# Comparative Analysis of Bacterial Foraging Optimization Algorithm with Simulated Annealing

**Nitin Kumar Jhankal[1], Dipak Adhyaru[2]**

[1]Global Institute of Technology, Jaipur, EPIP Sitapura Jaipur 302022, India

[2]Instrumentation & Control Department Institute of Technology, Nirma University, S.G Highway, Ahmedabad, Gujarat, India

**Abstract:** ***Bacterial foraging optimization algorithm (BFOA) has been widely accepted as a global optimization algorithm of current interest for optimization and control. BFOA is inspired by the social foraging behavior of Escherichia coli. BFOA has already drawn the attention of researchers because of its efficiency in solving real-world optimization problems arising in several application domains. Simulated annealing (SA) is a method for solving unconstrained and bound-constrained optimization problems. The method models the physical process of heating a material and then slowly lowering the temperature to decrease defects, thus minimizing the system energy. In present paper, a detailed explanation of this algorithm is given. Comparative analysis of BFOA with Simulated Annealing (SA) is presented.***

**Keywords:** Bacterial foraging, swarming, simulated annealing, optimization.

## 1. Introduction

Optimization problems defined by functions for which derivatives are unavailable or available at a prohibitive cost are appearing more and more frequently in computational science and engineering. Increasing complexity in mathematical modeling, higher sophistication of scientific computing, and abundance of legacy codes are some of the reasons why derivative-free optimization is currently an area of great demand.

In many physical applications, the true models or functions being optimized are extremely expensive to evaluate but, based e.g. on simplified physics or mesh coarsening, there are often surrogate models available, less accurate but cheaper to evaluate. In these circumstances, one would expect to design an optimization framework capable of extracting as much information as possible from the surrogate model while parsimoniously using the fine, true model to accurately guide the course of the optimization process. [1]

To solve the optimization problem efficient search or optimization algorithms are needed. There are many optimization algorithms which can be classified in many ways, depending on the focus and characteristics. Optimization algorithms can also be classified as deterministic or stochastic. If an algorithm works in a mechanical deterministic manner without any random nature, it is called deterministic. For such an algorithm, it will reach the same final solution if we start with the same initial point. Hill-climbing and downhill simplex are good examples of deterministic algorithms. On the other hand, if there is some randomness in the algorithm, the algorithm will usually reach a different point every time the algorithm is executed, even though the same initial point is used.

Search capability can also be a basis for algorithm classification. In this case, algorithms can be divided into local and global search algorithms. Local search algorithms typically converge towards a local optimum, not necessarily (often not) the global optimum, and such an algorithm is often deterministic and has no ability to escape from local optima. Simple hill-climbing is such an example. On the other hand, for global optimization, local search algorithms are not suitable, and global search algorithms should be used. A simple strategy such as hill-climbing with random restarts can turn a local search algorithm into an algorithm with global search capability. In essence, randomization is an efficient component for global search algorithms.

Obviously, algorithms may not exactly fit into each category. It can be a so-called mixed type or hybrid, which uses some combination of deterministic components with randomness, or combines one algorithm with another so as to design more efficient algorithms. [3]

### 1.1. Different type of Derivative free optimization

1) Genetic Algorithm
2) Simulated Annealing
3) Random Search Method
4) Swarm Optimization
5) Ant Colony Algorithm
6) Bacterial Foraging

Natural selection tends to eliminate animals with poor “foraging strategies” (methods for locating, handling, and ingesting food) and favor the propagation of genes of those animals that have successful forging strategies since they are more likely to enjoy reproductive success (they obtain enough food to enable them to reproduce). After many generations, poor foraging strategies are either eliminated or shaped into good ones (redesigned). Logically, such evolutionary principles have led scientists in the field of ”foraging theory” to hypothesize that it is appropriate to model the activity of foraging as an optimization process: A foraging animal takes actions to maximize the energy obtained per unit time spent foraging.

## 2. Basic Details: Bacterial Foraging Technology

### 2.1. Foraging: Element of Foraging Theory

Foraging theory is based on the assumption that animals search for and obtain nutrients in a way that maximizes their energy intake E per unit time T spent foraging. Hence, they try to maximize a function like

$$\frac{E}{T}$$

(or they maximize their long-term average rate of energy intake). Maximization of such a function provides nutrient sources to survive and additional time for other important activities (e.g., fighting, fleeing, mating, reproducing, sleeping, or shelter building). Optimal foraging theory formulates the foraging problem as an optimization problem and via computational or analytical methods [2].

### 2.2. Search Strategies for Foraging

Some animals are "cruise" or "ambush" searchers. For the cruise approach to searching, the forager moves continuously through the environment, constantly searching for prey at the boundary of the volume being searched (tuna fish and hawks are cruise searchers). In ambush search, the forager (e.g., a rattlesnake) remains stationary and waits for prey to cross into its strike range [2]. The search strategies of many species are actually between the cruise and ambush extremes.

### 2.3. Bacterial Foraging: E.coli

The *E. coli* bacterium has a plasma membrane, cell wall, and capsule that contains the cytoplasm and nucleoid (Figure 1). The pili (singular, pilus) are used for a type of gene transfer to other *E. coli* bacteria, and flagella (singular, flagellum) are used for locomotion [7]. The cell is about 1 μmin diameter and 2 μm in length [1]. The *E. coli* cell only weighs about 1 picogram and is about 70% water. *Salmonella typhimurium is* a similar type of bacterium [6].

### 2.4. Swimming and Tumbling via Flagella

Locomotion is achieved via a set of relatively rigid flagella that enable the bacterium to swim via each of them rotating in the same direction at about 100-200 revolutions per second (in control systems terms, we think of the flagellum as providing for actuation). Each flagellum is a left-handed helix configured so that as the base of the flagellum (i.e., where it is connected to the cell) rotates counterclockwise, as viewed from the free end of the flagellum looking toward the cell, it produces a force against the bacterium so it pushes the cell.

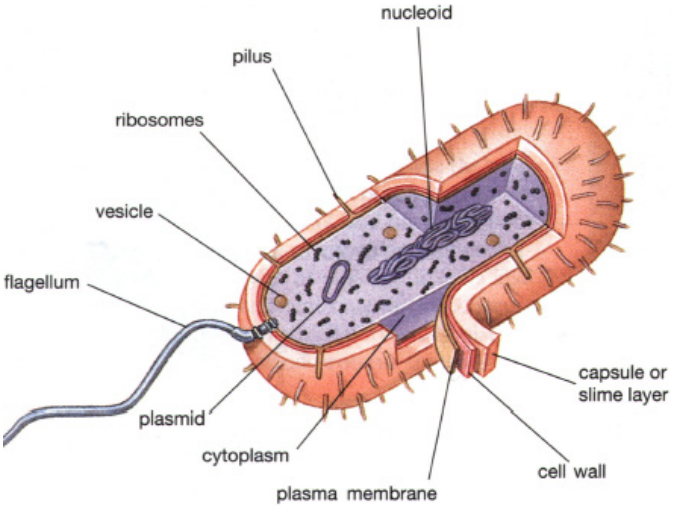


**Figure 1:** Bacterial foraging E.Coli [7]

An *E. coli* bacterium can move in two different ways; If the flagella rotate clockwise, each flagellum pulls on the cell, and the net effect is that each flagellum operates relatively independently of the others, and so the bacterium "tumbles" about (i.e., the bacterium does not have a set direction of movement and there is little displacement See Fig. 2. (a) [4]) If the flagella move counterclockwise, their effects accumulate by forming a bundle (it is thought that the bundle is formed due to viscous drag of the medium), and hence they essentially make a composite propeller and push the bacterium so that it runs (swims) in one direction (see Fig. 2 (a) [5]).

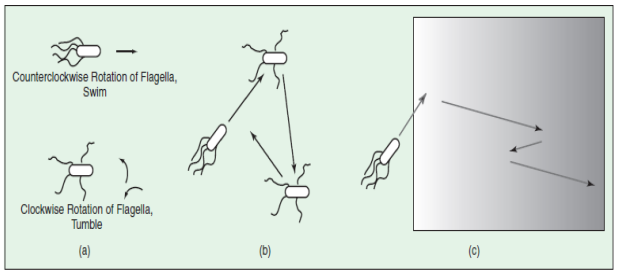


**Figure 2:** Swimming, tumbling and chemotactic behavior of E coli

## 3. Bacterial Foraging Optimization Algorithm

### 3.1. Steps for BFOA

**a) Chemotaxis:**
This process simulates the movement of an E.coli cell through swimming and tumbling via flagella. Suppose θi(j, k, l) represents the ith bacterium at jth chemotactic, kth reproductive, and lth elimination–dispersal step. C (i) is a scalar and indicates 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 unit length vector in the random direction.

**b) Swarming:**

Interesting group behavior has been observed for several motile species of bacteria including E.coli and S. typhimurium, where stable spatiotemporal patterns (swarms) are formed in semisolid nutrient medium. A group of E.coli cells arrange themselves in a traveling ring by moving up the nutrient gradient when placed amid a semisolid matrix with a single nutrient chemo-effecter. The cells when stimulated by a high level of succinate release an attractant aspartate, which helps them to aggregate into groups and, thus, move as concentric patterns of swarms with high bacterial density [8].

**c) 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.

**d) Elimination and Dispersal:**

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.

## 4. Simulated Annealing

Simulated annealing is a technique for combinatorial optimization problem, such as minimizing functions of very many variables. Because many real-world design problems can be cast in the form of such optimization problem, there is intense interest in general techniques for their solution. Simulated annealing is one such technique of rather recent vintage with an unusual pedigree: it is motivated by an analogy to the statistical mechanics of annealing in solids. To understand why such physics problem is of interest, consider how to coerce a solid into a low energy state. [3]

A low energy state usually means a highly ordered state, such as a crystal lattice. Simulated annealing techniques use an analogous set of “controlled cooling” operations for nonphysical optimization problems, in effect transformation a poor, unordered solution into a highly optimized, desirable solution. Simulated annealing offers an appealing physical analogy for the solution of optimization problems and more importantly, the potential to reshape mathematical insights from the domain of physics into insights for real optimization problem.

## 5. Example and Simulation

### 5.1. Function Optimization via Bacterial Foraging

As a simple illustrative example [2], we use the algorithm to try to Find minimum of function in Figure 4 ([15, 5] is the global minimum point, [20, 15] is a local minimum). Standard ideas from optimization theory can be used to set the algorithm parameters.

### 5.2. Simulation experiment

A function to be optimized is created with following MATLAB program:

```
function fposition=Live_fn(x)
    p=0; q=0;
    for k=1:5
    p=p+k*cos((k+1)*x(1)+k);
    q=q+k*cos((k+1)*x(2)+k);
    end
    fposition=p*q+(x(1)+1.42513)^2+(x(2)+.80032)^2;
```

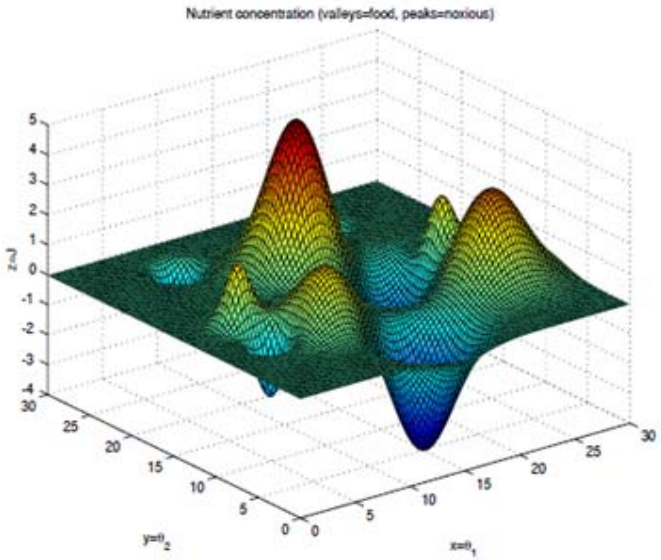


**Figure 3:** Function with multiple extremes point

The problem of optimization can be solved by two methods:

- Bacteria foraging optimization algorithm
- Simulated Annealing

Simulated annealing methods are methods proposed for the problem of finding, numerically, a point of the global minimum of a function defined on a subset of a k-dimensional Euclidean space. The motivation of the methods lies in the physical process of annealing, in which a solid is heated to a liquid state and, when cooled sufficiently slowly, takes up the configuration with minimal inner energy. Some of the literature, reports surprisingly good results when applying simulated annealing to difficult problems. The description of the results is often short on detail; not the number of steps in the method, but computer time is reported and the behavior of the function minimized, the position of local minima is difficult to ascertain.

As compared to that in Bacteria foraging optimization algorithm, we observe the behavior of the bacteria tumbling and swimming is expressed as chemotaxis equation which defines the movement of the bacteria. There is a swarming characteristic of the bacteria which is taken into account, where in the bacteria come together in large numbers in semisolid nutrient medium. On the basis of the above explanations, we observe that the Bacteria foraging algorithm differs from simulated annealing. Simulation results using both methods are presented in Fig. 4 and Fig. 5.

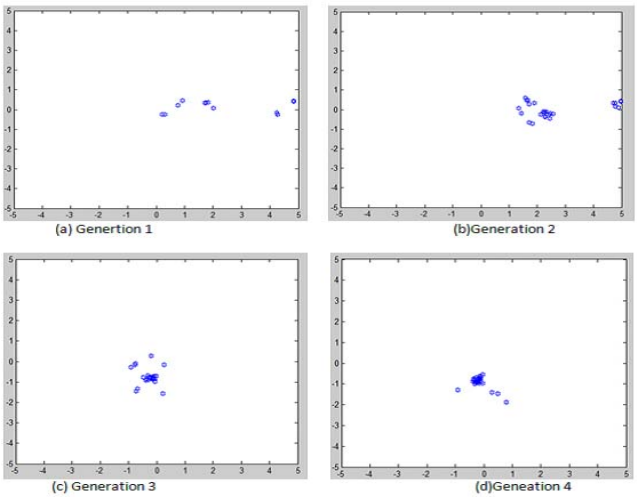


**Figure 4:** BFOA results

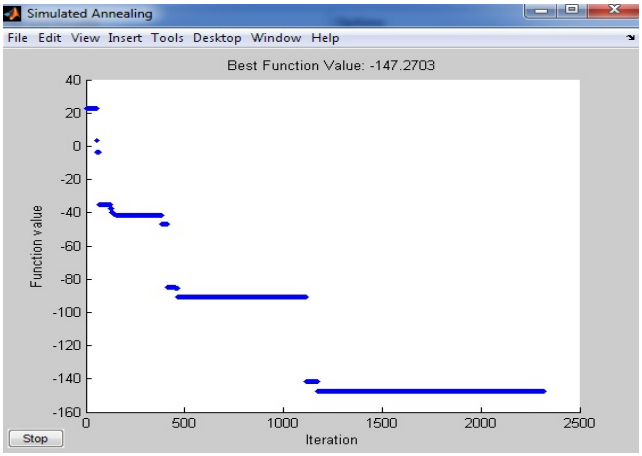


**Figure 5:** Fitness function plot by Simulated Annealing

### 5.3. Comment

We can see form the above results that the average fitness value for a simulated annealing algorithm is -147.270. According to the fitness function selected, lower the value better the generation. The best value of the fitness function achieved by bacterial foraging algorithm is -186.565 which is very good compare to simulated annealing. We can see from the above graphs of BFO that convergence to the minimum point occurs in about four generations while in case of SA after execution of around 100 generations we get minimum of a function. A comparative analysis of both algorithms is presented in the following table:

**Table 1:** Comparative analysis of BFOA and SA

| *S. No.* | *Factor* | *Bacterial Foraging Algorithm* | *Simulated Annealing* |
|---|---|---|---|
| 1. | Accuracy | More | Less |
| 2. | Fitness Function Value | -186.565 | -147.2703 |
| 3. | Time | 1.5340 sec | 3 sec |
| 4. | Optimum point | -1.2871<br>-0.7281 | -1.425<br>-7.081 |

## 6. Conclusion and Future Work

From above comparative data one can easily understand that Bacteria foraging optimization algorithm technique is better than the simulated annealing. Thus Bacterial Foraging algorithm, explains social foraging, Simulated annealing which thus makes it imperative to analyses strategies required for Global Optimization. Optimal Bacterial Foraging theory uses computational or analytical methods to provide an optimal foraging policy that specifies how foraging decisions are made. Hence the potential uses of Biomimcry of Bacterial Foraging optimization techniques are to develop adaptive controllers & co-operative control strategies for autonomous vehicles.

## References

[1] Jhankal, N.K., Adhyaru, D., "Bacterial foraging optimization algorithm: A derivative free technique," presented at the 2nd International Conference on current in technology, Ahmedabad, India, 8-10, Dec.2011.
[2] G. Kevin Passino,"Biomimicry of Bacterial Foraging for Distributed Optimization and Control," IEEE Control Systems Magazine, June 2002.
[3] V. Fabian, "Simulated annealing simulated," Computers math Applic. Vol. 33, no. 1/2, pp. 81-94, 1997.
[4] Stephens and J. Krebs, Foraging Theory. Princeton, NJ: Princeton univ. Press, 1986.
[5] W. O'Brien, H. Browman, and B. Evans, "Search strategies of foraging animals," Amer. Scientist," vol. 78, pp. 152-160, Mar. /Apr. 1990.
[6] H. Berg, Random Walks in Biology. Princeton, NJ: Princeton Univ. Press, 1993.
[7] D. De Rosier, "The turn of the screw: The bacterial flagellar motor," Cell, vol. 93, pp. 17-20, 1998.
[8] G. Lowe, M. Meister, and H. Berg, "Rapid rotation of flagellar bundles in swimming bacteria," Nature, vol. 325, pp. 637-640, Oct. 1987.
[9] Swagatam Das, Sambarta Dasgupta, Arijit Biswas, Ajith Abraham," On Stability of the Chemotactic Dynamics in Bacterial-Foraging Optimization Algorithm, '' IEEE Trans. Syst.,Man, Cybernet-Part A: systems and humans, vol. 39, no. 3, may 2009.

## Author Profile

**Nitin Kumar Jhankal** received the M.Tech in Instrumentation & Control Engineering from Institute of Technology, Nirma University in 2012 and B.E. degree in Electronic Instrumentation & Control Engineering from Govt. Engineering College Bikaner in 2009. He is now as an Assistant professor in Global Institute of Technology, Jaipur Rajasthan.

**Dipak Adhyaru** received his Ph.D, from IIT-Delhi. He is professor and Head in Instrumentation & Control Engineering, Department of Electrical Engineering, at Nirma University, Gujarat. His area of interest includes control engineering, soft computing and optimization. He published several research papers in International Journals and conferences.