Nature Inspired Optimization Technique: Bacterial Foraging Algorithm

Nikhil Kushwaha[1], Y. R. Sood[2], Suryakant[3] and Vimal Singh Bisht[4]
Department of Electrical Engineering, NIT Hamirpur, Himachal-Pradesh, India

Email-id: [1-4] nikhil.kushwaha85@gmail.com, yrsood@gmail.com, suryakantshukla8@gmail.com, bishtvimal@yahoo.com

Abstract— **In this era of modernization, deregulation and competition the scenario of the optimization techniques is being evolved dramatically towards the functional mimicry of nature. There are many nature inspired optimization algorithms such as PSO, Ant Colony, Genetic Algorithm, Evolutionary Techniques etc. Recently Bacterial foraging Optimization Algorithm has attracted a lot of attention as a high performance optimizer. In 2002, K. M. Passino proposed Bacterial Foraging Optimization Algorithm (BFOA)[1] for distributed optimization and control. One of the major driving forces of BFOA is the chemotactic movement of a virtual bacterium that models a trial solution of the optimization problem. The underlying biology behind the foraging strategy of E.coli is emulated in an extraordinary manner and used as a simple optimization algorithm. This paper presents the BFOA for *global optimization.***

Keywords— **Bacterial Foraging, global optimization, Chemotaxis, Optimization Algorithm.**

I. Introduction

Optimization is a computational science that studies techniques for searching the 'best' solutions. It has been widely employed in a large variety of fields, including transportation, manufacturing, physics, and medicine. To tackle complex search problems of the real world, scientists have been drawing inspiration from nature and natural creatures for years. 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 since they are more likely to enjoy reproductive success.

Since a foraging organism or animal takes necessary action to maximize the energy intake 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 solve real-world optimization problems. Based on this concept, Passino proposed an optimization technique known as the bacterial foraging optimization algorithm (BFOA) [1]. To date, BFOA has successfully been applied to real-world problems such as optimal controller design, harmonic estimation, transmission loss reduction, active power filter synthesis, and learning of artificial neural networks.

II. The Bacteria Foraging Optimization Algorithm

In the process of foraging, E. coli bacteria undergo four stages, namely, chemotaxis, swarming, reproduction, and elimination and dispersal. In search space, BFOA seek optimum value through the chemotaxis of bacteria, and realize quorum sensing via assemble function between bacterial, and satisfy the evolution rule of the survival of the fittest make use of reproduction operation, and use elimination-dispersal mechanism to avoiding falling into premature convergence.

A. 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 θi(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

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

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

B. Swarming

An interesting group behavior has been observed for several motile species of bacteria including E.coli and S. typhimurium. When a group of E. coli cells is placed in the center of a semisolid agar with a single nutrient chemo-effector, they move out from the center in a traveling ring of cells by moving up the nutrient gradient created by consumption of the nutrient by the group. To achieve this, function to model the cell-to-cell signaling via an attractant and a repellan. The mathematical representation for E.coli swarming [2] can be represented by:

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

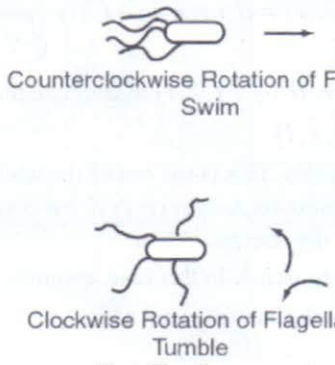

Fig.1 Flagella movements

$$= \sum_{i=1}^{S} [-d_{attract} \exp(-w_{attract} \sum_{m=1}^{P} (\theta_m - \theta_m^i)^2)]$$

$$+ \sum_{i=1}^{S} [h_{repellant} \exp(-w_{repellant} \sum_{m=1}^{P} (\theta_m - \theta_m^i)^2)]$$

where is the cost function value to be added to the actual cost function. S is the total number of bacteria and p is the number of parameters to be optimized which are present in each bacterium. $d_{attract}$ is the depth of the attractant released by the cell and $w_{attract}$ is a measure of the width of the attractant signal. $h_{repellant} = d_{attract}$ is the height of the repellant effect and $w_{repellant}$ is a measure of the width of the repellant.

C. Reproduction

According to the rules of evolution, individual will reproduce themselves in appropriate conditions in a certain way. For bacterial, a reproduction step takes place after all chemotactic steps.

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

Where J_{health}^i is the health of bacterium *i*. Sort bacteria and chemotactic parameters C(*i*) in order of ascending cost (higher cost means lower health). For keep a constant population size, bacteria with the highest J_{health} values die. The remaining bacteria are allowed to split into two bacteria in the same place.

D. Elimination-Dispersal

In the evolutionary process, elimination and dispersal events can occur such that bacteria in a region are killed or a group is dispersed into a new part of the environment due to some influence. They have the effect of possibly destroying chemotactic progress, but they also have the effect of assisting in chemotaxis, since dispersal may place bacteria near good food sources. From the evolutionary point of view, elimination and dispersal was used to guarantees diversity of individuals and to strengthen the ability of global optimization. In BFOA, bacteria are eliminated with a probability of ped.In order to keeping the number of bacteria in the population constant, if a bacterium is eliminated, simply disperse one to a random location on the optimization domain.

III. The BFOA Algorithm

[*Step 1*] *Initialization*

i. Number of parameters (p) to be optimized.
ii. Number of bacteria (S) to be used for searching the total region.
iii. Swimming length N_s after which tumbling of bacteria will be undertaken in a chemotactic loop.
iv. N_c the number of iterations to be undertaken in a chemotactic loop. (N_c>N_s).
v. N_{re} the maximum number of reproduction to be undertaken.
vi. N_{ed} the maximum number of elimination and dispersal events to be imposed over bacteria.
vii. P_{ed} the probability with which the elimination and dispersal events to be imposed over bacteria.
viii. The location of each bacterium P(1-p, 1-s, 1) which is specified be random numbers on [-1, 1].
ix. The value of C(*i*) which is assumed to be constant.
x. The values of $d_{attract}$, $w_{attract}$, $h_{repellant}$ and $w_{repellant}$

[*Step 2*] *Iterative algorithm for optimization*

This section models the bacteria population chemotaxis, swarming, reproduction, elimination and dispersal.

Elimination-dispersal loop: *l=l+1.*

Reproduction loop: *k=k+1.*

Chemotaxis loop: *j=j+1.*

a) For *i*=1, 2, …, S, calculate cost function value for each bacterium *i* as follows.
- Compute value of cost function *J(I, j, k, l)*. Let $J_{sw}(i, j, k, l) = J(i, j, k, l) + J_{cc}(\theta^i(j,k,l), P(j,k,l))$ (i.e., add on cell-to-cell attractant effect for swarming behaviour).
- Let $J_{last} = J_{sw}(i, j, k, l)$ to save this value since we may find a better cost via a run.
- End for stop loop.

b) For *i*=1, 2, ….., S take the tumbling/swimming decision
- 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].
- Move: let

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

Fixed step size in the direction of tumble for bacterium i is considered.

- Compute $J(i, j+1, k, l)$ and then let $J_{sw}(i, j, k, l)$ = $J(i, j, k, l) + J_{cc}(\theta^i(j,k,l), P(j,k,l))$
- Swim:

I. Let m=0; (counter for swim length)

II. While $m<N_s$ (have not climbed down too long)

- Let m=m+1
- If $J_{sw}(i, j+1, k, l)<J_{last}$ (if doing better), let $J_{last}=J_{sw}(i, j+1, k, l)$ 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 $\theta^i(j+1,k,l)$ to compute the new $J(i, j+1, k, l)$

- Else, let m=Ns. This is the end of the while statement.

c) Go to the next bacterium (i+1) if $i \neq S$ (i.e. go to b) to process the next bacterium.

If $j < N_c$, go to step 3. In this case, continue chemotaxis since the life of the bacteria s not over.

Reproduction

a) For the given k and l, and for each i=1, 2, ….S, let $J^i_{health} = \min_{j\in\{1\ldots N_c\}} \{J_{sw}(i,j,k,l)\}$ be the health of the bacterium i (a measure of how many nutrients got over its life time and how successful it was at avoiding noxious substance). Sort bacteria in order of ascending cost J_{health} (higher cost means lower health).

b) The S_r=S/2 bacteria with the highest J_{health} values die and other S_r bacteria with the best value split (and the copies that are made are placed at the same location as their parent)

If $k < N_{re}$ go to 2, in this case we have not reached the number of specified reproduction steps, so we start the next generation in the chemotactic loop.

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 a random location on the optimization domain.

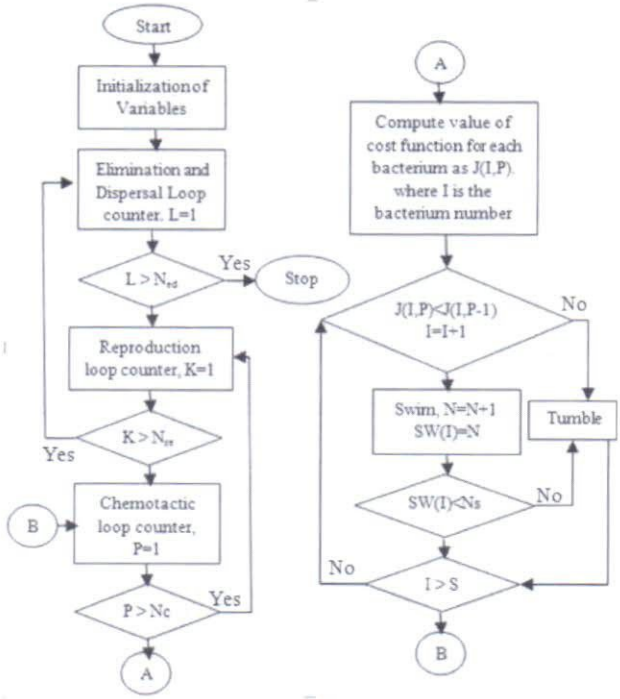

Fig. 2(a) Flow chart of Bacterial Foraging Algorithm

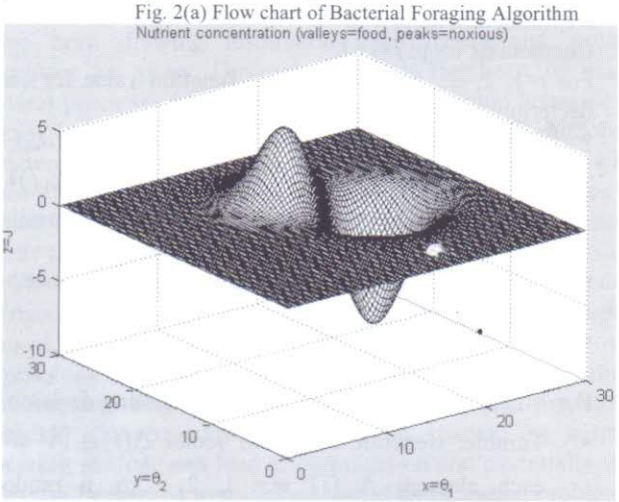

Fig.2(b) Contour of the test Function

Flow Chart of the above algorithm is shown in Fig. 2(a)

IV. SIMULATION AND RESULTS

The algorithm described above is simulated for the test function :

J = 5*exp(-0.1*((theta(1,1)-15)^2+(theta(2,1)-20)^2))-8*exp(-0.08*((theta(1,1)-20)^2+(theta(2,1)-15)^2)) in MATLAB 7.9.0. The following results are obtained when S=50, p=2, Nc=100, Ns=4, Nre=4, Sr=S/2, Ned=2, Ped=0.25.

Figure 2 shows the contour of the function where valleys are food and peaks are noxious. Figure 3 shows the bacteria movement towards the global optimization. Figure 4 shows the nutrients obtained by bacteria during life. Figure 5 shows the health of the bacteria i.e. Jhealth . Figure 6 shows the location of the bacteria P(1-p,1-s,1).

V. CONCLUSION

Bacterial Foraging Algorithm can be effectively used to solve optimization problems. The effectiveness of the BFOA is verified through simulation results on a test function and it provides a scope for more future work on this technique.

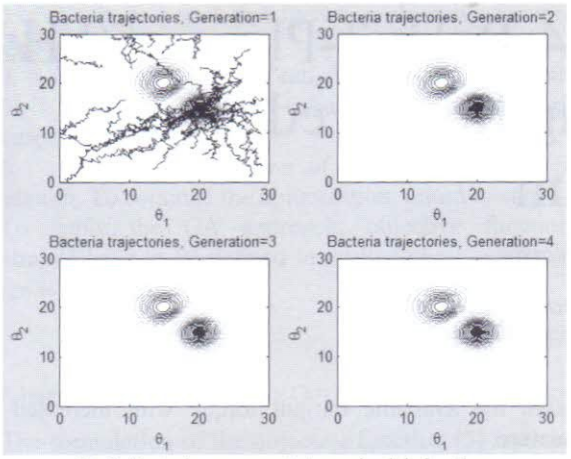

Fig.3: Bacteria movements towards global optima

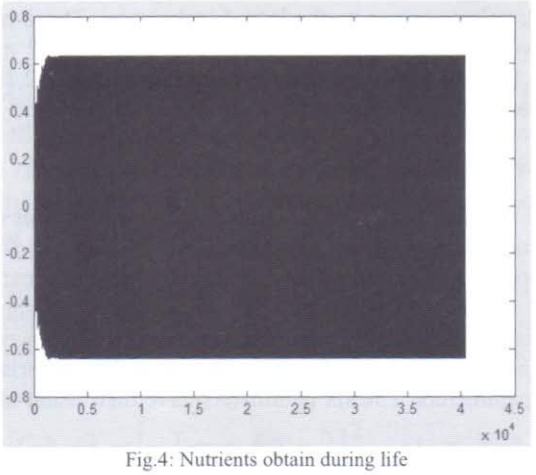

Fig.4: Nutrients obtain during life

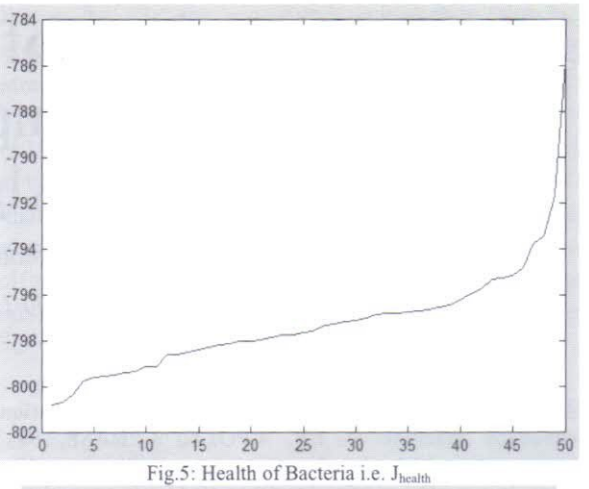

Fig.5: Health of Bacteria i.e. J_{health}

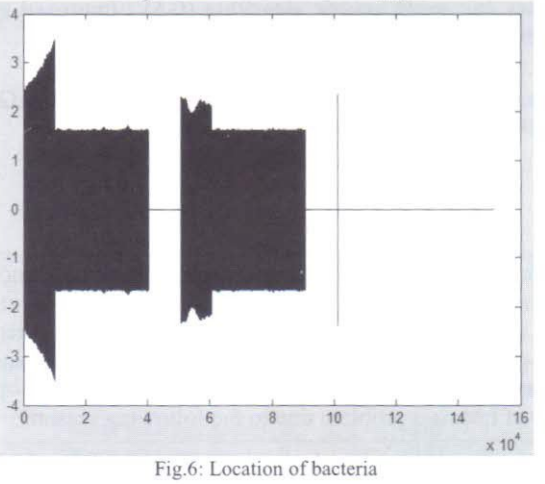

Fig.6: Location of bacteria

REFRENCES

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

[2] Y. Liu and K. M. Passino, Biomimicry of social foraging bacteria for distributed optimization: Models, principles, and emergent behaviors, *J. Optimization Theory Applicat.*, vol. 115, no. 3, pp. 603—628, Dec. 2002.

[3] D. H. Kim, A. Abraham, and J. H. Cho, A hybrid genetic algorithm and bacterial foraging approach for global optimization, *Inform. Sci.*, vol. 177, no. 18, pp. 3918—3937, 2007.

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

[5] M.Tripathy, S.Mishra, L.L.Lai and Q.P.Zhang, Transmissionloss reduction based on FACTS and bacteria foraging algorithm, in *Proc. PPSN*, 2006, pp. 222—231.

[6] S. Mishra, and C. N. Bhende, Bacterial foraging technique-based optimized active power lter for load compensation, *IEEE Trans. Power Delivery,* vol. 22, no. 1, pp. 457—465, Jan. 2007.

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

[8] B. D. Hughes, *Random Walks and Random Environments: Volume 1: Random Walks*. London, U.K.: Oxford Univ. Press, 1996.

[9] M. Avriel, *Nonlinear Programming: Analysis and Methods*.1sted.New York: Dover, 2003. ch. 10, sec. 10.1, pp. 288—299.

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

[11] J. Kennedy and R. C. Eberhart, "Particle Swarm Optimization," in *Proc. of the IEEE Int. Conf. on Neural Networks*. Piscataway, NJ: IEEE Service Center, 1995, pp. 1942–1948.

[12] Sambarta Dasgupta, Swagatam Das, Ajith Abraham, Adaptive Computational Chemotaxis in Bacterial Foraging Optimization: An Analysis, *IEEE Transactions on Evolutionary Computation,* VOL. 13, NO. 4, AUGUST 2009

[13] Qiaoling Wang, Xiao-Zhi Gao and Changhong Wang, "An Adaptive Bacterial Foraging Algorithm For Constrained Optimization," *International Journal of Innovative Computing, Information and Control,* Volume 6, Number 8, August 2010

[14] Hai Shen, Yunlong Zhu, Xiaoming Zhou, Haifeng Guo, Chunguang Chang, *Bacterial Foraging Optimization Algorithm with Particle Swarm Optimization Strategy for Global Numerical Optimization*, GEC'09, June 12–14, 2009, Shanghai, China.