Bacterial Foraging Optimization Algorithm: Theoretical Foundations, Analysis, and Applications

Chapter *in* Studies in Computational Intelligence · May 2009

DOI: 10.1007/978-3-642-01085-9_2

4 authors, including:

Swagatam Das
Indian Statistical Institute
489 PUBLICATIONS **23,129** CITATIONS

SEE PROFILE

Arijit Biswas
Indian Institute of Science Bangalore
22 PUBLICATIONS **1,453** CITATIONS

SEE PROFILE

Bacterial Foraging Optimization Algorithm: Theoretical Foundations, Analysis, and Applications

Swagatam Das[1], Arijit Biswas[1], Sambarta Dasgupta[1], and Ajith Abraham[2]

[1] Dept. of Electronics and Telecommunication Engg,
Jadavpur University, Kolkata, India
[2]Norwegian University of Science and Technology, Norway
swagatamdas19@yahoo.co.in, arijitbiswas87@gmail.com, sambartadg@gmail.com,
ajith.abraham@ieee.org

Abstract. Bacterial foraging optimization algorithm (BFOA) has been widely accepted as a global optimization algorithm of current interest for distributed optimization and control. BFOA is inspired by the social foraging behavior of *Escherichia coli*. BFOA has already drawn the attention of researchers because of its efficiency in solving real-world optimization problems arising in several application domains. The underlying biology behind the foraging strategy of *E.coli* is emulated in an extraordinary manner and used as a simple optimization algorithm. This chapter starts with a lucid outline of the classical BFOA. It then analyses the dynamics of the simulated chemotaxis step in BFOA with the help of a simple mathematical model. Taking a cue from the analysis, it presents a new adaptive variant of BFOA, where the chemotactic step size is adjusted on the run according to the current fitness of a virtual bacterium. Nest, an analysis of the dynamics of reproduction operator in BFOA is also discussed. The chapter discusses the hybridization of BFOA with other optimization techniques and also provides an account of most of the significant applications of BFOA until date.

1. Introduction

Bacteria Foraging Optimization Algorithm (BFOA), proposed by Passino [1], is a new comer to the family of nature-inspired optimization algorithms. For over the last five decades, optimization algorithms like Genetic Algorithms (GAs) [2], Evolutionary Programming (EP) [3], Evolutionary Strategies (ES) [4], which draw their inspiration from evolution and natural genetics, have been dominating the realm of optimization algorithms. Recently natural swarm inspired algorithms like Particle Swarm Optimization (PSO) [5], Ant Colony Optimization (ACO) [6] have found their way into this domain and proved their effectiveness. Following the same trend of swarm-based algorithms, Passino proposed the BFOA in [1]. Application of group foraging strategy of a swarm of *E.coli* bacteria in multi-optimal function optimization is the key idea of the new algorithm. Bacteria search for nutrients in a manner to maximize energy obtained per unit time. Individual bacterium also communicates with others by sending signals. A bacterium takes foraging decisions after considering two previous factors. The process, in which a bacterium moves by taking small steps while searching for nutrients, is called chemotaxis and key idea of BFOA is mimicking chemotactic movement of virtual bacteria in the problem search space.

Since its inception, BFOA has drawn the attention of researchers from diverse fields of knowledge especially due to its biological motivation and graceful structure. Researchers are trying to hybridize BFOA with different other algorithms in order to explore its local and global search properties separately. It has already been applied to many real world problems and proved its effectiveness over many variants of GA and PSO. Mathematical modeling, adaptation, and modification of the algorithm might be a major part of the research on BFOA in future.

This chapter is organized as follows: Section 2 provides the biological motivation behind the BFOA algorithm and outlines the algorithm itself in a comprehensive manner. Section 3 provides a simple mathematical analysis of the computational chemotaxis of BFOA in the framework of the classical

gradient descent search algorithm. A mathematical model of reproduction operator is furnished in section 4. Section 5 discusses the hybridization of BFOA with other soft computing algorithms. Section 6 provides an overview of the applications of BFOA in different fields of science and engineering. The chapter is finally summarized in Section 7.

2. The Bacteria Foraging Optimization Algorithm

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 [7, 8]. 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 above-mentioned algorithm the bacteria undergoes chemotaxis, where they like to move towards a nutrient gradient and avoid noxious environment. Generally the bacteria move for a longer distance in a friendly environment. Figure 1 depicts how clockwise and counter clockwise movement of a bacterium take place in a nutrient solution.

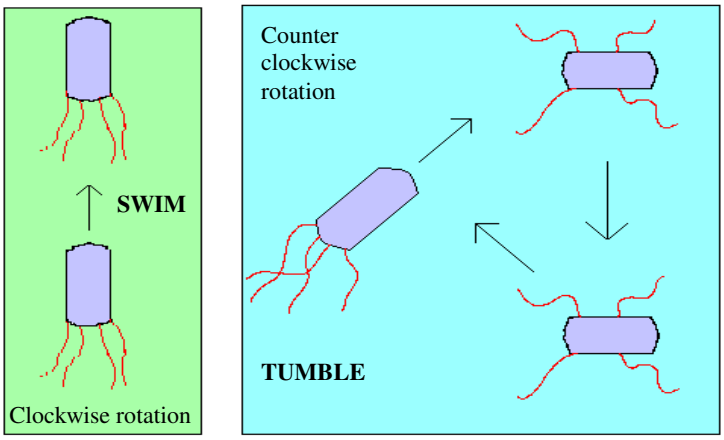


Fig.1. Swim and tumble of a bacterium

When they get food in sufficient, they are increased in length and in presence of suitable temperature they break in the middle to from 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 non-gradient optimization problem. A virtual bacterium is actually one trial solution (may be called a search-agent) that moves on the functional surface (see Figure 2) to locate the global optimum.

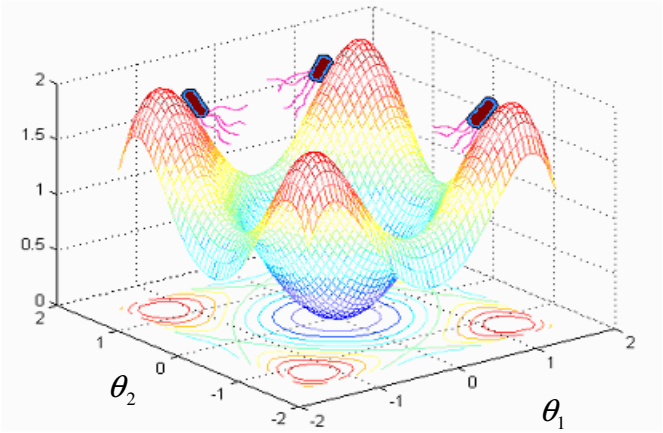


Fig. 2: A bacterial swarm on a multi-modal objective function surface.

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. Also let

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

Let $P(j,k,l)=\{\theta^i(j,k,l)\,|\,i=1,2,...,S\}$ represent the position of each member in the population of the S bacteria at the j-th chemotactic step, k-th reproduction step, and l-th elimination-dispersal event. Here, let $J(i,j,k,l)$ denote the cost at the location of the i-th bacterium $\theta^i(j,k,l)\in\Re^p$ (sometimes we drop the indices and refer to the i-th 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. Below we briefly describe the four prime steps in BFOA.

i) **Chemotaxis**: This process simulates the movement of an *E.coli* cell through swimming and tumbling via flagella. Biologically an *E.coli* bacterium can move in two different ways. It can swim for a period of time in the same direction or it may tumble, and alternate between these two modes of operation for the entire lifetime. Suppose $\theta^i(j,k,l)$ represents i-th bacterium at j-th chemotactic, k-th reproductive and l-th elimination-dispersal step. $C(i)$ is 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

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

where Δ indicates a vector in the random direction whose elements lie in [-1, 1].

ii) **Swarming**: An interesting group behavior has been observed for several motile species of bacteria including *E.coli* and *S. typhimurium*, where intricate and stable spatio-temporal 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 amidst 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.

$$J_{cc}(\theta, P(j,k,l)) = \sum_{i=1}^{S} J_{cc}(\theta, \theta^i(j,k,l))$$

$$= \sum_{i=1}^{S}[-d_{\text{attractant}}\exp(-w_{\text{attractant}}\sum_{m=1}^{p}(\theta_m - \theta_m^i)^2)] + \sum_{i=1}^{S}[h_{\text{repellant}}\exp(-w_{\text{repellant}}\sum_{m=1}^{p}(\theta_m - \theta_m^i)^2)] \quad (2)$$

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, S is the total number of bacteria, p is the number of variables to be optimized, which are present in each bacterium and $\theta = [\theta_1, \theta_2, \ldots, \theta_p]^T$ is a point in the p-dimensional search domain. $d_{\text{aatractant}}$, $w_{\text{attractant}}$, $h_{\text{repellant}}$, $w_{\text{repellant}}$ are different coefficients that should be chosen properly [1, 9].

iii) **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.

iv) **Elimination and Dispersal**: Gradual or sudden changes in the local environment where a bacterium population lives may occur due to various reasons e.g. a significant local rise of temperature may kill a group of bacteria that are currently in a region with a high concentration of nutrient gradients. Events can take place in such a fashion that all the bacteria in a region are killed or a group is dispersed into a new location. 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.

The pseudo-code as well as the flow-chart (Figure 3) of the complete algorithm is presented below:

The BFOA Algorithm

Parameters:

[Step 1] Initialize parameters *p, S, N_c, N_s, N_{re}, N_{ed}, P_{ed}, C(i)(i=1,2…S),* θ^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…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 $J_{last}=J(i, j, k, l)$ 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 $\Delta_m(i), m=1,2,..., p,$ a random number on [-1, 1].

[e] Move: Let

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

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 $J(i, j+1, k, l) < J_{last}$ (if doing better), let $J_{last} = J(i, j+1, k, l)$ and let

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

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 $i = 1,2,..., S$, let

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

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_{health} (higher cost means lower health).

[b] The S_r bacteria with the highest J_{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 $i = 1,2..., S$ 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.

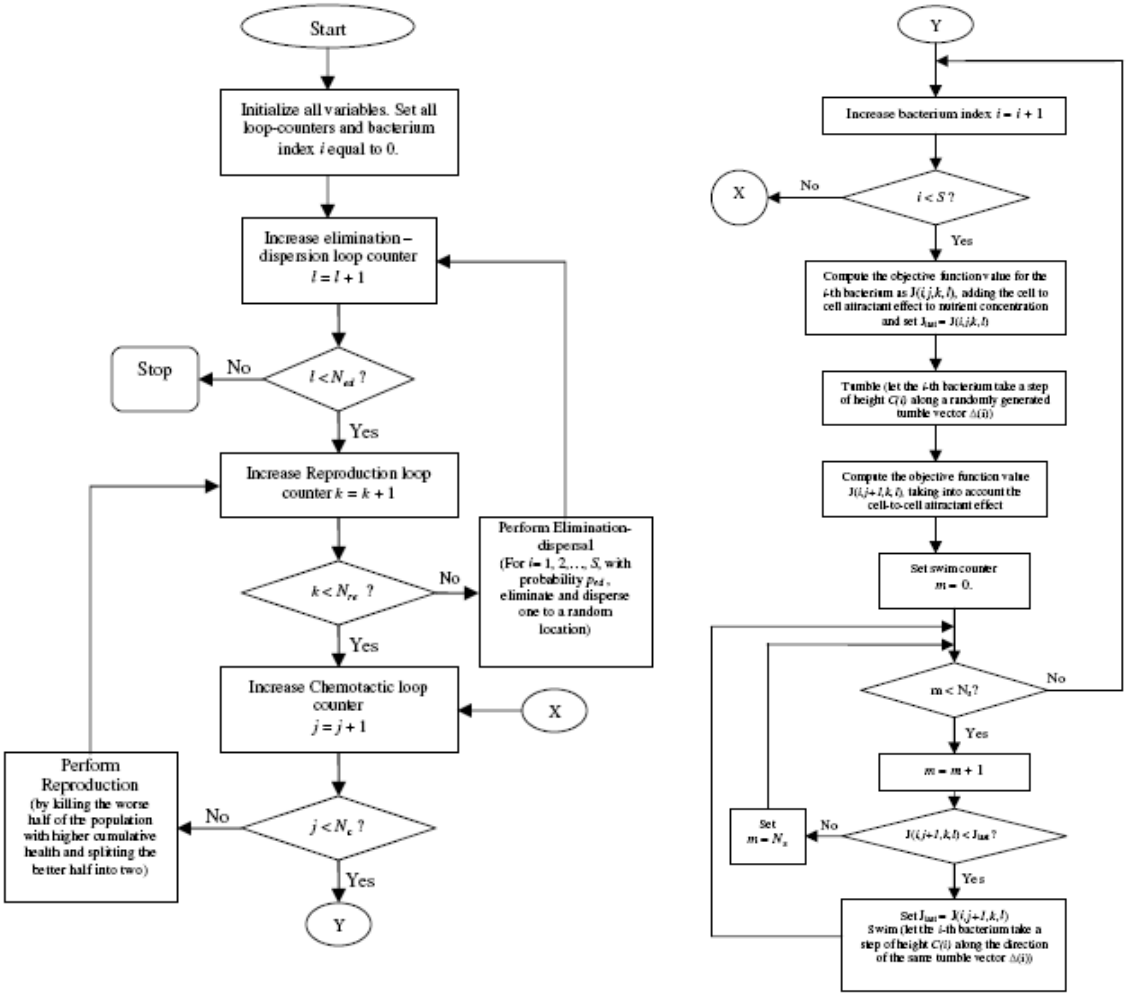


Fig.3: Flowchart of the Bacterial Foraging Algorithm

In Figure 4 we illustrate the behavior of a bacterial swarm on the constant cost contours of the two dimensional sphere model: $f(x_1, x_2) = x_1^2 + x_2^2$. Constant cost contours are curves in $x_1 - x_2$ plane along which $f(x_1, x_2) = x_1^2 + x_2^2 = \text{constant}$.

3. Analysis of the Chemotactic Dynamics in BFOA

Let us consider a single bacterium cell that undergoes chemotactic steps according to (1) over a single-dimensional 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 single dimensional case, without loss of generality. The bacterium lives in continuous time and at the *t*-th 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.

i) The objective function $J(\theta)$ is continuous and differentiable at all points in the search space.

The function is uni-modal in the region of interest and its one and only optimum (minimum) is located at $\theta = \theta_0$. Also $J(\theta) \neq 0$ for $\theta \neq \theta_0$.

ii) The chemotactic step size *C* is smaller than 1 (Passino himself took $C = 0.1$ in [8]).

iii) The analysis applies to the regions of the fitness landscape where gradients of the function are small i.e. near to the optima.

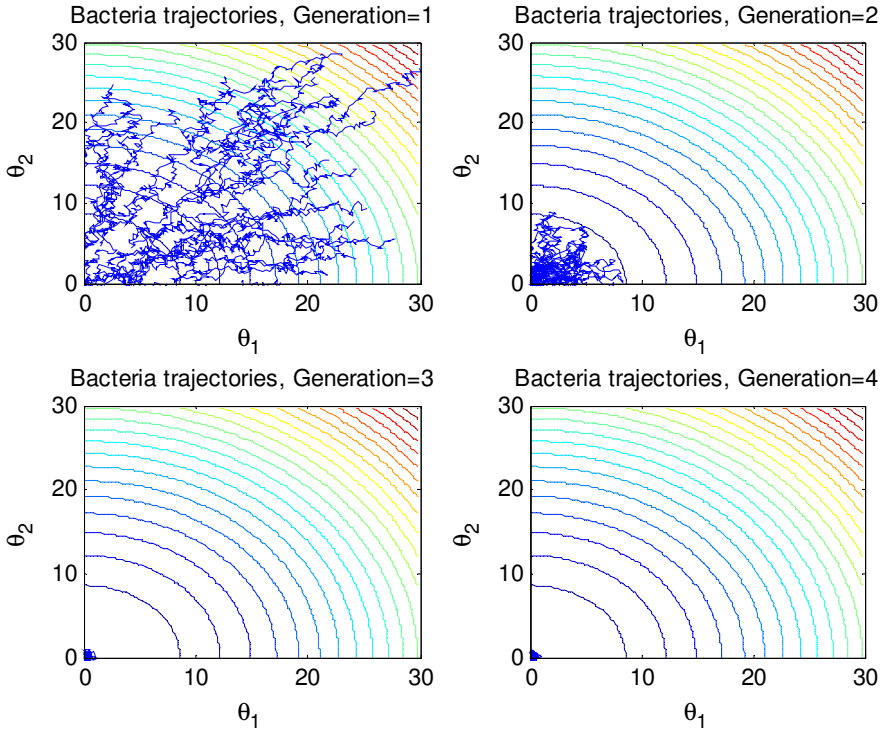


Fig. 4: Convergence behavior of virtual bacteria on the two-dimensional constant cost contours of the sphere model.

3.1 Derivation of Expression for Velocity:

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 number, which assumes only two values with equal probabilities. It is termed as the *direction of tumble* and is denoted by Δ. It can assume only two values 1 or –1 with equal probabilities. For one-dimensional optimization problem Δ is of unit magnitude. 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 sec. 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 sign of the quantity $J(\theta) - J(\theta + \Delta\theta)$ remains unchanged if Δt divides it. So, bacterium will change its position if and only if $\dfrac{J(\theta) - J(\theta + \Delta\theta)}{\Delta t}$

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 [10, 11]) defined as,

$$u(x) = 1, \text{ if } x > 0;$$
$$= 0, \text{ otherwise.} \quad (3)$$

Thus, $\Delta\theta = u(\frac{J(\theta) - J(\theta + \Delta\theta)}{\Delta t}).(C.\Delta)(\Delta t)$, where value of $\Delta\theta$ is 0 or $(C\Delta)(\Delta t)$ according to value of the unit step function. Dividing both sides of above relation by Δt we get,

$$\Rightarrow \frac{\Delta\theta}{\Delta t} = u[-\frac{\{J(\theta + \Delta\theta) - J(\theta)\}}{\Delta t}]C.\Delta \quad (4)$$

Velocity is given by, $V_b = \underset{\Delta t \to 0}{Lim} \frac{\Delta\theta}{\Delta t} = \underset{\Delta t \to 0}{Lim}[u\{-\frac{J(\theta + \Delta\theta) - J(\theta)}{\Delta t}\}.C.\Delta]$

$$\Rightarrow V_b = \underset{\Delta t \to 0}{Lim}[u\{-\frac{J(\theta + \Delta\theta) - J(\theta)}{\Delta\theta}\frac{\Delta\theta}{\Delta t}\}.C.\Delta]$$

as $\Delta t \to 0$ makes $\Delta\theta \to 0$, we may write, $V_b = [u\{-\left(\underset{\Delta\theta \to 0}{Lim} \frac{J(\theta + \Delta\theta) - J(\theta)}{\Delta\theta}\right)\left(\underset{\Delta t \to 0}{Lim} \frac{\Delta\theta}{\Delta t}\right)\}.C.\Delta]$

Again, $J(x)$ is assumed to be continuous and differentiable. $\underset{\Delta\theta \to 0}{Lim} \frac{J(\theta + \Delta\theta) - J(\theta)}{\Delta\theta}$ is the value of the gradient at that point and may be denoted by $\frac{dJ(\theta)}{d\theta}$ or G. Therefore we have:

$$V_b = u(-GV_b)C\Delta \quad (5)$$

where, $G = \frac{dJ(\theta)}{d\theta} =$ gradient of the objective function at θ.

In (5) argument of the unit step function is $-GV_b$. Value of the unit step function is 1 if G and V_b are of different sign and in this case the velocity is $C\Delta$. Otherwise, it is 0 making bacterium motionless. So (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

$$\phi(x) = \frac{1}{1 + e^{-kx}}$$

We note that, $$u(x) = \underset{k \to \infty}{Lt}\ \phi(x) = \underset{k \to \infty}{Lt}\ \frac{1}{1 + e^{-kx}} \quad (6)$$

Figure 5 illustrates how the logistic function may be used to approximate the unit step function used for decision-making in chemotaxis. For analysis purpose *k* cannot be infinity. We restrict ourselves to moderately large values of *k* (say *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),

$$V_b = \frac{C\Delta}{1 + e^{kGV_b}} \quad (7)$$

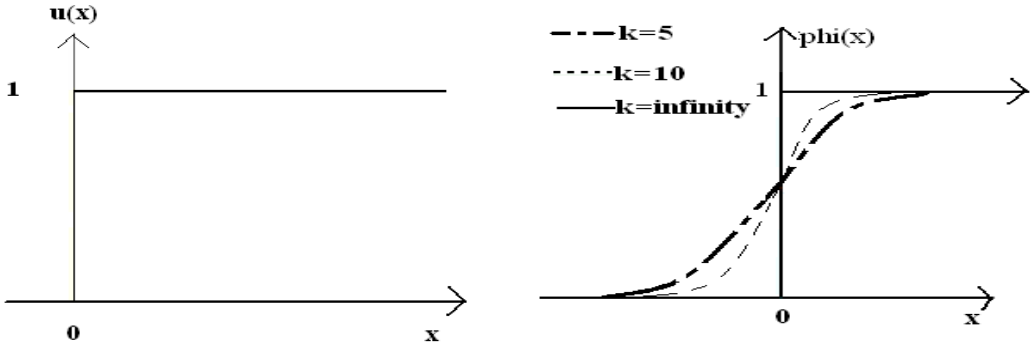


Fig. 5: The unit step and the logistic functions

According to assumptions (ii) and (iii), if *C* and *G* are very small and *k* ~10, then also we may 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,

$$\Rightarrow V_b = \frac{C.\Delta}{2} \frac{1}{1 + \frac{kGV_b}{2}}$$

$$\Rightarrow V_b = \frac{C.\Delta}{2}(1 - \frac{kGV_b}{2}) \qquad [\because |\frac{kGV_b}{2}| \ll 1, \text{ neglecting higher terms, } (1 + \frac{kGV_b}{2})^{-1} \approx (1 - \frac{kGV_b}{2})\ ]$$

After some manipulation we have,

$$\Rightarrow V_b = \frac{C\Delta}{2} \cdot \frac{1}{1 + \frac{kCG\ \Delta}{4}} \tag{8}$$

$$\Rightarrow V_b = \frac{C\Delta}{2}(1 - \frac{kGC\,\Delta}{4}) \qquad [\because |\frac{kGC\Delta}{4}| = |\frac{kGC}{4}| \ll 1, \text{ as } |\Delta| = 1 \text{ and neglecting the higher order terms.}]$$

$$\Rightarrow V_b = \frac{C\Delta}{2} - \frac{kGC^2\Delta^2}{8}$$

$$\Rightarrow V_b = -\frac{kC^2}{8}G + \frac{C\Delta}{2} \quad [\because \Delta^2 = 1] \tag{9}$$

Equation (9) is applicable to a single bacterium system and it does not take into account the cell-to-cell signaling effect. A more complex analysis for the two-bacterium system involving the swarming effect has been included at the appendix. It indicates that, a complex perturbation term is added to the dynamics of each bacterium due to the effect of the neighboring bacteria cells. However, the term becomes negligibly small for small enough values of *C* (~0.1) and the dynamics under these circumstances get practically reduced to that described in equation (9). In what follows, we shall continue the analysis for single bacterium system for better understanding of the chemotactic dynamics.

3.2 Experimental Verification of Expression for Velocity

Characteristic equation of chemotaxis (9) represents the dynamics of bacterium taking chemotactic steps. In order to verify how reliably the equation represents the motion of the virtual bacterium compare results obtained from (10) with that of according to BFOA. First the equation is expressed in iterative form, which is,

$$V_b(n) = \theta(n) - \theta(n-1) = -\frac{kC^2}{8}G(n-1) + \frac{C\Delta(n)}{2}$$

$$\Rightarrow \theta(n) = \theta(n-1) - \frac{kC^2}{8}G(n-1) + \frac{C\Delta(n)}{2} \quad (10)$$

where n is the iteration index. The tumble vector is also a function of iteration count (i.e. chemotactic step number) i.e. it is generated repeatedly for successive iterations. We have taken $J(\theta) = \theta^2$ as objective function for this experimentation. Bacterium was initialized at –2 i.e. $\theta(0) = -2$ and C is taken as 0.2. Gradient of $f(x)$ is $2x$. Therefore $G(n-1)$ may be replaced by $2\theta(n-1)$ .Finally for this specific case we get,

$$\theta(n) = (1 - \frac{kC^2}{4})\theta(n-1) + \frac{C\Delta(n)}{2} \quad (11)$$

We compute values of $\theta(n)$ for successive iterations according to above iterative relation. Also values of positions are noted following guidelines of BFOA. With current position is changed by $C\Delta$ if objective function value decreases for new position. Results have been presented in Figure 6. Figure 6 (a) shows position in successive iteration according to BFOA and as obtained from (11). Here also we have assumed position of bacterium changes linearly between two consecutive iterations. Mismatch between actual and predicted values is also shown. In Figure 6 (b) actual and predicted values of velocity is shown. 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 0. Difference between actual and predicted velocity is shown as error. Time lapsed between two consequent 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. Actually it is the time taken by the processor to perform numerical computations.

3.3 Chemotaxis and the Classical Gradient Decent Search

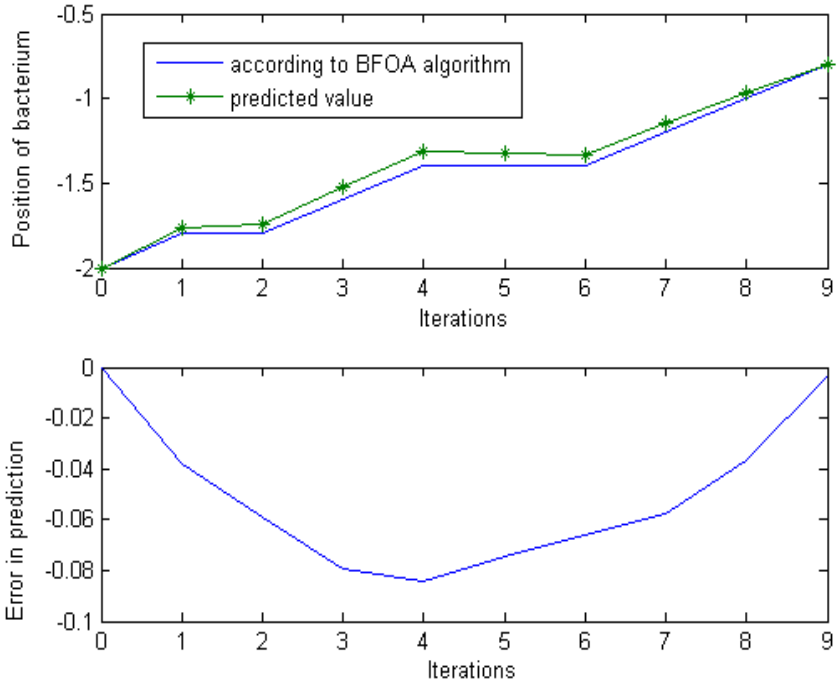
From expression (9) of Section 3.1, we get

$$V_b = -\frac{kC^2}{8}G + \frac{C\Delta}{2} \Rightarrow \frac{d\theta}{dt} = -\alpha' G + \beta' \quad (12)$$

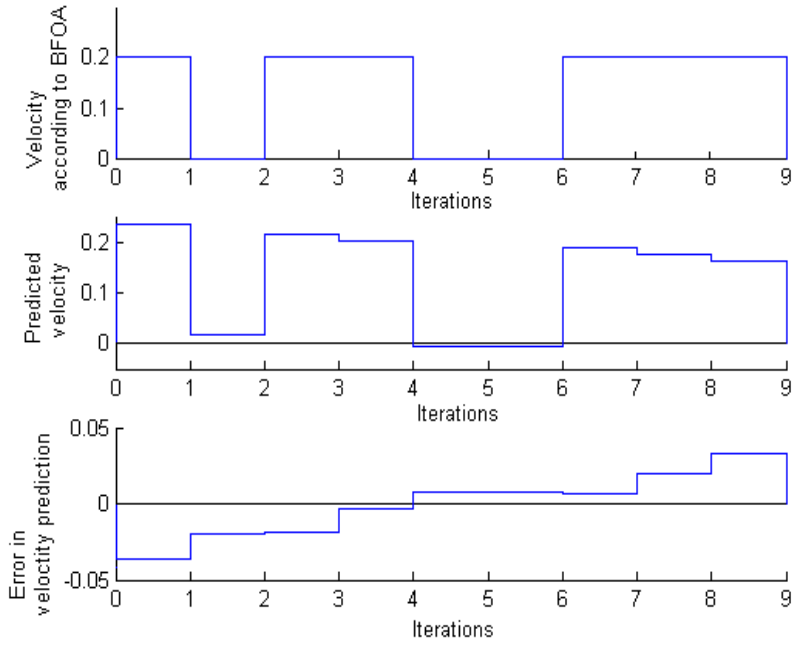
where α' is $\frac{-kC^2}{8}$ and β' is $\frac{C\Delta}{2}$. The classical gradient descent search algorithm is given by the following dynamics in single dimension [12]:

$$\frac{d\theta}{dt} = -\alpha.G + \beta \quad (13)$$

where, α is the learning rate and β is the momentum. Similarity between equations (12) and (13) suggests that chemotaxis may be considered a modified gradient descent search, where α' , a function of chemotactic step-size can be identified as the learning rate parameter.



(a) Graphs showing actual, predicted positions of bacterium and error in estimation over successive iterations.



(b) Similar plots for velocity of the bacterium.

Fig. 6: Comparison between actual and predicted motional state of the bacterium.

Already we have discussed that magnitude of gradient should be small within the region of our analysis. For chemotaxis of BFOA, when *G* becomes very mall, the gradient descent term $\alpha' G$ of

equation (12) becomes ineffective. But the random search term $\frac{C\Delta}{2}$ plays an important role in this context. From equation (12), considering $G \to 0$, we have

$$\frac{d\theta}{dt} = \frac{C.\Delta}{2} \neq 0 \tag{14}$$

So there is a convergence towards actual minima. The random search or momentum term $C\Delta/2$ in the RHS of equation (13) 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. But random search term may lead to change in position in the direction of increasing objective function value. If it happens then again magnitude of gradient increases and dominates the random search term.

3.4 Oscillation Problem: Need for Adaptive Chemotaxis

If magnitude of the gradient decreases consistently, near the optima or very close to the optima $\alpha' G$ of expression (12) becomes comparable to β. Then gradually β becomes dominant. When $|G| \to 0, |\frac{d\theta}{dt}| \approx |\beta| = |\frac{C\Delta}{2}| = \frac{C}{2} \quad \because |\Delta| = 1$. Let us assume the bacterium has reached close to the optimum. But since we obtain $|\frac{d\theta}{dt}| = \frac{C}{2}$, the bacterium does not stop taking chemotactic steps and oscillates about the optima. This crisis can be remedied if step size *C* is made adaptive according to the following relation,

$$C = \frac{|J(\theta)|}{|J(\theta)| + \lambda} = \frac{1}{1 + \lambda / |J(\theta)|}, \tag{15}$$

where λ is a positive constant. Choice of a suitable value for λ has been discussed in the next subsection. Here we have assumed that the global optimum of the cost function is 0. Thus from (25) we see, if $J(\theta) \to 0$, then $C \to 0$. So there would be no oscillation if the bacterium reaches optima because random search term vanishes as $C \to 0$. The functional form given in equation (15) causes *C* to vanish nears the optima. Besides, it plays another important role described below. From (15), we have, when $J(\theta)$ is large $\frac{\lambda}{|J(\theta)|} \to 0$ and consequently $C \to 1$.

The adaptation scheme presented in equation (15) has an important physical significance. If magnitude of cost function is large for an individual bacterium, it is in the vicinity of noxious substance. It will then try to move to a place with better nutrient concentration by taking large steps. On the other hand the bacterium, when in nutrient rich zone i.e. with small magnitude of the objective function value, tries to retain its position. Naturally, its step size becomes small.

The BFOA is made adaptive according to the above rule and its performance improved with respect to speed of convergence, quality of solution and rate of success rate.

3.5 A Special Case

If the optimum value of the objective function is not exactly zero, step-size adapted according to (15) may not vanish near optima. Step-size would shrink if the bacterium comes closer to the optima, but it may not approach zero always. To get faster convergence for such functions it becomes necessary to modify the adaptation scheme. Use of gradient information in the adaptation scheme i.e. making step-

size a function of the function-gradient (say $C = C(J(\theta), G)$) may not be practical enough, because in real-life optimization problems, we often deal with discontinuous and non-differentiable functions. In order to make BFOA a general black-box optimizer, our adaptive scheme should be a generalized one performing satisfactorily in these situations too. Therefore to accelerate the convergence under these circumstances, we propose an alternative adaptation strategy in the following way:

$$C = \frac{\left|J(\theta) - J_{best}\right|}{\left|J(\theta) - J_{best}\right| + \lambda} \tag{16}$$

J_{best} is the objective function value for the globally best bacterium (one with lowest value of objective function). $\left|J(\theta) - J_{best}\right|$ is the deviation in fitness value of an individual bacterium from global best. Expression (16) can be rearranged to give,

$$C = \frac{1}{1 + \dfrac{\lambda}{\left|J(\theta) - J_{best}\right|}}. \tag{17}$$

If a bacterium is far apart from the global best, $\left|J(\theta) - J_{best}\right|$ would be large making $C \approx 1 \because \dfrac{\lambda}{\left|J(\theta) - J_{best}\right|} \to 0$. On the other hand if another bacterium is very close to it, step size of that bacterium will almost vanish, because $\left|J(\theta) - J_{best}\right|$ becomes small and denominator of (17) grows very large. The scenario is depicted in Figure 7. BFOA with adaptive scheme of equation (15) is referred as ABFOA1 and the BFOA with adaptation scheme described in (17) is referred as ABFOA2.

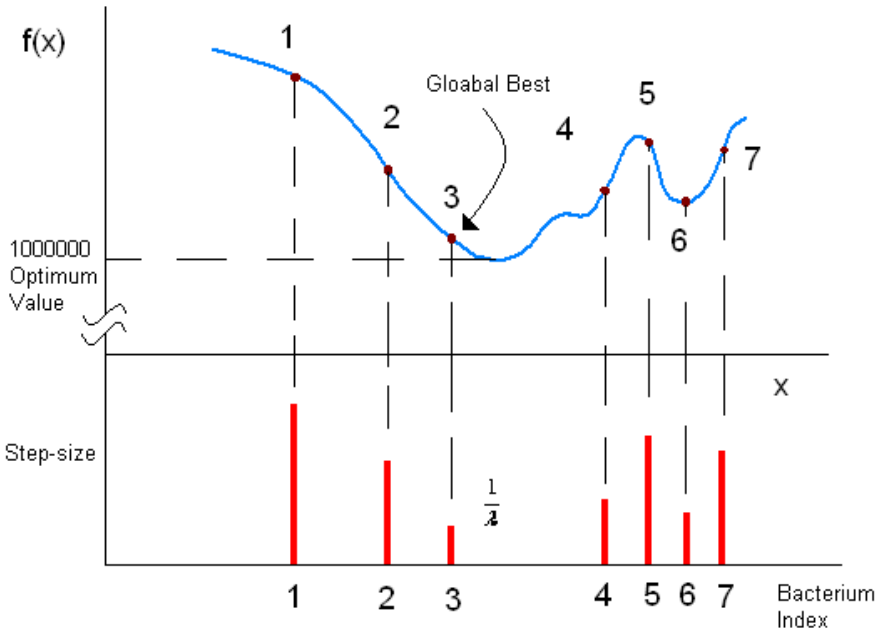


Fig. 7: An objective function with optimum value much greater than zero and a group of seven bacteria are scattered over the fitness landscape. Their step height is also shown.

Figure 7 shows how the step-size becomes large as objective function value becomes large for an individual bacterium. The bacterium with better function value tries to take smaller step and to retain its present position. For best bacterium of the swarm $\left|J(\theta) - J_{best}\right|$ is 0. Thus, from (17) its step-size