Applied Mathematics and Computation 215 (2010) 3343–3355

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

Analysis of the reproduction operator in an artificial bacterial foraging system

Arijit Biswas [a], Swagatam Das [a], Ajith Abraham [b,*], Sambarta Dasgupta [a]

[a] Department of Electronics and Telecommunication Engg., Jadavpur University, Kolkata, India
[b] Machine Intelligence Research Labs (MIR Labs), Scientific Network for Innovation and Research Excellence, P.O. Box 2259, Auburn, WA 98071, USA

ARTICLE INFO

Keywords:
Biological systems
Bacterial foraging
Reproduction
Selection
Global optimization
Computational chemotaxis

ABSTRACT

In his seminal paper published in 2002, Passino pointed out how individual and groups of bacteria forage for nutrients and how to model it as a distributed optimization process, which he called the Bacterial Foraging Optimization Algorithm (BFOA). One of the major driving forces of BFOA is the reproduction phenomenon of virtual bacteria each of which models a trial solution of the optimization problem. During reproduction, the least healthier bacteria (with a lower accumulated value of the objective function in one chemotactic lifetime) die and the other healthier bacteria each split into two, which then starts exploring the search place from the same location. This keeps the population size constant in BFOA. The phenomenon has a direct analogy with the selection mechanism of classical evolutionary algorithms. In this letter we provide a simple mathematical analysis of the effect of reproduction on bacterial dynamics. Our analysis reveals that the reproduction event contributes to the quick convergence of the bacterial population near optima.

1. Introduction

To tackle several complex search problems of real world, scientists have been looking into the nature for years-both as model and as metaphor-for inspiration. Optimization is at the heart of many natural processes like Darwinian evolution, group behavior of social insects and the foraging strategy of other microbial creatures. Natural selection tends to eliminate species with poor foraging strategies and favor the propagation of genes of species with successful foraging behavior, as they are more likely to enjoy reproductive success.

Since a foraging organism or animal takes necessary action to maximize the energy utilized per unit time spent for foraging, considering all the constraints presented by its own physiology such as sensing and cognitive capabilities, environment (e.g. density of prey, risks from predators, physical characteristics of the search space), the natural foraging strategy can lead to optimization and essentially this idea can be applied to real-world optimization problems. Based on this conception, Passino proposed an optimization technique known as Bacterial Foraging Optimization Algorithm (BFOA) [1–4] . Until date, the algorithm has successfully been applied to real world problems like optimal controller design [1,2], harmonic estimation [5], transmission loss reduction [6], pattern recognition [7], controller synthesis for active power filters [8], and power system optimization [9].

BFOA is a newly added member in the coveted realm of Swarm Intelligence [10–15], which also includes powerful optimization techniques like the Particle Swarm Optimization (PSO) [11,16] and Ant Colony Optimization (ACO) [17]. On the algorithmic front, several researchers extended the basic BFOA to deal with complex and multi-modal fitness landscapes,

* Corresponding author.
E-mail address: ajith.abraham@ieee.org (A. Abraham).

0096-3003/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.amc.2009.10.023

dynamical environments and to obtain efficient convergence behavior [18–23]. BFOA has also been hybridized with a few other state-of-the-art evolutionary computing techniques [24–26] in order to achieve robust and efficient search performances. An interesting characteristic feature of BFOA is that it has its own local search mechanism through the computational chemotaxis step and reproduction with elimination-dispersion helps in global search. Over certain real-world optimization problems, BFOA has been reported to outperform many powerful optimization algorithms like GA, PSO, etc. in terms of convergence speed and final accuracy, for example, see [5,8,9,25,26]. As pointed out by Das et al [27], unlike PSO and Differential Evolution (DE), the uniqueness of the stability criteria of BFOA remains in the fact that in order to ensure stability of the chemotactic dynamics in BFOA, the step-size parameter must be adjusted (i.e. made adaptive) according to the current location of the bacterium and its current fitness. The efficiency of the algorithm in solving real parameter optimization problems has made it a potential optimization algorithm, worth investing research time these days.

In [28], Dasgupta et al. have provided a mathematical model of the simulated chemotaxis operation of a simple bacterial system from the viewpoint of the classical gradient descent search [29]. Their analysis points out that the chemotaxis employed by classical BFOA, usually results in sustained oscillation, especially on flat fitness landscapes, when a bacterium cell is close to the optima. To accelerate the convergence speed of the group of bacteria near global optima, two simple schemes for adapting the chemotactic step-height were also proposed. This paper provides a simple mathematical analysis of another important step in BFOA, called reproduction. During reproduction, the bacterial population is at first sorted in order of ascending accumulated cost (value of the objective function to be optimized), then the worse half of the population containing least healthy bacteria is liquidated while all the members of the better half is split into two bacteria which start exploring the search space from the same location on the fitness landscape. As pointed out by Passino, this phenomenon finds analogy with the elitist-selection mechanism of the classical evolutionary algorithms (EA) [1,2,30]. Bacteria in the most favorable environment (i.e., near an optima) gain a selective advantage for reproduction through the cumulative cost. We focus our attention on a simple two-bacterial system working over a one-dimensional fitness landscape and verify the role of reproduction in the convergence behavior of the said population near global optima. Although the analysis may appear to have a limited scope, note that this article is the first of its kind and the issues of multi-bacterial population over a multi-dimensional fitness landscape are topics of further research. Here our primary objective is to provide important insight into the operational mechanism of the artificial bacterial foraging system, acting as a global function optimizer.

2. The bacterial foraging optimization algorithm

The bacterial foraging system consists of four principal mechanisms, namely chemotaxis, swarming, reproduction, and elimination-dispersal [1]. Below we briefly describe each of these processes and finally provide a pseudo-code of the complete algorithm.

(i) *Chemotaxis*: This process simulates the movement of an *E. coli* cell through swimming and tumbling via flagella. Biologically an *Escherichia 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 ith bacterium at jth chemotactic, kth reproductive and lth 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 *Salmonella typhimurium*, where intricate and stable spatio-temporal patterns (swarms) are formed in semisolid nutrient medium [31,32]. 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.

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

where $J_{cc}(\theta, P(j,k,l))$ is the objective function value to be added to the actual objective function (to be minimized) to present a time varying objective function, 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. $d_{\text{attractant}}$, $w_{\text{attractant}}$, $h_{\text{repellant}}$, $w_{\text{repellant}}$ are different coefficients that should be chosen properly.

(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 detailed pseudo-code of the complete algorithm is given below.

The BFOA Algorithm 1
Parameters:

Step 1 Initialize parameters p, S, N_c, N_s, N_{re}, N_{ed}, P_{ed}, $C(i)$ ($i = 1,2 \ldots S$), θ^i.
where,

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

Algorithm

Step 2 Elimination-dispersal loop: $l = l + 1$.

Step 3 Reproduction loop: $k = k + 1$.

Step 4 Chemotaxis loop: $j = j + 1$.

[a] For $i = 1, 2, \ldots, S$ take a chemotactic step for bacterium i as follows.

[b] Compute fitness function, $J(i, j, k, l)$. Let, $J(i, j, k, l) = J(i, j, k, l) + J_{cc}(\theta^i(j, k, l), P(j, k, l))$ (i.e. add on the cell-to cell attractant–repellant profile to simulate the swarming behavior), where, $J_{cc}$is defined in (2).

[c] Let $J_{last} = J(i, j, k, l)$ to save this value since we may find a better cost via a run.

[d] Tumble: generate a random vector $\Delta(i) \in \Re^p$ with each element $\Delta_m(i)$, $m = 1,2,\ldots,p$, a random number on $[-1, 1]$.

[e] Move: Let

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

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 $i = 1,2,\ldots,S$, 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 $i = 1,2 \ldots ,S$ with probability P_{ed}, eliminate and disperse each bacterium (this keeps the number of bacteria in the population constant). To do this, if a bacterium is eliminated, simply disperse another one to a random location on the optimization domain. If $l < N_{ed}$, then go to step 2; otherwise end.

3. Analysis of the reproduction step in BFOA

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 tth instant its position is given by $\theta(t)$. Below we list a few simplifying assumptions that were considered for the sake of gaining mathematical insight.

Assumptions 1

(i) The objective function $J(\theta)$ 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. 1). At the start of reproduction $\theta_1$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.

3.1. 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

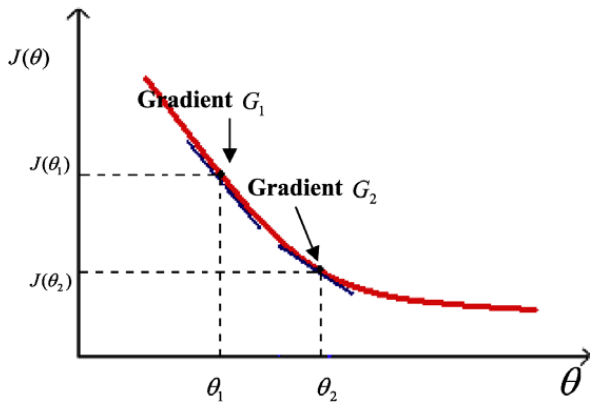

Fig. 1. A two-bacterium system on arbitrary fitness landscape.

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 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 Fig. 1.

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 $u(x)$ (also known as Heaviside step function [33]), which is defined as,

$$\begin{aligned} u(x) &= 1; \quad \text{if } x > 0 \\ &= 0; \quad \text{if } x < 0 \end{aligned} \tag{4}$$

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 $\Delta t(\Delta t \to 0)$ 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,

$$\begin{aligned} &\frac{\Delta\theta_1}{\Delta t} \propto (\theta_2 - \theta_1) \\ &\Rightarrow \frac{\Delta\theta_1}{\Delta t} = \bar{k}(\theta_2 - \theta_1) \quad [\text{where, } \bar{k} \text{ is the proportionality constant}] \\ &\Rightarrow \frac{\Delta\theta_1}{\Delta t} = 1 \cdot (\theta_2 - \theta_1) = (\theta_2 - \theta_1) \end{aligned} \tag{5}$$

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. How the first bacterium, a position situated at infinitesimal distance from it and the second bacterium (at different time instants) will look like in a single dimensional space is clearly depicted in Fig. 2. 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:

$$\frac{\Delta\theta_1}{\Delta t} = u\left[\int_0^t J(\theta_1)dt - \int_0^t J(\theta_1 + \Delta\theta_1)dt\right] \cdot (\theta_2 - \theta_1) \tag{6}$$

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

$$\frac{\Delta\theta_2}{\Delta t} = u\left[\int_0^t J(\theta_2)dt - \int_0^t J(\theta_2 + \Delta\theta_2)dt\right] \cdot (\theta_1 - \theta_2) \tag{7}$$

In Eq. (6), $\int_0^t J(\theta_1)dt$ 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.

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 $\int_0^t J(\theta_1 + \Delta\theta_1)$ is less than $\int_0^t J(\theta_1)$ 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 can rewrite Eq. (6) corresponding to bacterium 1 as,

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

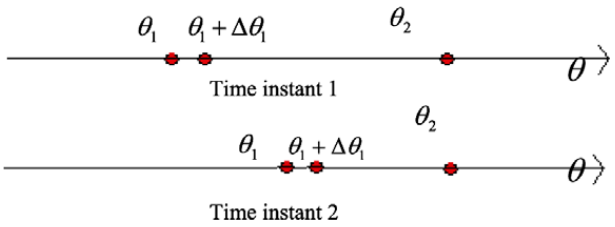

Fig. 2. Change of position of the bacteria during reproduction.

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

$$\Rightarrow \lim_{\substack{\Delta t\to 0\\ \Delta\theta_1\to 0}} \frac{\Delta\theta_1}{\Delta t} = \lim_{\substack{\Delta t\to 0\\ \Delta\theta_1\to 0}} u\left[-\int_0^t \frac{J(\theta_1+\Delta\theta_1)-J(\theta_1)}{\Delta t}dt\right]\cdot(\theta_2-\theta_1)$$

$$\Rightarrow \lim_{\substack{\Delta t\to 0\\ \Delta\theta_1\to 0}} \frac{\Delta\theta_1}{\Delta t} = \lim_{\substack{\Delta t\to 0\\ \Delta\theta_1\to 0}} u\left[-\int_0^t \frac{J(\theta_1+\Delta\theta_1)-J(\theta_1)}{\Delta\theta_1}\frac{\Delta\theta_1}{\Delta t}dt\right]\cdot(\theta_2-\theta_1)$$

Again, $J(x)$ is assumed to be continuous and differentiable. $\lim_{\Delta\theta\to 0}\frac{J(\theta_1+\Delta\theta_1)-J(\theta_1)}{\Delta\theta_1}$ 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,

$$\begin{aligned}&\Rightarrow \frac{d\theta_1}{dt} = u\left[-\int_0^t\left(\frac{dJ}{d\theta_1}\frac{d\theta_1}{dt}\right)dt\right]\cdot(\theta_2-\theta_1)\left[\text{where } \frac{d\theta_1}{dt} \text{ is the instantaneous velocity of the first bacterium}\right]\\ &\Rightarrow v_1 = u\left[-\int_0^t G_1 v_1 dt\right]\cdot(\theta_2-\theta_1) \quad [\text{where } v_1 = \frac{d\theta_1}{dt} \text{ and } G_1 \text{ is the gradient of } J \text{ at } \theta = \theta_1].\end{aligned} \tag{8}$$

Now in Eq. (6) 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 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 Eq. (6). 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 $\phi(x) = \frac{1}{1+e^{-kx}}$. We note that,

$$u(x) = \lim_{k\to\infty}\phi(x) = \lim_{k\to\infty}\frac{1}{1+e^{-kx}} \tag{9}$$

Fig. 3 illustrates how the logistic function may be used to approximate the unit step function used for decision making in reproduction.

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 $|k(t-1)^2| \ll 1$ 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

$$g(t) = \frac{2}{2+k(t-1)^2} \quad (\text{We can take } k = 5)$$

which is not an impulsive function just at t=1 rather a continuous function as shown in Fig. 4. Higher value of k will produce more effective result. Due to the presence of this function we see that $v_1\left(\text{i.e.}, \frac{d\theta_1}{dt}\right)$ will be maximum at $t = 1$ and decreases drastically when we move away from $t = 1$ in both sides.

So Eq. (8) is modified and becomes,

$$v_1 = u\left[-\int_0^t G_1 v_1 dt\right](\theta_2-\theta_1)\cdot\frac{2}{2+k(t-1)^2}. \tag{10}$$

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,

$$v_1 = u(M)(\theta_2-\theta_1)\cdot\frac{2}{2+k(t-1)^2}. \tag{11}$$

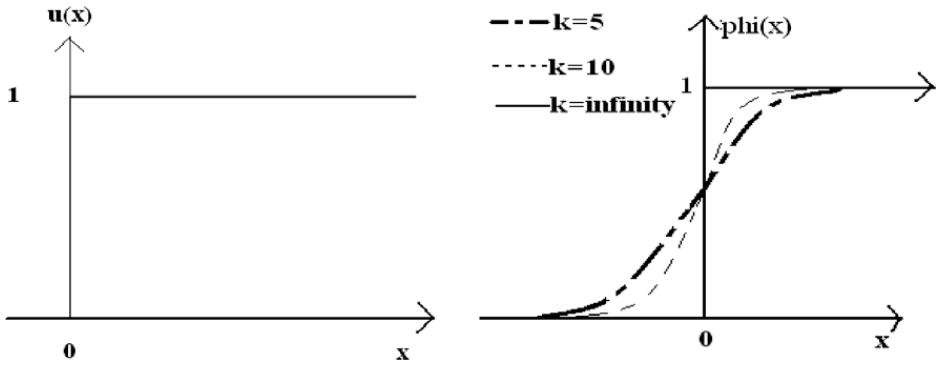

Fig. 3. The unit step and the logistic functions.

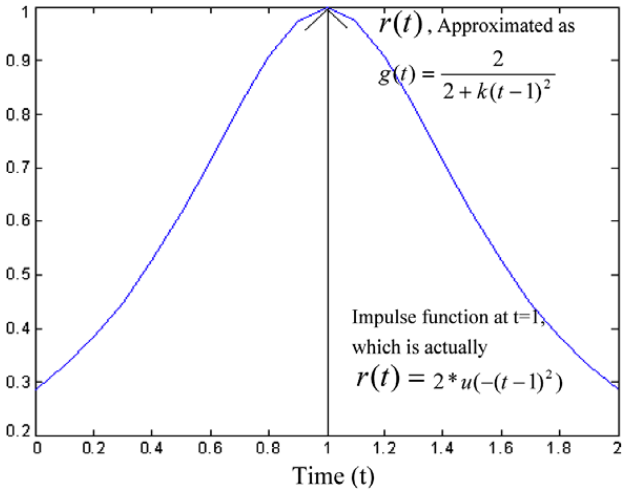

Fig. 4. Functions $r(t)$ and $g(t)$.

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 Eq. (11) we get,

$$v_1 = \frac{1}{2(1-\alpha M/2)}(\theta_2-\theta_1)\frac{2}{2(1+(k/2)(t-1)^2)} \Rightarrow \frac{v_1}{\theta_2-\theta_1}(1+(k/2)(t-1)^2) = \frac{1}{2}\left(1+\frac{\alpha M}{2}\right) \tag{12}$$

[$\because |\theta_2-\theta_1| \to 0$ but $|\theta_2-\theta_1| \neq 0$ also $\because |\frac{\alpha M}{2}| \ll 1$, neglecting higher order terms, $(1-\frac{\alpha M}{2})^{-1} \approx (1+\frac{\alpha M}{2})$]

Now the equation given by (12) is true for all values possible values of t, so we can differentiate both sides of it with respect to t and get,

$$\Rightarrow \frac{(\theta_2-\theta_1)\frac{dv_1}{dt} - v_1\left(\frac{d\theta_2}{dt}-\frac{d\theta_1}{dt}\right)}{(\theta_2-\theta_1)^2}(1+(k/2)(t-1)^2) + \frac{v_1}{\theta_2-\theta_1}k(t-1) = \frac{1}{4}\frac{d(\alpha M)}{dt} \tag{13}$$

Now, $\frac{d(CM)}{dt} = \frac{d(-\alpha\int_0^t v_1 G_1 dt)}{dt} = -\alpha v_1 G_1$ [By putting the expression for M and applying the Leibniz theorem for differentiating integrals]

So from (8), we get,

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

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,

$$\frac{dv_1}{dt} = -\frac{v_1^2}{\theta_2-\theta_1} - v_1\left[\frac{k(t-1)}{1+(k/2)(t-1)^2} + \frac{\alpha G_1(\theta_2-\theta_1)}{4(1+(k/2)(t-1)^2)} - \frac{v_2}{\theta_2-\theta_1}\right] \Rightarrow \frac{dv_1}{dt} = -Pv_1^2 - Qv_1 \tag{14}$$

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

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

$$\frac{dv_2}{dt} = -P'v_2^2 - Q'v_2 \tag{15}$$

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

3.2. Physical significance

A possible way to visualize the effect of the dynamics presented in Eqs. (14) and (15) 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

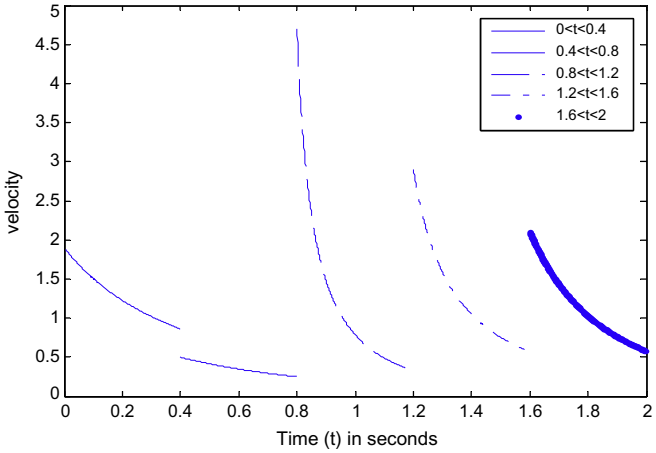

Fig. 5. Piece-wise change in velocity over small time intervals.

velocity of a bacterium (which is at the better place) has been plotted over five short time intervals in Fig. 5 (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.

Now if we study the second term in the expression of Q from Eq. (14) i.e. the term $\frac{\alpha G_1(\theta_2-\theta_1)}{4(1+(k/2)(t-1)^2)}$, as $G_1 \to 0$, $(\theta_2 - \theta_1)$ 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 Eq. (15) 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 Eqs. (14) and (15). After doing all this simplifications for getting a better mathematical insight, Eqs. (11) and (12) become,

$$\frac{dv_1}{dt} = -Pv_1^2 - Qv_1 \tag{16}$$

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

$$\frac{dv_2}{dt} = -P'v_2^2 - Q'v_2 \tag{17}$$

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

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.

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 do not 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 Fig. 6 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 [34–36].

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 [35,36]. Thus we see that reproduction was actually included in BFOA in order to facilitate grouped global search, which is explained from our small analysis.

3.3. 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 conver-

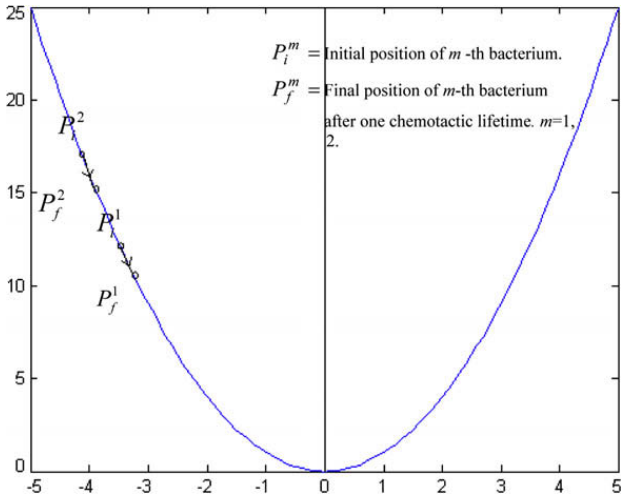

Fig. 6. Initial and final positions of the two bacteria (after one chemotactic lifetime).

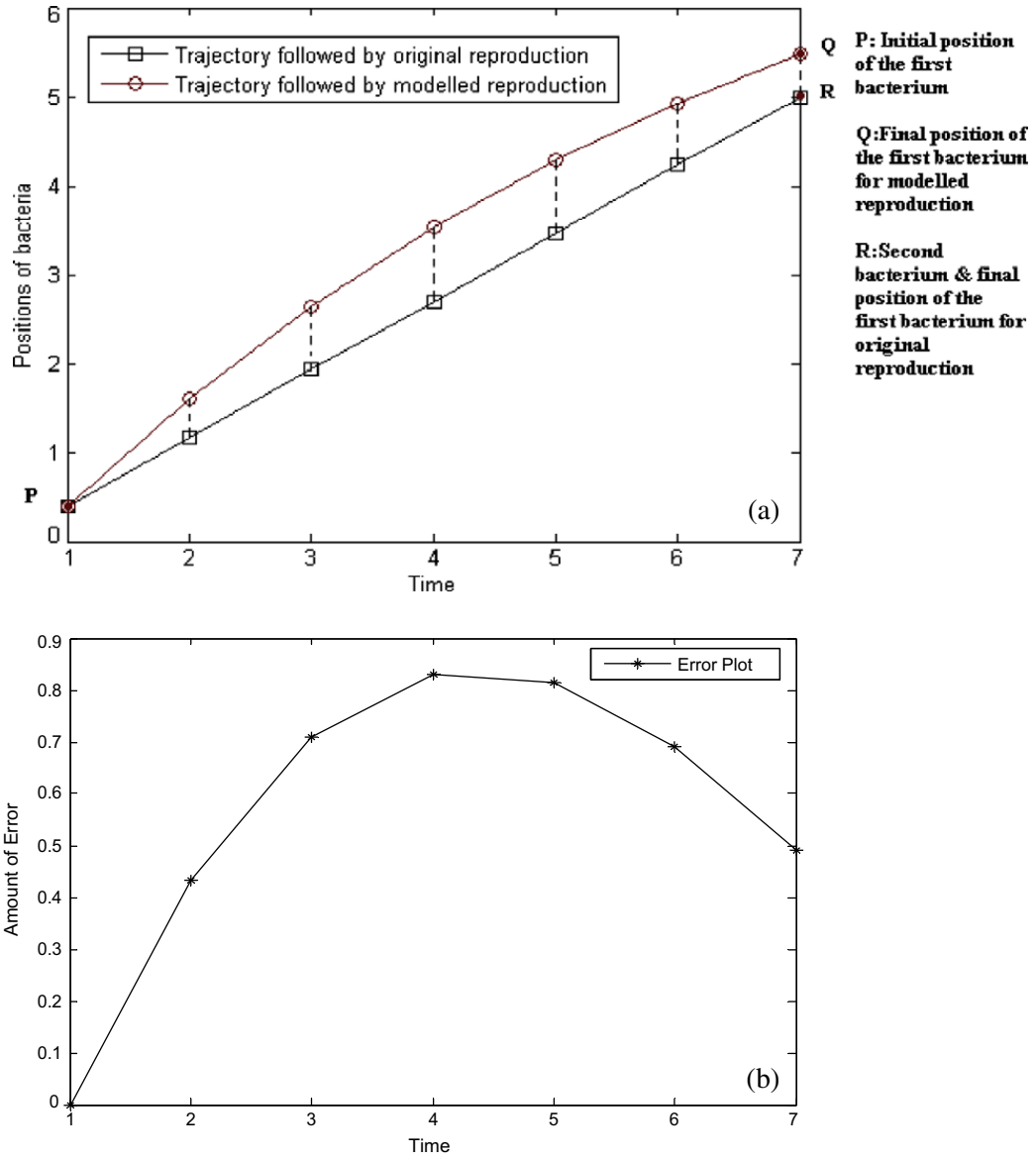

Fig. 7. (a) Original reproduction and modeled reproduction; (b) error in our model.

Table 1
Description of the benchmark functions used.

Function	Mathematical representation	Range of search	Optima
Sphere model	$f_1(\vec{x}) = \sum_{i=1}^{p} x_i^2$	$(-100, 100)^p$	$f_1(\vec{0}) = 0$
Schwefel's problem 2.22	$f_2(\vec{x}) = \sum_{i=1}^{p} \lvert x_i \rvert + \prod_{i=1}^{p} \lvert x_i \rvert$	$(-10, 10)^p$	$f_2(\vec{0}) = 0$
Rosenbrock's	$f_3(\vec{x}) = \sum_{i=1}^{p-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$(-100, 100)^p$	$f_3(\vec{1}) = 0$
Griewank	$f_4(\vec{x}) = \frac{1}{4000} \sum_{i=1}^{p} x_i^2 - \prod_{i=1}^{p} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$(-600, 600)^p$	$f_4(\vec{0}) = 0$
Ackley	$f_5(\vec{x}) = -20 \exp\left(-0.2\sqrt{(\frac{1}{p}\sum_{i=1}^{p} x_i^2)} - \exp(\frac{1}{p}\sum_{i=1}^{p} \cos 2\pi x_i) + 20 + e\right)$	$(-32, 32)^p$	$f_5(\vec{0}) = 0$
Rastrigin	$f_6(\vec{x}) = \sum_{i=1}^{p} [x_i^2 - 10\cos(2\pi x_i) + 10]$	$(-10, 10)^p$	$f_6(\vec{0}) = 0$
Six-Hump Camel-Back Function	$f_7(\vec{x}) = 4x_0^2 - 2.1x_0^4 + \frac{1}{3}x_0^6 + x_0x_1 - 4x_1^2 + 4x_1^4$	$(-5, 5)^2$	$f_7(0.08983, -0.7126) = -1.0316285$
Goldstein-Price Function	$f_8(\vec{x}) = \{1 + (x_0 + x_1 + 1)^2(19 - 14x_0 + 3x_0^2 - 14x_1 - 6x_0x_1 + 3x_1^2)\}\{30 + (2x_0 - 3x_1)^2(18 - 32x_0 + 12x_0^2 + 48x_1 - 36x_0x_1 + 27x_1^2)\}$	$(-2, 2)^2$	$f_8(0, 1) = 3.0000000$
Rotated Hyper-Ellipsoid function	$f_9(\vec{x}) = \sum_{i=1}^{p} \left(\sum_{j=1}^{i} x_j\right)^2$	$(-100, 100)^p$	$f_9(\vec{0}) = 0$
Step	$f_{10}(\vec{x}) = \sum_{i=1}^{p} (\lfloor x_i + 0.5 \rfloor)^2$	$(-100, 100)^p$	$f_{10}(\vec{q}) = 0, -\frac{1}{2} \leqslant q_i < \frac{1}{2}$
Shekel's Foxholes function	$f_{11}(\vec{x}) = \left[\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2} (x_i - a_{ij})^6}\right]^{-1}$ where $(a_{ij}) = \begin{pmatrix} -32 & -16 & 0 & 16 & 32 & -32 \ldots & 0 & 16 & 32 \\ -32 & -32 & -32 & -32 & -32 & -16 \ldots & 32 & 32 & 32 \end{pmatrix}$	(-65.536, 65.536)^2	$f_{11}(-32, -32) = 0.998$

gence 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.

3.4. Parallelism between modeled reproduction and original reproduction

Now for gaining further mathematical insight we do some simplifications over Eq. (16). The effect of reproduction is mostly pronounced around $t = 1$, so $(t - 1) \to 0$. Thus we can neglect the first expression in Q, which contains $(t - 1)$. Again we restrict our analysis to regions only where gradient is very low, i.e. $G_1 \to 0$. So we can also neglect the second expression in Q, which contains G_1. Thus we get a simplified version of the acceleration of the first bacterium as,

$$\frac{dv_1}{dt} = -\frac{v_1^2}{\theta_2 - \theta_1} + \frac{v_1 v_2}{\theta_2 - \theta_1} \Rightarrow \frac{d^2\theta_1}{dt^2} = \frac{\frac{d\theta_1}{dt}(v_2 - \frac{d\theta_1}{dt})}{(\theta_2 - \theta_1)} \tag{18}$$

This represents the modeled reproduction dynamics where the first bacterium possesses an acceleration given by Eq. (18). It starts from its initial position and stops when it comes very closer to the second bacterium and this motion occurs for a finite amount of time. But in original reproduction the first bacterium moves towards the second bacterium instantaneously, which is an impossible to model as a practical dynamics (as that needs an infinite velocity). So we consider the first bacterium to move towards the second one with a uniform speed for the same amount of time as taken in the modeled dynamics. With this we interconnect the original bacterial foraging and the modeled reproductive dynamics. In Fig. 7a we empirically try to provide a scenario where the modeled reproduction dynamics very closely imitate the original reproduction phenomenon. In Fig. 7b the error plot of our analysis is provided.

3.5. The adaptive reproduction BFOA (ARBFOA)

Following the same thought deduced in Section 3.3, we derive a new model for the BFO algorithm, called the Adaptive Reproduction Bacterial Foraging Optimization Algorithm (ARBFOA). In ARBFOA, the frequency of reproduction (determined by the number of chemotactic steps under each reproduction step) is made varying and it is gradually increased towards the end of the process. To do this the N_c (no of chemotactic steps in one reproduction loop) is gradually scaled down using the following formula,

$$N_c(k) = N_c(0) - (k - 1) * \Delta N_c \tag{19}$$

where, $N_c(k)$ is the no of chemotactic steps at the kth reproduction loop.
$N_c(0)$ is the initial no of chemotactic steps (when $k = 1$).
ΔN_c is the change in no of chemotactic steps and is a positive integral quantity.

We take $N_{ed} = 1$, that is elimination dispersal event to take place just once.

We also compare classical BFOA with ARBFOA over a test-suite of eleven well-known benchmark functions [37]. In Table 1, p represents the number of dimensions and we used p=15, 30, 45 and 60. Table 1 also summarizes the feasible search ranges used for all the functions and their theoretical optima. An asymmetrical initialization procedure has been used here following the work reported in [38].

Both the algorithms compared, use classical parameter setup as prescribed by Passino in [1], except the difference that the reproduction frequency in ARBFOA has been made adaptive according to (19). After performing a series of hand-tuning experiments, we found that keeping $\Delta N_c = 3$, provides considerably good result for all the benchmark functions dealt here. The parameter settings that were kept same for both the algorithms have been provided in Table 2.

Table 3 compares the algorithms on the quality of the optimum solution. The mean and the standard deviation (within parentheses) of the best-of-run values for 50 independent runs of both the two algorithms are presented. Each algorithm was run up to a predetermined maximum number of FEs (depending upon the complexity of the problem). The best solution in each case has been marked in bold.

We employed t-tests to compare the means of the results produced by the best ABFOA scheme and the best of the other competitor algorithms over each problem. The t-tests are quite popular among researchers in evolutionary computing and they are fairly robust to violations of a Gaussian distribution with large number of samples like 50 [39].

Table 2
Common parameter setup for BFOA and ARBFOA.

S	$N_c(0)/N_c$	N_s	N_{ed}	N_{re}	p_{ed}	$d_{\text{attractant}}$	$w_{\text{attractant}}$	$w_{\text{repellant}}$	$h_{repellant}$	ΔN_c
10	40	12	4	16	0.25	0.1	0.2	10	0.1	3

Table 3
Average and the standard deviation (in parentheses) of the best-of-run solution for 50 independent runs tested on eight benchmark functions.

Function	Dimensions	No. of FEs	Mean best value (standard deviation)		t-Value between ARBFOA and BFOA
			BFOA	ARBFO	
f_1	15	1×10^4	0.0016 (0.000035)	**0.0009 (0.000074)**	10.8687
	30	5×10^4	0.094 (0.0025)	**0.043 (0.0011)**	132.0340
	45	1×10^5	0.873 (0.1126)	**0.535 (0.0928)**	16.3798
	60	5×10^5	1.728 (0.2125)	**0.687 (0.1472)**	28.4754
f_2	15	1×10^4	0.0705 (0.0623)	**0.0361 (0.0155)**	3.7889
	30	5×10^4	0.216 (0.1254)	**0.112 (0.05465)**	5.3760
	45	1×10^5	0.873 (0.136)	**0.543 (0.1824)**	10.2560
	60	5×10^5	1.705 (0.762)	**1.239 (0.7285)**	3.1257
f_3	15	1×10^4	0.2654 (0.0152)	**0.0745 (0.4536)**	8.5653
	30	5×10^4	4.5354 (1.2644)	**2.5747 (1.7473)**	12.2545
	45	1×10^5	12.7659 (3.6846)	**6.5275 (3.9562)**	25.7567
	60	5×10^5	54.5457 (15.5275)	**21.3343 (5.8620)**	48.8651
f_4	15	1×10^4	0.2812 (0.0216)	**0.0321 (0.02264)**	3.0849
	30	5×10^4	0.3729 (0.0346)	**0.1823 (0.0946)**	2.5838
	45	1×10^5	0.6351 (0.0522)	**0.3069 (0.526)**	2.6417
	60	5×10^5	0.8324 (0.0764)	**0.5638 (0.3452)**	17.6261
f_5	15	1×10^4	0.9332 (0.0287)	**0.7613 (0.0542)**	161.740
	30	5×10^4	2.3243 (1.8833)	**0.7570 (0.5011)**	13.2057
	45	1×10^5	3.4564 (3.4394)	**1.3453 (0.1945)**	16.1300
	60	5×10^5	4.3247 (1.5613)	**2.6481 (0.4551)**	7.6454
f_6	15	1×10^4	0.4325 (0.0543)	**0.2313 (0.0274)**	47.4536
	30	5×10^4	1.4423 (1.3425)	**0.4570 (0.4563)**	34.2642
	45	1×10^5	2.3442 (1.5334)	**0.9354 (0.6647)**	25.1647
	60	5×10^5	2.9675 (2.5613)	**1.1335 (0.7447)**	7.6464
f_7	2	1×10^4	−1.025837 (0.000827)	**−1.031604 (0.000242)**	24.1644
f_8	2	1×10^4	3.156285 (0.109365)	**3.000012 (0.000032)**	7.1446
f_9	15	1×10^4	0.0285 (0.0152)	**0.0144 (0.0126)**	5.0499
	30	5×10^4	0.388 (0.2421)	**0.136 (0.0454)**	7.2341
	45	1×10^5	3.9925 (2.8329)	**1.4527 (1.3274)**	5.7409
	60	5×10^5	6.8234 (3.6231)	**2.3343 (1.2917)**	9.8424
f_{10}	15	1×10^4	0.4342 (0.0632)	**0.0368 (0.0532)**	132.759
	30	5×10^4	0.9227 (0.4829)	**0.4628 (0.3194)**	34.4256
	45	1×10^5	1.4538 (0.6927)	**0.8751 (0.8745)**	27.1753
	60	5×10^5	5.2428 (1.5032)	**1.6734 (0.3692)**	17.6454
f_{11}	2	1×10^4	1.056433 (0.01217)	0.9998023 (0.00825)	1.4536

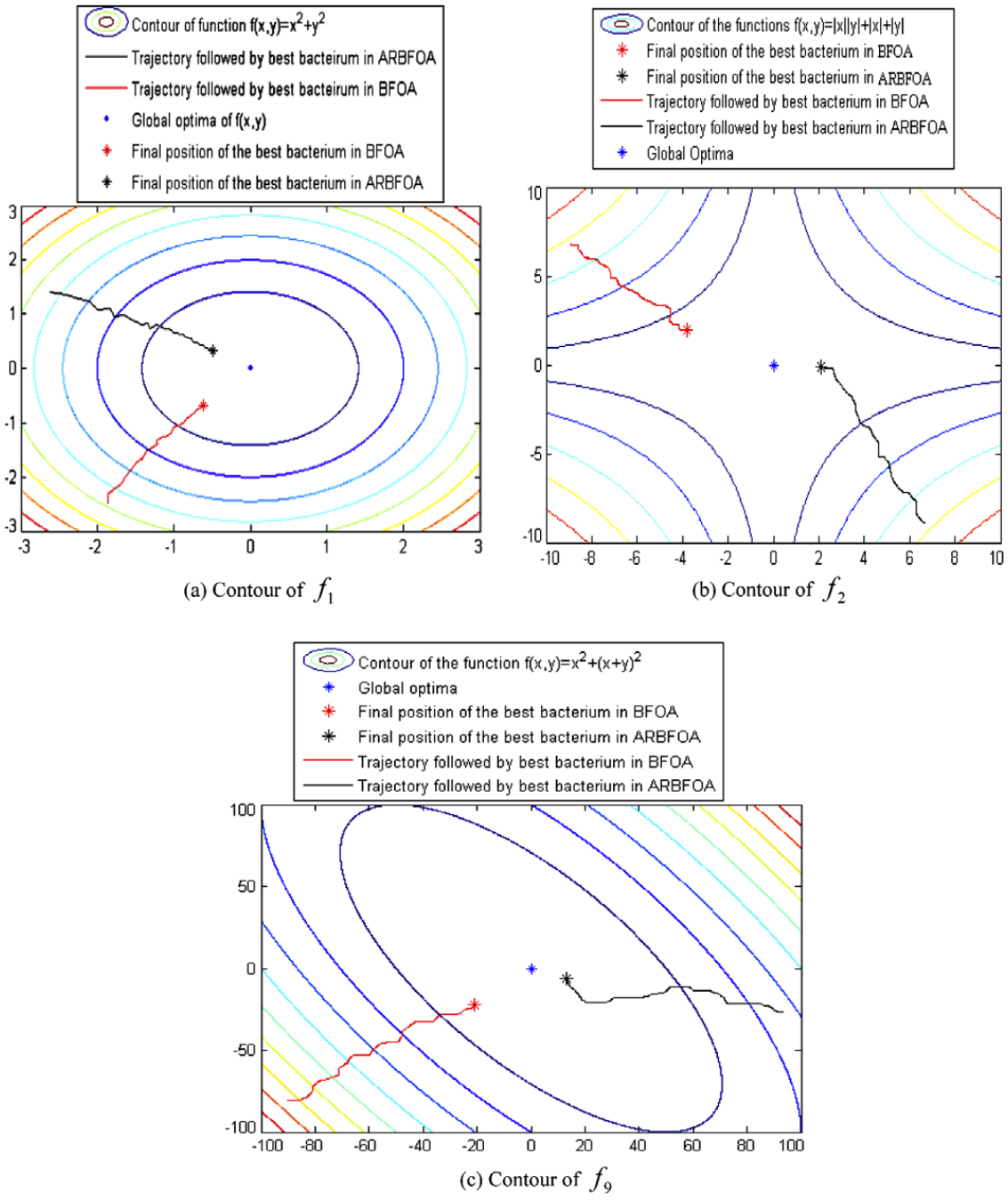

Fig. 8. Final position of best bacterium for BFOA and ARBFOA near the global optima.

When the *t*-value is higher than 1.645 (for degrees of freedom=49), there is a significant difference between the two algorithms with a 95% confidence level. The *t*-values between the ARBFOA and BFOA methods are shown in the sixth column of Table 3. We see that most *t*-values in this table are higher than 1.645. Therefore, the performance of the ARBFOA is statistically significantly better than that of other optimization methods with a 95% confidence level.

Fig. 8 depicts the locus of the best bacterium of classical BFOA and ARBFOA over the constant cost contours of the first three unimodal functions of Table 1. The functions are in 2 dimensions (to facilitate visualization) and on each of them BFOA and ARBFOA are run for the same number of FEs. We omit the similar plots for rest of the functions in order to save space. The plots indicate that ARBFOA is able to find the global optima quickly and more efficiently as compared to the classical BFOA.

4. Conclusions

We presented a simple mathematical analysis of the reproduction step used in the BFOA. For a two bacterial system, it has formulated the effect of the reproduction on bacterial dynamics in the form of two coupled differential equations. Although

it was not possible to have an explicit solution of the equations (as their coefficients vary with time in a complex manner), important conclusions regarding the search strategies of the bacterial population at the time of reproduction could be derived from the analysis and a new model of bacterial foraging namely ARBFOA is proposed which incorporates an adaptive reproduction rate. According to our limited experimental results, ARBFOA outperformed its classical counterpart on eight well-known benchmark functions in a statistically meaningful way. We would like to point out that this paper is a first step towards the mathematical analysis of the reproduction-dynamics in BFOA, which appears as an attractive global optimization technique of current interest. Future research should focus on extending the analysis presented here, to a group of bacteria working on a multi-dimensional fitness landscape and also include effect of the chemotaxis and elimination-dispersal events in the same.

References

[1] K.M. Passino, Biomimicry of bacterial foraging for distributed optimization and control, IEEE Control Systems Magazine (2002) 52–67.
[2] Y. Liu, K.M. Passino, Biomimicry of social foraging bacteria for distributed optimization: models, principles, and emergent behaviors, Journal of Optimization Theory And Applications 115 (3) (2002) 603–628.
[3] D.H. Kim, A. Abraham, J.H. Cho, A hybrid genetic algorithm and bacterial foraging approach for global optimization, Information Sciences 177 (18) (2007) 3918–3937.
[4] S. Das, A. Biswas, S. Dasgupta, A. Abraham, Bacterial Foraging Optimization Algorithm: Theoretical Foundations, Analysis, and Applications, Foundations of Computational Intelligence, Global Optimization, Studies in Computational Intelligence, vol. 3, Springer Verlag, Germany, 2009, pp. 23–55. ISBN: 978-3-642-01084-2.
[5] S. Mishra, A hybrid least square-fuzzy bacterial foraging strategy for harmonic estimation, IEEE Transactions on Evolutionary Computation 9 (1) (2005) 61–73.
[6] M. Tripathy, S. Mishra, L.L. Lai, Q.P. Zhang, Transmission loss reduction based on FACTS and bacteria foraging algorithm, PPSN (2006) 222–231.
[7] D.H. Kim, C.H. Cho, Bacterial foraging based neural network fuzzy learning, IICAI (2005) 2030–2036.
[8] S. Mishra, C.N. Bhende, Bacterial foraging technique-based optimized active power filter for load compensation, IEEE Transactions on Power Delivery 22 (1) (2007) 457–465.
[9] M. Tripathy, S. Mishra, Bacteria foraging-based to optimize both real power loss and voltage stability limit, IEEE Transactions on Power Systems 22 (1) (2007) 240–248.
[10] E. Bonabeau, M. Dorigo, G. Theraulaz, Swarm Intelligence: From Natural to Artificial Systems, Oxford University Press, New York, 1999.
[11] J. Kennedy, R. Eberhart, Y. Shi, Swarm Intelligence, Morgan Kaufmann, 2001.
[12] S. Sabat, Leandro dos Santos Coelho, A. Abraham, MESFET DC model parameter extraction using quantum particle swarm optimization, Microelectronics Reliability 49 (6) (2009) 660–666.
[13] H. Liu, A. Abraham, W. Zhang, A fuzzy adaptive turbulent particle swarm optimization, International Journal of Innovative Computing and Applications 1 (1) (2007) 39–47.
[14] H. Liu, A. Abraham, An hybrid fuzzy variable neighborhood particle swarm optimization algorithm for solving quadratic assignment problems, Journal of Universal Computer Science 13 (7) (2007) 1032–1054.
[15] H. Liu, A. Abraham, M. Clerc, Chaotic dynamic characteristics in Swarm intelligence, Applied Soft Computing Journal 7 (3) (2007) 1019–1026.
[16] J. Kennedy, R. Eberhart, Particle swarm optimization, Proceedings of the IEEE International Conference on Neural Networks (1995) 1942–1948.
[17] M. Dorigo, T. Stiizle, Ant Colony Optimization, MIT Press, Cambridge, MA, 2004.
[18] W.J. Tang, Q.H. Wu, J.R. Saunders, A novel model for bacteria foraging in varying environments, ICCSA 2006 Lecture Notes in Computer Science, vol. 3980, 2006, pp. 556–565.
[19] M.S. Li, W.J. Tang, W.H. Tang, Q.H. Wu, J.R. Saunders, Bacteria foraging algorithm with varying population for optimal power flow, Evo Workshops 2007, Lecture Notes in Computer Science, vol. 4448, 2007, pp. 32–41.
[20] M. Tripathy, S. Mishra, Bacteria foraging-based to optimize both real power loss and voltage stability limit, IEEE Transactions on Power Systems 22 (1) (2007) 240–248.
[21] L. Ulagammai, P. Vankatesh, P.S. Kannan, Prasad Padhy Narayana, Application of bacteria foraging technique trained and artificial and wavelet neural networks in load forecasting, Neurocomputing (2007) 2659–2667.
[22] Munoz, A. Mario, Lopez, A. Jesus, E. Caicedo, Bacteria foraging optimization for dynamical resource allocation in a multizone temperature experimentation platform, Anal. Des. of Intel. Sys. Using SC Tech., ASC 41, 2007, pp.427–435.
[23] H. Shen, Y. Zhu, X. Zhou, H. Guo, C. Chang, Bacterial foraging optimization algorithm with particle swarm optimization strategy for global numerical optimization, in: Proceedings of the First ACM/SIGEVO Summit on Genetic and Evolutionary Computation (Shanghai, China, June 12–14, 2009), GEC '09, ACM, New York, NY, pp. 497–504.
[24] D.H. Kim, A. Abraham, J.H. Cho, A hybrid genetic algorithm and bacterial foraging approach for global optimization, Information Sciences 177 (18) (2007) 3918–3937.
[25] A. Biswas, S. Dasgupta, S. Das, A. Abraham, Synergy of PSO and bacterial foraging optimization: a comparative study on numerical benchmarks, in: E. Corchado et al.(Eds.), Second International Symposium on Hybrid Artificial Intelligent Systems (HAIS 2007), Advances in Soft computing Series, Springer Verlag, Germany, Innovations in Hybrid Intelligent Systems, ASC 44, 2007, pp. 255–263.
[26] A. Biswas, S. Dasgupta, S. Das, A. Abraham, A synergy of differential evolution and bacterial foraging optimization for faster global search, International Journal on Neural and Mass-Parallel Computing and Information Systems-Neural Network World 8 (2007).
[27] S. Das, S. Dasgupta, A. Biswas, A. Abraham, A. Konar, On stability of the chemotactic dynamics in bacterial foraging optimization algorithm, IEEE Transactions on Systems Man and Cybernetics-Part A IEEE Press USA 39 (3) (2009) 670–679.
[28] S. Dasgupta, S. Das, A. Abraham, A. Biswas, Adaptive Computational Chemotaxis in Bacterial Foraging Optimization: An Analysis, IEEE Transactions on Evolutionary Computation, IEEE Press, USA 13 (4) (2009) 919-941.
[29] R. Fletcher, Practical Methods of Optimization, 2nd ed., John Wiley& Sons, Chichester, 1987.
[30] T. Back, D.B. Fogel, Z. Michalewicz, Handbook of Evolutionary Computation, IOP and Oxford University Press, Bristol, UK, 1997.
[31] E. Budrene, H. Berg, Dynamics of formation of symmetrical patterns by chemotactic bacteria, Nature 376 (1995) 49–53.
[32] Y. Blat, M. Eisenbach, Tar-dependent and-independent pattern formation by salmonella typhimurium, Journal of Bacteriology 177 (1995) 1683–1691.
[33] R.P. Anwal, Generalized Functions: Theory and Technique, second ed., Birkhãuser, Boston, MA, 1998.
[34] J.D. Murray, Mathematical Biology, Springer-Verlag, New York, 1989.
[35] E. Bonabeau, M. Dorigo, G. Theraulaz, Swarm Intelligence: From Natural to Artificial Systems, Oxford University Press, New York, 1999.
[36] A. Okubo, Dynamical aspects of animal grouping: swarms, schools, flocks, and herds, Advanced Biophysics 22 (1986) 1–94.
[37] X. Yao, Y. Liu, G. Lin, Evolutionary programming made faster, IEEE Transactions on Evolutionary Computation 3 (2) (1999) 82–102.
[38] P.J. Angeline, Evolutionary optimization versus particle swarm optimization: philosophy and the performance difference, Lecture Notes in Computer Science, vol. 1447, in: Proceedings of Seventh International Conference on. Evolutionary Programming–Evolutionary Programming VII, 1998, pp. 84–89.
[39] B. Flury, A First Course in Multivariate Statistics, Springer, 1997. p. 28.