On Stability of the Chemotactic Dynamics in Bacterial-Foraging Optimization Algorithm

Swagatam Das, Sambarta Dasgupta, Arijit Biswas,
Ajith Abraham, *Senior Member, IEEE*, and
Amit Konar, *Member, IEEE*

***Abstract*—Bacterial-foraging optimization algorithm (BFOA) attempts to model the individual and group behavior of *E.Coli* bacteria as a distributed optimization process. Since its inception, BFOA has been finding many important applications in real-world optimization problems from diverse domains of science and engineering. One key step in BFOA is the computational chemotaxis, where a bacterium (which models a candidate solution of the optimization problem) takes steps over the foraging landscape in order to reach regions with high-nutrient content (corresponding to higher fitness). The simulated chemotactic movement of a bacterium may be viewed as a guided random walk or a kind of stochastic hill climbing from the viewpoint of optimization theory. In this paper, we first derive a mathematical model for the chemotactic movements of an artificial bacterium living in continuous time. The stability and convergence-behavior of the said dynamics is then analyzed in the light of Lyapunov stability theorems. The analysis indicates the necessary bounds on the chemotactic step-height parameter that avoids limit cycles and guarantees convergence of the bacterial dynamics into an isolated optimum. Illustrative examples as well as simulation results have been provided in order to support the analytical treatments.**

***Index Terms*—Bacterial foraging, biological systems, computational chemotaxis, limit cycles, stability analysis.**

NOMENCLATURE

p	Dimension of the search space.
S	Total number of bacteria in the population.
Nc	Number of chemotactic steps.
N_s	Swimming length.
N_{re}	Number of reproduction steps.
N_{ed}	Number of elimination–dispersal events.
P_{ed}	Elimination–dispersal probability,
$C(i)$	Size of the step taken in the random direction specified by the tumble.

I. INTRODUCTION

For over the last five decades, metaheuristics like genetic algorithms (GAs) [1], [2], evolutionary programming [3], and evolutionary strategies [4], which draw their inspiration from evolution and natural genetics, have been dominating the realm of optimization algorithms. Recently, algorithms like particle swarm optimization (PSO) [5] and ant-colony optimization (ACO) [6], mimicking the collective behavior of social insects, have found their way into this domain and proved their effectiveness in solving several engineering optimization problems [7]. Following the same trend of nature-inspired computing, Passino *et al.* [8], [9] proposed the bacterial-foraging optimization algorithm (BFOA) in 2002. Unlike the classical evolutionary techniques, BFOA is based on the foraging theory of natural creatures that try to optimize (maximize) their energy intake per unit time spent for foraging, considering all the constraints presented by their own physiology, such as sensing and cognitive capabilities, and environment (e.g., density of prey, risks from predators, physical characteristics of the search space). Although BFOA has certain characteristics analogous to an evolutionary algorithm [8, p. 63], it is not directly connected to Darwinian evolution and natural genetics, which formed the basis of the GA-type algorithms in the early 1970s. To date, the algorithm has successfully been applied to several real-life problems like optimal controller design [8], [10], harmonic estimation [11], transmission-loss reduction [12], active-power-filter synthesis [13], and machine learning [14]. On the algorithmic front, extensions have been made to deal with complex and multimodal fitness landscapes and dynamical environments and to obtain efficient convergence behavior [15]–[19]. BFOA has also been hybridized with a few other state-of-the-art evolutionary computing techniques [10], [20], [21] in order to achieve robust and efficient search performances. Over certain real-world optimization problems, BFOA has been reported to outperform many powerful metaheuristics like GA, PSO, etc., in terms of convergence speed and final accuracy (for example, see [11], [13], [17], and [20]). The efficiency of the algorithm in solving real-parameter optimization problems has made it a potential optimization algorithm of current interest, worth investing research time. On the other hand, a downside to the algorithm is that it has a large number of control parameters as compared to PSO or ACO, and its performance critically depends on the choice of these parameters. Determining the suitable values of these control parameters necessitates a detailed mathematical analysis of the search operators of BFOA. This paper makes a humble attempt to contribute in this context.

One major step in BFOA is the simulated chemotactic movement. Chemotaxis is a foraging strategy that implements one type of local optimization where the bacteria try to climb up the nutrient concentration, avoid noxious substance, and search for ways out of neutral media. This step has much resemblance with a biased random-walk model [22]. The chemotactic operator employed in BFOA is supposed to guide the swarm to converge toward optima. In this paper, we make an attempt to find out under what conditions this local search strategy leads to a stable dynamics that can avoid limit cycles and asymptotically converge toward an optimum of the fitness landscape. The stability analysis has been undertaken using the Lyapunov's stability theorems from classical nonlinear control theory [23], [24]. Finally, we determine the bounds on the chemotactic step-size parameter C, which ensures asymptotic stability. Results of computer simulations have been provided in order to support the theoretical claims made in this paper. Although the analysis may appear to have a limited scope, note that this paper is the first of its kind, and the issues of multibacterial population over a multidimensional fitness landscape are topics of further research. In this paper, our primary objective is to gain important insight into the operational mechanism of the artificial bacterial-foraging system, acting as a function optimizer.

The rest of this paper is organized as follows. Section II describes the classical BFOA in sufficient details. In Section III, differential-equation model governing the motion of an individual bacterium in chemotaxis phase is derived. The model is then used to carry out stability analysis in Section IV. Results of computer simulations have

Manuscript received May 3, 2008; revised October 7, 2008. First published February 6, 2009; current version published April 17, 2009. This paper was recommended by Associate Editor J. Wu.

S. Das, S. Dasgupta, A. Biswas, and A. Konar are with the Department of Electronics and Telecommunication Engineering, Jadavpur University, Kolkata 700 032, India (e-mail: swagatamdas19@yahoo.co.in; sambartadg@gmail.com; arijitbiswas87@gmail.com; amitkonar_2030@rediffmail.com).

A. Abraham is with the Centre of Excellence for Quantifiable Quality of Service in Communication Systems, Centre of Excellence, Norwegian University of Science and Technology, 7491 Trondheim, Norway, and also with the Machine Intelligence Research Labs (MIR Labs), Scientific Network for Innovation and Research Excellence, Auburn, WA 98071 USA (e-mail: ajith.abraham@ieee.org).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCA.2008.2011474

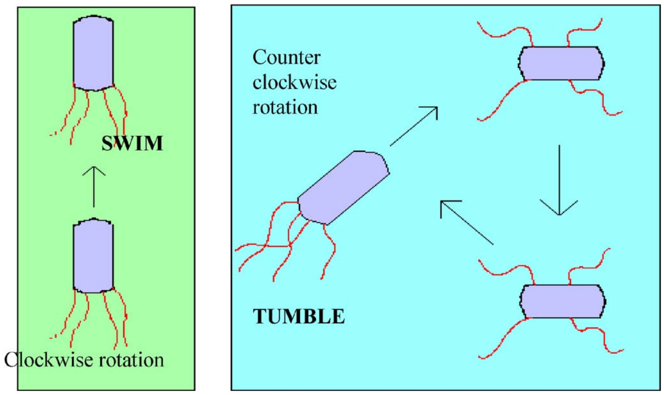


Fig. 1. Swim and tumble of a bacterium.

been presented and discussed in Section V. The analysis presented in this paper has been related with the stability criteria of two other state-of-the-art optimization algorithms in Section VI. Finally, this paper is concluded in Section VII.

II. Classical BFOA

During foraging of the real bacteria, locomotion is achieved by a set of tensile flagella. Flagella help an *E.coli* bacterium to tumble or swim, which are two basic operations performed by a bacterium at the time of foraging [25], [26]. When they rotate the flagella in the clockwise direction, each flagellum pulls on the cell. That results in the moving of flagella independently, and finally, the bacterium tumbles with lesser number of tumbling, whereas in a harmful place, it tumbles frequently to find a nutrient gradient. Moving the flagella in the counterclockwise direction helps the bacterium to swim at a very fast rate. In the aforementioned algorithm, the bacteria undergo chemotaxis, where they like to move toward a nutrient gradient and avoid noxious environment. Generally, the bacteria move for a longer distance in a friendly environment. Fig. 1 shows how clockwise and counterclockwise movements of a bacterium take place in a nutrient solution.

When they get food in sufficient amount, they are increased in length, and in presence of suitable temperature, they break in the middle to form an exact replica of itself. This phenomenon inspired Passino to introduce an event of reproduction in BFOA. Due to the occurrence of sudden environmental changes or attack, the chemotactic progress may be destroyed, and a group of bacteria may move to some other places or some other may be introduced in the swarm of concern. This constitutes the event of elimination–dispersal in the real bacterial population, where all the bacteria in a region are killed or a group is dispersed into a new part of the environment.

Now, suppose that we want to find the minimum of $J(\theta)$, where $\theta \in \Re^p$ (i.e., θ is a p-dimensional vector of real numbers), and we do not have measurements or an analytical description of the gradient $\nabla J(\theta)$. BFOA mimics the four principal mechanisms observed in a real bacterial system: chemotaxis, swarming, reproduction, and elimination–dispersal to solve this nongradient optimization problem. In the Nomenclature, we introduce the formal notations used in BFOA literature and then provide the complete pseudocode of the BFOA (a more detailed description of the steps of BFOA is out of the scope of this brief paper and can be found in [8]).

Let us define a chemotactic step to be a tumble followed by a tumble or a tumble followed by a run. Let j be the index for the chemotactic step. Let k be the index for the reproduction step. Let l be the index of the elimination–dispersal event.

Let $P(j,k,l) = \{\theta^i(j,k,l) | i = 1,2,\ldots,S\}$ represent the position of each member in the population of the S bacteria at the jth chemotactic step, kth reproduction step, and lth elimination–dispersal event. Here, let $J(i,j,k,l)$ denote the cost at the location of the ith bacterium $\theta^i(j,k,l) \in \Re^p$ (sometimes, we drop the indexes and refer to the ith bacterium position as θ^i). Note that we will interchangeably refer to J as being a "cost" (using terminology from optimization theory) and as being a nutrient surface (in reference to the biological connections). For actual bacterial populations, S can be very large (e.g., $S = 109$), but $p = 3$. In our computer simulations, we will use much smaller population sizes and will keep the population size fixed. BFOA, however, allows $p > 3$ so that we can apply the method to higher dimensional optimization problems. As follows, we briefly describe the four prime steps in BFOA. We also provide a pseudocode of the complete algorithm.

1) **Chemotaxis**: This process simulates the movement of an *E.coli* cell through swimming and tumbling via flagella. Suppose $\theta^i(j,k,l)$ represents the ith bacterium at jth chemotactic, kth reproductive, and lth elimination–dispersal step. $C(i)$ is a scalar and indicates the size of the step taken in the random direction specified by the tumble (run length unit). Then, in computational chemotaxis, the movement of the bacterium may be represented by

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

where Δ indicates a unit length vector in the random direction.

2) **Swarming**: An interesting group behavior has been observed for several motile species of bacteria including *E.coli* and *S. typhimurium*, where stable spatiotemporal patterns (swarms) are formed in semisolid nutrient medium. A group of *E.coli* cells arrange themselves in a traveling ring by moving up the nutrient gradient when placed amid a semisolid matrix with a single nutrient chemo-effecter. The cells when stimulated by a high level of *succinate* release an attractant *aspertate*, which helps them to aggregate into groups and, thus, move as concentric patterns of swarms with high bacterial density. The cell-to-cell signaling in *E. coli* swarm may be represented by the following function:

$$\begin{aligned} J_{cc}&\left(\theta, P(j,k,l)\right) \\ &= \sum_{i=1}^{S} J_{cc}\left(\theta, \theta^i(j,k,l)\right) \\ &= \sum_{i=1}^{S}\left[-d_{\text{attractant}} \exp\left(-w_{\text{attractant}} \sum_{m=1}^{p}\left(\theta_m - \theta_m^i\right)^2\right)\right] \\ &\quad + \sum_{i=1}^{S}\left[h_{\text{repellant}} \exp\left(-w_{\text{repellant}} \sum_{m=1}^{p}\left(\theta_m - \theta_m^i\right)^2\right)\right] \end{aligned} \tag{2}$$

where $J_{cc}(\theta, P(j,k,l))$ is the objective-function value to be added to the actual objective function (to be minimized) to present a time-varying objective function. The coefficients $d_{\text{attractant}}$, $w_{\text{attractant}}$, $h_{\text{repellant}}$, and $w_{\text{repellant}}$ control the strength of the cell-to-cell signaling. More specifically, $d_{\text{attractant}}$ is the depth of the attractant released by the cell, $w_{\text{attractant}}$ is a measure of the width of the attractant signal (a quantification of the diffusion rate of the chemical), $h_{\text{repellant}} = d_{\text{attractant}}$ is the height of the repellant effect (a bacterium cell also repels a nearby cell in the sense that it consumes nearby nutrients, and it is not physically possible to have two cells at the same location), and $w_{\text{repellant}}$ is a measure of the width of the repellant (for a detailed discussion on the function J_{cc}, please see [8]).

3) **Reproduction**: The least healthy bacteria eventually die while each of the healthier bacteria (those yielding lower value of the objective function) asexually split into two bacteria, which are then placed in the same location. This keeps the swarm size constant.
4) **Elimination and Dispersal**: To simulate this phenomenon in BFOA, some bacteria are liquidated at random with a very small probability while the new replacements are randomly initialized over the search space.

Pseudo-Code of BFOA

```
Parameters
[Step 1] Initialize parameters $p$, $S$, $N_c$, $N_s$, $N_{re}$, $N_{ed}$, $P_{ed}$,
         $C(i)(i = 1, 2, \ldots, S)$, $\theta^i$.
Algorithm:
[Step 2] Elimination–dispersal loop: $l = l + 1$
[Step 3] Reproduction loop: $k = k + 1$
[Step 4] Chemotaxis loop: $j = j + 1$
    [a] For $i = 1, 2, \ldots,$ S take a chemotactic step for bacterium
        $i$ as follows.
    [b] Compute fitness function, $J(i, j, k, l)$.
        Let, $J(i, j, k, l) = J(i, j, k, l) + J_{cc}(\theta^i(j, k, l), P(j, k, l))$
        (i.e., add on the cell-to-cell attractant–repellant profile
        to simulate the swarming behavior) where $J_{cc}$ is defined
        in (2).
    [c] Let $J_{\text{last}} = J(i, j, k, l)$ to save this value, since we may
        find a better cost via a run.
    [d] Tumble: generate a random vector $\Delta(i) \in \Re^p$ with each
        element $\Delta_m(i)$, $m = 1, 2, \ldots, p$, a random number on
        [−1, 1].
    [e] Move: Let
        $$\theta^i(j+1, k, l) = \theta^i(j, k, l) + C(i)\frac{\Delta(i)}{\sqrt{\Delta^{\mathrm{T}}(i)\Delta(i)}}.$$
        This results in a step of size $C(i)$ in the direction of the
        tumble for bacterium $i$.
    [f] Compute $J(i, j+1, k, l)$ and let $J(i, j+1, k, l) = J(i,$
        $j, k, l) + J_{cc}(\theta^i(j+1, k, l), P(j+1, k, l))$.
    [g] Swim
        i) Let $m = 0$ (counter for swim length).
        ii) While $m < N_s$ (if have not climbed down too long).
            • Let $m = m + 1$.
            • If $J(i, j+1, k, l) < J_{\text{last}}$ (if doing better), let $J_{\text{last}} =$
              $J(i, j+1, k, l)$ and let
              $$\theta^i(j+1, k, l) = \theta^i(j, k, l) + C(i)\frac{\Delta(i)}{\sqrt{\Delta^{\mathrm{T}}(i)\Delta(i)}}.$$
              And use this $\theta^i(j+1, j, k)$ to compute the new $J(i,$
              $j+1, k, l)$ as we did in [f]
            • Else, let $m = N_s$. This is the end of the while
              statement.
    [h] Go to next bacterium $(i + 1)$ if $i \neq S$ (i.e., go to [b] to
        process the next bacterium).
[Step 5] If $j < N_c$, go to step 4. In this case, continue chemotaxis
         since the life of the bacteria is not over.
[Step 6] Reproduction:
    [a] For the given $k$ and $l$, and for each $i = 1, 2, \ldots, S$, let
        $$J^i_{\text{health}} = \sum_{j=1}^{N_c+1} J(i, j, k, l)$$
        be the health of the bacterium $i$ (a measure of how many
        nutrients it got over its lifetime and how successful it
        was at avoiding noxious substances). Sort bacteria and
        chemotactic parameters $C(i)$ in order of ascending cost
        $J_{\text{health}}$ (higher cost means lower health).
    [b] The $S_r$ bacteria with the highest $J_{\text{health}}$ values die, and
        the remaining $S_r$ bacteria with the best values split (this
        process is performed by the copies that are made are
        placed at the same location as their parent).
[Step 7] If $k < N_{re}$, go to step 3. In this case, we have not
         reached the number of specified reproduction steps, so we
         start the next generation of the chemotactic loop.
[Step 8] Elimination–dispersal: For $i = 1, 2, \ldots, S$ with probability
         $P_{ed}$, eliminate and disperse each bacterium (this keeps the
         number of bacteria in the population constant). To do this,
         if a bacterium is eliminated, simply disperse another one to
         a random location on the optimization domain. If $l < N_{ed}$,
         then go to step 2; otherwise, end.
```

III. Modeling the Chemotactic Dynamics

Let us consider a single bacterium cell that undergoes chemotactic steps according to (1) over a 1-D objective-function space. Since each dimension in simulated chemotaxis is updated independently of others and the only link between the dimensions of the problem space are introduced via the objective functions, an analysis can be carried out on the 1-D case, without loss of generality. The bacterium lives in continuous time, and at the tth instant, its position is given by $\theta(t)$. Next, we list a few simplifying assumptions that have been considered for the sake of gaining mathematical insight.

1) The objective function $J(\theta)$ is continuous and differentiable at all points in the search space. The function is unimodal in the region of interest, and its one and only optimum (minimum) is located at $\theta = \theta_0$. In addition, $J(\theta) \neq 0$ for $\theta \neq \theta_0$.
2) The chemotactic step-size C is smaller than one (Passino himself took $C = 0.1$ in [8]).
3) The analysis applies to the regions of the fitness landscape where gradients of the function are small, i.e., near to the optima.

A. Analytical Treatment

Now, according to BFOA, the bacterium changes its position only if the modified objective-function value is less than the previous one, i.e., $J(\theta) > J(\theta + \Delta\theta)$, i.e., $J(\theta) - J(\theta + \Delta\theta)$ is positive. This ensures that bacterium always moves in the direction of decreasing objective-function value. A particular iteration starts by generating a random vector of unit length, termed as the *direction of tumble* and denoted by Δ. In case of a 1-D optimization problem, it can assume only two values, 1 or -1 with equal probabilities. In addition, since Δ is of unit magnitude, its value remains unchanged after dividing it by its magnitude or norm (as done in the algorithm). The bacterium moves by an amount of $C\Delta$ if objective-function value is reduced for new location. Otherwise, its position will not change at all. Assuming uniform rate of position change, if the bacterium moves $C\Delta$ in unit time, its position is changed by $(C\Delta)(\Delta t)$ in Δt seconds. It decides to move in the direction in which concentration of nutrient increases or, in other words, objective function decreases, i.e., $J(\theta) - J(\theta + \Delta\theta) > 0$. Otherwise, it remains immobile. We have assumed that Δt is an infinitesimally small positive quantity, thus the sign of the quantity $J(\theta) - J(\theta + \Delta\theta)$ remains unchanged if Δt divides it. Therefore, bacterium will change its position if, and only if, $(J(\theta) - J(\theta + \Delta\theta))/\Delta t$ is positive. This crucial decision-making (i.e., whether to take a step or not) activity of the bacterium can be modeled by a unit step function (also known as Heaviside step function [27]) defined as

$$\begin{aligned} u(x) &= 1, \qquad \text{if } x > 0 \\ &= 0, \qquad \text{otherwise.} \end{aligned} \tag{3}$$

Thus, $\Delta\theta = u((J(\theta) - J(\theta + \Delta\theta))/\Delta t) \cdot (C \cdot \Delta)(\Delta t)$, where value of $\Delta\theta$ is zero or $(C\Delta)(\Delta t)$ according to the value of the unit step function. Dividing both sides of the earlier relation by Δt, we get

$$\begin{aligned} \frac{\Delta\theta}{\Delta t} &= u\left[\frac{J(\theta) - J(\theta + \Delta\theta)}{\Delta t}\right] C \cdot \Delta \\ \Rightarrow \frac{\Delta\theta}{\Delta t} &= u\left[-\frac{\{J(\theta + \Delta\theta) - J(\theta)\}}{\Delta t}\right] C \cdot \Delta. \end{aligned} \tag{4}$$

Defining the velocity of the bacterium as $V_b = Lim_{\Delta t \to 0}(\Delta\theta/\Delta t)$ (naturally, here, we assume the time to be unidirectional, i.e., $\Delta t > 0$), we obtain

$$\begin{aligned} V_b &= \underset{\Delta t \to 0}{Lim} \frac{\Delta\theta}{\Delta t} = \underset{\Delta t \to 0}{Lim} \left[u\left\{-\frac{J(\theta+\Delta\theta) - J(\theta)}{\Delta t}\right\} \cdot C \cdot \Delta\right] \\ \Rightarrow V_b &= \underset{\Delta t \to 0}{Lim} \left[u\left\{-\frac{J(\theta+\Delta\theta) - J(\theta)}{\Delta\theta} \frac{\Delta\theta}{\Delta t}\right\} \cdot C \cdot \Delta\right] \end{aligned}$$

as $\Delta t \to 0$ makes $\Delta\theta \to 0$, we may write, $V_b = [u\{-(Lim_{\Delta\theta \to 0}((J(\theta + \Delta\theta) - J(\theta))/\Delta\theta))(Lim_{\Delta t \to 0}(\Delta\theta/\Delta t))\} \cdot C \cdot \Delta]$. Again, $J(\theta)$ is assumed to be continuous and differentiable, and thus, $Lim_{\Delta\theta \to 0}((J(\theta + \Delta\theta) - J(\theta))/\Delta\theta)$ is the value of the gradient at the point $\theta = \theta$. Therefore, we have

$$V_b = u(-GV_b)C\Delta \tag{5}$$

where $G = (dJ(\theta)/d\theta)$ = gradient of the objective function at $\theta = \theta$.

In (5), argument of the unit step function is $-GV_b$. Value of the unit step function is one if G and V_b are of different sign, and in this case, the velocity is $C\Delta$. Otherwise, it is zero, making bacterium motionless. Therefore, (5) suggests that bacterium will move the direction of negative gradient.

Since the unit step function $u(x)$ has a jump discontinuity at $x = 0$, to simplify the analysis further, we replace $u(x)$ with the continuous logistic function $\phi(x)$, where $\phi(x) = (1/(1 + e^{-kx}))$.

We note that

$$u(x) = \underset{k \to \infty}{Lt} \phi(x) = \underset{k \to \infty}{Lt} \frac{1}{1 + e^{-kx}}. \tag{6}$$

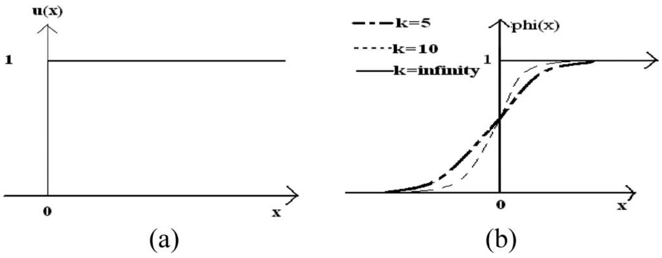


Fig. 2. Unit step and the logistic functions. (a) Unit step function. (b) Approximation with logistic function.

Fig. 2 shows how the logistic function approaches the unit step function as k tends to infinity. For analysis purpose, k cannot be infinity. We restrict ourselves to moderately large values of k (for example, $k = 10$) for which $\phi(x)$ fairly approximates $u(x)$. Thus, for moderately high values of k, $\phi(x)$ fairly approximates $u(x)$. Hence, from (5)

$$V_b = \frac{C\Delta}{1 + e^{kGV_b}}. \tag{7}$$

According to assumptions 2) and 3), if C and G are very small and $k \sim 10$, then we may also have $|kGV_b| \ll 1$. In that case, we neglect higher order terms in the expansion of e^{kgv_b} and have $e^{kgv_b} \approx 1 + kGV_b$. Substituting it in (7), we obtain

$$\begin{aligned} V_b &= \frac{C \cdot \Delta}{2 + kGV_b} \\ \Rightarrow V_b &= \frac{C \cdot \Delta}{2} \frac{1}{1 + \frac{kGV_b}{2}} \\ \Rightarrow V_b &= \frac{C \cdot \Delta}{2}\left(1 - \frac{kGV_b}{2}\right) \end{aligned}$$

$$\left[\because \left|\frac{kGV_b}{2}\right| \ll 1, \qquad \text{neglecting higher terms} \quad \left(1 + \frac{kGV_b}{2}\right)^{-1} \approx \left(1 - \frac{kGV_b}{2}\right)\right].$$

After some manipulation, we have

$$\begin{aligned} V_b &= \frac{2C \cdot \Delta}{4 + kGC\Delta} \qquad (8) \\ \Rightarrow V_b &= \frac{C\Delta}{2} \frac{1}{1 + \frac{kCG\Delta}{4}} \\ \Rightarrow V_b &= \frac{C\Delta}{2}\left(1 - \frac{kGC\Delta}{4}\right) \end{aligned}$$

$$\left[\because \left|\frac{kGC\Delta}{4}\right| = \left|\frac{kGC}{4}\right| \ll 1, \text{as } |\Delta| = 1 \quad \text{and neglecting the higher order terms.}\right]$$

$$\begin{aligned} \Rightarrow V_b &= \frac{C\Delta}{2} - \frac{kGC^2\Delta^2}{8} \\ \Rightarrow V_b &= \frac{d\theta}{dt} = -\frac{kC^2}{8}G + \frac{C\Delta}{2} \qquad [\because \Delta^2 = 1]. \end{aligned} \tag{9}$$

Equation (9) represents the fundamental dynamics of the computational chemotaxis step in BFOA. Equation (9) is applicable to a single-bacterium system and is independent of the objective function (as long as the function obeys the assumptions listed and it does not take into account the cell-to-cell signaling effect). In what follows, our stability-analysis procedures will be mostly centered on this equation. From (9), we get

$$V_b = -\frac{kC^2}{8}G + \frac{C \cdot \Delta}{2} \Rightarrow \frac{d\theta}{dt} = -\alpha' G + \beta' \tag{10}$$

where α' is $-kC^2/8$ and β' is $C\Delta/2$. The classical gradient-descent-search algorithm is given by the following dynamics in single dimension [10]:

$$\frac{d\theta}{dt} = -\alpha \cdot G + \beta \tag{11}$$

where α is the learning rate and β is the momentum [28]. Similarity between (10) and (11) suggests that chemotaxis may be considered as s modified gradient descent search, where α', a function of chemotactic step-size, can be identified as the learning-rate parameter. Note that the random-search or momentum term $(C \cdot \Delta)/2$ in the right-hand side of (9) provides an additional feature to the classical gradient descent search. When gradient becomes very small, the random term dominates over gradient decent term, and the bacterium changes its position. However, random-search term may lead to change in position in the direction of increasing objective-function value. If it happens, then, again, the magnitude of gradient increases and dominates the random-search term.

B. Experimental Verification of the Chemotactic Dynamics as Given by (9)

In order to verify how reliably does (9) represent the motion of a virtual bacterium, we compare results obtained from (9) with that obtained using the actual BFOA iterations. First, we express (9) in iterative (discrete time) form given by

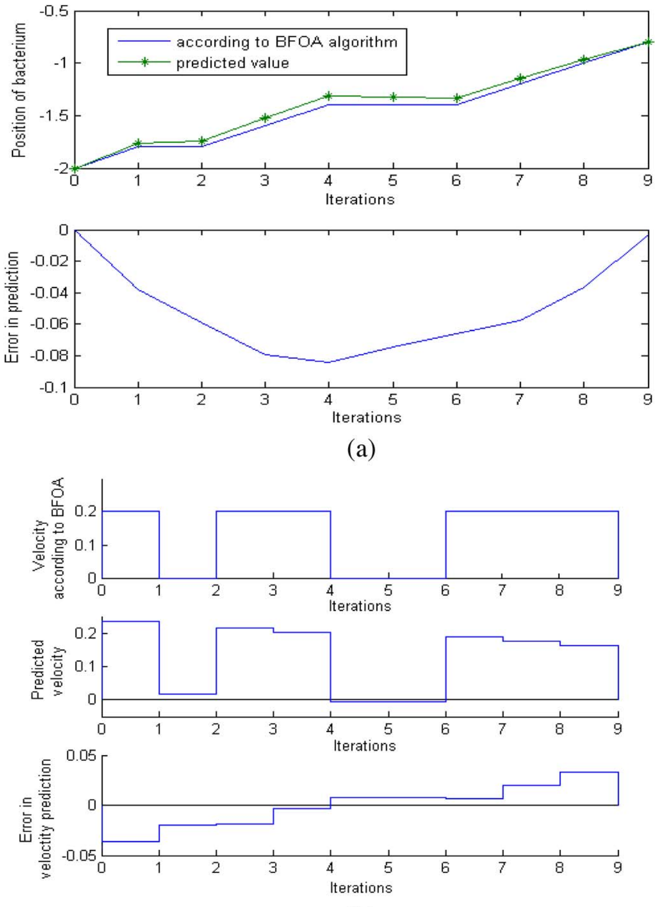
$$\begin{aligned} V_b(p) &= \theta(p) - \theta(p-1) = -\frac{kC^2}{8}G(p-1) + \frac{C\Delta(p)}{2} \\ \Rightarrow \theta(p) &= \theta(p-1) - \frac{kC^2}{8}G(p-1) + \frac{C\Delta(p)}{2} \end{aligned} \tag{12}$$

where p is the iteration index. The tumble vector $\Delta(p)$ is also a function of iteration count (i.e., chemotactic step number) as it is generated anew for successive iterations. We have taken $J(\theta) = \theta^2$ as objective function for this simulation study. The bacterium was initialized at -2, i.e., $\theta(0) = -2$, and C is taken as 0.2. Here, the gradient of $J(\theta)$ is 2θ. Therefore, $G(p-1)$ may be replaced by $2\theta(p-1)$. Finally, for this specific case, we get

$$\theta(p) = \left(1 - \frac{kC^2}{4}\right)\theta(p-1) + \frac{C\Delta(p)}{2}. \tag{13}$$

We compute values of $\theta(n)$ for successive iterations according to earlier iterative relation. In addition, values of positions are noted following guidelines of BFOA. The current position is changed by $C\Delta$ if objective-function value decreases for new position. Results are shown in Fig. 3. Fig. 3(a) shows position in successive iteration according to BFOA and as obtained from (13). Here, also, we have assumed position of bacterium changes linearly between two subsequent iterations. Mismatch between the actual and predicted values is shown in the same figure. Fig. 3(b) shows the actual and predicted values of velocity. Velocity is assumed to be constant between two successive iterations. According to BFOA, magnitude of velocity is either C (0.2 in this case) or zero. Difference between actual and predicted velocity is shown as error. Time lapsed between two subsequent iterations is spent for computation and is termed as unit time. This may be perceived as the time required by a bacterium to measure nutrient content of a new point on fitness landscape. It is the time taken by the processor to perform numerical computations.

Fig. 3(a) and (b) shows that (9) can adequately model the dynamics of a bacterium, which is taking chemotactic steps in BFOA.



(a)

(b)

Fig. 3. Comparison between actual and predicted motional states of the bacterium. (a) Plots showing actual and predicted positions of bacterium and error in estimation over successive iterations. (b) Similar plots for velocity of the bacterium.

IV. Stability Analysis

In this section, we analyze the stability of the chemotactic dynamics represented by (9) using the concept of Lyapunov stability theorems [23]. We begin this treatment by explaining some basic concepts and their interpretations from the standard literature on nonlinear control theory [24], [29]. We denote a vector variable by $\vec{x}$ instead of θ and a scalar function of the vector variable as $f(\vec{x})$ instead of $J(\theta)$ to cope with the standard notations of the literature on control theory.

Definition 4.1: A point $\vec{x} = \vec{x}_e$ is called an *equilibrium state*, if the dynamics of the system is given by

$$\frac{d\vec{x}}{dt} = f(\vec{x}(t))$$

becomes zero at $\vec{x} = \vec{x}_e$ for any t, i.e., $f(\vec{x}_e(t)) = 0$. The equilibrium state is also called equilibrium (stable) point in D-dimensional hyperspace, when the state $\vec{x}_e$ has D components.

Definition 4.2: A scalar function $V(\vec{x})$ is said to be *positive definite* with respect to the point $\vec{x}_e$ in the region $\|\vec{x} - \vec{x}_e\| \le K$, if $V(\vec{x}) > 0$ at all points of the region except at $\vec{x}_e$ where it is zero.

Definition 4.3: A scalar function $V(\vec{x})$ is said to be *negative definite* if $-V(\vec{x})$ is positive definite.

Definition 4.4: A dynamics $(d\vec{x}/dt) = f(\vec{x}(t))$ is *asymptotically stable* at the equilibrium point $\vec{x}_e$, if we have the following conditions.

1) It is stable in the sense of Lyapunov, i.e., for any neighborhood $S(\varepsilon)$ surrounding $\vec{x}_e$ ($S(\varepsilon)$ contains points $\vec{x}$ for which

$\|\vec{x}-\vec{x}_e\| \leq \varepsilon$), where there is a region $S(\delta)$ ($S(\delta)$ contains points $\vec{x}$ for which $\|\vec{x}-\vec{x}_e\| \leq \delta$), $\delta < \varepsilon$, such that trajectories of the dynamics starting within $S(\delta)$ do not leave $S(\varepsilon)$ as time $t \to \infty$.

2) The trajectory starting within $S(\delta)$ converges to the origin as time t approaches infinity.

The sufficient condition for stability of a dynamics can be obtained from the Lyapunov's theorem, presented as follows.

Lyapunov's Stability Theorem [23], [26]: Given a scalar function $V(\vec{x})$ and some real number $\varepsilon > 0$, such that, for all $\vec{x}$ in the region $\|\vec{x}-\vec{x}_e\| \leq \varepsilon$, the following conditions hold.

1) $V(\vec{x}_e) = 0$.
2) $V(\vec{x}) > 0$ for $\vec{x} \neq \vec{x}_e$, i.e., $V(\vec{x})$ is positive definite.
3) $V(\vec{x})$ has continuous first partial derivatives with respect to all components of $\vec{x}$.

Then, the equilibrium state $\vec{x}_e$ of the system $(d\vec{x}/dt) = f(\vec{x}(t))$ is as follows.

1) *Asymptotically stable* if $(dV/dt) < 0$, i.e., dV/dt is negative definite.
2) *Asymptotically stable* in the large if $(dV/dt) < 0$ for $\vec{x} \neq \vec{x}_e$, and in addition, $V(\vec{x}) \to \infty$ as $\|\vec{x}-\vec{x}_e\| \to \infty$.

Remark: Lyapunov stability analysis is based on the idea that if the total energy in the system continually decreases, then the system will asymptotically reach the zero energy state associated with an equilibrium point of the system. A system is said to be asymptotically stable if all the states approach the equilibrium state with time.

Theorem 4.1 (Main Result): Let the bacterial dynamics be represented by (9), and $\theta = \theta_0$ is the single optimum (minimum) in the region of search. Then, this optimum is asymptotically stable if

$$\begin{aligned} C &> \tfrac{4}{k}\left|\tfrac{\theta-\theta_0}{J(\theta)}\right|, && \text{if } \theta \neq \theta_0. \\ &= 0, && \text{if } \theta = \theta_0. \end{aligned} \tag{14}$$

Proof: In order to determine the equilibrium point for the system, we set (by Definition 4.1)

$$\begin{aligned} &\frac{d\theta}{dt} = 0 \\ &\Rightarrow -\frac{kC^2}{8}G + \frac{C\Delta}{2} = 0. \end{aligned} \tag{15}$$

Since the bacterium is expected to converge at the optimum of the fitness landscape, we have the equilibrium point $\theta_e = \theta_0$ and also the function gradient $G = 0$ at this point. Putting $G = 0$ in (15), we obtain $C = 0$. Thus, the step-height C should become zero at $\theta = \theta_0$ for the equilibrium point to be located at the desired optimum, i.e.,

$$C = 0, \qquad \text{if } \theta = \theta_0. \tag{16}$$

This criterion is intuitively appealing also from the perspective of an optimization algorithm. Once reaching the optimum of the unimodal fitness landscape, the bacterium is expected to stay there, and hence, it should not take any more chemotactic steps or, in other words, its chemotactic step-size C should become zero.

Now, to test the stability, consider a scalar function

$$V(\theta) = \frac{kC^2}{8}J(\theta) - \frac{C\Delta}{2}(\theta - \theta_0) \tag{17}$$

where $J(\theta)$ is the objective function. In order to qualify as a Lyapunov energy function, $V(\theta)$ must be a positive-definite function with respect to the equilibrium point θ_0. Thus, by Definition 4.2, $V(\theta)$ must satisfy the relation $V(\theta_0) = 0$, and $V(\theta) > 0$ if $\theta \neq \theta_0$.

As $C = 0$ at $\theta = \theta_0$, we have

$$V(\theta_0) = \frac{kC^2}{8}J(\theta_0) - \frac{C\Delta}{2}(\theta_0 - \theta_0) = \frac{kC^2}{8}J(\theta_0) = 0.$$

Now, for the second condition to be satisfied, we should have

$$\begin{aligned} &\tfrac{kC^2}{8}J(\theta) - \tfrac{C\Delta}{2}(\theta - \theta_0) > 0 && \forall\, \theta \neq \theta_0 \\ &\quad\Rightarrow \tfrac{kC}{4}J(\theta) > (\theta - \theta_0)\Delta && \forall\, \theta \neq \theta_0 \\ &&& [\text{as } C > 0 \text{ for all positions other than optima}]. \end{aligned} \tag{18}$$

Now, by assumption 1), $J(\theta) \neq 0$ for all $\theta \neq \theta_0$, and also, noting that $k > 0$, dividing both sides of (16) by $kJ(\theta)/4$, we get

$$C > \frac{4(\theta-\theta_0)\Delta}{kJ(\theta)} \qquad \forall\, \theta \neq \theta_0. \tag{19}$$

If the right-hand side of (17) is negative, it will lead to a trivial condition as step-height C is always positive.

Now

$$\begin{aligned} &\left|\frac{4(\theta-\theta_0)\Delta}{kJ(\theta)}\right| \geq \frac{4(\theta-\theta_0)\Delta}{kJ(\theta)} \\ &\Rightarrow \frac{4}{k}\left|\frac{\theta-\theta_0}{J(\theta)}\right| \geq \frac{4(\theta-\theta_0)\Delta}{kJ(\theta)} \qquad [\text{as } |\Delta| = 1]\,. \end{aligned}$$

Therefore, if C satisfies the relation $C > (4/k)|(\theta-\theta_0)/J(\theta)|$ for all $\theta \neq \theta_0$, then $C > (4/k)|(\theta-\theta_0)/J(\theta)| \geq (4(\theta-\theta_0)\Delta)/kJ(\theta)$ for all $\theta \neq 0$, i.e., condition (19) is automatically satisfied.

Thus, provided that C satisfies conditions (16) and (19), $V(\theta)$ is a Lyapunov energy function and

$$\frac{dV}{dt} = \frac{dV}{d\theta} \cdot \frac{d\theta}{dt}. \tag{20}$$

Now, differentiating both sides of (15) with θ, we have

$$\frac{dV}{d\theta} = \frac{kC^2}{8} \cdot \frac{dJ(\theta)}{d\theta} - \frac{C \cdot \Delta}{2} = -\left(-\frac{kC^2}{8} \cdot G + \frac{C \cdot \Delta}{2}\right) \tag{21}$$

substituting values of $dV/d\theta$ and $d\theta/dt$ from (19) and (9), respectively, into (18), we get

$$\frac{dV}{dt} = -\left(-\frac{kC^2}{8}G + \frac{C\Delta}{2}\right)^2 < 0 \qquad \forall\, \theta \neq \theta_0. \tag{22}$$

In addition, $dV/dt = 0$ if $\theta = \theta_0$ [as $C = 0$ and $G = 0$ at $\theta = \theta_0$]. Thus, by Definition 4.3, dV/dt is negative definite. Therefore, we can infer that the bacterial dynamics of (9) exhibits an asymptotically stable behavior with respect to the optimum $\theta = \theta_0$ if the step size satisfies conditions (14) and (17) simultaneously. This completes the proof. ■

V. Computer-Simulation Results

In Section IV, we have derived the criterion for asymptotic stability of a bacterium with respect to an optimum of the search space. In this section, we investigate what happens to the dynamics of the bacterium if this criterion is met and whether the bacterium shows unstable or oscillatory behavior otherwise, with the help of computer simulations. Consider the case of a single bacterium taking chemotactic steps over 1-D fitness landscape of the function $J(\theta) = \theta^2$, where the single optimum located at $\theta = \theta_0 = 0$. Let the bacterium start from $\theta = -0.5$ and start taking chemotactic steps of height $C = 0.2$ following the directives of the actual BFOA. Now, as step size remains constant, condition given in (12) is violated at some point of time.

TABLE I
VALUES OF C AND $C_{\text{threshold}}$ OVER SUCCESSIVE ITERATIONS

Iteration	Position	Actual step size C	Threshold step size $C_{threshold}$	Remarks
1	-0.5	0.2	0.061538	(14) is satisfied
2	-0.3	0.2	0.102564	(14) is satisfied
3	-0.3	0.2	0.102564	(14) is satisfied
4	-0.1	0.2	0.307692	(14) is violated and limit-cyclic behavior
5	-0.1	0.2	0.307692	(14) is violated and limit-cyclic behavior
6	0.1	0.2	0.307692	(14) is violated and limit-cyclic behavior
7	0.1	0.2	0.307692	(14) is violated and limit-cyclic behavior

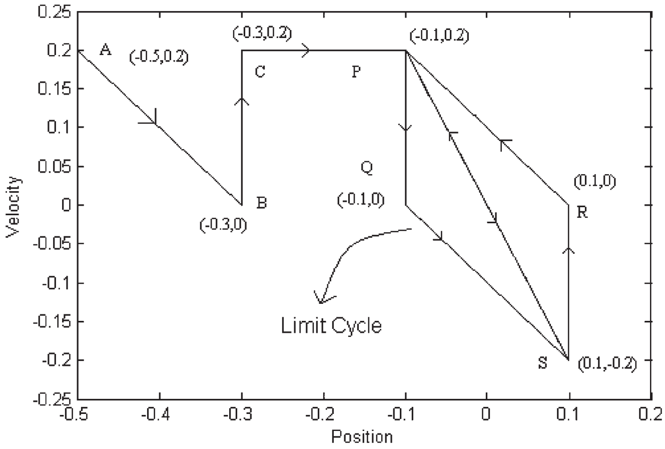


Fig. 4. Phase trajectory constructed according to algorithm not maintaining (14).

TABLE II
VARIOUS STATES AND SET OF DIRECTION OF TUMBLE USED FOR SIMULATION

State of phase-trajectory or points on Figure 3	Position	Velocity	Direction of tumble Δ
A	-0.5	0.2	+1.0
B	-0.3	0.0	-0.1
C	-0.3	0.2	+0.1
P	-0.1	0.2	+0.1
Q	-0.1	0.0	-0.1
S	0.1	-0.2	+0.1
R	0.1	0.0	+0.1

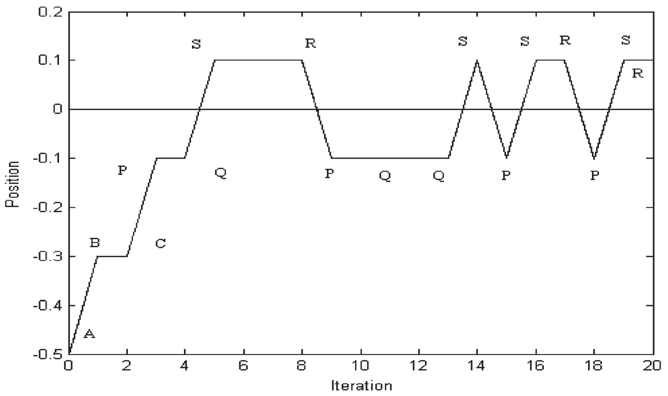


Fig. 5. Variation of position with time for the bacterium of Fig. 4.

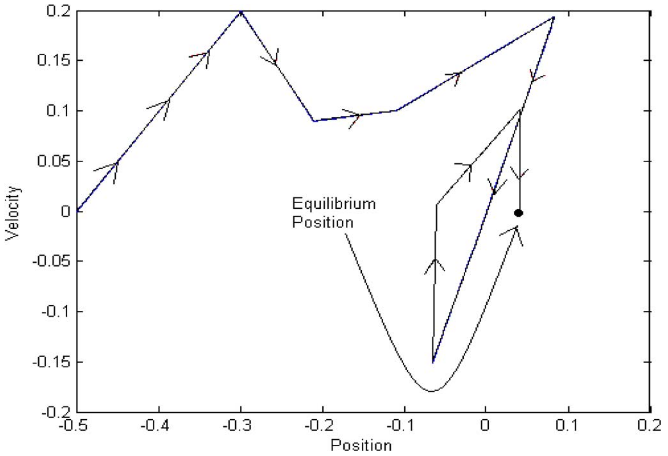
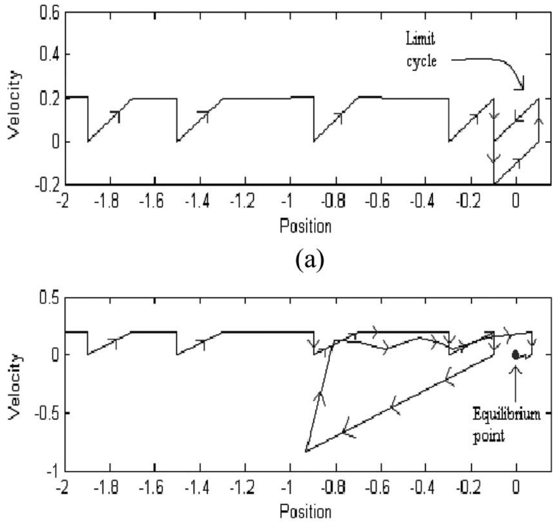


Fig. 6. Phase trajectory constructed for bacterium satisfying condition (14).



(a)

(b)

Fig. 7. Phase trajectories of a single bacterium over the objective function $J(\theta) = 1 - e^{-\theta^2}$. (a) Limit cyclic behavior of the bacterium, not satisfying condition (14). (b) Stable behavior of the bacterium, satisfying condition (14).

Let $C_{\text{threshold}} = (4/k)|(\theta - \theta_0)/J(\theta)|$. Then, according to (12), the bacterium should exhibit stable dynamic behavior near the optima as long as $C > C_{\text{threshold}}$. Table I shows, with changing positions of bacterium, varying values of $C_{\text{threshold}}$. We have assumed that $k = 130$. Fig. 4 shows the phase trajectory (plot of velocity versus position) of a bacterium.

A brief explanation to the nature of the phase trajectory shown in Fig. 4 may be given in the following way. The bacterium starts from the initial position $\theta = -0.5$, and this initial position is marked as point A in the phase trajectory. Now, in each iteration, a direction of tumble Δ (which, in this paper, can be either 1 or -1) is generated randomly. Note that, due to the greedy nature of computational chemotaxis, the bacterium can really move only if Δ leads it to the direction of nondecreasing fitness (i.e., nonincreasing objective-function value). The values of Δ and the positions and velocities of the bacterium at successive time-steps (as used in Fig. 4) have been reported in Table II.

In the very first iteration, the bacterium takes a step of size 0.2 and reaches $\theta = -0.3$. Then, in the second iteration, it does not move (as doing so would increase the function value), and its velocity drops to zero. This situation is represented as point B in phase trajectory. The line AB makes an angle of $-45°$ with the position axis. Next, it takes a chemotactic step. This state can be seen in C. After taking the step, it reaches P. Now, the bacterium can change position by

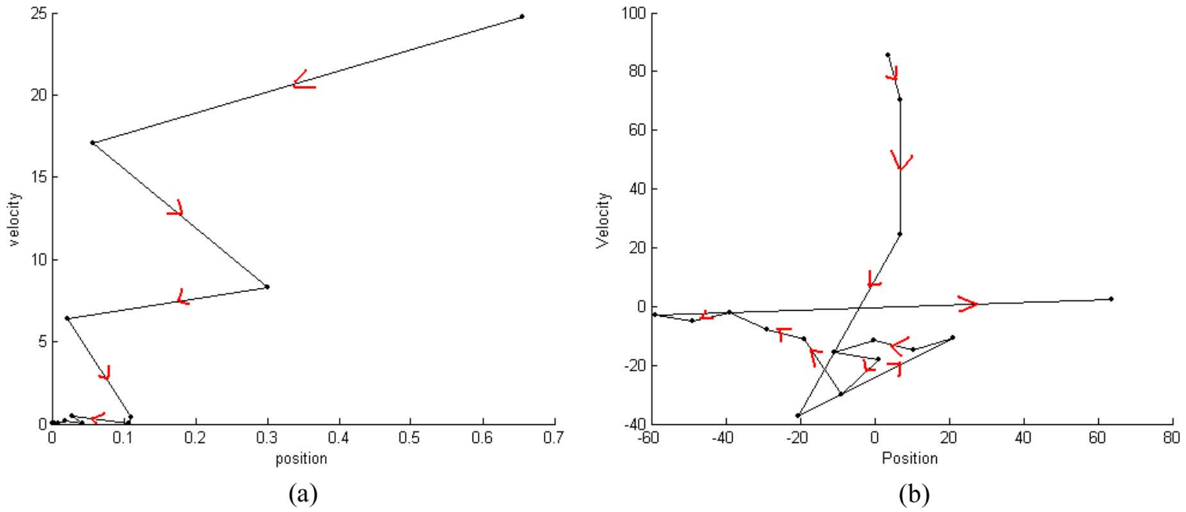


Fig. 8. Particle trajectories in phase plane for PSO over the objective function $J(x) = x^2$. (a) Stable behavior for $c_1 + c_2 = 2$ and $w = 0.2$ [obeying condition (25)]. (b) Unstable behavior for $c_1 + c_2 = 3.5$ and $w = 0.9$.

an amount C or $-C$, which are 0.2 and -0.2 in this case. These cases have been shown in P and S. Otherwise, it remains immobile and velocity becomes zero. These cases can be observed in Q and R. The bacterium makes transition between these points in cyclic order. Here, in states P, Q, R, and S, the objective-function value remains constant, and the distance of the bacterium from the optimum is also constant. Still, it continues to change its position. From Table I, we can predict that, after reaching $\theta = -0.1$, the bacterium should show asymptotically unstable behavior. Experimentally, we observe that the bacterium enters stable limit cycles after reaching that position (please see Fig. 3). Fig. 5 shows how the position of the bacterium θ varies with iteration time-step.

Finally, we observe what happens if the condition mentioned in (14) is satisfied, i.e., $C < (4/k)|(\theta - \theta_0)/J(\theta)|$ for all θ in the feasible search range. In this case, we take $C = C_{\text{threshold}} + \xi$ for each iteration, where $\xi = 0.01$ is a small positive bias. Initial position is again $\theta = -0.5$. Phase trajectory, constructed for this case, has been provided as shown in Fig. 6, and we observe that it converges and shows no oscillatory behavior.

In Fig. 7, we show phase-trajectories for another function $J(\theta) = 1 - e^{-\theta^2}$. In addition, we observe that if condition (12) is not met, the bacterium gets trapped into limit cycle [Fig. 7(a)], and if the condition is satisfied, then it asymptotically converges to the optimum, as shown in Fig. 7(b).

Please note that the semigreedy nature of the chemotactic dynamics is responsible for the oscillatory behavior near the optimum, when step-size does not satisfy the Lyapunov's stability criterion.

VI. Relation With the Stability Criteria of Other Popular Metaheuristics

Determining the stability criteria for population-based metaheuristics is a challenging problem at its own right. Previously, the stability of another powerful swarm-intelligence algorithm called PSO has been extensively studied for both deterministic and stochastic dynamics in works like [30]–[32]. Usually, just like we did in Section IV for BFOA, for PSO also, the stability criteria are formulated as suitable bounds over the control parameters. In PSO, each *particle* is defined as a potential solution to a problem in d-dimensional space with a memory of its previous best position and the best position among all particles, in addition to a velocity component. At each iteration, the particles are combined to adjust the velocity along each dimension, which in turn is used to compute the new particle position. The particle dimension in single dimension may be given by

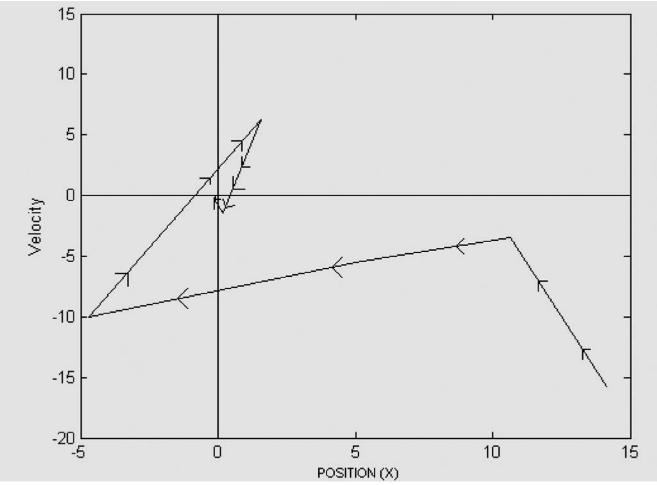


Fig. 9. Phase trajectory of the median order vector (in a population of size $NP = 11$) for objective function $J(x) = x^2$.

$$v_{t+1} = \omega v_t + \alpha_t^l \left(p_t^l - x_t\right) + \alpha_t^g \left(p_t^g - x_t\right) \tag{23}$$

$$x_{t+1} = x_t + v_{t+1} \tag{24}$$

where v_t is the velocity of the particle at the tth iteration, x_t is the particle position at the tth iteration, p_t^l is the personal (local) best position of the particle so far achieved until iteration t, and p_t^g is the global best position among all particles at iteration t. $\alpha_t^l \sim (0, c_1)$ and $\alpha_t^g \sim (0, c_2)$ are random parameters with uniform distributions where c_1 and c_2 are constants known as acceleration coefficients. In [32], Kadirkamanathan *et al.* analyzed the stability of particle dynamics without the deterministic restrictions using the Lyapunov stability theorems. The stability criterion was formulated as

$$c_1 + c_2 < \left(\frac{2\left(1 - 2|w| + w^2\right)}{1 + w}\right). \tag{25}$$

Fig. 8(a) and (b) shows the stable and unstable behaviors of a particle in phase plane (velocity versus position) for two different sets of parameters c_1, c_2, and w over the same objective function $J(x) = x^2$ which we also used to test the stability criteria of BFOA.

Another state-of-the-art evolutionary algorithm, which has gained wide popularity these days, is the differential evolution (DE) [33], [34]. Since its advent in 1995, DE has found several interesting applications in engineering optimization problems (e.g., see [35]–[38]). The population dynamics of DE has been extensively studied, and the stability aspects were investigated by Dasgupta *et al.* in [39] and [40]. The results indicate that the search agents (also called *vectors* in DE literature) remain stable and asymptotically converge to an optimum of the search volume for the two parameters F (scale factor) and Cr (crossover rate) remaining below one, which is the usual range of their values. The phase trajectory of the median order vector (in a population of size $NP = 11$) has been shown in Fig. 9 on the function $J(x) = x^2$ for the most popular DE/rand/1/bin scheme [33].

Unlike PSO and DE, the uniqueness of the stability criteria of BFOA remains in the fact that in order to ensure stability of the chemotactic dynamics in BFOA, the step-size parameter C must be adjusted (i.e., made adaptive) according to the current location of the bacterium and its current fitness as shown in (14).

VII. Conclusion

In this paper, we have presented a simple mathematical model of the computational chemotaxis operation in BFOA, which emerges as a prominent optimization technique of current interest. The Lyapunov's stability theorems were applied to derive the conditions of asymptotic stability of a bacterium near an isolated optimum of the fitness landscape. Computer simulations over two 1-D unimodal objective functions illustrate how the bacterium bursts into oscillations around the optimum instead of converging to the same, when the stability criteria derived here are not satisfied. We also note that in classical BFOA, where the step-size is usually kept constant, at some point of time, the step-size violates the conditions of asymptotic stability, and the bacterium starts oscillating around the optimum, instead of converging to it. This calls for some adaptation schemes, which may adjust the step-size *on the run*, thus avoiding the limit cycles. Future work should focus on extending the analysis undertaken here, to a multibacterial swarm working on a multidimensional fitness landscape. Another avenue is to include the effects of reproduction and elimination–dispersal events in the same mathematical model, in order to judge their effects on stability of the group dynamics. Some adaptation schemes for online adjustment of the chemotactic step-size (that guarantees convergence to the optimum) over different objective functions should also be investigated in future.

References

[1] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: Univ. Michigan Press, 1975.

[2] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Boston, MA: Kluwer, 1989.

[3] L. J. Fogel, A. J. Owens, and M. J. Walsh, *Artificial Intelligence Through Simulated Evolution*. Hoboken, NJ: Wiley, 1966.

[4] H.-P. Schwefel, *Evolution and Optimum Seeking*. New York: Wiley, 1995.

[5] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, 1995, pp. 1942–1948.

[6] M. Dorigo and L. M. Gambardella, "Ant colony system: A cooperative learning approach to the traveling salesman problem," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 53–66, Apr. 1997.

[7] F. T. S. Chan and M. K. Tiwari, *Swarm Intelligence: Focus on Ant and Particle Swarm Optimization*. Vienna, Austria: I-Tech Edu. Publishing, 2007.

[8] K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," *IEEE Control Syst. Mag.*, vol. 22, no. 3, pp. 52–67, Jun. 2002.

[9] Y. Liu and K. M. Passino, "Biomimicry of social foraging bacteria for distributed optimization: Models, principles, and emergent behaviors," *J. Optim. Theory Appl.*, vol. 115, no. 3, pp. 603–628, Dec. 2002.

[10] D. H. Kim, A. Abraham, and J. H. Cho, "A hybrid genetic algorithm and bacterial foraging approach for global optimization," *Inf. Sci.*, vol. 177, no. 18, pp. 3918–3937, Sep. 2007.

[11] S. Mishra, "A hybrid least square-fuzzy bacterial foraging strategy for harmonic estimation," *IEEE Trans. Evol. Comput.*, vol. 9, no. 1, pp. 61–73, Feb. 2005.

[12] M. Tripathy, S. Mishra, L. L. Lai, and Q. P. Zhang, "Transmission loss reduction based on facts and bacteria foraging algorithm," in *Parallel Problem Solving From Nature (PPSN IX)*, ser. Lecture Notes in Computer Science, vol. 4193. Berlin: Springer-Verlag, 2006, pp. 222–231.

[13] S. Mishra and C. N. Bhende, "Bacterial foraging technique-based optimized active power filter for load compensation," *IEEE Trans. Power Del.*, vol. 22, no. 1, pp. 457–465, Jan. 2007.

[14] D. H. Kim and C. H. Cho, "Bacterial foraging based neural network fuzzy learning," in *Proc. IICAI*, 2005, pp. 2030–2036.

[15] W. J. Tang, Q. H. Wu, and J. R. Saunders, "A novel model for bacteria foraging in varying environments," in *Proc. ICCSA*, 2006, vol. 3980, pp. 556–565.

[16] M. S. Li, W. J. Tang, W. H. Tang, Q. H. Wu, and J. R. Saunders, "Bacteria foraging algorithm with varying population for optimal power flow," in *Proc. Evo Workshops*, 2007, vol. 4448, pp. 32–41.

[17] M. Tripathy and S. Mishra, "Bacteria foraging-based solution to optimize both real power loss and voltage stability limit," *IEEE Trans. Power Syst.*, vol. 22, no. 1, pp. 240–248, Feb. 2007.

[18] M. Ulagammai, P. Vankatesh, P. S. Kannan, and N. P. Padhy, "Application of bacterial foraging technique trained artificial and wavelet neural networks in load forecasting," *Neurocomputing*, vol. 70, no. 16–18, pp. 2659–2667, Oct. 2007.

[19] M. A. Munoz, J. A. Lopez, and E. Caicedo, "Bacteria foraging optimization for dynamical resource allocation in a multizone temperature experimentation platform," *Anal. Des. Intell. Syst. Using SC Tech., ASC*, vol. 41, pp. 427–435, 2007.

[20] A. Biswas, S. Dasgupta, S. Das, and A. Abraham, "Synergy of PSO and Bacterial foraging optimization: A comparative study on numerical benchmarks," in *Proc. 2nd Int. Symp. HAIS*, E. Corchado *et al.*, Ed. Berlin, Germany: Springer-Verlag, 2007, vol. 44, pp. 255–263.

[21] A. Biswas, S. Dasgupta, S. Das, and A. Abraham, "A synergy of differential evolution and bacterial foraging optimization for faster global search," *Int. J. Neural Mass-Parallel Comput. Inf. Syst.—Neural Network World*, vol. 17, no. 6, pp. 607–626, 2007.

[22] B. D. Hughes, *Random Walks and Random Environments*. London, U.K.: Oxford Univ. Press, 1996.

[23] W. Hahn, *Theory and Application of Lyapunov's Direct Method*. Englewood Cliffs, NJ: Prentice–Hall, 1963.

[24] W. M. Haddad and V. Chellaboina, *Nonlinear Dynamical Systems and Control: A Lyapunov-Based Approach*. Princeton, NJ: Princeton Univ. Press, 2008.

[25] H. Berg and D. Brown, "Chemotaxis in escherichia coli analysed by three-dimensional tracking," *Nature*, vol. 239, no. 5374, pp. 500–504, Oct. 1972.

[26] H. Berg, *Random Walks in Biology*. Princeton, NJ: Princeton Univ. Press, 1993.

[27] R. P. Anwal, *Generalized Functions: Theory and Technique*, 2nd ed. Boston, MA: Birkhãuser, 1998.

[28] J. A. Snyman, *Practical Mathematical Optimization: An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms*. New York: Springer-Verlag, 2005.

[29] B. C. Kuo, *Automatic Control Systems*. Englewood Cliffs, NJ: Prentice–Hall, 1987.

[30] M. Clerc and J. Kennedy, "The particle swarm—Explosion, stability, and convergence in a multidimensional complex space," *IEEE Trans. Evol. Comput.*, vol. 6, no. 1, pp. 58–73, Feb. 2002.

[31] I. C. Trelea, "The particle swarm optimization algorithm: Convergence analysis and parameter selection," *Inf. Process. Lett.*, vol. 85, no. 6, pp. 317–325, Mar. 2003.

[32] V. Kadirkamanathan, K. Selvarajah, and P. J. Fleming, "Stability analysis of the particle dynamics in particle swarm optimizer," *IEEE Trans. Evol. Comput.*, vol. 10, no. 3, pp. 245–255, Jun. 2006.

[33] K. Price, R. Storn, and J. Lampinen, *Differential Evolution—A Practical Approach to Global Optimization*. Berlin, Germany: Springer-Verlag, 2005.

[34] J. Lampinen, "A bibliography of differential evolution algorithm," Lappeenranta Univ. Technol., Dept. Inf. Technol., Lab. Inf. Process., Lappeenranta, Finland, 1999. Tech. Rep. [Online]. Available: http://www.lut.fi/~jlampine/debiblio.htm

[35] B. V. Babu and K. K. N. Sastry, "Estimation of heat transfer parameters in a trickle-bed reactor using differential evolution and orthogonal collocation," *Comput. Chem. Eng.*, vol. 23, no. 3, pp. 327–339, Feb. 1999.

[36] R. Angira and B. V. Babu, "Optimization of process synthesis and design problems: A modified differential evolution approach," *Chem. Eng. Sci.*, vol. 61, no. 14, pp. 4707–4721, Jul. 2006.

[37] B. V. Babu and R. Angira, "Modified Differential Evolution (MDE) for optimization of non-linear chemical processes," *Comput. Chem. Eng.*, vol. 30, no. 6/7, pp. 989–1002, May 2006.

[38] B. V. Babu, P. G. Chakole, and J. H. Syed Mubeen, "Multiobjective Differential Evolution (MODE) for optimization of adiabatic styrene reactor," *Chem. Eng. Sci.*, vol. 60, no. 17, pp. 4822–4837, Sep. 2005.

[39] S. Dasgupta, A. Biswas, S. Das, and A. Abraham, "The population dynamics of differential evolution: A mathematical model," in *Proc. IEEE CEC, IEEE WCCI*, 2008, pp. 1439–1446.

[40] S. Dasgupta, S. Das, A. Abraham, and A. Biswas, "On stability and convergence of the population-dynamics in differential evolution," in *AI Commun.*, 2009, to be published.