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

Swagatam Das, Arijit Biswas, Sambarta Dasgupta, and Ajith Abraham

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

Swagatam Das, Arijit Biswas, and Sambarta Dasgupta
Department of Electronics and Telecommunication Engineering,
Jadavpur University, Kolkata, India

Ajith Abraham
Norwegian University of Science and Technology, Norway
e-mail: ajith.abraham@ieee.org

A. Abraham et al. (Eds.): Foundations of Comput. Intel. Vol. 3, SCI 203, pp. 23–55.
springerlink.com

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.

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

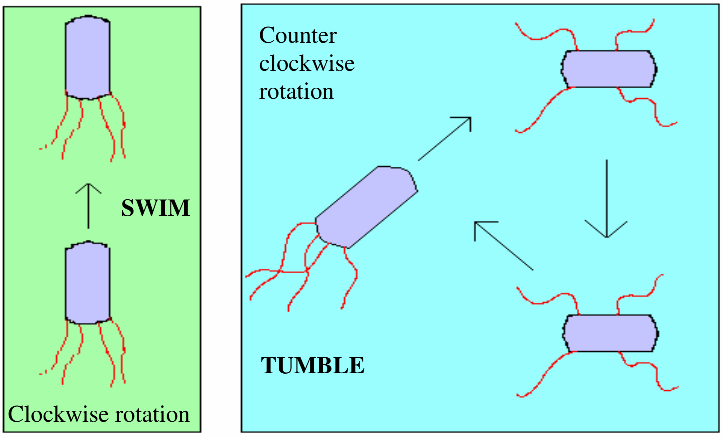


Fig. 1 Swim and tumble of a bacterium

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.

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.

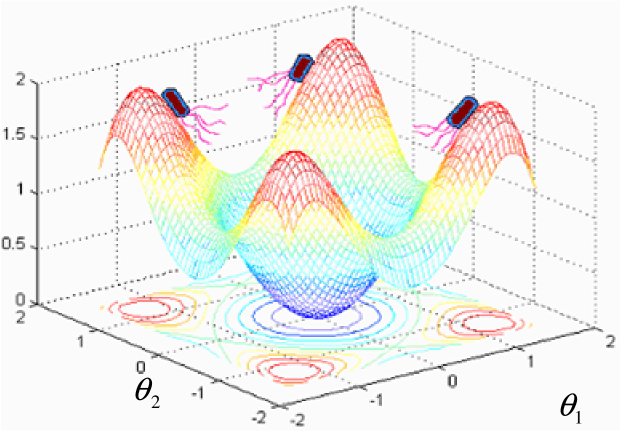


Fig. 2 A bacterial swarm on a multi-modal objective function surface

Let $P(j,k,l)=\{\theta^i(j,k,l) \mid i=1,2,...,S\}$ 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

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

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.

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

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

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, S is the total number of bacteria, p is the number of variables to be optimized, which are present in each bacterium and $\theta = [\theta_1, \theta_2, ..., \theta_p]^T$ 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

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

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.

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 $f(x_1, x_2) = x_1^2 + x_2^2 =$ constant.

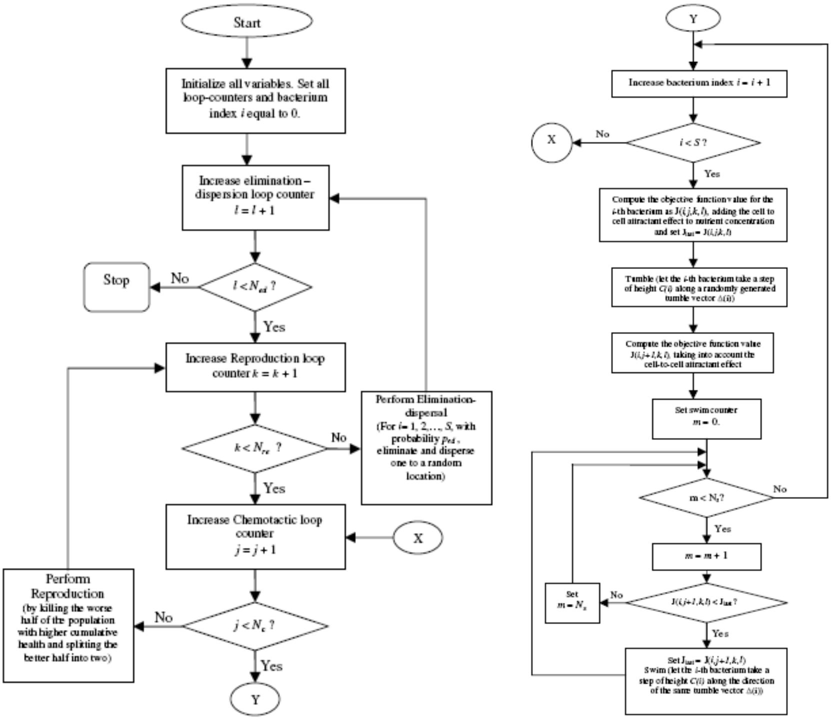

Start
Initialize all variables. Set all loop-counters and bacterium index i equal to 0.
Increase elimination–dispersion loop counter $l = l + 1$
$l < N_{ed}$?
No
Stop
Yes
Increase Reproduction loop counter $k = k + 1$
$k < N_{re}$?
No
Perform Elimination-dispersal (For i= 1, 2,…, S, with probability p_{ed}, eliminate and disperse one to a random location)
Yes
Increase Chemotactic loop counter $j = j + 1$
X
$j < N_c$?
No
Perform Reproduction (by killing the worse half of the population with higher cumulative health and splitting the better half into two)
Yes
Y
Y
Increase bacterium index $i = i + 1$
$i < S$?
No
X
Yes
Compute the objective function value for the i-th bacterium as $J(i,j,k,l)$, adding the cell to cell attractant effect to nutrient concentration and set $J_{last} = J(i,j,k,l)$
Tumble (let the i-th bacterium take a step of height $C(i)$ along a randomly generated tumble vector $\Delta(i)$)
Compute the objective function value $J(i,j+1,k,l)$, taking into account the cell-to-cell attractant effect
Set swim counter $m = 0$.
m < N_s?
No
Yes
$m = m + 1$
$J(i,j+1,k,l) < J_{last}$?
No
Set $m = N_s$
Yes
Set $J_{last} = J(i,j+1,k,l)$ Swim (let the i-th bacterium take a step of height $C(i)$ along the direction of the same tumble vector $\Delta(i)$)

Fig. 3 Flowchart of the Bacterial Foraging Algorithm

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]).

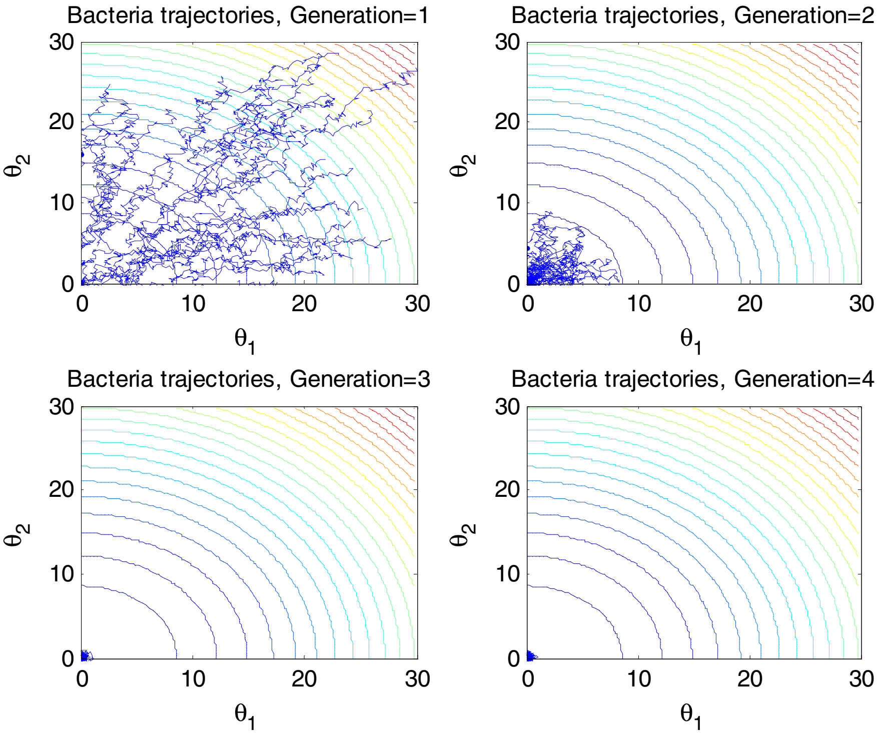


Fig. 4 Convergence behavior of virtual bacteria on the two-dimensional constant cost contours of the sphere model

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

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 $\frac{J(\theta) - J(\theta + \Delta\theta)}{\Delta t}$ 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,

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

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,

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

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:

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

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

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),

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

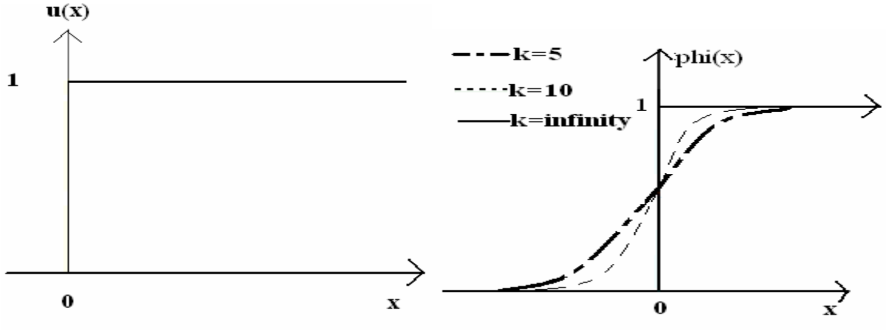


Fig. 5 The unit step and the logistic functions

According to assumptions (ii) and (iii), if *C* and *G* are very small and *k* ~10, then also we may have $|kGV_b| << 1$.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,

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

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

After some manipulation we have,

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

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

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

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

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,

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

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

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,

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

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

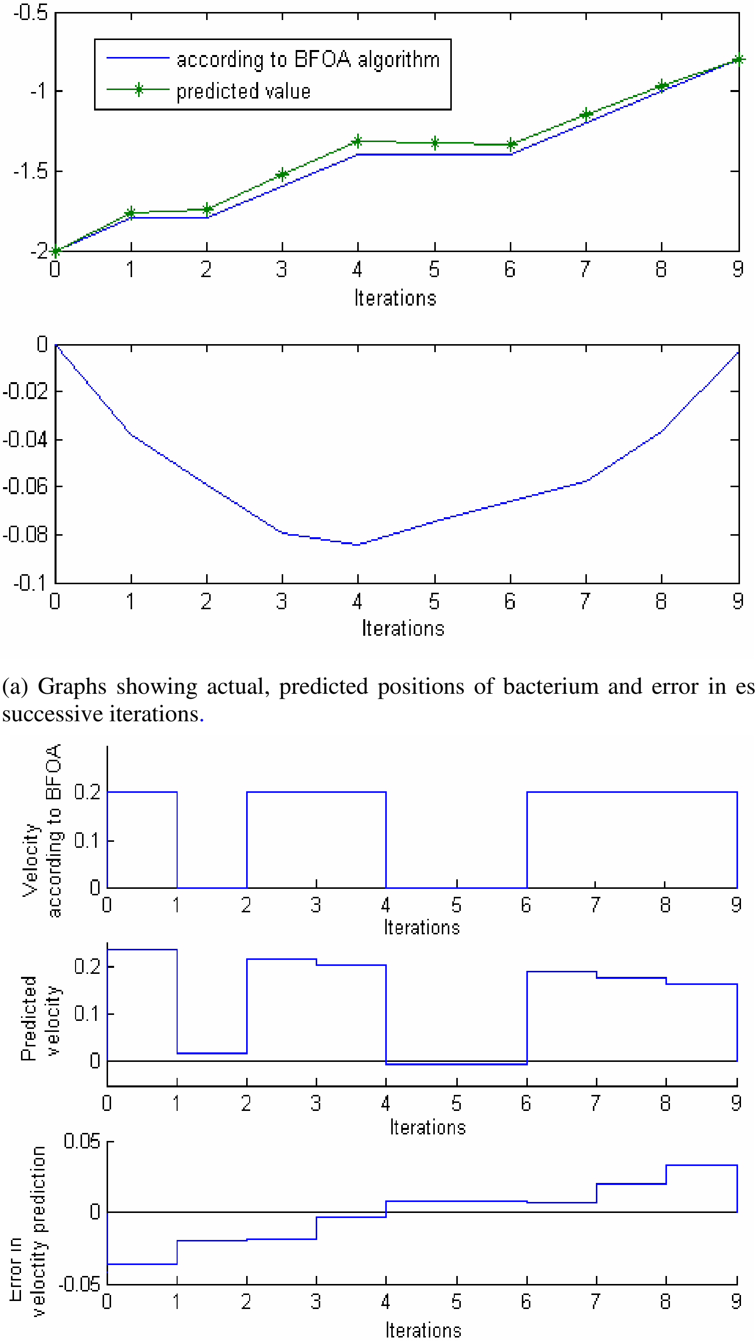
From expression (9) of Section 3.1, we get

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

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]:

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

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:

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

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

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

If a bacterium is far apart from the global best, $|J(\theta) - J_{best}|$ would be large making $C \approx 1 \because \frac{\lambda}{|J(\theta) - J_{best}|} \to 0$. On the other hand if another bacterium is very close to it, step size of that bacterium will almost vanish, because $|J(\theta) - J_{best}|$ becomes small and denominator of (17) grows very large. The scenario is

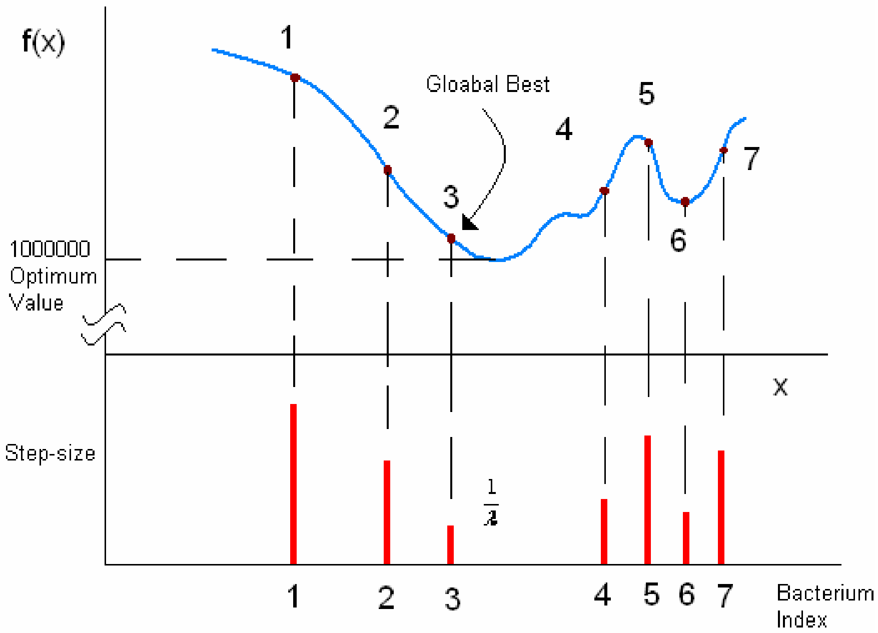


Fig. 7 An objective function with optimum value much greater than zero and a group of seven bacteria are scattered over the fitness landscape. Their step height is also shown

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.

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 $|J(\theta) - J_{best}|$ is 0. Thus, from (17) its step-size is $1/\lambda$, which is quite small. The adaptation scheme bears a physical significance too. A bacterium located at relatively less nutrient region of fitness landscape will take large step sizes to attain better fitness. Whereas, another bacterium located at a location, best in regard to nutrient content, is unlikely to move much.

In real world optimization problems optimum value of objective function is very often found to be zero. In those cases adaptation scheme of (15) works satisfactorily. But for functions, which do not have a moderate optimum value, (16) should be used for better convergence. Note that neither of two proposed schemes contains derivative of objective function so they can be used for discontinuous and non-differentiable functions as well. In [13], Dasgupta *et al.* have established the efficacy of the adaptive BFO variants by comparing their performances with classical BFOA, its other state-of-the-art variants, a recently proposed variant of PSO and a standard real-coded GA on numerical benchmarks as well as one engineering optimization problem.

4 Analysis of the Reproduction Step in BFOA

This section presents a simple mathematical analysis of the reproduction operator of BFOA for a two-bacterium system [14]. Let us consider a small population of two bacteria that sequentially undergoes the four basic steps of BFOA over a one-dimensional objective function. The bacteria live in continuous time and at the *t*-th instant its position is given by $\theta(t)$. Below we list a few assumptions that were considered for the sake of gaining mathematical insight into the dynamics of reproduction.

Assumptions:

i) The objective function *J(θ)* is continuous and differentiable at all points in the search space.
ii) The analysis applies to the regions of the fitness landscape where gradients of the function are small i.e., near to the optima. The region of fitness landscapes between θ_1 and θ_2 is monotonous at the time of reproduction.
iii) During reproduction, two bacteria remain close to each other and one of them must not superpose on another (i.e. $|\theta_2 - \theta_1| \to 0$ may happen due to reproduction but $\theta_2 \neq \theta_1$. Suppose P and Q represent the respective positions of the two bacteria as shown in fig.6). At the start of reproduction θ_1 and θ_2 remain apart from each other but as the process progresses they come close to each other gradually.

iv) The bacterial system lives in continuous time.

4.1 Analytical Treatment

In our two bacterial system, $\theta_1(t)$ and $\theta_2(t)$ represent the position of the two bacteria at time t and $J(\theta_1), J(\theta_2)$ denote the cost function values at those positions respectively. During reproduction, the virtual bacterium with a relatively larger value of the cost function (for a minimization problem) is liquidated while the other is split into two. These two offspring bacteria start moving from the same location. Hence in effect, through reproduction the least healthy bacteria shift towards the healthier bacteria. Health of a bacterium is measured in terms of the accumulated cost function value, possessed by the bacterium until that time instant. The accumulated cost may be mathematically modeled as $\int_0^t J(\theta_1(t))dt$.

For a minimization problem, higher accumulated cost represents that a bacterium

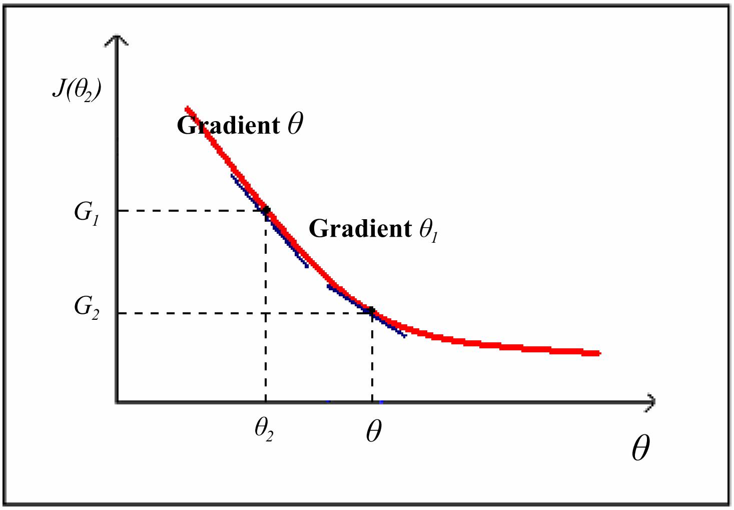


Fig. 8 A two-bacterium system on arbitrary fitness landscape

did not get as many nutrients during its lifetime of foraging and hence is not as "healthy" and thus unlikely to reproduce .The two-bacterial system working on a single-dimensional fitness landscape has been depicted in Figure 8.

To simulate the bacterial reproduction we have to take a decision on which bacterium will split in next generation and which one will die. This decision may be modeled with the help of the well-known unit step function $u(x)$ defined in equation (3). In what follows, we shall denote $\theta_1(t)$ and $\theta_2(t)$ as θ_1 and θ_2 respectively. Now if we consider that $\Delta\theta_1$ is the infinitesimal displacement ($\Delta\theta_1 \to 0$) of the first bacterium in infinitesimal time Δt $(\Delta t \to 0)$ towards the second bacterium in favorable condition i.e. when the second is healthier than the first one, then the instantaneous velocity of the first one is given by, $\frac{\Delta\theta_1}{\Delta t}$. Now when we are trying to model reproduction we assume the instantaneous velocity of the worse bacterium to be proportional with the distance between the two bacteria, i.e. as they come closer their velocity decreases but this occurs unless we incorporate the decision making part. So, if the first bacterium is the worse one then,

$$\frac{\Delta\theta_1}{\Delta t} \infty (\theta_2 - \theta_1)$$

$$\Rightarrow \frac{\Delta\theta_1}{\Delta t} = \bar{k}(\theta_2 - \theta_1) \qquad \text{[Where, } \bar{k} \text{ is the proportionality constant]}$$

$$\Rightarrow \frac{\Delta\theta_1}{\Delta t} = 1.(\theta_2 - \theta_1) = (\theta_2 - \theta_1) \tag{18}$$

[If we assume that $\bar{k} = 1 \ \sec^{-1}$]

Since we are interested in modeling a dynamics of the reproduction operation, the decision making i.e. whether one of the bacteria will move towards the other, can not be discrete i.e. it is not possible to check straightaway whether the other bacterium is at a better position or not. So a bacterium (suppose θ_1) will be checking whether a position situated at an infinitesimal distance from θ_1 is healthier or not and then it will move (see Figure 9). The health of first bacterium is given by the integral of $J(\theta_1)$ from zero to time t and the same for the differentially placed position is given by the integral of $J(\theta_1 + \Delta\theta_1)$ from zero to time t. Then we may model the decision making part with the unit step function in the following way:

$$\frac{\Delta\theta_1}{\Delta t} = u[\int_0^t J(\theta_1)dt - \int_0^t J(\theta_1 + \Delta\theta_1)dt].(\theta_2 - \theta_1) \tag{19}$$

Similarly, when we consider the second bacterium, we get,

$$\frac{\Delta\theta_2}{\Delta t} = u[\int_0^t J(\theta_2)dt - \int_0^t J(\theta_2 + \Delta\theta_2)dt].(\theta_1 - \theta_2) \tag{20}$$

In equation (19), $\int_0^t J(\theta_1)dt$ represents the health of the first bacterium at the time instant *t* and $\int_0^t J(\theta_1 + \Delta\theta_1)dt$ represents the health corresponding to $(\theta_1 + \Delta\theta_1)$ at the time instant *t*. We are going to carry out calculations with the equation for bacterium 1 only, as the results for other bacterium can be obtained in a similar fashion.

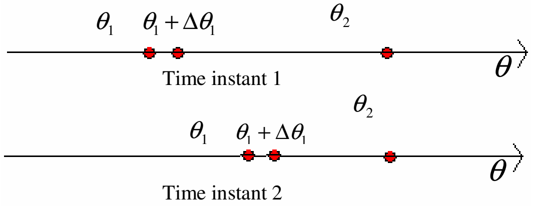


Fig. 9 Change of position of the bacteria during reproduction

Since we are considering only the monotonous part of any function, so if θ_2 is at a better position, then any position, in-between θ_1 and θ_2 , has a lesser objective function value compared to θ_1. So we may conclude $J(\theta_1 + \Delta\theta_1)$ is less than $J(\theta_1)$. In that case we can imagine that $\int_0^t J(\theta_1 + \Delta\theta_1).dt$ is less than $\int_0^t J(\theta_1).dt$ as t is not too high, the functional part under consideration is monotonous and change of $\theta_1 + d\theta_1$ with respect to t is same as that of θ_1. We rewrite the equation (19) corresponding to bacterium 1 as,

$$\Rightarrow \frac{\Delta\theta_1}{\Delta t} = u[-\int_0^t \frac{J(\theta_1 + \Delta\theta_1) - J(\theta_1)}{\Delta t} dt](\theta_2 - \theta_1)$$

[$\because$ $\Delta t > 0$. We know for a positive constant Δt, $u(\frac{x}{\Delta t}) = u(x)$ as x and $\frac{x}{\Delta t}$ are of same sign and unit step function depends only upon sign of the argument.]

$$\Rightarrow \underset{\substack{\Delta t \to 0 \\ \Delta\theta_1 \to 0}}{Lt} \frac{\Delta\theta_1}{\Delta t} = \underset{\substack{\Delta t \to 0 \\ \Delta\theta_1 \to 0}}{Lt} u[-\int_0^t \frac{J(\theta_1 + \Delta\theta_1) - J(\theta_1)}{\Delta t} dt].(\theta_2 - \theta_1)$$

$$\Rightarrow \underset{\substack{\Delta t \to 0 \\ \Delta\theta_1 \to 0}}{Lt} \frac{\Delta\theta_1}{\Delta t} = \underset{\substack{\Delta t \to 0 \\ \Delta\theta_1 \to 0}}{Lt} u[-\int_0^t \frac{J(\theta_1 + \Delta\theta_1) - J(\theta_1)}{\Delta\theta_1} \frac{\Delta\theta_1}{\Delta t} dt].(\theta_2 - \theta_1) \qquad (21)$$

Again, $J(x)$ is assumed to be continuous and differentiable. $\underset{\Delta\theta \to 0}{Lim} \frac{J(\theta_1 + \Delta\theta_1) - J(\theta_1)}{\Delta\theta_1}$ is the value of the gradient at that point and may be denoted by $\frac{dJ(\theta_1)}{d\theta_1}$ or G_1. So we write,

$$\Rightarrow \frac{d\theta_1}{dt} = u[-\int_0^t (\frac{dJ}{d\theta_1} \frac{d\theta_1}{dt}) dt].(\theta_2 - \theta_1)$$ [Where $\frac{d\theta_1}{dt}$ is the instantaneous velocity of the first bacterium]

$$\Rightarrow v_1 = u[-\int_0^t G_1 v_1 dt].(\theta_2 - \theta_1) \qquad (22)$$

[Where $v_1 = \frac{d\theta_1}{dt}$ and G_1 is the gradient of J at $\theta = \theta_1$.]

Now in equation (19) we have not yet considered the fact that the event of reproduction is taking place at t=1 only. So we must introduce a function of

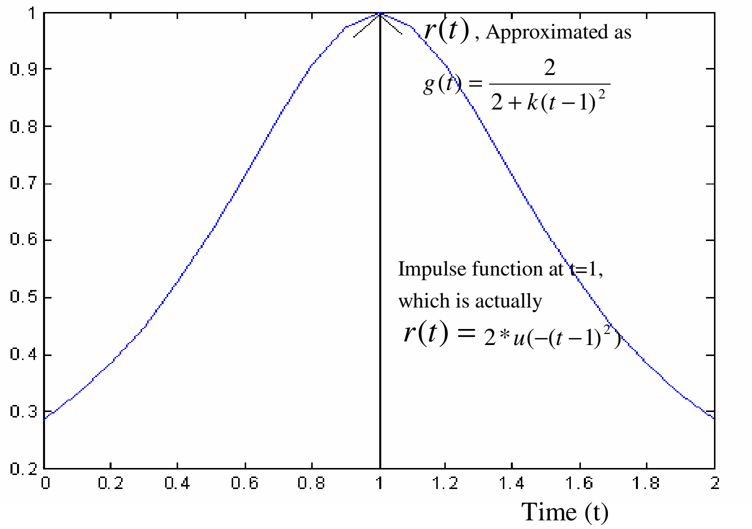


Fig. 10 Function r (t) and g(t)

time $r(t) = 2*u(-(t-1)^2)$ (unit step) ($u(-(t-1)^2)$ is multiplied with 2 for getting $r(t) = 1$, not 0.5, when t=1) in product with the right hand side of equation (19). This provides a sharp impulse of strength 1 unit at time t = 1. Now it is well known that $u(x)$ may be approximated with the continuous logistic function $\phi(x)$, where $\phi(x) = \dfrac{1}{1+e^{-kx}}$.

We note that,

$$u(x) = \underset{k\to\infty}{Lt}\,\phi(x) = \underset{k\to\infty}{Lt}\,\frac{1}{1+e^{-kx}} \tag{23}$$

Following this we may write:

$$r(t) = 2*u(-(t-1)^2) \approx \frac{2}{1+e^{k(t-1)^2}}$$

For moderately large value of *k*, since $t \to 1$, we can have $\left|k(t-1)^2\right| << 1$ and thus $e^{k(t-1)^2} \approx 1+k(t-1)^2$. Using this approximation of the exponential term we may replace the unit step function $r(t)$ with another continuous function *g*(t) where

$$g(t) = \frac{2}{2+k(t-1)^2}, \qquad \text{(we can take } k = 10\text{)}$$

which is not an impulsive function just at t = 1 rather a continuous function as shown in Figure 10. Higher value of k will produce more effective result. Due to the presence of this function we see that $v_1(i.e, \frac{d\theta_1}{dt})$ will be maximum at t =1 and decreases drastically when we move away from t =1 in both sides.

So equation (22) is modified and becomes,

$$v_1 = u[-\int_0^t G_1 v_1 dt](\theta_2 - \theta_1).\frac{2}{2+k(t-1)^2} \tag{24}$$

For ease of calculation we denote the term within the unit step function as $M = -\int_0^t G_1 v_1 dt$ to obtain,

$$v_1 = u(M)(\theta_2 - \theta_1).\frac{2}{2+k(t-1)^2} \tag{25}$$

Since $u(M) = \underset{\alpha\to\infty}{\text{Lt}} \frac{1}{1+e^{-\alpha M}}$

We take a smaller value of α for getting into the mathematical analysis (say $\alpha = 10$). Since, we have the region, under consideration with very low gradient and the velocity of the particle is low, (so product $G_1 v_1$ is also small enough), and the time interval of the integration is not too large (as the time domain under consideration is not large), so we can write, by expanding the exponential part and neglecting the higher order terms:

$$u(M) = \frac{1}{1+(1-\alpha M)}$$

$$= \frac{1}{2(1-\alpha M/2)}$$

Putting this expression in equation (25) we get,

$$v_1 = \frac{1}{2(1-\alpha M/2)}(\theta_2 - \theta_1)\frac{2}{2(1+(k/2)(t-1)^2)}$$

$$\Rightarrow \frac{v_1}{\theta_2 - \theta_1}((1+(k/2)(t-1)^2) = \frac{1}{2}(1+\frac{\alpha M}{2}) \tag{26}$$

[$\because |\theta_2 - \theta_1| \to 0$ but $|\theta_2 - \theta_1| \neq 0$

also $\because \left|\frac{\alpha M}{2}\right|$ <<1, neglecting higher order terms, $(1-\frac{\alpha M}{2})^{-1} \approx (1+\frac{\alpha M}{2})$]

Now the equation given by (26) is true for all values possible values of *t*, so we can differentiate both sides of it with respect to *t* and get,

$$\Rightarrow \frac{(\theta_2-\theta_1)\frac{dv_1}{dt}-v_1(\frac{d\theta_2}{dt}-\frac{d\theta_1}{dt})}{(\theta_2-\theta_1)^2}((1+(k/2)(t-1)^2)+\frac{v_1}{\theta_2-\theta_1}k(t-1)=\frac{1}{4}\frac{d(\alpha M)}{dt} \tag{27}$$

Now, $\frac{d(CM)}{dt}=\frac{d(-\alpha\int_0^t v_1G_1dt)}{dt}=-\alpha v_1G_1$ [By putting the expression for *M* and applying the Leibniz theorem for differentiating integrals]

So from (27), we get,

$$\frac{(\theta_2-\theta_1)\frac{dv_1}{dt}-v_1(\frac{d\theta_2}{dt}-\frac{d\theta_1}{dt})}{(\theta_2-\theta_1)^2}((1+(k/2)(t-1)^2)+\frac{v_1}{\theta_2-\theta_1}k(t-1)=-\frac{1}{4}\alpha v_1G_1$$

Putting $\frac{d\theta_1}{dt}=v_1$ and $\frac{d\theta_2}{dt}=v_2$ after some further manipulations (where we need to cancel out $(\theta_2-\theta_1)$, which we can do as $|\theta_2-\theta_1|\to 0$ towards the end of reproduction but never $|\theta_2-\theta_1|\neq 0$ according to assumption (iii)), we get,

$$\frac{dv_1}{dt}=-\frac{v_1^2}{\theta_2-\theta_1}-v_1[\frac{k(t-1)}{1+(k/2)(t-1)^2}+\frac{\alpha G_1(\theta_2-\theta_1)}{4(1+(k/2)(t-1)^2)}-\frac{v_2}{\theta_2-\theta_1}]$$

$$\Rightarrow \frac{dv_1}{dt}=-Pv_1^2-Qv_1 \tag{28}$$

Where, $P=\frac{1}{\theta_2-\theta_1}$ and $Q=(\frac{k(t-1)}{1+(k/2)(t-1)^2}+\frac{\alpha G_1(\theta_2-\theta_1)}{4(1+(k/2)(t-1)^2)}-\frac{v_2}{\theta_2-\theta_1})$

The above equation is for the first bacterium and similarly we can derive the equation for the second bacterium, which looks like,

$$\frac{dv_2}{dt}=-P^{'}v_2^2-Q^{'}v_2\ , \tag{29}$$

where, $P^{'}=\frac{1}{\theta_1-\theta_2}$ and $Q^{'}=(\frac{k(t-1)}{1+(k/2)(t-1)^2}+\frac{\alpha G_2(\theta_1-\theta_2)}{4(1+(k/2)(t-1)^2)}-\frac{v_1}{\theta_1-\theta_2})$

4.2 Physical Significance

A possible way to visualize the effect of the dynamics presented in equations (28) and (29) is to see how the velocities of the bacteria vary over short time intervals over which the coefficients *P* and *Q* can be assumed to remain fairly constant. The velocity of a bacterium (which is at the better place) has been plotted over five short time intervals in Figure 11(*P* and *Q* are chosen arbitrarily in those intervals). Note that at the time of reproduction (t = 1) the graph is highly steep indicating sharp decrease in velocity.

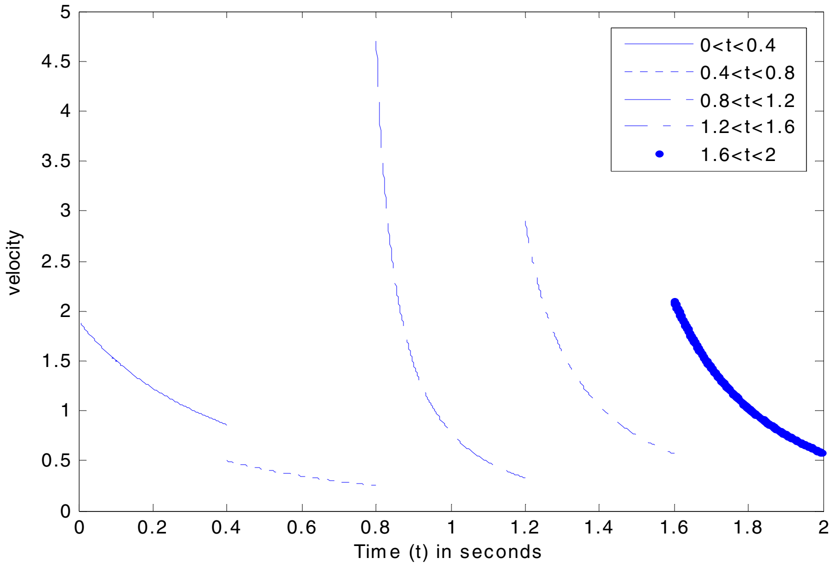


Fig. 11 Piece-wise change in velocity over small time intervals

Now if we study the second term in the expression of Q from equation (28) i.e. the term $\frac{\alpha G_1(\theta_2 - \theta_1)}{4(1+(k/2)(t-1)^2)}$, as $G_1 \rightarrow 0$, $(\theta_2 - \theta_1)$ is also small and α is not taken to be very large. At the denominator also we have got some divisors greater than 1. So the term becomes insignificantly small and all we can neglect it from Q. In equation (29) also we can similarly neglect the term $\frac{\alpha G_2(\theta_1 - \theta_2)}{4(1+(k/2)(t-1)^2)}$ from $Q^{'}$. Again we assume, the velocity of both the particles to be negative for the time being. So we can replace, $v_1 = -|v_1|$ and

$v_2 = -|v_2|$ in Q and Q' in equations (28) and (29). After doing all this simplifications for getting a better mathematical insight, equations (28) and (29) become,

$$\frac{dv_1}{dt} = -Pv_1^2 - Qv_1 \ , \tag{30}$$

where, $P = \dfrac{1}{\theta_2 - \theta_1}$ and $Q = (\dfrac{k(t-1)}{1+(k/2)(t-1)^2} + \dfrac{|v_2|}{\theta_2 - \theta_1})$

$$\frac{dv_2}{dt} = -P'v_2^2 - Q'v_2 \ , \tag{31}$$

where, $P' = \dfrac{1}{\theta_1 - \theta_2}$ and $Q' = (\dfrac{k(t-1)}{1+(k/2)(t-1)^2} + \dfrac{|v_1|}{\theta_1 - \theta_2})$.

Now, for $\theta_2 > \theta_1$ P and Q are both positive. That means the first bacterium slows down very quickly. Whereas the second particle has P' and Q' (assuming the other term independent of $(\theta_1 - \theta_2)$ in Q' is lesser than this) both negative. That means this bacterium accelerates. This acceleration is hopefully towards the first bacterium.

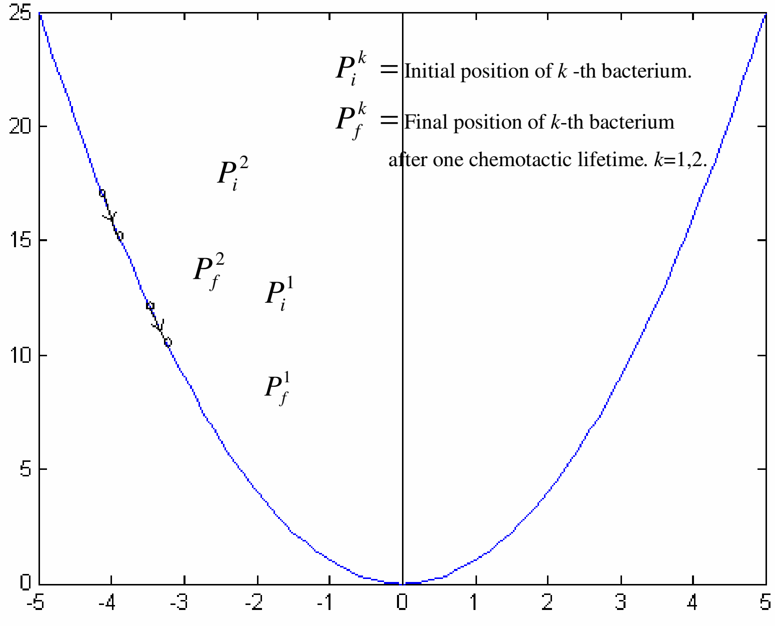


Fig. 12 Initial and final positions of the two bacteria (after one chemotactic lifetime)

Since the rate of change of velocity of bacterium 1 and 2 are dependent on $(\theta_2 - \theta_1)$ and $(\theta_1 - \theta_2)$ respectively, it is evident that the distance between the two bacteria guides their dynamics. If we assume, $\theta_2 > \theta_1$ and they don't traverse too long, the first bacterium is healthier (less accumulated cost) than the second one, when the function is decreasing monotonically in a minimization problem and also the time rate change of first bacterium is less than that of the second (as depicted in Figure 12 clearly, where we take $J(\theta) = \theta^2$).

So at the time of reproduction, in a two bacteria system, the healthier bacterium when senses that it is in a better position compared to its fellow bacterium, it hopes that the optima might be very near so it slows down and its search becomes more fine-tuned. This can be compared with the real bacterium involved in foraging. Whenever it senses that food might be nearby then it obviously slows down and searches that place thoroughly at cost of some time [15 - 17].

The second bacterium moves away from that place with a high acceleration quite naturally getting the information from the first bacterium that the fitter place is away from its present position. In biological system for grouped foraging when one member of the group share information from its neighbors it tries to move towards the best position found out by the neighboring members [15].

Thus we see that reproduction was actually included in BFOA in order to facilitate grouped global search, which is explained from our small analysis.

4.3 Avoiding Premature Convergence

Again if we observe the bacterium at the better position more carefully we will be seeing, that this has a tendency to decelerate at a very high rate and it becomes at rest very quickly. Now when it is near the optima, we can conclude that as $t \to \infty$, $v_{better} \to 0$ (velocity of the better one). Thus as it reaches the optima it stabilize without any further oscillation. Thus reproduction helps the better bacterium to stabilize at the optima.

But the darker side of this fact lies in premature convergence i.e. the best bacterium can converge towards a local optima and the search process gets disturbed. So we understand that at the start of search process reproduction can cause premature convergence but the same can lead to a stable system if applied near the global optima. So we suggest an adaptive scheme related to reproduction operator. The reproduction rate should be made adaptive and it should be increased gradually towards the end of this search process. This has been proved experimentally.

5 Hybridization of BFOA with Other Approaches

We have a handful of optimization algorithms for applying in practical problems but as we know from NFL (No Free Lunch theorem) [18] that no algorithm can perform satisfactorily well over every possible optimization problems. Some algorithms are inspired by natural evolution whereas some are by natural flocking of birds or swarming of bees. Some algorithm can have an extremely good local search behavior while some other can have a good global search property. This may be the reason why hybridization of different algorithms can give better performance as compared to the parent algorithms.

In 2007, Kim *et al.* proposed a hybrid approach involving genetic algorithm (GA) and BFOA for function optimization [19]. The proposed algorithm outperformed both GA and BFOA over a few numerical benchmarks and a practical PID tuner design problem.

Biswas *et al.* coupled BFOA and PSO to develop a new algorithm called BSO (Bacterial Swarm Optimization) [20]. This algorithm provided some very good results when tested over a set of benchmark problems and a difficult engineering problem of spread spectrum radar poly-phase code design. BSO performs local search through the chemotactic movement operation of BFOA whereas a PSO operator accomplishes the global search over the entire search space. In this way it balances between exploration and exploitation, enjoying best of both the worlds. In BSO, after undergoing a chemo-tactic step, a PSO operator also mutates each bacterium. In this phase, the bacterium is stochastically attracted towards the globally best position found so far in the entire population at current time and also towards its previous heading direction. The PSO operator uses only the globally best position found by the entire population to update the velocities of the bacteria and eliminates term involving the personal best position as the local search in different regions of the search space is already taken care of by the chemo-tactic operator of BFOA.

The chemotaxis step of BFOA have been hybridized with another powerful optimization algorithm of current interest called the Differential Evolution (DE) [21] and gave rise to an algorithm known as CDE (Chemotactic Differential Evolution) [22]. Biswas *et al.* proved efficiency of this algorithm too on a set of optimization problems, both numerical benchmark and practical. In this algorithm a bacterium undergoes a differential mutation step just after one chemotaxis step and the rest is kept similar to that of the original BFOA algorithm. Thus each of the bacteria explores the fitness landscape more carefully.

6 Applications of BFOA

Ulagammai *et al.* applied BFOA to train a Wavelet-based Neural Network (WNN) and used the same for identifying the inherent non-linear characteristics of power system loads [23]. In [24], BFOA was used for the dynamical resource allocation in a multiple input/output experimentation platform, which mimics a temperature grid plant and is composed of multiple sensors and actuators organized in zones. Acharya *et al.* proposed a BFOA based Independent Component Analysis (ICA) [25] that aims at finding a linear representation of non-gaussian data so that the components are statistically independent or as independent as possible. The proposed scheme yielded better mean square error performance as compared to a CGAICA (Constrained Genetic Algorithm based ICA). Chatterjee *et al.* reported

Table 1 A Summary of State-of-the-art research works on BFOA

Area of research	Sub-topic	Researchers	References
Hybridization	BFOA-GA, BFOA-PSO, BFOA-DE Hybridization	Dong Hwa Kim, Jae Hoon Cho, Ajith Abraham, Swagatam Das, Arijit Biswas, Sambarta Dasgupta,	[19], [20], [22]
Mathematical Analysis	Chemotaxis, Reproduction, modeling in varying and dynamics environment	Swagatam Das, Sambarta Dasgupta, Arijit Biswas, Ajith Abraham, W. J. Tang, Q. H. Wu, J. R. Saunders	[13], [14], [30], [31]
Modification of BFOA	Adaptive chemotactic step size, modified step size using Hybrid least square-Fuzzy Logic, advanced BFOA using fuzzy logic and clonal selection, BFOA in dynamic environments, BFOA with varying population, cooperative approach to BFOA	Kevin M. Passino, Sambarta Dasgupta, Arijit Biswas, Swagatam Das, Ajith Abraham, Dong Hwa Kim, Jae Hoon Cho, S. Mishra, W. J Tang, Q H Wu, J R Saunders, Carlos Fernandes, Vitorino Ramos, Agostinho C. Rosa, Hanning Chen, Yunlong Zhu, Kunyuan Hu.	[1], [9], [19], [29], [27], [28], [32], [37]
Application in the field of electrical engineering and Control	Optimization of real power loss and voltage stability and distribution static compensator, Harmonic estimation, Active power filter for load compensation, dynamic resource allocation in multi-zone temperature experimentation, PID controller design,	S. Mishra, M. Tripathi, C.N. Bhende, L.L Lai, Mario A. Munoz, Jesus A. Lopez, Eduardo Caicedo, Dong Hwa Kim	[27], [28], [32], [33], [34]
Filtering Problem	Application of BFOA to extended Kalman filter based simultaneous localization and mapping problems	Amitava Chatterjee, Fumitoshi Matsuno	[26]
Learning and Neural network problems	Wavelet neural network training, Optimal learning of Neuro fuzzy structure, Parameter optimization of extreme learning machine	M. Ulagammai, P. Venkatesh, P.S. Kannan, Narayan Prasad Padhy, D.H Kim, Jae-Hoon Cho, Dae-Jong Lee	[23], [35],
Pattern Recognition	Circle detection with Adaptive BFOA, Independent component analysis	Sambarta Dasgupta, Arijit Biswas, Swagatam Das, Ajith Abraham, D P Acharya, G Panda, S Mishra, Y V S Laxmi	[36], [25]
Scheduling Problem	BFOA for job shop scheduling	Chunguo Wu, Na Zhang, Jingqing Jiang, Jinhui Yang and Yanchun Liang	[38]

an interesting application of BFOA in [26] to improve the quality of solutions for the extended Kalman Filters (EKFs), such that the EKFs can offer to solve simultaneous localization and mapping (SLAM) problems for mobile robots and autonomous vehicles.

Tripathy and Mishra proposed an improved BFO algorithm for simultaneous optimization of the real power losses and Voltage Stability Limit (VSL) of a mesh power network [27]**.** In their modified algorithm, firstly, instead of the average value, the minimum value of all the chemotactic cost functions is retained for deciding the bacterium's health. This speeds up the convergence, because in the average scheme described by Passino [1], it may not retain the fittest bacterium for the subsequent generation. Secondly for swarming, the distances of all the bacteria in a new chemotactic stage are evaluated from the globally optimal bacterium to these points and not the distances of each bacterium from the rest of the others, as suggested by Passino [1]. Simulation results indicated the superiority of the proposed approach over classical BFOA for the multi-objective optimization problem involving the UPFC (Unified Power Flow Controller) location, its series injected voltage, and the transformer tap positions as the variables. Mishra and Bhende used the modified BFOA to optimize the coefficients of Proportional plus Integral (PI) controllers for active power filters [28]. The proposed algorithm was found to outperform a conventional GA with respect to the convergence speed.

Mishra, in [29], proposed a Takagi-Sugeno type fuzzy inference scheme for selecting the optimal chemotactic step-size in BFOA. The resulting algorithm, referred to as Fuzzy Bacterial Foraging (FBF), was shown to outperform both classical BFOA and a Genetic Algorithm (GA) when applied to the harmonic estimation problem. However, the performance of the FBF crucially depends on the choice of the membership function and the fuzzy rule parameters [29] and there is no systematic method (other than trial and error) to determine these parameters for a given problem. Hence FBF, as presented in [29], may not be suitable for optimizing any benchmark function in general. In Table 1 we summarize the current researches on different aspects and applications of BFOA.

7 Conclusions

Search and optimization problems are ubiquitous through the various realms of science and engineering. This chapter has provided a comprehensive overview of one promising real-parameter optimization algorithm called the Bacterial Foraging Optimization Algorithm (BFOA). BFOA is currently gaining popularity due to its efficacy over other swarm and evolutionary computing algorithms in solving engineering optimization problems. It mimics the individual as well as grouped foraging behavior of *E.coli* bacteria that live in our intestine.

The chapter first outlines the classical BFOA in sufficient details. It then develops a simple mathematical model of the simulated chemotaxis operation of BFOA. With the help of this model it analyses the chemotactic dynamics of a single bacterium moving over a one-dimensional fitness landscape. The analysis indicates that the chemotactic dynamics has some striking similarity with the classical gradient descent search although the former never uses an analytic expression of the derivative of the objective function. A problem of oscillations near the optimum is identified from the presented analysis and two adaptation rules for the chemotactic step-height have been proposed to promote the quick convergence of the algorithm near the global optimum of the search space. The chapter also provides an analysis of the reproduction step of BFOA for a two-bacterium system. The analysis reveals how the dynamics of reproduction helps in avoiding premature convergence.

In recent times, a symbiosis of swarm intelligence with other computational intelligence algorithms has opened up new avenues for the next generation computing systems. The chapter presents an account of the research efforts aiming at hybridizing BFOA with other popular optimization techniques like PSO, DE, and GA for improved global search and optimization. It also discusses the significant applications of BFOA in diverse domains of science and engineering. The content of the chapter reveals that engineering search and optimization problems including those from the fields of pattern recognition, bioinformatics, and machine intelligence will find new dimensions in the light of swarm intelligence techniques like BFOA.

References

[1] Passino, K.M.: Biomimicry of Bacterial Foraging for Distributed Optimization and Control. IEEE Control Systems Magazine, 52–67 (2002)

[2] Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Harbor (1975)

[3] Fogel, L.J., Owens, A.J., Walsh, M.J.: Artificial Intelligence through Simulated Evolution. John Wiley, Chichester (1966)

[4] Rechenberg, I.: Evolutionsstrategie 1994. Frommann-Holzboog, Stuttgart (1994)

[5] Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks, pp. 1942–1948 (1995)

[6] Dorigo, M., Gambardella, L.M.: Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. IEEE Transactions on Evolutionary Computation 1(1), 53–66 (1997)

[7] Berg, H., Brown, D.: Chemotaxis in escherichia coli analysed by three-dimensional tracking. Nature 239, 500–504 (1972)

[8] Berg, H.: Random Walks in Biology. Princeton Univ. Press, Princeton (1993)

[9] Liu, Y., Passino, K.M.: Biomimicry of Social Foraging Bacteria for Distributed Optimization: Models, Principles, and Emergent Behaviors. Journal of Optimization Theory And Applications 115(3), 603–628 (2002)

[10] Abramowitz, M., Stegun, I.A. (eds.): Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables. Dover, New York (1972)

[11] Bracewell, R.: Heaviside's Unit Step Function, H(x), The Fourier Transform and Its Applications, 3rd edn., pp. 57–61. McGraw-Hill, New York (1999)

[12] Snyman, J.A.: Practical Mathematical Optimization: An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms. Springer Publishing, Heidelberg (2005)

[13] Dasgupta, S., Das, S., Abraham, A., Biswas, A.: Adaptive Computational Chemotaxis in Bacterial Foraging Optimization: An Analysis. IEEE Transactions on Evolutionary Computation (in press, 2009)

[14] Abraham, A., Biswas, A., Dasgupta, S., Das, S.: Anaysis of Reproduction Operator in Bacterial Foraging Optimization. In: IEEE Congress on Evolutionary Computation CEC 2008, IEEE World Congress on Computational Intelligence, WCCI 2008, pp. 1476–1483. IEEE Press, USA (2008)

[15] Murray, J.D.: Mathematical Biology. Springer, New York (1989)

[16] Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence: From Natural to Artificial Systems. Oxford Univ. Press, New York (1999)

[17] Okubo, A.: Dynamical aspects of animal grouping: swarms, schools, flocks, and herds. Advanced Biophysics 22, 1–94 (1986)

[18] Wolpert, D.H., Macready, W.G.: No Free Lunch Theorems for Optimization. IEEE Transactions on Evolutionary Computation 1(1), 67–82 (1997)

[19] Kim, D.H., Abraham, A., Cho, J.H.: A hybrid genetic algorithm and bacterial foraging approach for global optimization. Information Sciences 177(18), 3918–3937 (2007)

[20] Biswas, A., Dasgupta, S., Das, S., Abraham, A.: Synergy of PSO and Bacterial Foraging Optimization: A Comparative Study on Numerical Benchmarks. In: Corchado, E., et al. (eds.) Second International Symposium on Hybrid Artificial Intelligent Systems (HAIS 2007), Innovations in Hybrid Intelligent Systems, ASC. Advances in Soft computing Series, vol. 44, pp. 255–263. Springer, Germany (2007)

[21] Storn, R., Price, K.: Differential evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. Journal of Global Optimization 11(4), 341–359 (1997)

[22] Biswas, A., Dasgupta, S., Das, S., Abraham, A.: A Synergy of Differential Evolution and Bacterial Foraging Algorithm for Global Optimization. Neural Network World 17(6), 607–626 (2007)

[23] Ulagammai, L., Vankatesh, P., Kannan, P.S., Padhy, N.P.: Application of Bacteria Foraging Technique Trained and Artificial and Wavelet Neural Networks in Load Forecasting. Neurocomputing, 2659–2667 (2007)

[24] Munoz, M.A., Lopez, J.A., Caicedo, E.: Bacteria Foraging Optimization for Dynamical resource Allocation in a Multizone temperature Experimentation Platform. In: Anal. and Des. of Intel. Sys. using SC Tech., ASC, vol. 41, pp. 427–435 (2007)

[25] Acharya, D.P., Panda, G., Mishra, S., Lakhshmi, Y.V.S.: Bacteria Foaging Based Independent Component Analysis. In: International Conference on Computational Intelligence and Multimedia Applications. IEEE Press, Los Alamitos (2007)

[26] Chatterjee, A., Matsuno, F.: Bacteria Foraging Techniques for Solving EKF-Based SLAM Problems. In: Proc. International Control Conference (Control 2006), Glasgow, UK, August 30- September 01 (2006)

[27] Tripathy, M., Mishra, S.: Bacteria Foraging-Based to Optimize Both Real Power Loss and Voltage Stability Limit. IEEE Transactions on Power Systems 22(1), 240–248 (2007)

[28] Mishra, S., Bhende, C.N.: Bacterial Foraging Technique-Based Optimized Active Power Filter for Load Compensation. IEEE Transactions on Power Delivery 22(1), 457–465 (2007)

[29] Mishra, S.: A hybrid least square-fuzzy bacterial foraging strategy for harmonic estimation. IEEE Trans. on Evolutionary Computation 9(1), 61–73 (2005)

[30] Tang, W.J., Wu, Q.H., Saunders, J.R.: A Novel Model for Bacteria Foraging in Varying Environments. In: Gavrilova, M.L., Gervasi, O., Kumar, V., Tan, C.J.K., Taniar, D., Laganá, A., Mun, Y., Choo, H. (eds.) ICCSA 2006. LNCS, vol. 3980, pp. 556–565. Springer, Heidelberg (2006)

[31] Biswas, A., Das, S., Dasgupta, S., Abraham, A.: Stability Analysis of the Reproduction Operator in Bacterial foraging Optimization. In: IEEE/ACM International Conference on Soft Computing as Transdisciplinary Science and Technology (CSTST 2008), Paris, France, pp. 568–575. ACM Press, New York (2008)

[32] Fernandes, C., Ramos, V., Agostinho, C.: Varying the Population Size of Artificial Foraging Swarms on Time Varying Landscapes. In: Duch, W., Kacprzyk, J., Oja, E., Zadrożny, S. (eds.) ICANN 2005. LNCS, vol. 3696, pp. 311–316. Springer, Heidelberg (2005)

[33] Tripathy, M., Mishra, S., Lai, L.L., Zhang, Q.P.: Transmission Loss Reduction Based on FACTS and Bacteria Foraging Algorithm. In: PPSN, pp. 222–231 (2006)

[34] Mishra, S., Bhende, C.N.: Bacterial Foraging Technique-Based Optimized Active Power Filter for Load Compensation. IEEE Transactions on Power Delivery 22(1), 457–465 (2007)

[35] Kim, D.H., Cho, C.H.: Bacterial Foraging Based Neural Network Fuzzy Learning. In: IICAI 2005, pp. 2030–2036 (2005)

[36] Dasgupta, S., Biswas, A., Das, S., Abraham, A.: Automatic Circle Detection on Images with an Adaptive Bacterial Foraging Algorithm. In: 2008 Genetic and Evolutionary Computation Conference, GECCO 2008, pp. 1695–1696. ACM Press, New York (2008)

[37] Chen, H., Zhu, Y., Hu, K., He, X., Niu, B.: Cooperative Approaches to Bacterial Foraging Optimization. In: ICIC (2), pp. 541–548 (2008)

[38] Wu, C., Zhang, N., Jiang, J., Yang, J., Liang, Y.: Improved Bacterial Foraging Algorithms and Their Applications to Job Shop Scheduling Problems. In: Beliczynski, B., Dzielinski, A., Iwanowski, M., Ribeiro, B. (eds.) ICANNGA 2007. LNCS, vol. 4431, pp. 562–569. Springer, Heidelberg (2007)