPBFO and the best results of the other algorithms regarding the indexes “Mean” for different benchmark functions

Functions	Dim	p(F-test)	p(t-test)	significance	Two-tailed P
Sphere	D=30	1.43e-04	3.98e-04	YES+	0.05
Function	D=60	1.00e-06	6.00e-06	YES+	0.05
Quartic	D=30	0.005	0.063	NO+	0.05
Function	D=60	2.00e-06	4.20e-05	YES+	0.05
Schwefel's	D=30	1.40e-05	1.63e-08	YES+	0.05
Problem 2.22	D=60	7.41e-07	2.35e-12	YES+	0.05
Rosenbrock	D=30	0.002	0.062	NO-	0.05
Function	D=60	0.008	0.026	YES-	0.05
Rosenbrock	D=30	—	—	SAME	0.05
Function	D=60	—	—	SAME	0.05
Griewank	D=30	0.041	0.330	NO-	0.05
Function	D=60	0.041	0.330	NO-	0.05
Ackley	D=30	0.559	1.52e-11	YES+	0.05
Function	D=60	3.32e-07	7.72e-10	YES+	0.05
Schwefel's	D=30	7.00e-06	0.051	NO-	0.05
Problem 2.26	D=60	7.20e-05	2.06e-16	YES-	0.05

SLPSO settings: The inertia weight $\omega = 0.9$, acceleration coefficient $\eta = 1.496$, the minimum selection ratio for each operator $\gamma = 0.01$, the initial learning probability $P_l = 1.0$, the initial update frequency $U_f = 10$.

PPBO settings: The acceleration coefficient $c_1 = c_2 = 2$; $w^s = 0.9$, $w^e = 0.4$; the topology of PPBO is (5,10), which is a ring consisting of 5 subpopulations, each containing 10 particles.

distABC settings: The control parameter $limit = D \times S/2$.

SABC-GB settings: The control parameter $limit = 0.6 * (S/2) * D$; $LP = 10$.

Tables 7 and 8 show the comparison among the experimental results based on 8 basic benchmark functions with 30, 60 dimensions. From the experimental results, the proposed LPBFO algorithm converges fast and achieves significantly better results in comparison with the best results of the other four algorithms on 4 benchmark functions: Sphere, Quartic, Schwefel's Problem 2.22 and Ackley function. The results shown in Tables 7 and 8 indicate that the SLPSO, distABC, SABC-GB and LPBFO can converge to 0 on Rastrigin function, which means that these algorithms all can obtain the best result. Only on three functions are the results of LPBFO are worse than the best results achieved by the other algorithms: Rosenbrock (SLPSO), Griewank (distABC), Schwefel's Problem 2.26 (SABC-GB).

It can be seen from the formulation of Rosenbrock function that the first term $100(x_{i+1} - x_i)^2$ mainly influences

the function value, which means that only the values among different dimensions in the candidate solutions are almost the same, the function value is smaller. Therefore, by choosing one dimension to updating the candidate solution, LPBFO failed to achieve the best results. However, by taking the global optimal solution (gbest) and the local optimal solution (pbest) into account, SLPSO uses different strategies to guide particles to move to cope with different situations in the search space. Therefore, based on the self-learning strategy and communication between particles, SLPSO algorithm gets the best results on Rosenbrock function.

The Griewank function and Schwefel's Problem 2.26 are multimodal functions, where algorithms may easily fall into a local optimum when intending to solve the functions. The distABC algorithm uses the mean and standard deviation of the selected two solution to obtain a new candidate solution. The SABC-GB algorithm uses several different candidate solution generating strategies and self-adaptive strategies to obtain a new candidate solution. Both algorithms use multiple candidate solutions and use the communication between particles, which effectively reduce the probability of algorithm falling into local optimum. As a result, combining multiple solutions information to find the global optimum, distABC and SABC-GB have achieved the best results on Griewank function and Schwefel's Problem 2.26.

In general, it is obvious that the proposed LPBFO algorithm yields competitive results in comparison with the best results of the other four algorithms.

Table 10 Comparison between LPBFO and other LPBFO variants for 20 times independent runs tested on 8 basic benchmark functions with 30 dimensions

Functions	Metrics	LPBFO1	LPBFO2	LPBFO3	LPBFO
Sphere Function	Best	1.41e−34	1.53e−06	5.81e−55	1.61e−71
	Median	7.08e−34	2.01e−06	5.92e−53	1.73e−70
	Worst	7.62e−33	3.38e−06	3.41e−52	2.46e−67
	Mean	1.86e−33	2.24e−06	1.00e−52	1.28e−68
	Std	2.40e−33	5.78e−07	1.16e−52	5.49e−68
Quartic Function	Best	1.15e−74	2.86e−11	1.57e−114	3.30e−143
	Median	7.20e−73	1.11e−10	4.02e−112	4.98e−139
	Worst	2.47e−72	3.56e−10	2.53e−103	1.69e−135
	Mean	8.41e−73	1.45e−10	2.54e−104	9.35e−137
	Std	8.88e−73	1.07e−10	8.00e−104	3.77e−136
Schwefel's Problem 2.22	Best	6.30e−19	5.41e−02	1.11e−29	4.60e−37
	Median	2.28e−18	5.63e−01	1.53e−27	2.22e−36
	Worst	4.93e−18	4.55e+00	6.01e−27	1.75e−35
	Mean	2.12e−18	1.25e+00	1.69e−27	4.81e−36
	Std	1.22e−18	1.62e+00	1.89e−27	5.38e−36
Rosenbrock Function	Best	4.08e−02	2.30e+01	8.65e−05	1.09e−04
	Median	1.39e+01	2.79e+01	2.59e+00	9.70e−01
	Worst	2.15e+01	9.54e+01	1.92e+01	7.00e+01
	Mean	1.11e+01	3.90e+01	5.77e+00	7.13e+00
	Std	9.34e+00	2.59e+01	6.85e+00	1.61e+01
Rastrigin Function	Best	0.00e+00	3.58e+01	0.00e+00	0.00e+00
	Median	0.00e+00	5.92e+01	0.00e+00	0.00e+00
	Worst	0.00e+00	1.00e+02	0.00e+00	0.00e+00
	Mean	0.00e+00	6.44e+01	0.00e+00	0.00e+00
	Std	0.00e+00	2.17e+01	0.00e+00	0.00e+00
Griewank Function	Best	0.00e+00	6.47e−07	0.00e+00	0.00e+00
	Median	0.00e+00	1.11e−02	0.00e+00	0.00e+00
	Worst	9.86e−03	4.67e−02	1.48e−02	7.40e−03
	Mean	9.86e−04	1.53e−02	1.48e−03	3.70e−04
	Std	3.12e−03	1.43e−02	4.67e−03	1.70e−03
Ackley Function	Best	1.70e−08	9.97e+00	2.49e−14	1.42e−14
	Median	7.43e−07	1.40e+01	3.20e−14	1.78e−14
	Worst	2.81e−06	1.99e+01	9.95e−14	2.84e−14
	Mean	1.02e−06	1.37e+01	4.12e−14	1.83e−14
	Std	1.08e−06	2.96e+00	2.29e−14	4.04e−15
Schwefel's Problem 2.26	Best	3.79e+02	3.19e+03	2.73e+03	2.38e+02
	Median	9.77e+02	4.37e+03	3.67e+03	8.34e+02
	Worst	1.67e+03	5.42e+03	4.54e+03	2.62e+03
	Mean	9.88e+02	4.31e+03	3.72e+03	9.48e+02
	Std	4.04e+02	8.01e+02	5.00e+02	5.31e+02

4.4 Comparison regarding the t-test

The same as Section 4.2, the F-test and t-test were performed in this section. “Yes+” means that the results of LPBFO are significantly better than the results of other

algorithms. “NO+” indicates that the results of LPBFO are better than the results of other algorithms but there exists no significant difference. “Yes−” means that the results of LPBFO are significantly worse than the results of other algorithms. “NO−” indicates that the results of LPBFO are

Table 11 Comparison between LPBFO and other other LPBFO variants for 20 times independent runs tested on 8 basic benchmark functions with 60 dimensions

Functions	Metrics	LPBFO1	LPBFO2	LPBFO3	LPBFO
Sphere Function	Best	1.33e-14	1.88e-05	7.05e-25	2.26e-33
	Median	5.83e-14	3.91e-05	1.04e-23	2.37e-32
	Worst	7.18e-14	4.72e-05	2.36e-22	8.71e-32
	Mean	4.97e-14	3.41e-05	4.04e-23	3.04e-32
	Std	2.28e-14	1.04e-05	7.19e-23	2.77e-32
Quartic Function	Best	1.29e-34	1.39e-08	2.72e-51	4.28e-67
	Median	2.52e-33	1.74e-08	1.43e-49	3.56e-65
	Worst	5.08e-32	8.88e-08	3.29e-47	1.74e-63
	Mean	1.10e-32	2.58e-08	5.59e-48	2.74e-64
	Std	1.85e-32	2.29e-08	1.11e-47	5.16e-64
Schwefel's Problem 2.22	Best	1.86e-08	4.91e+00	6.76e-13	1.77e-17
	Median	3.04e-08	9.76e+00	1.20e-12	3.14e-17
	Worst	4.27e-08	1.76e+01	2.06e-12	8.41e-17
	Mean	3.07e-08	1.10e+01	1.31e-12	3.56e-17
	Std	8.01e-09	4.14e+00	5.14e-13	1.82e-17
Rosenbrockf Function	Best	1.63e+00	5.29e+01	3.19e+00	3.67e-01
	Median	9.33e+00	5.87e+01	2.03e+01	1.17e+01
	Worst	1.56e+02	1.90e+02	2.23e+01	1.43e+02
	Mean	2.50e+01	8.69e+01	1.77e+01	2.74e+01
	Std	4.65e+01	4.49e+01	6.53e+00	3.77e+01
Rastrigin Function	Best	2.90e-04	6.27e+01	5.88e-11	0.00e+00
	Median	2.31e-01	1.11e+02	1.07e-09	0.00e+00
	Worst	1.99e+00	1.79e+02	4.11e-07	0.00e+00
	Mean	4.43e-01	1.15e+02	6.25e-08	0.00e+00
	Std	6.30e-01	3.90e+01	1.38e-07	0.00e+00
Griewank Function	Best	4.06e-05	8.52e-06	0.00e+00	0.00e+00
	Median	1.03e-03	1.54e-05	0.00e+00	0.00e+00
	Worst	1.18e-02	1.73e-02	7.40e-02	9.90e-03
	Mean	5.31e-03	4.94e-03	7.40e-03	4.93e-03
	Std	3.57e-03	6.87e-03	2.34e-03	2.20e-03
Ackley Function	Best	3.18e-06	1.43e+01	1.29e-08	4.97e-14
	Median	9.04e-06	1.99e+01	3.06e-08	8.53e-14
	Worst	4.48e-05	1.99e+01	5.56e-08	1.17e-13
	Mean	1.27e-05	1.92e+01	3.25e-08	8.37e-14
	Std	1.19e-05	1.80e+00	1.62e-08	1.89e-14
Schwefel's Problem	Best	5.92e+03	6.22e+03	8.48e+03	4.18e+03
	Median	6.42e+03	8.64e+03	9.01e+03	5.39e+03
	Worst	7.25e+03	1.16e+04	9.54e+03	6.91e+03
	Mean	6.51e+03	8.81e+03	9.01e+03	5.46e+03
	Std	6.21e+02	1.94e+03	3.88e+02	7.35e+02

worse than the results of other algorithms but there exists no significant difference. "SAME" means that LPBFO and the best one in other algorithms have obtained the same optimal solution.

Based on the results of statistical tests in Table 9, LPBFO algorithm is significantly better than the results of other algorithms in 7 experiments (YES+); LPBFO algorithm is significantly worse than the results of other

algorithms in 2 experiments (YES-); there exists no significant difference between LPBFO algorithm and the results of other algorithms in 7 experiments (NO+, NO- and SAME). The above analysis suggests that the LPBFO algorithm is still the best one among the six algorithms and it possess competitive performance in comparison with other algorithms.

4.5 Discussion on LPBFO algorithm

In this subsection, we offer a thorough analysis on the proposed LPBFO algorithm and all the algorithms were tested using the same parameters with the previous experiments.

In order to overcome the insufficiencies in BFO algorithm, the proposed LPBFO algorithm introduces three improvements: the adaptive Lévy flight step-size, the improved search direction, the modified swarming process, where each improvement has different effect. We constructed three LPBFO algorithm variants (LPBFO1, LPBFO2, LPBFO3) to analyze how each improvement influences the LPBFO algorithm. All the algorithms were tested on 8 benchmark functions with 30, 60 dimensions and each algorithm were repeated 20 times.

The difference between LPBFO1 algorithm and LPBFO algorithm is that instead of using the adaptive Lévy flight step-size as the chemotaxis step-size, LPBFO1 algorithm uses the fixed step-size $C(i) = 0.1$. Different with LPBFO algorithm, LPBFO2 algorithm uses the tumbling direction in the original BFO algorithm as the search direction which is randomly selected from the search space. In the modified swarming process of LPBFO algorithm, the global best solution is used to guide the movement of the bacteria; LPBFO3 algorithm used a randomly selected solution to complete the swarming process. Tables 10 and 11 show the experimental results obtained by each algorithm in the 20 independent runs. As can be seen from Tables 10 and 11, LPBFO with the three improvements obtained the best solution than other LPBFO variants, where each LPBFO variant (LPBFO1, LPBFO2, LPBFO3) uses two improvements. Therefore, any improvement being less worsen the result, which means that the three improvements working together can achieve the best result.

Among the three LPBFO variants (LPBFO1, LPBFO2, LPBFO3), LPBFO3 achieved the best solution, followed by LPBFO1 and the last is LPBFO2. Without using the global best solution in the modified swarming process, LPBFO3 still obtained good results which means that the third improvement has minor impact on LPBFO. From Tables 10 and 11, without using the improved search direction, LPBFO2 obtained the worst solutions than LPBFO1 and LPBFO3. Therefore, we can conclude that the second improvement (searching one dimension one time) has the

most impact on LPBFO. Moreover, the adaptive Lévy flight step-size has certain impact on LPBFO and the global best solution has the least impact on LPBFO.

5 Conclusion

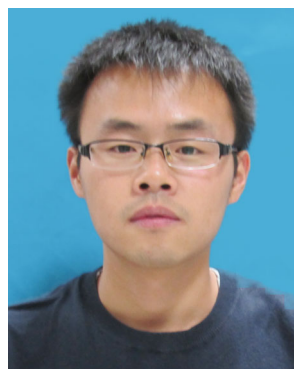
In this paper, we proposed an improved BFO algorithm (LPBFO) to accelerate the convergence rate and improve the precision of the solution. The adaptive Lévy Flight step-size is set as the chemotaxis step-size to balance local search and global search. To avoid the oscillation phenomenon among different dimensions and maximize the search ability of the algorithm, each bacterium choose one dimension randomly to tumble. To improve the accuracy of convergence, the information of global best solution is used to guide the movement of the bacteria. Finally, simulation experiments were conducted to assess the effectiveness of the LPBFO algorithm based on eight benchmark functions. The experiments verified that LPBFO algorithm significantly outperforms other bionic algorithms in the literature in terms of t-test.

Acknowledgements This research is supported by the NSFC (National Nature Science Foundation of China) under grant no. 61573213, 61473174, 61473179, by the Natural Science Foundation of Shandong Province under grant no. ZR2015PF009, ZR2017PF008, by the China Postdoctoral Science Foundation under grant no. 2017M612270, by Shandong Province Science and Technology Development Program under grant no. 2014GGX103038, and Special Technological Program of Transformation of Initiatively Innovative Achievements in Shandong Province under grant no. 2014ZZCX04302.

References

1. Babaoglu I (2015) Artificial bee colony algorithm with distribution-based update rule. *Appl Soft Comput* 34(C):851–861
2. Bermejo E, Cordón O, Damas S, Santamaría J (2015) A comparative study on the application of advanced bacterial foraging models to image registration. *Inf Sci* 295:160–181
3. Biswas A, Dasgupta S, Das S, Abraham A (2007) Synergy of pso and bacterial foraging optimization a comparative study on numerical benchmarks. *Adv Soft Comput* 44:255–263
4. Chen H, Niu B, Ma L, Su W, Zhu Y (2014) Bacterial colony foraging optimization. *Neurocomputing* 137(2):268–284
5. Chen H, Zhu Y, Hu K (2010) Multi-colony bacteria foraging optimization with cell-to-cell communication for rfid network planning. *Appl Soft Comput* 10(2):539–547
6. Chen H, Zhu Y, Hu K, Ma L (2014) Bacterial colony foraging algorithm: Combining chemotaxis, cell-to-cell communication, and self-adaptive strategy. *Inf Sci* 273(3):73–100
7. Chen YP, Li Y, Wang G, Zheng YF, Xu Q, Fan JH, Cui XT (2017) A novel bacterial foraging optimization algorithm for feature selection. *Expert Syst with Appl Intern J* 83(C):1–17

8. Dasgupta S, Das S, Abraham A, Biswas A (2009) Adaptive computational chemotaxis in bacterial foraging optimization: an analysis. *IEEE Trans Evol Comput* 13(4):919–941
9. Karaboga D (2005) An idea based on honey bee swarm for numerical optimization, technical report - tr06. Erciyes university, engineering faculty computer engineering department
10. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: *Proceedings International conference on neural networks, 1995*. IEEE. vol 4, pp 1942–1948 vol 4
11. Lai X, Zhou Y (2018) An adaptive parallel particle swarm optimization for numerical optimization problems. *Neural Computing and Applications*
12. Li C, Yang S, Nguyen TT (2012) A self-learning particle swarm optimizer for global optimization problems. *IEEE Trans Syst Man Cybern Part B* 42(3):627–646
13. Liu L, Shan L, Dai Y, Liu C, Qi Z (2018) A modified quantum bacterial foraging algorithm for parameters identification of fractional-order system. *IEEE Access* 6:6610–6619
14. Liu L, Shan L, Dai Y, Liu C, Zhidong QI (2018) Improved quantum bacterial foraging algorithm for tuning parameters of fractional-order pid controller. *J Syst Eng Electron* 29(1):166–175
15. Edwards A, Phillips R, Watkins NW, Freeman MP, Murphy EJ, Afanasyev V, Buldyrev SV et al (2007) Revisiting Lévy flight search patterns of wandering albatrosses, bumblebees and deer. *Nature* 449(7165):1044–1048
16. Mantegna RN (1994) Fast, accurate algorithm for numerical simulation of Lévy stable stochastic processes. *Phys Rev E Stat Phys Plasmas Fluids Relat Interdiscip Topics* 49(5):4677
17. Mishra S (2005) A hybrid least square-fuzzy bacterial foraging strategy for harmonic estimation. *IEEE Trans Evol Comput* 9(1):61–73
18. Mishra S, Bhende CN (2006) Bacterial foraging technique-based optimized active power filter for load compensation. *IEEE Trans Power Delivery* 22(1):457–465
19. Niu B, Wang H, Tan J (2011) Improved BFO with adaptive chemotaxis step for global optimization. In: *Proceedings of 2011 Seventh International Conference on Computational Intelligence and Security (CIS)*, pp 76–80
20. Panda R, Naik MK (2015) A novel adaptive crossover bacterial foraging optimization algorithm for linear discriminant analysis based face recognition. *Appl Soft Comput* 30(C):722–736
21. Passino KM (2002) Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Syst* 22(3):52–67
22. Rajni ChanaI. (2013) Bacterial foraging based hyper-heuristic for resource scheduling in grid computing. *Futur Gener Comput Syst* 29(3):751–762
23. Ratnaweera A, Halgamuge SK, Watson HC (2004) Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. *IEEE Trans Evol Comput* 8(3):240–255
24. Reynolds AM, Frye MA (2007) Free-flight odor tracking in drosophila is consistent with an optimal intermittent scale-free search. *Plos One* 2(4):e354
25. Sakhthivel VP, Bhuvanewari R, Subramanian S (2010) Bacterial foraging technique based parameter estimation of induction motor from manufacturer data. *Electr Mach Power Syst* 38(6):657–674
26. Supriyono H, Tokhi MO (2010) Bacterial foraging algorithm with adaptable chemotactic step size. In: *Second international conference on computational intelligence, communication systems and networks*, pp 72–77
27. Tang K, Xiao X, Wu J, Yang J, Luo L (2017) An improved multilevel thresholding approach based modified bacterial foraging optimization. *Appl Intell* 46(1):1–13
28. Turanoglu B, Akkaya G (2018) A new hybrid heuristic algorithm based on bacterial foraging optimization for the dynamic facility layout problem. *Expert Syst Appl* 98:93–104
29. Viswanathan GM, Afanasyev V (1996) Lévy flight search patterns of. *Nature*
30. Viswanathan GM, Buldyrev SV, Havlin S, Da LM, Raposo EP, Stanley HE (1999) Optimizing the success of random searches. *Nature* 401(6756):911
31. Xu X, Chen HL (2014) Adaptive computational chemotaxis based on field in bacterial foraging optimization. *Soft Comput* 18(4):797–807
32. Xue Y, Jiang J, Zhao B, Ma T (2018) A self-adaptive artificial bee colony algorithm based on global best for global optimization. *Soft Comput* 22(9):2935–2952
33. Yang C, Ji J, Liu J, Yin B (2016) Bacterial foraging optimization using novel chemotaxis and conjugation strategies. *Inf Sci* 363(C):72–95
34. Zhao F, Jiang X, Zhang C, Wang J (2015) A chemotaxis-enhanced bacterial foraging algorithm and its application in job shop scheduling problem. *Int J Comput Integr Manuf* 28(10):1106–1121



Bao Pang is a Ph.D. candidate in School of Control Science and Engineering at Shandong University. His current research interests include swarm robotics, intelligent control, and machine learning.



Yong Song received his Ph.D. degree in Pattern Recognition and Intelligent System from Shandong University in 2012. He is currently an associate professor in the School of Mechanical, Electrical and Information Engineering, Shandong University at Weihai. His current research interests include swarm intelligence robotics, machine learning, and neural network.



Chengjin Zhang received his Ph.D. degree in Industrial Automation from Northeastern University in 1997. He is currently a professor in both the School of Control Science and Engineering, Shandong University and the School of Mechanical, Electrical and Information Engineering, Shandong University at Weihai. His current research interests include the adaptive identification and control, time-varying and nonlinear systems, intelligent robot control, and bioinformatics.



Runtao Yang received his Ph.D. degree in Bioinformatics and Intelligent System from Shandong University in 2016. He is currently a lecture in the School of Mechanical, Electrical and Information Engineering, Shandong University at Weihai. His current research interests include swarm intelligence robotics, Bioinformatics, and system biology.



Hongling Wang is a Ph.D. candidate in School of Control Science and Engineering at Shandong University. Her current research interests include search and rescue robotics, intelligent robot control, and simultaneous localization and mapping.