

Adaptive Computational Chemotaxis in Bacterial Foraging Algorithm

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

Department of Electronics and Telecommunication Engineering, Jadavpur University, Kolkata, India

Center of Excellence for Quantifiable Quality of Service*

Norwegian University of Science and Technology, Norway

ajith.abraham@ieee.org

Abstract

Some researchers have illustrated how individual and groups of bacteria forage for nutrients and to model it as a distributed optimization process, which is called the Bacterial Foraging Optimization (BFOA). One of the major driving forces of BFOA is the chemotactic movement of a virtual bacterium, which models a trial solution of the optimization problem. In this article, we analyze the chemotactic step of a one dimensional BFOA in the light of the classical Gradient Descent Algorithm (GDA). Our analysis points out that chemotaxis employed in BFOA may result in sustained oscillation, especially for a flat fitness landscape, when a bacterium cell is very near to the optima. To accelerate the convergence speed near optima we have made the chemotactic step size C adaptive. Computer simulations over several numerical benchmarks indicate that BFOA with the new chemotactic operation shows better convergence behavior as compared to the classical BFOA.

Keywords: bacterial foraging, stochastic gradient descent, computational chemotaxis

1. Introduction

A foraging organism 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 as well as the environment. BFOA has been successfully applied to real world problems like optimal controller design [1, 2, 3], harmonic estimation [4], transmission loss reduction [5], and machine learning [6]. A major step in the BFOA is the simulated chemotactic movement. Basically, chemotaxis is a foraging strategy that implements a type of local optimization where the bacteria try to climb up the nutrient concentration, avoid noxious substance and search for ways out of neutral media. This step has much resemblance with biased random walk model.

Mishra, in [4], proposed a Takagy-Sugeno fuzzy inference scheme for selecting the optimal chemotactic step size in BFOA. In this article we have provided a simple mathematical analysis of the computational chemotaxis in the framework of the classical gradient descent search [7, 8] algorithm. We have analytically shown that a chemotactic step height varying as the function of the current fitness value leads to better convergence especially on the flatter region of the fitness landscape. The rest of the paper is organized as follows. In Section 2 we outline the classical BFOA and provide the mathematical analysis of the chemotactic movement of a one-dimensional bacterium and propose the adaptive chemotaxis for BFOA. Section 3 provides detailed comparison between the classical and the newly developed chemotaxis operator on a test-suit of several benchmarks. Finally the paper is concluded in Section 4.

2. Bacterial Foraging Algorithm

In BFOA the coordinates of a bacterium represent an individual solution to the optimization problem. Such a set of trial solution converges towards global optima following a foraging group dynamics of the bacteria population. Each bacterium first undergoes a chemotactic movement as long as it goes in the direction of positive nutrient gradient (i.e., decreasing fitness). After a certain number of complete swims the best half of the population undergoes reproduction eliminating the rest of the population. In order to escape local optima an elimination-dispersal event is carried out where some bacteria are liquidated at random with a very small probability while the new replacements are randomly initialized over the search space.

2.1. Analysis of the Computational Chemotaxis

Let us consider a single bacterium cell that undergoes chemotactic steps according to (1) over a one dimensional objective function. The bacterium lives in

continuous time and at the t -th instant its position is given by $\theta(t)$. Below we list a few assumptions for the sake of gaining mathematical insight.

- i) The objective function $J(\theta)$ is continuous and differentiable at all points in the search space.
- ii) The chemotactic step size is not very large.
- iii) The analysis applies to the regions of the fitness landscape where gradients of the function are small i.e., near to the optima.

According to assumption (iii), analysis is restricted within the region alike to that of the shaded region, as shown in Figure 1. In Figure 1, the green arrow represents velocity of the bacterium and the blue arrow shows the gradient vector. It is to be noted that velocity vector does not necessarily coincide with the gradient vector. Initially, the bacterium was at point P and it moves to point Q. Here, the vector PQ shows the direction of velocity of the bacterium.

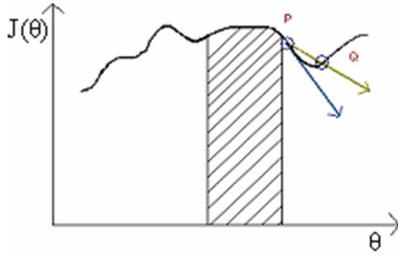


Figure 1. A continuous, one-dimensional fitness landscape for BFOA.

2.2. Analytical Treatment

Let, at time t position of an individual bacterium be θ and value of objective function (to be minimized) be $J(\theta)$. Also assume that, after an infinitesimal time interval Δt , its position changes by an amount $\Delta \theta$. Then the new value of the objective function becomes $J(\theta + \Delta \theta)$. According to the chemotactic step used in BFOA, the bacteria 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. The decision-making (i.e. whether to take a step or not) activity of bacteria can be modeled by unit step function as,

$$u(J(\theta) - J(\theta + \Delta \theta)) = 1, \text{ if } J(\theta) - J(\theta + \Delta \theta) > 0 \\ = 0, \text{ otherwise}$$

$$\text{Thus, } \Delta \theta = u(J(\theta) - J(\theta + \Delta \theta)) \cdot C \cdot \Delta \quad (1)$$

Where, C indicates the chemotactic step height and Δ = Direction of tumble (Here it can assume only two values 1 or -1 with uniform probabilities).

Dividing both sides of (1) by Δt we get,

$$\frac{\Delta \theta}{\Delta t} = \frac{u[J(\theta) - J(\theta + \Delta \theta)]}{\Delta t} \cdot C \cdot \Delta \\ \Rightarrow \frac{\Delta \theta}{\Delta t} = u \left[-\frac{J(\theta + \Delta \theta) - J(\theta)}{\Delta t} \right] \cdot C \cdot \Delta \quad (2)$$

[$\because \Delta t > 0$. We know for a positive constant a , $u\left(\frac{x}{a}\right) = u(x)$]

Now velocity of the bacterium is given by,

$$V_b = \lim_{\Delta t \rightarrow 0} \frac{\Delta \theta}{\Delta t} = \lim_{\Delta t \rightarrow 0} u \left[-\frac{J(\theta + \Delta \theta) - J(\theta)}{\Delta t} \right] \cdot C \cdot \Delta \\ \text{by (2), } V_b = \lim_{\Delta t \rightarrow 0} u \left[-\frac{J(\theta + \Delta \theta) - J(\theta)}{\Delta \theta} \frac{\Delta \theta}{\Delta t} \right] \cdot C \cdot \Delta \quad (3)$$

and then from(3),

$$V_b = \theta' = u \left[-\frac{dJ(\theta)}{d\theta} \frac{d\theta}{dt} \right] C \Delta \quad (4)$$

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

Where, $G = J'(\theta)$ = gradient of the objective function at θ . 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,

$$u(x) = \lim_{k \rightarrow \infty} \phi(x) = \lim_{k \rightarrow \infty} \frac{1}{1 + e^{-kx}} \quad (6)$$

2.3. Error Estimation and Analysis Limitations

Due to the approximation of the unit step function, small error has been introduced in the analysis. Again we have simplified the function for some special cases (assumptions (ii) and (iii)). Here, magnitude of maximum possible value of error in estimation of $\frac{d\theta}{dt}$

equal to $|\frac{C\Delta}{2}| = \frac{C}{2} \cdot |\Delta| = 1$. We have assumed $u(x)$ is

approximately equal to $\frac{1}{1 + e^{-kx}}$, with $k=10$. For this

value of k , $\phi(x)$ fairly approximates $u(x)$. Hence from

$$(5), V_b = \frac{C \Delta}{1 + e^{kGV_b}} \quad (7)$$

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

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

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

$$V_b = \frac{2C \cdot \Delta}{4 + kGC \cdot \Delta} \quad (8)$$

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

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

$$\Rightarrow V_b = \frac{C \Delta}{2} - \frac{kGC^2 \Delta^2}{8} \quad [\Delta^2 = 1]$$

$$\Rightarrow V_b = -\frac{kC^2}{8} G + \frac{C \Delta}{2} \quad (9)$$

If x is small, we may again approximate the logistic function with the following equation of a straight line as: $u(x) = \frac{k}{4}x + \frac{1}{2}$ (10)

These simplifications are already done in (7) to (9) in that case $x = V_b G$. The straight line, which approximates logistic function as shown in Figure 2, intersects graph of the logistic function at two points A and B. But when $|x| > OA$ or $|x| > OC$, the error in the decision of our analysis gradually increases. So we must restrict our analysis within the region AC i.e. magnitude of GV_b has certain limits. As shown in Figure 2, x must lie between A and C. Otherwise our analysis becomes erroneous. After imposing constraints on (10) we get,

$$x \frac{k}{4} + \frac{1}{2} \leq 1 \text{ and } x \frac{k}{4} + \frac{1}{2} \geq 0$$

After solving above inequalities we get

$$-\frac{2}{k} \leq x \leq \frac{2}{k} \quad (11)$$

Substituting, x by GV_b in (11) we get,

$$|V_b| \leq \frac{2}{k|G|} \quad (12)$$

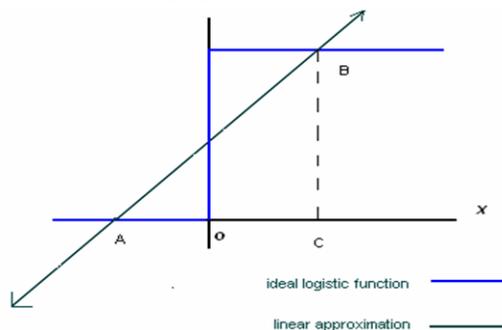


Figure 2. The region of error due to approximation of the unit step with the logistic functions

From (9) and (12) we get,

$$\frac{2}{k|G|} \geq |V_b| = \left| \frac{C \Delta}{2} - \frac{kC^2}{8} G \right|$$

We know, for any two numbers a and b , $|a - b| \geq |a| - |b|$. Hence,

$$\left| \frac{C \Delta}{2} - \frac{kC^2}{8} G \right| \geq \left| \frac{C \Delta}{2} \right| - \left| \frac{kC^2}{8} G \right| = \frac{C}{2} - \frac{kC^2}{8} |G|.$$

[$C > 0, k > 0$ and Δ can assume values 1 and -1 randomly, giving $|\Delta| = 1$]

Incorporating inequality (11) in above we get,

$$\frac{2}{k|G|} \geq \frac{C \Delta}{2} - \frac{kC^2}{8} G \geq \frac{C}{2} - \frac{kC^2}{8} |G| \quad (13)$$

Again, as, $|a - b| \geq |b| - |a|$

$$\left| \frac{C \Delta}{2} - \frac{kC^2}{8} G \right| \geq \left| \frac{kC^2}{8} G \right| - \left| \frac{C \Delta}{2} \right| = \frac{kC^2}{8} |G| - \frac{C}{2}$$

Incorporating inequality (12) in above one we get,

$$\frac{2}{k|G|} \geq \left| \frac{C \Delta}{2} - \frac{kC^2}{8} G \right| \geq \frac{kC^2}{8} |G| - \frac{C}{2} \quad (14)$$

Inequality (14) implies, $\frac{2}{k|G|} \geq \frac{C}{2} - \frac{kC^2}{8} |G|$

$\Rightarrow (k|G|C - 2)^2 + 12 \geq 0$. It is trivially true.

Inequality (14) implies,

$$\frac{2}{k|G|} \geq \frac{kC^2}{8} |G| - \frac{C}{2}$$

$$\Rightarrow k^2 |G|^2 C^2 - 4k|G|C - 16 \leq 0$$

Solving the inequality, we get,

$$-\frac{2}{k|G|}(\sqrt{5} - 1) \leq C \leq \frac{2}{k|G|}(1 + \sqrt{5})$$

But as $C > 0$, $\therefore 0 < C \leq \frac{2}{k|G|}(1 + \sqrt{5})$

So, $C_{\max} = \frac{2}{k|G|}(1 + \sqrt{5})$

Now, let us assume within our domain of analysis $|G|_{\max}$ be the maximum possible magnitude of gradient.

$\frac{2}{k|G|}(1 + \sqrt{5})$ term is minimized when

$|G| = |G|_{\max}$. Our analysis is valid if chemotactic step size is less than or equals to this minimum value i.e.

$\frac{2}{k|G|_{\max}}(1 + \sqrt{5})$. So we define maximum allowable

value of chemotactic step size as $C_{\max} = \frac{2}{k|G|_{\max}}(1 + \sqrt{5})$. If $|G|_{\max}$ is large, the

maximum allowable step size almost vanishes making our analysis invalid for moderately small values of step size. From this consideration we should also restrict

domain of analysis within the region with moderate value of gradient.

2.4. Chemotaxis and the Classical Gradient Decent Search

From (9) we get, $V_b = -\frac{kC^2}{8}G + \frac{C\Delta}{2}$

$$\Rightarrow \frac{d\theta}{dt} = -\alpha'G + \beta \quad (15)$$

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:

$$\frac{d\theta}{dt} = -\alpha G \quad (16)$$

where α is the learning rate. Similarity between equation (15) and (16) suggests that chemotaxis is a modified CGDA where α' , which is a function of chemotactic step size, can be identified as the learning rate of chemotaxis. As discussed in Section 2.3, the magnitude of gradient should be small within the region of our analysis. So we choose point P in the one dimensional fitness landscape shown in Figure 3 as the operating point for our analysis. For chemotaxis of BFOA, when G becomes very small, the gradient descent term $\alpha'G$ of (15) becomes ineffective. But the random search term $\frac{C\Delta}{2}$ plays an important role in this context. From (15), considering, $G \rightarrow 0$, $\frac{d\theta}{dt} = \frac{C\Delta}{2} \neq 0$. So there is a convergence towards actual minima as shown in Figure 4. The random search term $\frac{C\Delta}{2}$ in the RHS of (8) provides an improvement to the classical gradient descent search. When gradient becomes very small, the random term dominates over gradient decent term and bacteria changes its position. But random search term may lead to change in position in the direction of increasing cost function value. If it happens then again magnitude of gradient increases and dominates the random search term.

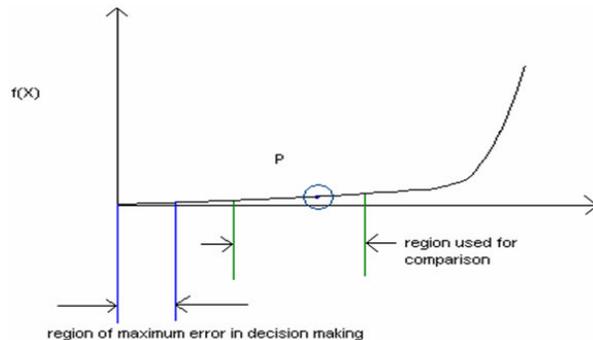


Figure 3. A sample fitness landscape for studying the computational chemotaxis

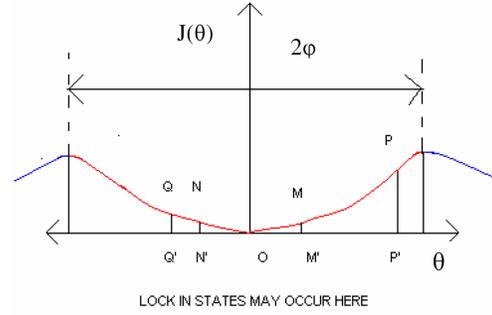


Figure 4. Fitness landscape for an even objective function

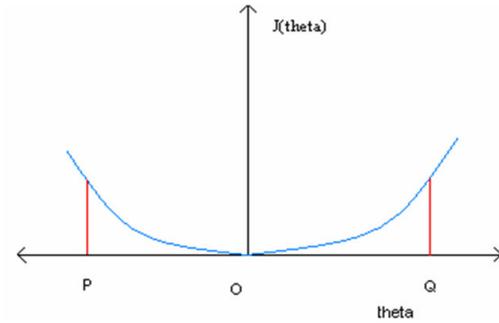


Figure 5. Fitness landscape for the function $J(\theta) = \theta^2$

2.5. Oscillation Problem: Need for Adaptive Chemotaxis

If magnitude of gradient decreases consistently, near the optima or very close to the optima $-\alpha'G$ of (15) becomes comparable to β . Then gradually β becomes dominant.

$$\text{When } |G| \rightarrow 0, \left| \frac{d\theta}{dt} \right| = |\beta| = \frac{C\Delta}{2} = \frac{C}{2} \quad [\because |\Delta| = 1]$$

Let us assume the bacterium has reached close to the optimum. But as we have derived $\left| \frac{d\theta}{dt} \right| = \frac{C}{2}$, the bacterium does not stop taking chemotactic steps. It oscillates about the optima. This crisis can be remedied if step size C is made adaptive according to the following equation,

$$C = \frac{|J(\theta)|}{|J(\theta)| + \lambda} = \frac{1}{1 + \frac{\lambda}{|J(\theta)|}} \quad (17)$$

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 (16) we see, if $J(\theta) \rightarrow 0$, then $C \rightarrow 0$. So there would be no oscillation if the bacterium reaches optima because random search term vanishes as $C \rightarrow 0$. The functional form given in equation (16) causes C to vanish nears the optima.

Besides, it plays another important role described below. From (16), we have,

When $J(\theta)$ is large $\frac{\lambda}{|J(\theta)|} \rightarrow 0$ and consequently,

$C \rightarrow 1$. This has an important physical significance. A bacterium in the vicinity of noxious substance (high value of the cost function) takes larger chemotactic steps. On the other hand, bacterium near high nutrient concentration (i.e. low value of cost function) is much reluctant to take chemotactic steps.

2.6 Adaptive Chemotaxis for Avoiding Lock-in State

Let us consider an even function $J(\theta)$ (as shown in Figure 5), which has its minima at $\theta=0$ and its minimum value is also equals to 0. Let us also assume the function is increasing in the interval $[0, \phi]$. (e. g. $J(\theta) = \theta^2$ is an even function where it is increasing in the interval $[0, \infty)$). So in this case $\phi \rightarrow \infty$. A special case of stagnation may occur within the region $(-\phi, \phi)$ i.e. where the function is increasing. We here refer to this situation as *lock in*. The *lock in* condition arises when a bacterium has reached somewhat near to the optima of a function and then its further movements are not possible due to comparatively large step size. Consider a bacterium to be at M' at the current time step, such that $OM' = x$ and $(ON' = x)$ and $x < \phi$. Let the chemotactic step height taken by the bacterium be C . Now if $C > 2x$ the bacterium may move on to points like P or Q on the fitness landscape (corresponding to $\Delta = +1/-1$ and such that $M'P' = M'Q' = C$). As can be observed from figure 5, in both the cases value of the objective function increases than the previous value ($PP' > MM'$ and $QQ' > MM'$). Hence according to the BFOA directives, the bacterium cannot jump to either of the locations P or Q . In, the limiting case, for $C = 2x$, the bacterium will oscillate between points M and N (as in this case N and Q coincide).

The same events occur if initially the bacterium is at N i.e. if its present coordinate $x < 0$. Hence taking into account both the cases, the condition for lock-in may be put forward as $C \geq 2|x|$ where x indicates the present coordinate of the bacterium.

Example 1:

Suppose we have to minimize a one-dimensional function $J(\theta) = \theta^2$ (Figure 5). Let in Figure 6, $|PO| = |QO| = |\theta|$. We also assume that step size is C and the bacterium is currently at the position $\theta = \theta$ i.e. it is

at Q . Now in classical chemotaxis, three cases may arise as described below:

Case I: Let step-size $C = 2|\theta|$. Then, for $\Delta = -1$, the bacterium should move to P . But as in this case its objective function value remains same, it does not come to P , but stays at Q [as $J(\theta) = J(-\theta) = \theta^2$]. As $\Delta = 1$ tries to shift the bacterium to the right (where the objective function value increases again) it again stays at Q . Hence the bacterium gets trapped at Q .

Case II: Let $C > 2|\theta|$. In that case, bacterium remains immobile for both values of Δ . Here step size is constant and greater than $2|\theta|$. If the bacterium moves in any one of the two directions, value of the objective function increases. So the bacterium is trapped.

Case III: Let, $C < 2|\theta|$. In this case, bacterium will move to some point in the left of origin. But, C is fixed (say, 0.5). So, after certain iterations any one of CASE II and I must arise.

Now consider the situation when the step-size has been made adaptive according to (16). Then we have

$$C = \frac{\theta^2}{\theta^2 + \lambda}$$

The lock in states never occur if for all possible values of θ , if $C < 2|\theta| \Rightarrow \frac{|\theta|^2}{|\theta|^2 + \lambda} < 2|\theta|$

$$\Rightarrow |\theta|^2 < 2|\theta|(|\theta|^2 + \lambda) \quad [\because |\theta|^2 + \lambda > 0]$$

$$\Rightarrow \lambda > \frac{|\theta|}{2} - |\theta|^2$$

Let, us consider, $f(|\theta|) = \frac{|\theta|}{2} - |\theta|^2$. Maximum value of function is obtained when, $\frac{df(|\theta|)}{d|\theta|} = 0 \Rightarrow \theta = \frac{1}{4}$. Putting,

$$|\theta| = \frac{1}{4} \text{ we get, } \arg \max_{\theta} f(|\theta|) = f\left(\frac{1}{4}\right) = \frac{1}{16}$$

Hence, for all values of θ , if $\lambda > f(|\theta|) \Rightarrow \lambda > 1/16$, then no trapping or sustained oscillations of the bacterium cell will arise near the global optimum. In this case the bacterium cell will follow a trajectory as depicted in Figure 6.

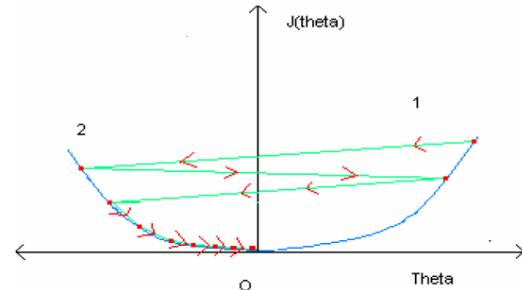


Figure 6. Convergence towards the global optima for adaptive step size in chemotaxis

Now, suppose we take, $\lambda = 0.0630 (> \frac{1}{16})$. Let, $\theta = 0.1$

initially. Then in Table 1, we present how the value of θ changes over subsequent chemotactic steps, when the step-size has been made adaptive.

Chemotactic step	θ	Step-size C
1	0.1000	0.1369
2	-0.0370	0.0213
3	-0.0160	0.0040
4	-0.0120	0.0020
5	-0.010	0.0015
6	-0.0097	0.0017
7	-0.0082	0.0011
8	-0.0063	0.0006
9	-0.0057	0.0005
10	-0.0052	0.0004

Table 1. Variation of bacterium position θ with chemotactic steps for adaptive step size C

3. Experimental Results

We present the results of Adaptive BFOA (ABFOA) and the classical BFOA using a test-suite of six well-known benchmark functions - Sphere function f_1 , Rosenbrock f_2 , Rastrigin f_3 , Griewank f_4 , Ackley f_5 and Shekel's Foxholes function f_6 [9]. All the benchmark functions except f_5 have their global minima at the origin or very near to the origin. For f_5 the global minimum is at (-31.95,-31.95) and its value is 0.998. This function has only 2 dimensions. An asymmetrical initialization procedure has been used here following the work reported in [10]. BFOA and ABFOA used classical parameter setup as prescribed in [1], except the difference that the chemotactic step size in ABFOA has been made adaptive according to (16). After performing a series of hand-tuning experiments, we found that keeping $\lambda = 4000$, provides considerably good result for nearly all benchmark functions dealt here. The chemotactic step-size $C(i)$ was kept at 0.1 in the classical BFOA. Rest of the parameter settings that were kept same for both the algorithms are as follows: $w_{attract} = 0.2$, $w_{repellant} = 10$, $h_{repellant} = 0.1$, $S = 50$, $N_c = 100$, $N_s = 4$, $N_{ed} = 1$ and $p_{ed} = 0.25$. The comparative study presented here, focuses on the following performance metrics: (a) the quality of the final solution (b) the convergence speed (measured in terms of the number of fitness function evaluations (FE's)) (c) the frequency of hitting the optima. Twenty independent runs of each of the algorithms were carried out and the average and the standard deviation of the best-of-run values were recorded.

	D	No. of FE's	Mean best value (standard deviation)	
			BFOA	ABFOA
f_1	15	1×10^5	0.001 (0.00)	0.001(0.00)
	30	5×10^5	0.084 (0.0025)	0.045 (0.0061)
	45	1×10^6	0.673 (0.1126)	0.435 (0.0928)
	60	5×10^6	1.728 (0.2125)	0.827 (0.1472)
f_2	15	1×10^5	26.705 (11623)	11.561 (2.355)
	30	5×10^5	58.216 (14.3254)	40.212 (6.5465)
	45	1×10^6	96.873 (26.136)	54.543 (10.8824)
	60	5×10^6	154.705 (40.1632)	91.239 (12.7285)
f_3	15	1×10^5	6.9285 (2.0952)	1.0044 (0.6126)
	30	5×10^5	17.0388 (4.8421)	3.3316 (0.5454)
	45	1×10^6	30.9925 (7.8329)	6.4527 (1.3274)
	60	5×10^6	33.8234 (9.6231)	8.3343 (0.2917)
f_4	15	1×10^5	0.2812 (0.0216)	0.0321 (0.02264)
	30	5×10^5	0.3729 (0.0346)	0.2414 (0.5107)
	45	1×10^6	0.6351 (0.0522)	0.3069 (0.4526)
	60	5×10^6	0.8324 (0.0764)	0.5638 (0.3452)
f_5	15	1×10^5	0.9332 (0.0287)	0.7613 (0.0542)
	30	5×10^5	2.3243 (1.8833)	0.8038 (0.5512)
	45	1×10^6	3.4564 (3.4394)	1.3432 (0.1945)
	60	5×10^6	4.3247 (1.5613)	2.4224 (0.4551)
f_6	2	1×10^5	1.056433 (0.01217)	0.999832 (0.00167)

Table 2. Average and the standard deviation of the best-of-run solution

Different maximum number of FE's was used according to the complexity of the problem. Table 2 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 25 independent runs of each of the four algorithms are presented. Each algorithm was run up to a predetermined maximum number of Function Evaluations –FE's (depending upon the complexity of the problem). The best solution in each case has been shown in bold. Table 3 shows results of unpaired t -tests between the competing algorithms in each case (standard error of difference of the two means, 95% confidence interval of this difference, the t value, and the two-tailed P value). For all cases in Table 3, sample size = 25 and degrees of freedom = 48.

Fn, Dim	Std. Err	t	95% Conf. Intvl	Two-tailed P	Significance
$f_1, 30$	0.001	29.579	0.036349 to 0.041651	< 0.0001	Extremely significant
$f_1, 45$	0.029	8.1555	0.179324 to 0.296676	< 0.0001	Extremely significant
$f_1, 60$	0.052	17.427	0.797049 to 1.004951	< 0.0001	Extremely significant
$f_2, 15$	0.639	23.685	13.8584338 to 16.4295662	< 0.0001	Extremely significant
$f_2, 30$	3.150	5.7154	11.670356 to 24.337644	< 0.0001	Extremely significant
$f_2, 45$	3.498	4.3658	-22.30540 to -8.23860	< 0.0001	Extremely significant
$f_2, 60$	5.153	8.7855	-55.63537 to -34.91263	< 0.0001	Extremely significant
$f_3, 15$	0.064	16.690	-1.2011367 to -0.9428633	<0.0001	Extremely Significant
$f_3, 30$	0.192	1.7899	-0.04242 to 0.73042	0.0798	Not quite Significant
$f_3, 45$	1.513	14.468	-24.9330 to -18.8487	< 0.0001	Extremely significant
$f_3, 60$	1.513	2.5724	-6.932087 to -0.849713	0.0132	Significant
$f_4, 15$	0.006	3.0849	0.0062682 to 0.0297318	0.0034	Very Significant
$f_4, 30$	0.022	2.5838	0.012333 to 0.098867	0.0129	Significant
$f_4, 45$	0.023	2.6417	-0.108662 to -0.014738	0.0111	Significant
$f_4, 60$	0.020	17.626	-0.3972779 to -0.3159221	<0.0001	Extremely Significant
$f_5, 15$	0.001	161.74	-0.0881827 to -0.086017	<0.0001	Extremely Significant
$f_5, 30$	0.011	13.205	-0.1671923 to -0.1230077	<0.0001	Extremely Significant
$f_5, 45$	0.689	16.130	9.727915 to 12.498485	<0.0001	Extremely Significant
$f_5, 60$	0.136	2.1524	-0.565934 to -0.019266	0.0364	Significant
$f_6, 2$	0.002	23.038	0.05166125 to 0.06154075	<0.0001	Extremely Significant

Table 3. Results of unpaired t-tests on the data of Table 2.

It is interesting to see from Tables 2 and 3 that the proposed ABFOA performed very well when compared to the classical BFOA in a statistically significant way. Table 4 shows, for all test functions and all algorithms, the number of runs (out of 25) that managed to find the optimum solution within a given tolerance or cutoff value. In Table 4, we also report the mean number of function evaluations (FEs) and standard deviations (within parentheses) required by each algorithm to converge within the prescribed cut-off value. In each case, the mean is calculated over the corresponding number of runs that managed to converge within the cut-off value.

The cut-off value was determined from Table 4 as an objective function value that is slightly higher than the worst value found by any of the competitor algorithms. In Figure 7, we have graphically presented the rate of convergence of the contestant methods for all the functions (in 60 dimensions). Tables 2 to 4 and Figure 7 clearly indicates that the ABFOA can yield better quality solutions for all the benchmark problems

considered here, consuming less computational time as compared to the classical BFOA. Since both the contestant algorithms start from the same initial population, both of them use common parametric setup, the difference in their performance must have resulted from the use of adaptive chemotactic step height in ABFOA. This observation also agrees with the simplified analytical treatment provided in Section 2, which points out that the adaptive chemotactic operator has an edge over the classical chemotaxis, especially in context to the convergence behavior of the algorithm, very near to the optima.

Func	Dim	Threshold Value	No. of runs converging to the cut-off, Mean No. of FEs Required and (Std Deviation)	
			BFOA	ABFOA
f_1	15	0.001	25, 48109.20 (41.873)	25, 31465.48 (19.492)
	30	0.090	50, 77372.54 (343.547)	50, 72093.12 (102.763)
	45	0.700	50, 119476.80 (3316.982)	50, 974005.08 (920.276)
	60	1.800	50, 212381.64 (331.282)	50, 15136.40 (45.932)
f_2	15	30.00	31, 95920.88 (344.390)	43, 92004.28 (142.009)
	30	60.00	25, 319432.20 (317.912)	34, 210202.96 (321.87)
	45	100.00	15, 911235.80 (1012.282)	28, 818932.12 (314.928)
	60	160.00	12, 4146674.50 (1291.203)	22, 293093.40 (346.561)
f_3	15	8.00	50, 42298.44 (131.291)	50, 39231.60 (124.397)
	30	20.00	36, 419827.40 (63.292)	43, 350635.80 (98.290)
	45	32.00	23, 242938.97 (823.208)	40, 121093.50 (98.403)
	60	50.00	10, 7142938.40 (8323.208)	41, 2368039.77 (1462.659)
f_4	15	0.300	34, 92827.85 (223.839)	43, 70938.79 (116.349)
	30	0.500	28, 432094.25 (52.491)	28, 350948.75 (56.872)
	45	0.700	10, 925367.30 (134.268)	13, 840291.34 (142.903)
	60	1.000	14, 4328701.56 (23.221)	10, 3425949.10 (55.715)
f_5	15	1.000	50, 4573.64 (3.492)	50, 7745.34 (13.569)
	30	5.000	50, 8768.44 (23.383)	50, 15049.30 (82.298)
	45	15.000	45, 49089.66 (324.482)	50, 127940.38 (232.98)
	60	20.000	31, 460729.03 (171.265)	50, 3119847.26 (873.389)
f_6	25	1.10	50, 44372.34 (67.482)	50, 28923.72 (9.409)

Table 4. No. of successful runs, mean no. of FE's and standard deviation (in paranthesis) required to converge.

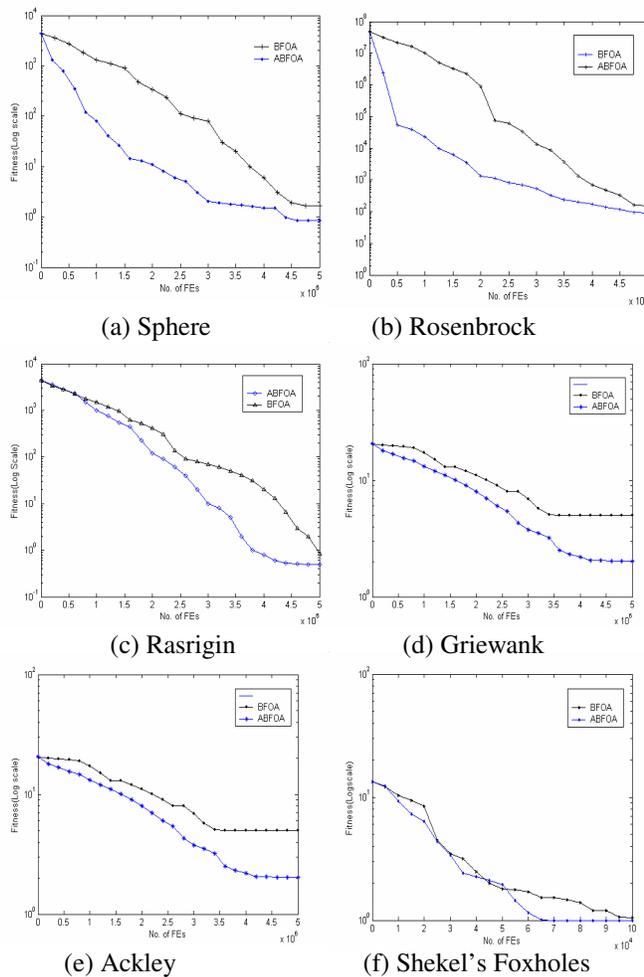


Figure 7. Progress to the optimum solution (all plots are for dimension = 60, except Shekel's foxhole that is 2-dimensional)

4. Conclusions

This paper presented a simple mathematical analysis of the computational chemotaxis, used in the BFOA. It also proposed a simple scheme to adapt the chemotactic step-size in BFOA with a view to improving its convergence behavior without imposing additional requirements in terms of the no. of FEs. It has analytically been shown that the proposed adaptation scheme can avoid the oscillation around the optima or the stagnation near optima for a one dimensional bacterium cell. The classical BFOA was compared with the adaptive BFOA over a test-bed of six well-known numerical benchmarks. Following performance metrics were used: (a) solution quality, (b) speed of convergence, and (c) frequency of hitting the optimum. The adaptive BFOA was shown to outperform its classical counterpart in a statistically

meaningful way for all of the tested problems. The future research may 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 cell-to-cell attractor repellent profile in the same. A mathematical analysis of other steps of the BFOA like reproduction, elimination-dispersal etc may provide valuable guidelines for choosing/tuning the parameters of the algorithm.

5. References

- [1] Passino, K. M.: Biomimicry of Bacterial Foraging for Distributed Optimization and Control, *IEEE Control Systems Magazine*, 52-67, (2002).
- [2] Liu, Y. and Passino, K. M.: Biomimicry of Social Foraging Bacteria for Distributed Optimization: Models, Principles, and Emergent Behaviors, *Journal of Optimization Theory And Applications*: Vol. 115, No. 3, pp. 603–628, December 2002
- [3] Kim, D. H., Abraham, A., Cho, J. H.: A hybrid genetic algorithm and bacterial foraging approach for global optimization, *Information Sciences*, Vol. 177 (18), 3918-3937, (2007).
- [4] Mishra, S.: A hybrid least square-fuzzy bacterial foraging strategy for harmonic estimation. *IEEE Trans. on Evolutionary Computation*, vol. 9(1): 61-73, (2005).
- [5] Tripathy, M., Mishra, S., Lai, L. L. and Zhang, Q. P.: Transmission Loss Reduction Based on FACTS and Bacteria Foraging Algorithm. *PPSN*, 222-231, (2006).
- [6] Kim, D. H., Cho, C. H.: Bacterial Foraging Based Neural Network Fuzzy Learning. *IICAI 2005*, 2030-2036.
- [7] Avriel, M.: *Nonlinear Programming: Analysis and Methods*, Dover Publishing (2003).
- [8] Snyman, J. A.: *Practical Mathematical Optimization: An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms*. Springer Publishing (2005).
- [9] Yao, X., Liu, Y., Lin, G. Evolutionary programming made faster, *IEEE Transactions on Evolutionary Computation*, vol 3, No 2, 82-102, (1999).
- [10] Angeline, P. J.: Evolutionary optimization versus particle swarm optimization: Philosophy and the performance difference, *Lecture Notes in Computer Science* (vol. 1447), Proceedings of 7th International Conference on Evolutionary Programming – Evolutionary Programming VII (1998) 84-89.