

Stability Analysis of Chemotaxis Dynamics in Bacterial Foraging Optimization over Multi-dimensional Objective Functions

Cuicui Yang^a, Junzhong Ji^{a,*}, Sanjiang Li^b

^aBeijing Municipal Key Laboratory of Multimedia and Intelligent Software Technology, Faculty of Information Technology, Beijing University of Technology, Beijing, China

^bFaculty of Engineering and Information Technology, University of Technology Sydney, Australia.

Abstract

Bacterial foraging optimization (BFO) has been proved to be an efficient optimization method and successfully applied to a variety of fields in the real world. In BFO, the chemotaxis process is a complex and close combination of swimming and tumbling, and plays a crucial role in searching better solutions. A previous study has modeled the dynamics of the chemotaxis mechanism mathematically, and investigated the stability and convergence behavior of the chemotaxis dynamics over the one-dimensional objective function by Lyapunov stability theorem. However, this study appears to be very limited from a practical point of view, and how to extend their study to the multidimensional objective function is a challenge. To solve it, we present a stability analysis of chemotaxis dynamics in BFO over the multi-dimensional objective function in this paper. First, the general mathematical model of the chemotaxis mechanism over the multi-dimensional objective function is created. Secondly, this paper uses the general descent search to analyze the general mathematical model, and points out two necessary conditions for avoiding the bacterium to trap into a non-optimal solution. And then, the stability and convergence of the chemotaxis dynamics, represented by the general mathematical model, is proved by using Lyapunov stability theorem. Finally, empirical research is conducted to validate the above theoretical analysis.

Keywords: Bacterial foraging optimization, chemotaxis dynamics, multi-dimensional objective function, general mathematical model, Lyapunov stability theorem.

1. Introduction

The metaheuristic search technology based on swarm intelligence has been increasing in popularity due to its ability to solve a variety of complex scientific and engineering problems [1–4]. Such technology models the social behavior of certain living creatures, where each individual is simple, has limited cognitive capability, and communicates only locally with others and environment, but the swarm as a whole can act in a coordinated way without a coordinator or an external commander, and yield an intelligent behavior to obtain global optima. Inspired by the biological behavior of bacteria searching for nutrient-rich areas, Passino [5] developed a bacterial foraging optimization (BFO) algorithm in 2002, where each bacteria represents a feasible solution of an optimization problem. It starts with a population of bacteria, each of which is randomly generated during an initialization phase. Then, the bacterial population iteratively tries to find an optimal solution by four mechanisms, namely, chemotaxis, swarming, reproduction, and elimination and dispersal. Up to now, BFO has been successfully applied to a variety of fields in the real world [6–14], for example, image registration [7], prediction of stock market indices [9], vehicle routing [11], and visual imaging [13], etc.

While BFO has been established as a popular optimization technique by a lot of empirical studies, not much research has so far been devoted to theoretically analyze its search mechanism and convergence property, which is very important for developing more effective methods [15]. To the best of our knowledge, there are only several research works that focus on the theoretical analyses of the properties of BFO. In [16, 17], researchers presented mathematical analysis of the chemotaxis mechanism over the one-dimensional (1-D) objective function (i.e., 1-D individual) in light of the classical gradient descent search. Following the same train of thought, Abraham et al. made a mathematical analysis of the reproduction mechanism in 1-D case [18]. In [19], the stability and convergence behavior of the chemotaxis mechanism was analyzed on the 1-D objective function using Lyapunov stability theorem.

It is obvious that these existing theoretical studies on BFO establish the theoretical analysis of properties of BFO over the 1-D objective function, just like many theoretical studies on the swarm intelligence algorithm prefer to make analysis on the 1-D objective function [20–23]. However, although the theoretical studies over the 1-D objective function give insights into some mathematically relevant properties of algorithms, they appear to be very limited from a practical point of view [24]. That is because the 1-D function optimization is a kind of relatively simple and basic problem, and there exist abundant optimization problems of complex functions with multi-dimensions in

*Corresponding author

Email address: jjz01@bjut.edu.cn (Junzhong Ji)

science and engineering. The authors [16–19] also acknowledged this point of view, and take it as a challenging future work to generalize their studies to multi-dimensional objective functions.

To break through the limitation, this paper makes a humble attempt on the chemotaxis mechanism in BFO, and has tried to extend the theoretical analysis of the chemotaxis mechanism to multi-dimensional objective functions based on the similar train of thoughts as in the 1-D case [19]. First, this paper derives the general mathematical model of a single bacterium performing the chemotaxis process over arbitrary dimensional functions. Then it uses the general descent search to analyze the general mathematical model, and points out two necessary conditions for avoiding the bacterium to trap into a non-optimal solution. Subsequently, the stability and convergence of the chemotaxis dynamics, represented by the general mathematical model, is proved by using Lyapunov stability theorem when the bacterium is very near the equilibrium state. Finally, for ease of visualization and without loss of generality, two 2-D objective functions are used to do the empirical research to validate the above theoretical analysis.

The rest of this paper is structured as follows. Section II briefly introduces the classical BFO algorithm. Section III derives and analyzes the general movement model of a single bacterium performing the chemotaxis process over multi-dimensional objective functions. The general model is proved to be asymptotically stable near the equilibrium state by using Lyapunov stability theorem in Section IV. Afterwards, Section V carries out experimental simulations to verify the theoretical results. Finally, Section VI concludes this paper and outlines future research work.

2. Classical BFO

The classical BFO algorithm operates under four principal mechanisms: chemotaxis, swarming, reproduction, and elimination and dispersal. In the following, we will briefly introduce each of these mechanisms in the case of finding the minimum of an objective function (also called fitness function), $J(\mathbf{x})$, where $\mathbf{x} = (x_1, x_2, \dots, x_p, \dots, x_n) \in \mathbf{R}^n$ (i.e., \mathbf{x} is an n -dimensional vector of real numbers), and finally provide a pseudo-code of the complete BFO algorithm.

2.1. Chemotaxis

Chemotaxis simulates the movement of E.coli bacteria through tumbling and swimming via flagella. A bacterium tumbles in a random direction, searching for food. If food is abundant in the selected direction, the bacterium will swim in this direction until the food supply worsens or the bacterium reaches the specified maximum number of steps. This bacterium movement is expressed as

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

where $\mathbf{x}^i(j, k, l)$ represents the position of the i th bacterium at the j th chemotaxis, k th reproduction, and l th elimination

and dispersal step (sometimes, we drop the indexes and use \mathbf{x}^i to represent the position of the i th bacterium); $\Phi(i) = (\phi_1(i), \phi_2(i), \dots, \phi_p(i), \dots, \phi_n(i))$ represents a random direction, where each component $\phi_p(i)$ ($p = 1, 2, \dots, n$) is a random number in $[-1, 1]$; and the expression $\frac{\Phi(i)}{\sqrt{\Phi^T(i)\Phi(i)}}$ is to compute a unit length vector which is regarded as the direction of the tumbling behavior; $C(i)$ is the step size in the random direction for bacterium i . It should be noted that the step size $C(i)$ is usually a constant, and many state-of-the-art studies adopt adaptive step size to improve the performance of BFO [9, 25–27]. When $J(\mathbf{x}^i(j+1, k, l)) < J(\mathbf{x}^i(j, k, l))$, the i th bacterium will swim another step of size $C(i)$ in the same direction according to Eq.(1). Swimming continues until the bacterium reaches the maximum number of steps, N_s , or its objective function value does not decrease.

2.2. Swarming

When each bacterium moves, it always releases attractant signals to make other bacteria swim toward it. Meanwhile, it also releases repellant signals to keep other bacteria at a safe distance. A cell-to-cell signaling function (whose value should be added to the objective function value) is used to model this phenomenon as below:

$$\begin{aligned} J_{cc}(\mathbf{x}^i, P(j, k, l)) &= \sum_{r=1}^S J_{cc}^r(\mathbf{x}^i, \mathbf{x}^r) \\ &= \sum_{r=1}^S \left[-d_{att} \exp \left(-w_{att} \sum_{p=1}^n (x_p^i - x_p^r)^2 \right) \right] \\ &\quad + \sum_{r=1}^S \left[h_{rep} \exp \left(-w_{rep} \sum_{p=1}^n (x_p^i - x_p^r)^2 \right) \right] \end{aligned} \quad (2)$$

where S is the number of the bacterial population; $P(j, k, l) = \{\mathbf{x}^r(j, k, l) | r = 1, 2, \dots, S\}$ represents the position of each individual in the population of S bacteria at the j th chemotaxis, k th reproduction, and l th elimination and dispersal step; d_{att} , w_{att} , h_{rep} and w_{rep} are different coefficients that should be properly chosen. It is noted that swarming is an optional mechanism. Some studies do not consider the swarming operation to keep the algorithm computationally simple without seriously impacting the solution accuracy [9, 28].

2.3. Reproduction

The bacteria grow longer in accordance with the increase in the absorption of nutrients in the chemotactic steps. Under appropriate conditions, some bacteria will die, and others that have obtained adequate nutrients will asexually split into two same bacteria. To model this reproduction phenomenon, let the number of chemotactic steps, N_c , be the lifetime of the bacterium, and $S_r = S/2$ be the number of bacteria that have accumulated adequate nutrients to copy for themselves. After N_c chemotactic steps, a reproduction step is instigated. First, the bacterial population is sorted in an ascending order according to the health of bacteria. Then, each of the S_r healthiest bacteria is split into two bacteria, which are placed at the same location, while the remaining S_r bacteria die, to maintain a constant

population size. The health function is used to compute the accumulated objective value of the i th bacterium over its lifetime, and is defined as

$$J_{health}^i = \sum_{j=1}^{N_c+1} J(i, j, k, l), \quad (3)$$

where $J(i, j, k, l) = J(\mathbf{x}^i(j, k, l))$, and J_{health}^i represents the accumulated objective value of the i th bacterium, and the lower the value of J_{health}^i , the healthier the bacterium i .

2.4. Elimination and dispersal

With changes to the local environment that a population of bacteria live in, a group of bacteria may be dispersed into a new environment to find better food sources. To simulate this phenomenon, an elimination and dispersal step is taken after N_{re} reproduction steps. Each bacterium in the population may be dispersed to a new location with probability P_{ed} . The rule is shown below:

$$\mathbf{x} = \begin{cases} \mathbf{x}', & \text{if } q < P_{ed} \\ \mathbf{x}, & \text{otherwise} \end{cases} \quad (4)$$

where \mathbf{x} is the current solution associated with a bacterium; \mathbf{x}' is a new solution generated at random in the n -dimensional search space; q is a random number uniformly distributed in $[0,1]$. That is, for each bacterium, if the number generated randomly is smaller than P_{ed} , it will move to a new location and get a new random solution, otherwise, it will keep the original solution unchanged.

According to [5], the detailed pseudo-code of the complete algorithm is given in Algorithm 1.

3. Model the chemotaxis dynamics over multi-dimensional functions

This section provides a theoretical analysis of the chemotaxis dynamics over multi-dimensional objective functions. We first derive the mathematical models of a single bacterium performing the chemotaxis process over the n -dimensional (i.e., arbitrary dimensional) case. Then we analyze the general mathematical model with an arbitrary dimension in light of the general descent search. As mentioned in [19], five similar simplifying assumptions have been made to gain mathematical insights, and they are listed in the following. It should be noted that we drop the indexes of $C(i)$ and $\Phi(i)$ in the chemotaxis process defined in Eq.(1), and use C and Φ to represent the step size and the tumble direction, respectively, since only a single bacterium is considered in this section.

- 1) The bacterium lives in continuous time, and its position is denoted by an n -dimensional vector $\mathbf{x}(t) = (x_1(t), x_2(t), \dots, x_n(t))$ at the t th instant.
- 2) The chemotactic step C satisfies $0 \leq C < 1$ (in the seminal paper of BFO, $C = 0.1$ was used [5]).

Algorithm 1: The classical BFO algorithm

1 Initialization:

- a). **Set parameters:** $S, N_s, N_c, N_{re}, N_{ed}, P_{ed}$.
- b). **Let $j = k = l = 0$ (three counters)**
- c). **Initialize the bacterial population:** randomly distribute each bacterium $\mathbf{x}^i(0, 0, 0)$ across the domain of the optimization problem, and set $\mathbf{x}^{best} = \mathbf{x}^0(0, 0, 0)$.

2 Elimination and dispersal loop: $l = l + 1$

3 Reproduction loop: $k = k + 1$

4 Chemotaxis loop: $j = j + 1$

- 5 For bacterium $i = 1, 2, \dots, S$, perform a **chemotaxis operator**.
 - a). Compute $J(\mathbf{x}^i(j, k, l))$, let $J(\mathbf{x}^i(j, k, l)) = J(\mathbf{x}^i(j, k, l)) + J_{cc}(\mathbf{x}^i(j, k, l), P(j, k, l))$ (i.e., add the swarming behavior).
 - b). Let $J_{last} = J(\mathbf{x}^i(j, k, l))$ to save this value since it is possible to find a better objective value via a run.
 - c). **Tumble:** randomly generate a n -dimensional vector $\Phi(i)$.
 - d). **Move:** Make a move according to Eq.(1), then compute $J(\mathbf{x}^i(j+1, k, l))$, let $J(\mathbf{x}^i(j+1, k, l)) = J(\mathbf{x}^i(j+1, k, l)) + J_{cc}(\mathbf{x}^i(j+1, k, l), P(j+1, k, l))$.
 - e). **Swim as follows:**
 - i) Let $m = 0$ (The swimming counter).
 - ii) While $m < N_s$
 - Let $m = m + 1$.
 - If $J(\mathbf{x}^i(j+1, k, l)) < J_{last}$, let $J_{last} = J(\mathbf{x}^i(j+1, k, l))$, keep on the move according to Eq.(1), then use the new $\mathbf{x}^i(j+1, k, l)$ to compute the new $J(\mathbf{x}^i(j+1, k, l))$, and let $J(\mathbf{x}^i(j+1, k, l)) = J(\mathbf{x}^i(j+1, k, l)) + J_{cc}(\mathbf{x}^i(j+1, k, l), P(j+1, k, l))$.
 - Else, let $m = N_s$.
 - f). If $J(\mathbf{x}^i(j+1, k, l)) < J(\mathbf{x}^{best})$, then $\mathbf{x}^{best} = \mathbf{x}^i(j+1, k, l)$.
 - g). Go to the next bacterium ($i+1$) if $i < S$.
 - 6 If $j < N_c$, go to step 4.
 - 7 **Reproduction**
 - a). Compute the health value for each bacterium according to Eq.(3).
 - b). Sort bacteria based on the health values in descending order.
 - c). Abandon S_r bacteria with higher health values and split each of another S_r bacteria into exactly same two ones.
 - 8 If $k < N_{re}$, go to step 3.
 - 9 **Elimination and dispersal**

For each bacterium $i = 1, 2, \dots, S$ may be dispersed into a new location with a probability P_{ed} according to Eq.(4).
 - 10 If $l < N_{ed}$, go to step 2. Else **return** the optimal solution \mathbf{x}^{best} .

3) The bacterium moves at a uniform rate. $\Phi = (\phi_1, \phi_2, \dots, \phi_n)$ is an n -dimensional (n -D) unit vector so that we need not to calculate a unit vector by $\frac{\Phi}{\sqrt{\Phi^T \Phi}}$. Thus, if the bacterium moves $C \cdot \Phi$ in a unit time, its position will be changed by $\Delta \mathbf{x} = (C \cdot \Phi) \cdot (\Delta t)$ in Δt time, i.e., in component-wise manner, $(\Delta x_1, \Delta x_2, \dots, \Delta x_n) = (C \cdot \phi_1 \cdot \Delta t, C \cdot \phi_2 \cdot \Delta t, \dots, C \cdot \phi_n \cdot \Delta t)$.

4) The objective function $J(\mathbf{x})$ is a continuous differentiable function in the search space and a unimodal function in the region of interest, and its unique optimum $J(\mathbf{x}) = 0$ is located at $\mathbf{x} = \mathbf{x}^e$.

5) The analysis applies to the regions where the objective function is close to the optimum, i.e., the gradients of the objective function are very small.

3.1. Derive the general model

At each chemotactic step in classical BFO algorithm, the bacterium first randomly generates a direction vector of unit

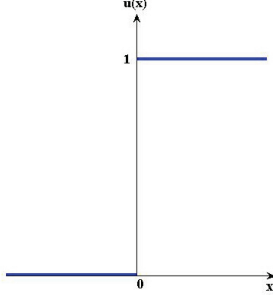


Figure 1: Unit step function

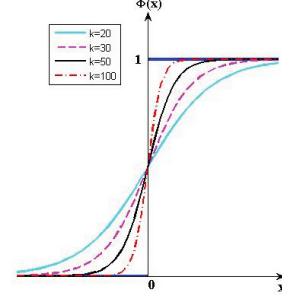


Figure 2: Logistic function (the larger the parameter k , the more close to $u(x)$)

length (i.e., the direction of tumble) denoted by Φ . In the n -D optimization problem, Φ has n components, i.e., $\Phi = (\phi_1, \phi_2, \dots, \phi_n)$, and has the relation $\sum_{i=1}^n \phi_i^2 = 1$. Then, the bacterium decides whether to move in the Φ direction by judging the change of the objective function value. If the new objective function value is less than the previous one, i.e., $J(\mathbf{x} + \Delta \mathbf{x}) < J(\mathbf{x})$ (in component-wise manner, $J(x_1 + \Delta x_1, x_2 + \Delta x_2, \dots, x_n + \Delta x_n) < J(x_1, x_2, \dots, x_n)$), the bacterium will change its position; otherwise, it will stay in place. To model this crucial decision-making activity of the bacterium, we use the unit step function [29] which is shown in Fig.1 and defined as

$$u(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{otherwise} \end{cases}$$

Thus, according to the assumption 3), we have

$$\Delta \mathbf{x} = u(J(\mathbf{x}) - J(\mathbf{x} + \Delta \mathbf{x})) \cdot (C \cdot \Phi) \cdot (\Delta t)$$

From this equation, the amount $\Delta \mathbf{x}$ that the bacterium moves is equal to 0 or $(C \cdot \Phi) \cdot (\Delta t)$, depending on the sign of the quantity of $(J(\mathbf{x}) - J(\mathbf{x} + \Delta \mathbf{x}))$. For clarity and convenience, we use the following component form in most cases:

$$\Delta x_p = u(J(x_1, \dots, x_n) - J(x_1 + \Delta x_1, \dots, x_n + \Delta x_n)) \cdot C \cdot \phi_p \cdot \Delta t \quad \forall p = 1, 2, \dots, n \quad (5)$$

According to Eq.(5), each component has similar expression. In this equation, Δt is assumed to be an infinitesimally small positive quantity, so the sign of $(J(x_1, \dots, x_n) - J(x_1 + \Delta x_1, \dots, x_n + \Delta x_n))$ remains unchanged if it is divided by Δt . After both sides of Eq.(5) is divided by Δt , we obtain

$$\begin{aligned} \frac{\Delta x_p}{\Delta t} &= u \left(\frac{J(x_1, \dots, x_n) - J(x_1 + \Delta x_1, \dots, x_n + \Delta x_n)}{\Delta t} \right) \cdot C \cdot \phi_p \\ &= u \left(- \frac{J(x_1 + \Delta x_1, \dots, x_n + \Delta x_n) - J(x_1, \dots, x_n)}{\Delta t} \right) \cdot C \cdot \phi_p \end{aligned}$$

Defining the velocity of the bacterium as $\mathbf{v} = (v_1, v_2, \dots, v_n)$, and each component as $v_p = \frac{dx_p}{dt} = \lim_{\Delta t \rightarrow 0} \left(\frac{\Delta x_p}{\Delta t} \right)$ ($\forall p = 1, 2, \dots, n$), we get

$$\begin{aligned} v_p &= \lim_{\Delta t \rightarrow 0} \frac{\Delta x_p}{\Delta t} \\ &= \lim_{\Delta t \rightarrow 0} \left[u \left(- \frac{J(x_1 + \Delta x_1, \dots, x_n + \Delta x_n) - J(x_1, \dots, x_n)}{\Delta t} \right) \cdot C \cdot \phi_p \right] \end{aligned}$$

In accordance with the assumption 4), $J(x_1, x_2, \dots, x_n)$ is a continuous differentiable function, thus we apply the theory of differential and integral calculus, and obtain $J(x_1 + \Delta x_1, x_2 + \Delta x_2, \dots, x_n + \Delta x_n) - J(x_1, x_2, \dots, x_n) = J'_{x_1} \Delta x_1 + J'_{x_2} \Delta x_2 + \dots + J'_{x_n} \Delta x_n + o(\rho)$, where $\rho = \sqrt{\Delta x_1^2 + \dots + \Delta x_n^2}$, and $o(\rho)$ represents the infinitesimal of higher order of ρ . Hence, the equation above can be written as

$$\begin{aligned} v_p &= \lim_{\Delta t \rightarrow 0} \left[u \left(- \frac{J'_{x_1} \Delta x_1 + \dots + J'_{x_n} \Delta x_n + o(\rho)}{\Delta t} \right) \cdot C \cdot \phi_p \right] \\ &= \lim_{\Delta t \rightarrow 0} \left[u \left(- \left(J'_{x_1} \frac{\Delta x_1}{\Delta t} + \dots + J'_{x_n} \frac{\Delta x_n}{\Delta t} + \frac{o(\rho)}{\Delta t} \right) \right) \cdot C \cdot \phi_p \right] \\ &= u \left(- \left(J'_{x_1} \lim_{\Delta t \rightarrow 0} \frac{\Delta x_1}{\Delta t} + \dots + J'_{x_n} \lim_{\Delta t \rightarrow 0} \frac{\Delta x_n}{\Delta t} + \lim_{\Delta t \rightarrow 0} \frac{o(\rho)}{\Delta t} \right) \right) \\ &\quad \cdot C \cdot \phi_p \\ &= u(- (J'_{x_1} v_1 + \dots + J'_{x_n} v_n)) \cdot C \cdot \phi_p \quad (6) \end{aligned}$$

Moreover, for ease of calculation, the step function $u(x)$ is usually approximated with the continuous logistic function $\phi(x)$:

$$u(x) = \lim_{k \rightarrow \infty} \phi(x) = \lim_{k \rightarrow \infty} \frac{1}{1 + e^{-kx}}$$

Fig.2 illustrates the logistic function for different values of k . It is seen from this figure that a larger k makes the logistic function $\phi(x)$ be more close to the unit step function $u(x)$. However, for ease of analysis, k cannot be infinity, so we choose a moderately large value (for example $k = 100$) with which $\phi(x)$ can fairly approximate $u(x)$. Hence, from Eq.(6), we get

$$v_p = \frac{C \phi_p}{1 + e^{k(J'_{x_1} v_1 + \dots + J'_{x_n} v_n)}}$$

According to the assumption 5), J'_{x_p} ($\forall p = 1, 2, \dots, n$) are very small, and $k = 100$, so $k(J'_{x_1} v_1 + \dots + J'_{x_n} v_n) \ll 1$, and

$\left| \frac{k(J'_{x_1} v_1 + \dots + J'_{x_n} v_n)}{2} \right| \ll 1$. By neglecting higher order terms in the expansions, $e^{k(J'_{x_1} v_1 + \dots + J'_{x_n} v_n)} \approx 1 + k(J'_{x_1} v_1 + \dots + J'_{x_n} v_n)$, and $\left(1 + \frac{k(J'_{x_1} v_1 + \dots + J'_{x_n} v_n)}{2} \right)^{-1} \approx \left(1 - \frac{k(J'_{x_1} v_1 + \dots + J'_{x_n} v_n)}{2} \right)$. Hence, the preceding equation is changed into

$$\begin{aligned} v_p &\approx \frac{C\phi_p}{2 + k(J'_{x_1} v_1 + \dots + J'_{x_n} v_n)} \\ \Rightarrow v_p &\approx \frac{C\phi_p}{2} \cdot \frac{1}{1 + \frac{k(J'_{x_1} v_1 + \dots + J'_{x_n} v_n)}{2}} \\ \Rightarrow v_p &\approx \frac{C\phi_p}{2} \cdot \left(1 - \frac{k(J'_{x_1} v_1 + \dots + J'_{x_n} v_n)}{2} \right) \end{aligned} \quad (7)$$

Due to $\Phi = (\phi_1, \phi_2, \dots, \phi_n)$ is a vector of unit length, at least one component is not zero. Thus, if the bacterium moves in the Φ direction, at least one component of $\mathbf{v} = (v_1, v_2, \dots, v_n)$ is nonzero. Supposing $\phi_1 \neq 0$ and $v_1 \neq 0$, the following relation is obtained from Eq.(6):

$$\frac{v_p}{v_1} = \frac{\phi_p}{\phi_1} \Rightarrow v_p = \frac{\phi_p}{\phi_1} \cdot v_1 \quad (\forall p = 1, 2, \dots, n)$$

From the two previous equations, we have

$$\begin{aligned} v_p &\approx \frac{C\phi_p}{2} \cdot \left(1 - \frac{k(J'_{x_1} v_1 + J'_{x_2} \cdot \frac{\phi_2}{\phi_1} v_1 + \dots + J'_{x_n} \cdot \frac{\phi_n}{\phi_1} v_1)}{2} \right) \\ \Rightarrow v_p &\approx \frac{C\phi_p}{2} - \frac{C}{2} \cdot \frac{v_p \phi_1}{v_1} \cdot \frac{k(J'_{x_1} v_1 + J'_{x_2} \cdot \frac{\phi_2}{\phi_1} v_1 + \dots + J'_{x_n} \cdot \frac{\phi_n}{\phi_1} v_1)}{2} \\ \Rightarrow v_p &\approx \frac{C\phi_p}{2} - \frac{kC(J'_{x_1} \phi_1 + \dots + J'_{x_n} \phi_n) v_p}{4} \end{aligned}$$

It is easy to solve v_p from the equation above, and get

$$\begin{aligned} v_p &\approx \frac{2C\phi_p}{4 + kC(J'_{x_1} \phi_1 + \dots + J'_{x_n} \phi_n)} \\ \Rightarrow v_p &\approx \frac{C\phi_p}{2} \cdot \frac{1}{1 + \frac{kC(J'_{x_1} \phi_1 + \dots + J'_{x_n} \phi_n)}{4}} \end{aligned}$$

As a vector of unit length, $|\Phi| = \sqrt{\Phi \cdot \Phi} = \sqrt{\phi_1^2 + \phi_2^2 + \dots + \phi_n^2} = 1$, so $|\phi_p| \leq 1$ ($\forall p = 1, 2, \dots, n$). According to the assumptions 2) and 5), $0 \leq C < 1$, J'_{x_p} ($\forall p = 1, 2, \dots, n$) is very small, and $k = 100$, then $\frac{kC(J'_{x_1} \phi_1 + \dots + J'_{x_n} \phi_n)}{4} \ll 1$. By neglecting higher order terms, $\left(1 + \frac{kC(J'_{x_1} \phi_1 + \dots + J'_{x_n} \phi_n)}{4} \right)^{-1} \approx \left(1 - \frac{kC(J'_{x_1} \phi_1 + \dots + J'_{x_n} \phi_n)}{4} \right)$. Therefore,

$$\begin{aligned} v_p &\approx \frac{C\phi_p}{2} \left(1 - \frac{kC(J'_{x_1} \phi_1 + \dots + J'_{x_n} \phi_n)}{4} \right) \\ \Rightarrow v_p &\approx -\frac{kC^2 \phi_p}{8} (J'_{x_1} \phi_1 + \dots + J'_{x_p} \phi_p) + \frac{C\phi_p}{2} \\ \Rightarrow v_p &\approx -\frac{kC^2 \phi_p}{8} (\text{grad}J(\mathbf{x}) \cdot \Phi) + \frac{C\phi_p}{2} \quad (\forall p = 1, 2, \dots, n) \end{aligned} \quad (8)$$

where $\text{grad}J(\mathbf{x}) = (J'_{x_1}, J'_{x_2}, \dots, J'_{x_n})$ is the gradient of the objective function $J(\mathbf{x})$ at $\mathbf{x} = (x_1, x_2, \dots, x_n)$.

Each v_p is a component of \mathbf{v} along ϕ_p direction, and we can write it as below

$$v_p \approx \left(-\frac{kC^2}{8} (\text{grad}J(\mathbf{x}) \cdot \Phi) + \frac{C}{2} \right) \phi_p \quad (\forall p = 1, 2, \dots, n) \quad (9)$$

Thus, we have

$$\mathbf{v} = \frac{d\mathbf{x}}{dt} \approx \left(-\frac{kC^2}{8} (\text{grad}J(\mathbf{x}) \cdot \Phi) + \frac{C}{2} \right) \cdot \Phi \quad (10)$$

i.e.,

$$\mathbf{v} = \frac{d\mathbf{x}}{dt} \approx \left(-\frac{kC^2}{8} \cdot \frac{\partial J}{\partial \Phi} + \frac{C}{2} \right) \cdot \Phi \quad (11)$$

Through the above derivation, we have obtained the general model of the chemotaxis dynamics over multi-dimensional objective functions. The general model is represented by Eq.(10) or Eq.(11), and its component-wise manner is expressed as Eq.(8) or Eq.(9).

3.2. Analyze the general model

It is exciting to find out that the general model of the chemotaxis dynamics represented by Eq.(11) is essentially consistent with the line search descent method [30], which is shown

$$\frac{d\mathbf{x}}{dt} = \left(-\alpha \cdot \frac{\partial J}{\partial \Phi} + \beta \right) \cdot \Phi \quad (12)$$

where α represents the learning rate, and β is the momentum [19]. The way that how the descent direction Φ is chosen determines different descent methods [30]. When the descent direction Φ is identical with the opposite direction of the gradient, Eq.(12) becomes the famous steepest descent algorithm. The consistency between Eq.(11) and Eq.(12) suggests that the chemotaxis process may be viewed as a kind of the line search descent method.

As [30] have pointed out, to decrease the objective function value, it is needed that $\partial J / \partial \Phi < 0$, i.e., $\text{grad}J(\mathbf{x}) \cdot \Phi < 0$. Due to $\text{grad}J(\mathbf{x}) \cdot \Phi = |\text{grad}J(\mathbf{x})| |\Phi| \cos \theta$, where θ is the angle between two vectors $\text{grad}J(\mathbf{x})$ and Φ , thus, to decrease the objective function value and locate a better solution, θ must satisfies the relation $\pi/2 < \theta \leq \pi$.

For ease of visualization and without loss of generality, Fig.(3) takes the 2-D objective function as an example to illustrate this fact. In Fig.(3), L_1, L_2, L_3, L_4 , and L_5 are different contour lines of the objective function. Point O represents the optimal solution $\mathbf{x}^e = (x_1^e, x_2^e)$. At present, the bacterium arrives at point A on the level curve L_1 . It should be noted that only the objective function value within the region surrounded by the level curve L_1 is smaller than that of point A . To reduce the objective function value, the bacterium will select a new direction for moving. For the multi-dimensional case, there are numerous directions. Here, we only plot four different directions,

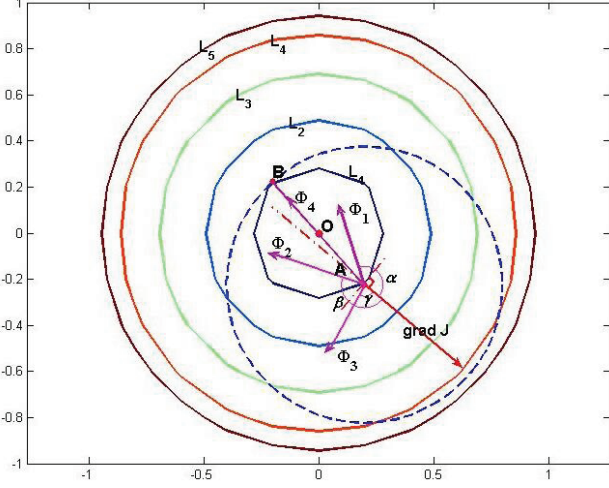


Figure 3: The movement state of the bacterium at point A in the search space

Φ_1 , Φ_2 , Φ_3 , and Φ_4 , to explain this fact. It is clear from the figure that the angles between the three directions, Φ_1 , Φ_2 and Φ_4 , and the gradient vector $gradJ(\mathbf{x})$ meet the relation $\pi/2 < \theta \leq \pi$. So if the bacterium chooses one of them, it has a chance to enter the region surrounded by the level curve L_1 , and decrease the objective function value. But if the bacterium picks out the Φ_3 direction, it will keep unchanged since the angle between Φ_3 and the gradient vector $gradJ(\mathbf{x})$ is in a range of $0 \leq \theta \leq \pi/2$ that would not make the bacterium step into the area surrounded by the level curve L_1 , and can not reduce the objective function value.

Actually, in addition to a right descent direction, a proper multiplier (i.e., the amount of movement in the selected direction) is another key factor in decreasing the objective function value [30]. For the chemotaxis dynamics, the multiplier corresponds to the step size C . With a too large step size C , the bacterium would not move even if it finds a right direction. Fig.(3) shows clearly this point. In Fig.(3), the bacterium is at point A. The longest segment that starts from A, and is surrounded by the level curve L_1 is AB whose length is symbolized by d . Obviously, the value of d changes as the bacterium moves. The more close the bacterium to the optimal solution, the smaller the value of d . The blue dashed curve is a circle that takes the point A as the center, and the length d as the radius. The relation between d and C influences the movement of the bacterium. To be specific, there are the following three situations.

- $C > d$: The bacterium keeps in place because the bacterium would go outside the area surrounded by the level curve L_1 , and increase the objective function value no matter which direction it moves toward.
- $C = d$: The bacterium would get to the blue dashed curve no matter which direction it moves toward. On the blue dashed curve, only point B has the same objective function value as point A, and all the other points have larger objective function values than point A. In other words, the objective function value of any point on the blue dashed

curve is not smaller than that of point A. Thus, the bacterium still keeps in place in this situation.

- $C < d$: If the angle between the selected direction and the gradient direction belongs to the range of $0 \leq \theta \leq \pi/2$ (that is, $gradJ(\mathbf{x}) \cdot \Phi \geq 0$), the bacterium remains motionless for the reason that it would not step into the area surrounded by the level curve L_1 , and can not decrease the objective function value. If the angle between the selected direction and the gradient direction is in the range of $\pi/2 < \theta \leq \pi$ (that is, $gradJ(\mathbf{x}) \cdot \Phi < 0$), the bacterium may have chance to enter the area surrounded by the level curve L_1 , and get a smaller objective function value. However, it is merely a possibility, since $C < d$ and $gradJ(\mathbf{x}) \cdot \Phi < 0$ are only two necessary conditions for the bacterium entering the area with smaller objective function value. Besides the two necessary conditions, the longest distance of moving in a selected direction is also related to the movement of the bacterium.

Through the above analysis, it comes to a beneficial conclusion as below. There are two necessary key conditions to ensure that the bacterium has chance to move toward the direction of decreasing the objective function value: (1) The angle between the moving direction and the gradient direction must satisfy the relation $\pi/2 < \theta \leq \pi$ (that is, $gradJ(\mathbf{x}) \cdot \Phi < 0$), and (2) The step size C must be smaller than the longest distance d between the current position and any other location on the same contour (i.e., $C < d$). If the two conditions are met, the bacterium is likely to move to a new position associated a better solution; otherwise, the bacterium does not move. To be noted, the two key conditions have different requirements. The key condition (1) naturally demands a proper direction for the bacterium to locate a better solution. The key condition (2) spontaneously requires a varying step size C , since the value of d becomes smaller and smaller with the bacterium getting closer to the optimal solution, and therefore the step size C should be tuned from a coarse mode to a fine mode and get smaller and smaller to meet the relation $C < d$. In fact, these two key conditions are just the theoretical bases of many studies on developing more efficient BFO algorithms. For example, some research works focus on setting good directions by introducing new strategies to the classical BFO algorithm [31, 32]; some others present new adaptive step size C to improve the classical BFO algorithm [9, 25–27].

4. Stability Analysis Using Lyapunov Stability Theorem

Lyapunov stability theory is an important tool to study the stability of the nonlinear dynamic system in the field of modern control [33–36]. This section will use Lyapunov stability theory to establish the stability of chemotaxis dynamics over the multi-dimensional objective function defined in Eq.(10) or Eq.(11).

4.1. Lyapunov stability theory

Some basic concepts about the Lyapunov stability theory are first introduced according to the standard literature on modern control theory [33, 34].

Definition 1 (Equilibrium state (point)). A state (point) $\mathbf{x}^e \in \mathbf{R}^n$ is called an equilibrium state (point) of the dynamic system described by

$$\frac{d\mathbf{x}}{dt} = f(\mathbf{x}(t)) \quad \mathbf{x}(t_0) = \mathbf{x}^0 \quad (13)$$

if for any t , the following condition holds:

$$f(\mathbf{x}^e(t)) \equiv 0$$

Definition 2 (Stable in the sense of Lyapunov). A dynamic system defined in Eq.(13) is stable in the sense of Lyapunov at the equilibrium state \mathbf{x}^e if for any $\varepsilon > 0$ there exists a $\phi > 0$ such that

$$\|\mathbf{x}^0 - \mathbf{x}^e\| \leq \phi \rightarrow \|\mathbf{x}(t; \mathbf{x}^0, t_0) - \mathbf{x}^e\| < \varepsilon, \quad \forall t > t_0 \quad (14)$$

Explanation: $\mathbf{x}(t; \mathbf{x}^0, t_0)$ represents the trajectories that starting from the initial state \mathbf{x}^0 at time t_0 . A dynamic system being stable at the equilibrium state \mathbf{x}^e in the sense of Lyapunov requires that for any neighborhood $S(\mathbf{x}^e, \varepsilon)$ (i.e., $\|\mathbf{x} - \mathbf{x}^e\| \leq \varepsilon$), there would be a neighborhood $S(\mathbf{x}^e, \phi)$, such that the trajectories $\mathbf{x}(t; \mathbf{x}^0, t_0)$ starting within $S(\mathbf{x}^e, \phi)$ does not leave $S(\mathbf{x}^e, \varepsilon)$ as time $t \rightarrow \infty$.

Definition 3 (Asymptotically stable). A dynamic system defined in Eq.(13) is asymptotically stable at the equilibrium state \mathbf{x}^e if

- 1) it is stable in the sense of Lyapunov, and
- 2) $\lim_{t \rightarrow \infty} \|\mathbf{x}(t; \mathbf{x}^0, t_0) - \mathbf{x}^e\| = 0$.

Definition 4 (Positive definite function). A scalar function $V : \mathbf{R}^n \rightarrow \mathbf{R}$ is said to be a positive definite function in the region $\Omega = \{\mathbf{x} \mid \|\mathbf{x} - \mathbf{x}^e\| \leq M\}$ if

- 1) $V(\mathbf{x}^e) = 0$, and
- 2) $V(\mathbf{x}) > 0, \forall \mathbf{x} \in \Omega$ and $\mathbf{x} \neq \mathbf{x}^e$.

Definition 5 (Negative definite function). A scalar function $V : \mathbf{R}^n \rightarrow \mathbf{R}$ is said to be a negative definite function in the region $\Omega = \{\mathbf{x} \mid \|\mathbf{x} - \mathbf{x}^e\| \leq M\}$ if

- 1) $V(\mathbf{x}^e) = 0$, and
- 2) $V(\mathbf{x}) < 0, \forall \mathbf{x} \in \Omega$ and $\mathbf{x} \neq \mathbf{x}^e$.

Theorem 1 (Lyapunov stability theorem). Consider a scalar function $V : \mathbf{R}^n \rightarrow \mathbf{R}$, and some real number $\varepsilon > 0$, such that, for all \mathbf{x} in the region $\|\mathbf{x} - \mathbf{x}^e\| \leq \varepsilon$, if the following conditions hold:

- 1) V has continuous first partial derivatives with respect to all components of \mathbf{x} .
- 2) V is a positive definite function.
- 3) dV/dt is a negative definite function.

Then, the system defined in Eq.(13) is asymptotically stable at the equilibrium state \mathbf{x}^e .

Remark: This theorem is based on an energetic standpoint, and is inspired by an intuitive physical fact of the classical mechanics vibration system, i.e., the total energy of the system reduces continually with time, till reaching the equilibrium state (at this time, the energy of the system is zero). The scalar function V that satisfies the three conditions listed in **Theorem 1** is often called Lyapunov function.

4.2. The stability proof of the chemotaxis dynamics

Theorem 2 (Main result). The chemotaxis dynamics represented by Eq.(10) or Eq.(11) is asymptotically stable at the equilibrium state \mathbf{x}^e .

Proof: In the following, we will use the Lyapunov stability theorem (i.e., **Theorem 1**) to determine the stability of the chemotaxis dynamics. To do this, one of the difficulties is that establishing a scalar function regarding the chemotaxis dynamics defined in Eq.(10) or Eq.(11). Here, we successfully find a scalar function as follows:

$$V(\mathbf{x}) = \frac{kC^2\phi_1^2}{8}J(\mathbf{x}) - \frac{C\phi_1^2}{2}((\mathbf{x} - \mathbf{x}^e) \cdot \Phi) \quad (15)$$

and its component form is

$$V(\mathbf{x}) = \frac{kC^2\phi_1^2}{8}J(\mathbf{x}) - \frac{C\phi_1^3}{2}(x_1 - x_1^e) - \frac{C\phi_1^2\phi_2}{2}(x_2 - x_2^e) - \dots - \frac{C\phi_1^2\phi_n}{2}(x_n - x_n^e)$$

where

$$\begin{cases} C = 0, & \text{if } \mathbf{x} = \mathbf{x}^e \\ 0 < C < d, & \text{if } \text{grad}J(\mathbf{x}) \cdot \Phi < 0 \text{ and } (\mathbf{x} - \mathbf{x}^e) \cdot \Phi < 0 \\ C > \frac{4}{k} \frac{|\mathbf{x} - \mathbf{x}^e|}{J(\mathbf{x})}, & \text{other cases} \end{cases} \quad (16)$$

The key to determining the stability of the system is verifying that V meets the three conditions in **Theorem 1**. Next, we will check each condition individually.

Condition 1: According to the assumption 4), V has continuous first partial derivatives with respect to all components of \mathbf{x} . Thus, the first condition is satisfied automatically.

Condition 2: In light of **Definition 4**, we must judge that V satisfies the relation $V(\mathbf{x}^e) = 0$, and $V(\mathbf{x}) > 0$ for any $\mathbf{x} \neq \mathbf{x}^e$ in order to confirm that V is a positive definite function.

When $\mathbf{x} = \mathbf{x}^e$, we have

$$V(\mathbf{x}) = \frac{kC^2\phi_1^2}{8}J(\mathbf{x}^e) - \frac{C\phi_1^2}{2}((\mathbf{x}^e - \mathbf{x}^e) \cdot \Phi) = 0$$

Moreover, at this time, $\text{grad}J(\mathbf{x}) = \vec{0}$ and $\mathbf{v} = d\mathbf{x}/dt = \vec{0}$. Thus, from Eq.(11),

$$\begin{aligned} \mathbf{v} &= \frac{d\mathbf{x}}{dt} \approx \left(-\frac{kC^2}{8}(\text{grad}J(\mathbf{x}) \cdot \Phi) + \frac{C}{2} \right) \cdot \Phi = \vec{0} \\ \Rightarrow \mathbf{v} &= \frac{d\mathbf{x}}{dt} \approx \left(-\frac{kC^2}{8}(\vec{0} \cdot \Phi) + \frac{C}{2} \right) \cdot \Phi = \vec{0} \\ \Rightarrow \frac{C \cdot \Phi}{2} &= \vec{0} \\ \Rightarrow C &= 0 \end{aligned}$$

The criterion, $C = 0$ at $\mathbf{x} = \mathbf{x}^e$, means that the bacterium does not take any more chemotactic steps once it arrives at the

optimum \mathbf{x}^e of the unimodal objective function. This intuitively reflects a desired performance for an optimization algorithm. That is, the algorithm is expected to do nothing when it finds the optimum.

When $\mathbf{x} \neq \mathbf{x}^e$, to make $V(\mathbf{x}) > 0$, we should have

$$\frac{kC^2\phi_1^2}{8}J(\mathbf{x}) - \frac{C\phi_1^2}{2}((\mathbf{x} - \mathbf{x}^e) \cdot \Phi) > 0 \quad \forall \mathbf{x} \neq \mathbf{x}^e$$

In the equation above, $C > 0$ for all the points except \mathbf{x}^e , and we have assumed that $\phi_1 \neq 0$ beforehand. Thus, we only need to guarantee

$$\frac{kC}{4}J(\mathbf{x}) > (\mathbf{x} - \mathbf{x}^e) \cdot \Phi \quad \forall \mathbf{x} \neq \mathbf{x}^e$$

Since $k = 100$, and $J(\mathbf{x}), C > 0$ for all the points except \mathbf{x}^e , we have $\frac{kC}{4}J(\mathbf{x}) > 0$ when $\mathbf{x} \neq \mathbf{x}^e$. Thus, if $(\mathbf{x} - \mathbf{x}^e) \cdot \Phi < 0$, the equation above always stands up. However, we impose a restriction on C (requiring $0 < C < d$) when $\text{grad}J(\mathbf{x}) \cdot \Phi < 0$ in order to continuously reduce the objective function value according to the analysis in Section III.

For other cases except $(\mathbf{x} - \mathbf{x}^e) \cdot \Phi < 0$ and $\text{grad}(\mathbf{x}) \cdot \Phi < 0$, to let $V(\mathbf{x}) > 0$, we only need to assure

$$C > \frac{4}{k} \frac{(\mathbf{x} - \mathbf{x}^e) \cdot \Phi}{J(\mathbf{x})} \quad \forall \mathbf{x} \neq \mathbf{x}^e$$

Now,

$$\begin{aligned} & \left| \frac{4}{k} \frac{(\mathbf{x} - \mathbf{x}^e) \cdot \Phi}{J(\mathbf{x})} \right| \geq \frac{4}{k} \frac{(\mathbf{x} - \mathbf{x}^e) \cdot \Phi}{J(\mathbf{x})} \\ \Rightarrow & \left| \frac{4}{k} \frac{(\mathbf{x} - \mathbf{x}^e) \cdot \Phi}{J(\mathbf{x})} \right| \geq \frac{4}{k} \frac{(\mathbf{x} - \mathbf{x}^e) \cdot \Phi}{J(\mathbf{x})} \quad [as \ |\Phi| = 1] \\ \Rightarrow & \frac{4}{k} \frac{|\mathbf{x} - \mathbf{x}^e|}{J(\mathbf{x})} \geq \frac{4}{k} \frac{(\mathbf{x} - \mathbf{x}^e) \cdot \Phi}{J(\mathbf{x})} \end{aligned}$$

Hence, if C is required to satisfy the relation $C > \frac{4}{k} \frac{(\mathbf{x} - \mathbf{x}^e) \cdot \Phi}{J(\mathbf{x})}$, then $C > \frac{4}{k} \frac{|\mathbf{x} - \mathbf{x}^e|}{J(\mathbf{x})}$ can fully meet this requirement.

Therefore, according to **Definition 4**, V is a positive definite function, and the second condition is proved to be satisfied.

Condition 3: Taking the derivative of V with t , we get

$$\begin{aligned} \frac{dV}{dt} &= \frac{\partial V}{\partial x_1} \cdot \frac{dx_1}{dt} + \frac{\partial V}{\partial x_2} \cdot \frac{dx_2}{dt} + \dots + \frac{\partial V}{\partial x_n} \cdot \frac{dx_n}{dt} \\ &= \sum_{p=1}^n \frac{\partial V}{\partial x_p} \cdot \frac{dx_p}{dt} \end{aligned} \quad (17)$$

The next step is to deal with $\frac{\partial V}{\partial x_p}$ and $\frac{dx_p}{dt}$ ($p = 1, 2, \dots, n$), respectively.

$$\begin{aligned} \frac{\partial V}{\partial x_1} &= \frac{kC^2\phi_1^2}{8}J'_{x_1} - \frac{C\phi_1^3}{2} \\ \frac{\partial V}{\partial x_p} &= \frac{kC^2\phi_1^2}{8}J'_{x_p} - \frac{C\phi_1^2\phi_p}{2} \quad (p = 2, \dots, n) \\ \frac{dx_1}{dt} &= v_1 \\ \frac{dx_p}{dt} &= v_p = \frac{\phi_p}{\phi_1} \cdot v_1 \quad (p = 2, \dots, n) \end{aligned}$$

Based on the four equations above, we obtain

$$\begin{aligned} \frac{\partial V}{\partial x_1} \cdot \frac{dx_1}{dt} &= \left(\frac{kC^2\phi_1^2}{8}J'_{x_1} - \frac{C\phi_1^3}{2} \right) \cdot v_1 \\ \frac{\partial V}{\partial x_p} \cdot \frac{dx_p}{dt} &= \left(\frac{kC^2\phi_1^2}{8}J'_{x_p} - \frac{C\phi_1^2\phi_p}{2} \right) \cdot v_p \\ &= \left(\frac{kC^2\phi_1^2}{8}J'_{x_p} - \frac{C\phi_1^2\phi_p}{2} \right) \cdot \frac{\phi_p}{\phi_1} v_1 \\ &= \left(\frac{kC^2\phi_1\phi_p}{8}J'_{x_p} - \frac{C\phi_1\phi_p^2}{2} \right) \cdot v_1 \end{aligned}$$

From Eq.(17)

$$\begin{aligned} \frac{dV}{dt} &= \left(\frac{kC^2\phi_1^2}{8}J'_{x_1} + \sum_{p=2}^n \left(\frac{kC^2\phi_1\phi_p}{8}J'_{x_p} \right) - \frac{C\phi_1^3}{2} \right. \\ &\quad \left. - \sum_{p=2}^n \left(\frac{C\phi_1\phi_p^2}{2} \right) \right) \cdot v_1 \\ &= \left(\frac{kC^2\phi_1}{8} \cdot \sum_{p=1}^n (J'_{x_p} \phi_p) - \frac{C\phi_1}{2} \cdot \sum_{p=1}^n \phi_p^2 \right) \cdot v_1 \end{aligned}$$

Since $\Phi = (\phi_1, \phi_2, \dots, \phi_n)$ is a vector of unit length, $\Phi \cdot \Phi = \sum_{p=1}^n \phi_p^2 = 1$. Then,

$$\frac{dV}{dt} = \left(\frac{kC^2\phi_1}{8} (\text{grad}J(\mathbf{x}) \cdot \Phi) - \frac{C\phi_1}{2} \right) \cdot v_1$$

where $\text{grad}J(\mathbf{x}) = (J'_{x_1}, J'_{x_2}, \dots, J'_{x_n})$ is the gradient of the objective function at $\mathbf{x} = (x_1, x_2, \dots, x_n)$.

According to Eq.(8), $v_1 = -\frac{kC^2\phi_1}{8}(\text{grad}J(\mathbf{x}) \cdot \Phi) + \frac{C\phi_1}{2}$. Hence, the previous equation is changed into

$$\frac{dV}{dt} = - \left(-\frac{kC^2\phi_1}{8}(\text{grad}J(\mathbf{x}) \cdot \Phi) + \frac{C\phi_1}{2} \right)^2$$

$$i.e., \quad \frac{dV}{dt} = -v_1^2$$

When $\mathbf{x} = \mathbf{x}^e$, we have $C = 0$, then get $v_1 = 0$ according to Eq.(8), and finally obtain $dV/dt = -v_1^2 = 0$. When $\mathbf{x} \neq \mathbf{x}^e$, dV/dt meets the relation $dV/dt < 0$ as we have assumed

Table 1: The positions and their information of bacterium iterations with a step size $C = 0.2$ over $J(\mathbf{x}) = x_1^2 + x_2^2$

Iteration	Position	Step size (C)	Velocity	Tumble direction (Φ)	grad $J(\mathbf{x})$	grad $J(\mathbf{x}) \cdot \Phi$	d
1	(-0.5, -0.4)	0.2	0.2	$(\frac{1}{2}, \frac{\sqrt{3}}{2})$	(-1, -0.8)	-1.1928	1.2806
2	(-0.4, -0.2268)	0.2	0.2	$(\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2})$	(-0.8, -0.4536)	-0.2449	0.9196
3	(-0.2586, -0.3682)	0.2	0	$(-\frac{\sqrt{11}}{4}, -\frac{\sqrt{5}}{4})$	(-0.5172, -0.7364)	0.8405	0.8999
4	(-0.2586, -0.3682)	0.2	0.2	$(\frac{\sqrt{3}}{3}, \frac{\sqrt{6}}{3})$	(-0.5172, -0.7364)	-0.8999	0.8999
5	(-0.1431, -0.2049)	0.2	0	$(-\frac{\sqrt{7}}{4}, \frac{3}{4})$	(-0.2862, -0.4098)	-0.1180	0.4998
6	(-0.1431, -0.2049)	0.2	0.2	$(-\frac{1}{4}, \frac{\sqrt{15}}{4})$	(-0.2862, -0.4098)	-0.1626	0.4998
7	(-0.1931, -0.0113)	0.2	0.2	$(\frac{\sqrt{30}}{6}, \frac{\sqrt{6}}{6})$	(-0.3862, -0.0226)	-0.3618	0.3869
8	(-0.0105, 0.0703)	0.2	0	Any direction	(-0.0210, 0.1406)	--	0.1422

 Table 2: The positions and their information of bacterium iterations with an adjusted step size $C = 0.02$ over $J(\mathbf{x}) = x_1^2 + x_2^2$

Iteration	Position	Step size (C)	Velocity	Tumble direction (Φ)	grad $J(\mathbf{x})$	grad $J(\mathbf{x}) \cdot \Phi$	d
9	(-0.0105, 0.0703)	0.02	0.02	$(\frac{\sqrt{30}}{6}, -\frac{\sqrt{6}}{6})$	(-0.0210, 0.1406)	-0.0766	0.1422
10	(0.0078, 0.0621)	0.02	0	$(\frac{\sqrt{7}}{4}, \frac{3}{4})$	(0.0156, 0.1242)	0.0517	0.1252
11	(0.0078, 0.0621)	0.02	0.02	$(\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2})$	(0.0156, 0.1242)	-0.0768	0.1252
12	(0.0219, 0.0480)	0.02	0.02	$(-\sqrt{\frac{1}{3}}, -\sqrt{\frac{2}{3}})$	(0.0438, 0.0960)	-0.0807	0.1055
13	(0.0104, 0.0249)	0.02	0.02	$(-\sqrt{\frac{2}{5}}, -\sqrt{\frac{3}{5}})$	(-0.0126, -0.0560)	-0.0719	0.0540
14	(0.0052, -0.0049)	0.02	0	Any direction	(0.0104, -0.0098)	--	0.0193

$v_1 \neq 0$ in section III. Thus, according to **Definition 5**, dV/dt is a negative definite function, and the third condition is also proved to be satisfied.

Based on the analysis above, V defined in Eq.(15) has met the three conditions in **Theorem 1**, and is a Lyapunov function. Therefore, the chemotaxis dynamics represented by Eq.(10) or Eq.(11) is asymptotically stable at the equilibrium state \mathbf{x}^e . Here, there are a couple of things to note about the Lyapunov function V . One thing is that the restriction on the step size in Eq.(16) makes the step size C look complicated and seem to be a function of \mathbf{x} , but it is actually not. The step size C is only a constant which varies at the different stages of the optimization process, and would not change with \mathbf{x} from moment to moment. The experimental section illustrates the change of the step size. The other thing is that in control theory, if a dynamic system is asymptotically stable at the equilibrium state, there must be a Lyapunov function regarding the dynamic system, but the Lyapunov function is not the only one. In other words, the existence of a Lyapunov function is only a sufficient condition for the stability of the dynamic system, and not a necessary and sufficient condition. Hence, the Lyapunov function V defined in Eq.(15) is merely a tool to prove the asymptotically stability of the chemotaxis dynamics at the equilibrium state \mathbf{x}^e , and the restriction on the step size in Eq.(16) is only one criterion. There may be another Lyapunov function of regarding the chemotaxis dynamics and another criterion of the step size. Given this, in experimental section, we only focus on how to validate the two key conditions obtained in section III, and ignore the specific restriction, such as Eq.(16).

5. simulation results

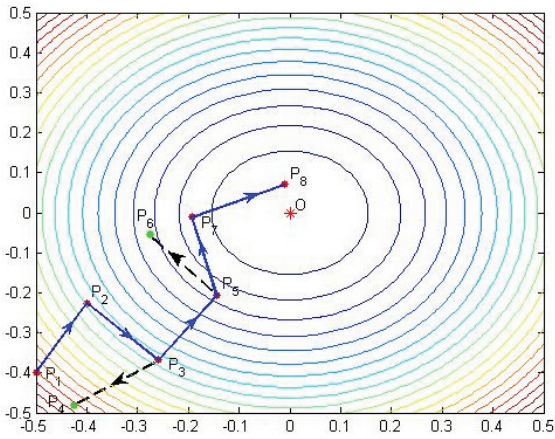
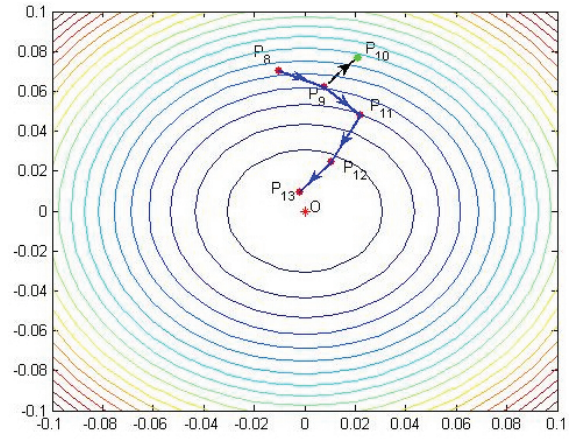
For the bacterium to continuously move toward the equilibrium state, this section gives two examples to verify the two

necessary conditions obtained by theoretical analysis in Section III. The two necessary conditions are: (1) The angle between the moving direction and the gradient direction must satisfy the relation $\pi/2 < \theta \leq \pi$ (that is, $\text{grad}J(\mathbf{x}) \cdot \Phi < 0$), and (2) The step size C must be smaller than the longest distance d between the current position and any other location on the same contour (i.e., $C < d$). The two examples are specific to the 2-D case since it is ease of visualization and without loss of generality.

One example is the objective function $J(\mathbf{x}) = x_1^2 + x_2^2$, where the single optimum is located at $\mathbf{x}^e = (0, 0)$. The gradient of this objective function is $\text{grad}J(\mathbf{x}) = (2x_1, 2x_2)$ at point $\mathbf{x} = (x_1, x_2)$. The longest distance d between the current position and any other position on the same contour is calculated by $d = 2 \cdot \|\mathbf{x} - \mathbf{x}^e\|$. Let the bacterium move under the guidance of the actual BFO algorithm, and start at $(x_1, x_2) = (-0.5, -0.4)$. With the bacterium moving, Tables 1 and 2 list the positions that the bacterium arrives at and some position-relevant information. As Table 1 shown, the bacterium moves with a step size $C = 0.2$ in the early stages of the optimization process. When the bacterium reaches the position $(-0.2586, -0.3682)$ in the third iteration, it selects the direction $(-\sqrt{11}/4, -\sqrt{5}/4)$ to move. However, $(-\sqrt{11}/4, -\sqrt{5}/4)$ is a direction of increasing the objective function value, so the bacterium keeps in place (i.e., its velocity is 0). At this time, $\text{grad}J(\mathbf{x}) \cdot \Phi > 0$. That is, when $\text{grad}J(\mathbf{x}) \cdot \Phi > 0$, the bacterium does not move because doing so would increase the objective function value. Obviously, this result supports the necessary condition (1). When the bacterium gets to the position $(-0.1431, -0.2049)$ in the fifth iteration, despite the fact that it obeys the two conditions since $\text{grad}J(\mathbf{x}) \cdot \Phi = -0.1180 < 0$ and $C = 0.2 < d = 0.4998$, it still does not move. This illustrates that the two conditions are only necessary conditions and not sufficient conditions for the bacterium to reduce the objective function value. Once the bacterium arrives at the position $(-0.0105, 0.0703)$ in the eighth

Table 3: The positions and their information of bacterium iterations with a step size $C = 0.2$ over $J(\mathbf{x}) = 1 - e^{-(x_1^2+x_2^2)}$

Iteration	Position	Step size (C)	Velocity	Tumble direction (Φ)	grad $J(\mathbf{x})$	grad $J(\mathbf{x}) \cdot \Phi$	d
1	(0.4, -0.6)	0.2	0.2	$(-\frac{1}{2}, \frac{\sqrt{3}}{2})$	(0.4756, -0.7134)	-0.8556	1.4422
2	(0.3, -0.4268)	0.2	0.2	$(\sqrt{\frac{3}{7}}, \sqrt{\frac{4}{7}})$	(0.4570, -0.6502)	-0.1923	1.0434
3	(0.4309, -0.2756)	0.2	0.2	$(-\sqrt{\frac{5}{6}}, -\sqrt{\frac{1}{6}})$	(0.6634, -0.4243)	-0.4324	1.0230
4	(0.2483, -0.3572)	0.2	0	$(\sqrt{\frac{3}{5}}, \sqrt{\frac{2}{5}})$	(0.6001, -0.8632)	-0.0811	0.9131
5	(0.2483, -0.3572)	0.2	0.2	$(-\sqrt{\frac{2}{3}}, \sqrt{\frac{1}{3}})$	(0.6001, -0.8632)	-0.7375	0.9131
6	(0.0850, -0.2417)	0.2	0	$(-\frac{1}{4}, -\frac{\sqrt{15}}{4})$	(0.1592, -0.4527)	0.3985	0.5124
7	(0.0850, -0.2417)	0.2	0.2	$(-\frac{\sqrt{3}}{2}, \frac{1}{2})$	(0.1592, -0.4527)	-0.4735	0.5124
8	(-0.0882, -0.1417)	0.2	0	$(\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2})$	(-0.1716, -0.2756)	0.2220	0.3292
9	(-0.0882, -0.1417)	0.2	0.2	$(\sqrt{\frac{8}{11}}, \sqrt{\frac{3}{11}})$	(-0.1716, -0.2756)	-0.2903	0.3292
10	(0.0824, -0.0373)	0.2	0	Any direction	(0.1635, -0.0740)	--	0.1809


 Figure 4: The moving trajectory of the bacterium with a step size $C = 0.2$ over $J(\mathbf{x}) = x_1^2 + x_2^2$

 Figure 5: The moving trajectory of the bacterium with a step size $C = 0.02$ over $J(\mathbf{x}) = x_1^2 + x_2^2$

iteration, no matter which direction it chooses, it never moves. Because the step size $C = 0.2$ is too large for the bacterium, which goes against the necessary condition (2) and leads to the bacterium trapped into the position $(-0.0105, 0.0703)$. At this moment, to make the bacterium jump out of this position and keep walking towards the optimal solution, the step size should be adjusted to meet the necessary condition (2). As Table 2 shown, given a smaller step size $C = 0.02$ ($C = 0.02 < d = 0.1422$), the bacterium again is able to go on moving from the position $(-0.0105, 0.0703)$. When the bacterium arrives at the position $(0.0052, -0.0049)$ in the fourteenth iteration, it once again get trapped into stagnating because the value of d gradually gets smaller with the bacterium approaching the optimal solution, and the step size $C = 0.02$ no longer meets the necessary condition (2). If the step size C continues to be adjusted to satisfy the condition (2), the bacterium will keep on moving towards the optimal solution. This fact exactly suggests that a constant step size is not conducive to search the optimum. Because a constant step size easily makes the bacterium trapped into a non-optimal position with the bacterium getting closer and closer to the optimum. That is why many researches explore the self-adaptive modes of the step size [9, 25–27].

To more clearly show the movement of the bacterium, we

plot it in Figs. 4 and 5. In the two figures, the red asterisk indicates the optimum (i.e., the equilibrium state), marked by “O”. The closed curves represent different contour lines. The blue line with the arrow is the trajectory of the bacterium, and the black dotted line with the arrow represents the wrong direction that the bacterium selects at a certain position. As seen from Fig. 4, the bacterium starts from the position P_1 $(-0.5, -0.4)$, and gets to P_2 $(-0.4, -0.2268)$ by taking a step size of 0.2 in the direction $(1/2, \sqrt{3}/2)$. Then it moves along the direction $(\sqrt{2}/2, -\sqrt{2}/2)$, and arrives at P_3 $(-0.2586, -0.3682)$. At this position, the bacterium first chooses a direction $(-\sqrt{11}/4, -\sqrt{5}/4)$ which makes $\text{grad}J(\mathbf{x}) \cdot \Delta > 0$. If the bacterium walks in this direction, it will reach P_4 , where the corresponding objective function value is larger than that at the position P_3 . So this direction is a wrong direction, and the bacterium has to reselect a new direction $(\sqrt{3}/3, \sqrt{6}/3)$, and gets to P_5 $(-0.1431, -0.2049)$. Next, the bacterium intends to move in the direction $(-\sqrt{7}/4, 3/4)$ at first. However, in this situation, although the two necessary conditions are met (i.e., $\text{grad}J(\mathbf{x}) \cdot \Phi < 0$ and $C < d$), moving in this direction is still a bad choice. Because if doing so, the bacterium will arrive at the position P_6 , where the objective function value is larger than that on the position P_5 . Therefore,

Table 4: The positions and their information of bacterium iterations with an adjusted step size $C = 0.03$ over $J(\mathbf{x}) = 1 - e^{-(x_1^2+x_2^2)}$

Iteration	Position	Step size (C)	Velocity	Tumble direction (Φ)	grad $J(\mathbf{x})$	grad $J(\mathbf{x}) \cdot \Phi$	d
11	(0.0824, -0.0373)	0.03	0.03	$(-\frac{\sqrt{3}}{2}, \frac{1}{2})$	(0.1635, -0.0740)	-0.1786	0.1809
12	(0.0564, -0.0223)	0.03	0.03	$(-\sqrt{\frac{5}{6}}, -\sqrt{\frac{1}{6}})$	(0.1132, -0.0448)	-0.0850	0.1213
13	(0.0290, -0.0345)	0.03	0	$(\sqrt{\frac{2}{3}}, \sqrt{\frac{1}{3}})$	(0.0581, -0.0691)	0.0075	0.0901
14	(0.0290, -0.0345)	0.03	0.03	$(-\sqrt{\frac{1}{5}}, \sqrt{\frac{4}{5}})$	(0.0581, -0.0691)	-0.0438	0.0901
15	(0.0156, -0.0077)	0.03	0.03	$(-\sqrt{\frac{5}{7}}, \sqrt{\frac{2}{7}})$	(0.0312, -0.0154)	-0.0346	0.0348
16	(-0.0098, 0.0083)	0.03	0	Any direction	(-0.0196, 0.0166)	--	0.0257

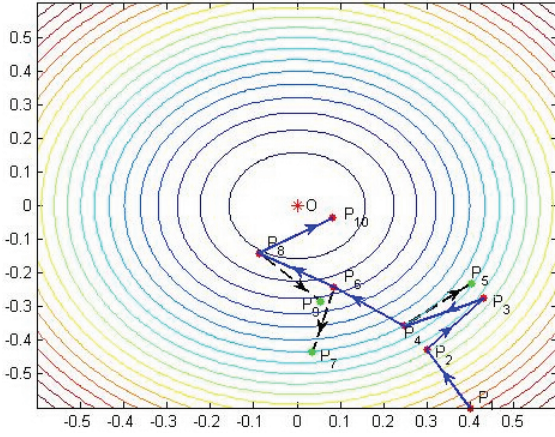


Figure 6: The moving trajectory of the bacterium with a step size $C = 0.2$ over $J(\mathbf{x}) = 1 - e^{-(x_1^2+x_2^2)}$

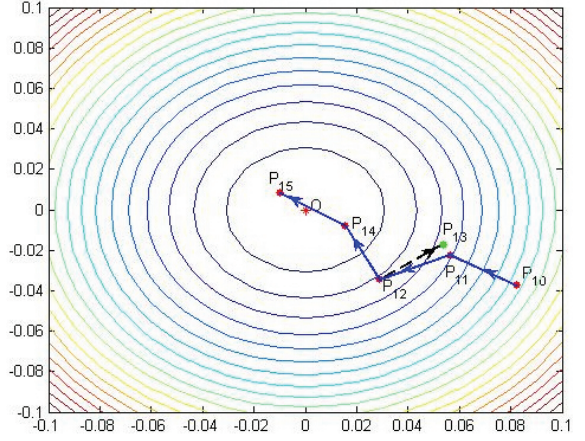


Figure 7: The moving trajectory of the bacterium with a step size $C = 0.03$ over $J(\mathbf{x}) = 1 - e^{-(x_1^2+x_2^2)}$

the bacterium again chooses a new direction $(-1/4, \sqrt{15}/4)$, and gets to the position $P_7 (-0.1931, -0.0113)$. Finally, the bacterium reaches the position $P_8 (-0.0105, 0.0703)$ by taking a step in the direction $(\sqrt{30}/6, \sqrt{6}/6)$. But the bacterium is trapped into this position due to that the step size $C = 0.2$ is too large, and has violated the necessary condition $C < d$. To escape from the position P_8 and go on moving towards the optimal solution, the bacterium has to adjust its step size to a smaller value (e.g. $C = 0.02$) which meets the necessary condition $C < d$. As shown in 5, the bacterium successfully jumps out of P_8 with the smaller step of 0.02, and arrives at P_9 by taking a step size in the direction $(\sqrt{30}/6, -\sqrt{6}/6)$. Afterward, the bacterium tries to move in the direction $(\sqrt{7}/4, 3/4)$, but it fails because moving in this direction will increase the objective function value. So the bacterium has to reselect a new direction $(\sqrt{2}/2, -\sqrt{2}/2)$, and gets to the position P_{11} . Subsequently, the bacterium reaches the position P_{13} by moving twice, one towards the direction $(-\sqrt{1/3}, -\sqrt{2/3})$ and another towards the direction $(-\sqrt{2/5}, -\sqrt{3/5})$. However, while gradually getting closer to the optimal solution, the bacterium once again gets trapped into this non-optimal position due to the step size $C = 0.02$ no longer meets the necessary condition $C < d$.

The other example is the objective function $J(\mathbf{x}) = 1 - e^{-(x_1^2+x_2^2)}$, where the single optimum is also located at $\mathbf{x}^e = (0, 0)$. The gradient of this objective function is given by $\text{grad}J(\mathbf{x}) = (2x_1 \cdot e^{-(x_1^2+x_2^2)}, 2x_2 \cdot e^{-(x_1^2+x_2^2)})$. The longest distance d between

the current position and any other position on the same contour is also calculated by $d = 2 \cdot \|\mathbf{x} - \mathbf{x}^e\|$. Similarly, let the bacterium move under the guidance of the actual BFO algorithm, and start at the position $(x_1, x_2) = (0.4, -0.6)$. Tables 3 and 4 list the positions that the bacterium arrives at and some position-relevant information. Figs. 6 and 7 clearly show the moving trajectory of the bacterium. Among them, Table 3 and Fig. 6 illustrate the movement of the bacterium with a step size of 0.2 in the early stage of the optimization process. From Table 3 and Fig. 6, it is easy to find that the bacterium tries twice to select a good moving direction when it gets to the position P_4 in the fourth iteration. On the first try, the bacterium fails to move because taking a step in the selected direction $(\sqrt{3/5}, \sqrt{2/5})$ will increase the objective function value despite the two necessary conditions are met. Besides, the bacterium also takes two tries at the position P_6 in the sixth iteration and at the position P_8 in the eighth iteration, respectively. The reason for the failures on the first try is that the selected directions lead to $\text{grad}J(\mathbf{x}) \cdot \Phi > 0$, which goes against the necessary condition (1). When the bacterium arrives at the position P_{10} , it gets trapped into this position due to that $C = 0.2 > d = 0.1809$, which violates the necessary condition (2). To continue to search for a better solution, the bacterium has to change its step size into a smaller value to meet the condition $C < d$. Table 4 and Fig. 6 show the movement of the bacterium with a smaller step size of 0.03 in the latter stage of the optimization process. In this stage,

the bacterium also goes through a failure attempt at the position P_{12} . The bacterium first selects a direction $(\sqrt{2/3}, \sqrt{1/3})$, but this direction makes $gradJ(\mathbf{x}) \cdot \Delta > 0$ which violates the condition (1) and would increase the objective function value. At last, the movement of this stage ends up with the bacterium trapped into the position P_{15} for the same reason that the step size $C = 0.03 > d = 0.0257$, which again violates the necessary condition (2).

In the above two examples, the two necessary conditions obtained by the theoretical analysis in Section III have been visually and effectively validated. The two examples clearly illustrate that a proper direction and a varying step size are crucially important to make the bacterium move towards the optimum, which are not only the theoretical basis of studies on developing effective BFO algorithms, but are also the reasons behind the success of these studies [9, 25–27, 31, 32]. That is, to make the bacterium pretty close to the optimal solution, the step size should be adjusted every once in a while except to select the correct direction.

6. Conclusion

In this paper, we have extended the theoretical analysis of the chemotaxis mechanism in BFO algorithm from 1-D objective functions to multi-dimensional objective functions. We first derived the general mathematical model of a single bacterium performing the chemotaxis process over multi-dimensional objective functions. Then we analyzed the general mathematical model in light of the general descent search, and pointed out two necessary conditions for the bacterium continuously moving toward the optimum. Afterwards, the Lyapunov stability theorem was applied to prove the asymptotically stability of the chemotaxis dynamics over multi-dimensional objective functions near the equilibrium state (i.e., the optimal solution). At last, the experimental simulations took two representative examples to validate the two necessary conditions. Both the theoretical analysis and experimental simulations suggested that a right direction and a varying step size (i.e., a adaptation scheme) are very essential for the bacterium to search the optimal solution.

In future work, we will focus on developing superior BFO algorithms for the single objective function optimization problem, the multi-objective function optimization problem, and the combinational optimization problem based on the theoretical analysis.

Compliance with Ethical Standard

Funding: This work was partly supported by the NSFC Research Program (61672065, 613750592), the Beijing Municipal Education Research Plan key project (Beijing Municipal Fund Class B) (KZ201410005004), and the Beijing Postdoctoral Research Foundation (2017-ZZ-024).

Conflict of Interest: All the authors declare that there is no conflict of interest.

Ethical approval: This article does not contain any studies with human participants or animals performed by any of the authors.

References

- [1] Hinchey MG, Sterritt R, Rouff C (2007) Swarms and swarm intelligence. *IEEE Computer* 40(4):111-113.
- [2] Roach JH, Marks RJ, Thompson BB (2015) Recovery from sensor failure in an evolving multiobjective swarm. *IEEE Trans Syst Man Cybern Syst* 45(1):170-174.
- [3] Todorovic N, Petrovic S (2013) Bee colony optimization algorithm for nurse rostering. *IEEE Trans Syst Man Cybern Syst* 43(2):467-473.
- [4] Fu Y, Ding M, Zhou C, Hu H (2013) Route planning for unmanned aerial vehicle (UAV) on the sea using hybrid differential evolution and quantum-behaved particle swarm optimization. *IEEE Trans Syst Man Cybern Syst* 43(6):1451-1465.
- [5] Passino KM (2002) Biomimicry of bacterial foraging for distributed optimization and control. *Control Systems* 22(3): 52-67.
- [6] Bermejo E, Cordon O, Damas S, Santamaría J (2013) Quality time-of-flight range imaging for feature-based registration using bacterial foraging. *Appl Soft Comput* 13(6):3178-3189.
- [7] Bermejo E, Cordon O, Damas D, Santamaría J (2005) A Comparative study on the application of advanced bacterial foraging models to image registration. *Inf. Sci.* 295:160-181.
- [8] Devi S, Geethanjali M (2014) Application of modified bacterial foraging optimization algorithm for optimal placement and sizing of distributed generation. *Expert Syst Appl* 41(6): 2772-2781.
- [9] Majhi R, Panda G, Majhi B, Sahoo G (2009) Efficient prediction of stock market indices using adaptive bacterial foraging optimization (ABFO) and BFO based techniques. *Expert Syst Appl* 36(6): 10097-10104.
- [10] Panda R, Naik MK (2015) A novel adaptive crossover bacterial foraging optimization algorithm for linear discriminant analysis based face recognition. *Appl Soft Comput* 30: 722-736.
- [11] Tan L, Lin F, Wang H (2015) Adaptive comprehensive learning bacterial foraging optimization and its application on vehicle routing problem with time windows. *Neurocomputing* 151(3): 1208-1215.
- [12] Verma OP, Hanmandlu M, Kumar P, Chhabra S, Jindal A (2011) A novel bacterial foraging technique for edge detection. *Pattern Recogn Lett* 33(8): 1187-1196.
- [13] Oyekan J, Gu D, Hu H (2013) Visual imaging of invisible hazardous substances using bacterial inspiration. *IEEE Trans Syst Man Cybern Syst* 43(5): 1105-1115.
- [14] Yang C, Ji J, Liu J, Liu J, Yin B (2016) Structural learning of bayesian networks by bacterial foraging optimization. *Int J Approx Reason* 69: 147-167.
- [15] Das S, Suganthan PN (2011) Differential evolution: a survey of the state-of-the-art. *IEEE Trans Evol Comput* 15(1): 4-31.
- [16] Dasgupta S, Biswas A, Abraham A, Das S (2008) Adaptive computational chemotaxis in Bacterial foraging algorithm. In: *Proceedings of IEEE International Conference Complex, Intelligent and Software Intensive Systems*, pp 64-71.
- [17] Dasgupta S, Das S, Abraham S, Biswas A (2009) Adaptive computational chemotaxis in bacterial foraging optimization: an analysis. *IEEE Trans Evol Comput* 13(4): 919-941.
- [18] Abraham A, Biswas A, Dasgupta S, Das S (2008) Analysis of reproduction operator in bacterial foraging optimization algorithm. In: *Proceedings of IEEE Congress on Evolutionary Computation*, pp 1476-1483.
- [19] Das S, Dasgupta S, Biswas A, Abraham A, Konar A (2009) On the stability of the chemotactic dynamics in bacterial-foraging optimization algorithm. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans* 39(3): 670-679.
- [20] Kennedy J, Eberhart R (1995) Particle swarm optimization. In: *Proceedings of the 1995 IEEE International Conference on Neural Networks*, pp 1942-1948.
- [21] Clerc M, Kennedy J (2002) The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans Evol Comput* 6(1): 58-73.

- [22] Trelea IC (2002) The particle swarm optimization algorithm: convergence analysis and parameter selection. *Inform Process Lett* 85(6): 317-325.
- [23] Kadirkamanathan V, Selvarajah K, Fleming PJ (2006) Stability analysis of the particle dynamics in particle swarm optimizer. *IEEE Trans Evol Comput* 10(3): 245-255.
- [24] Poli R, Kennedy J, Blackwell T (2007) Particle swarm optimization. *Swarm Intelligence* 1(1): 33-57.
- [25] Chen H, Niu B, Ma L, Zhu Y (2014) Bacterial colony foraging optimization. *Neurocomputing* 37: 268-284.
- [26] Chen H, Zhu Y, Hu K, Ma K (2014) Bacterial colony foraging algorithm: combining chemotaxis, cell-to-cell communication, and self-adaptive strategy. *Inf Sci* 273: 73-100.
- [27] Yang C, Ji J, Liu J, Yin B (2016) Bacterial foraging optimization using chemotaxis and conjugation strategies. *Inf Sci* 363: 72-95.
- [28] Hernández-Ocana B, Pozos-Parra MDP, Mezura-Montes E (2016) Improved modified bacterial foraging optimization algorithm to solve constrained numerical optimization problems. *Appl Math Inf Sci* 10(2) 607-622.
- [29] Kanwal RP (2012) *Generalized Functions Theory and Technique: Theory and Technique*. Springer Science & Business Media.
- [30] Snyman J (2005) *Practical mathematical optimization: an introduction to basic optimization theory and classical and new gradient-based algorithms*. Springer Science & Business Media.
- [31] Biswas A, Dasgupta S, Das S, Abraham A (2007) Synergy of PSO and bacterial optimization—a comparative study on numerical benchmarks. *Innovations in Hybrid Intelligent Systems*, Springer Berlin Heidelberg, pp 255-263.
- [32] Benbouabdallah K, Zhu Q (2013) Bacterial foraging oriented by particle swarm optimization of a Lyapunov-based controller for mobile robot target tracking. In: *Proceedings of 2013 Ninth International Conference on Natural Computation* pp 506-511.
- [33] Kuo BC (1987) *Automatic control systems*. Prentice Hall PTR.
- [34] Haddad WM, Chellaboina VS (2008) *Nonlinear dynamical systems and control: a Lyapunov-based approach*. Princeton University Press.
- [35] Shen Q, Shi P (2016) Output consensus control of multiagent systems with unknown nonlinear dead zone. *IEEE Trans Syst Man, Cybern Syst* 46(10): 1329-1337.
- [36] Wang D, Liu D, Zhang Q, Zhao D (2016) Data-Based adaptive critic designs for nonlinear robust optimal control With uncertain dynamics. *IEEE Trans Syst Man, Cybern Syst* 46(11): 1544-1555.