Genetic Algorithm based Bacterial Foraging Approach for Optimization

Nikhil Kushwaha
NIT Hamirpur
Department of Electrical Engineering

Vimal Singh Bisht
NIT Hamirpur
Department of Electrical Engineering

Gautam Shah
3rd author's affiliation
1st line of address
2nd line of address

ABSTRACT
Bacterial foraging optimization algorithm (BFOA) has been widely accepted as a global optimization algorithm of current interest for distributed optimization and control. BFOA is inspired by the social foraging behavior of Escherichia coli. BFOA has already drawn the attention of researchers because of its efficiency in solving real world optimization problems arising in several application domains. The underlying biology behind the foraging strategy of E.coli is emulated in an extraordinary manner and used as a simple optimization algorithm. This paper proposes a genetic algorithm (GA) based bacterial foraging (BF) algorithms for function optimization. The proposed method using test functions and the performance of the algorithm is studied with an emphasis on mutation, crossover, variation of step sizes, chemotactic steps, and the lifetime of the bacteria.

General Terms
Chemotaxis, Cross-over and Mutation, Foraging is an optimization process created by evolution.

Keywords
Genetic algorithm, Bacterial foraging technique, optimization

1. INTRODUCTION
In the last decade, approaches based on genetic algorithms (GA) have received increased attention from the academic and industrial communities for dealing with optimization problems that have been shown to be intractable using conventional problem solving techniques [6,7,8,9,11,12,13,15]. A typical task of a GA is to find the best values of a predefined set of free parameters associated with either a process model or a control vector. System identification is one of the active areas of GA research [3,4,14,16,7]. Recent surveys of genetic algorithms, relating to improvements in the search process with respect to control system engineering problems, can be found in [14,16,17]. GA has also been used extensively to optimize nonlinear systems.

Usually, a possible solution to a specific problem is encoded as an individual (or a chromosome), which consists of a group of genes. Each individual represents a point in the search space and a possible solution to the problem can be formulated. A population consists of a finite number of individuals and each individual is decided by a fitness evaluation. Using this fitness value and suitable genetic operators, a new population is generated iteratively, with each iteration referred to as a generation. The GA uses basic genetic operators such as crossover and mutation to produce the genetic composition of a population. The crossover operator produces two offspring (new candidate solutions) by recombining the information from two parents. As the mutation operation is a random alteration of some gene values in an individual, the allele of each gene is a candidate for mutation, and its applicability is determined by the mutation probability. In the literature, much research has gone into the enhancement of conventional genetic algorithms [2,7,24].

Bacteria Foraging Optimization Algorithm (BFOA), proposed by Passino [1], is a new comer to the family of nature-inspired optimization algorithms. For over the last five decades, optimization algorithms like Genetic Algorithms (GAs) [3], Evolutionary Programming (EP) [3], Evolutionary Strategies (ES) [4], which draw their inspiration from evolution and natural genetics, have been dominating the realm of optimization algorithms. Recently natural swarm inspired algorithms like Particle Swarm Optimization (PSO) [2], Ant Colony Optimization (ACO) [4] have found their way into this domain and proved their effectiveness. Following the same trend of swarm-based algorithm Passino proposed the BFOA in [1]. Application of group foraging strategy of a swarm of E.coli bacteria in multi-optimal function optimization is the key idea of the new algorithm. Bacteria search for nutrients in a manner to maximize energy obtained per unit time. Individual bacterium also communicates with others by sending signals. A bacterium takes foraging decisions after considering two previous factors. The process, in which a bacterium moves by taking small steps while searching for nutrients, is called chemotaxis and key idea of BFOA is mimicking chemotactic movement of virtual bacteria in the problem search space.

Since its inception, BFOA has drawn the attention of researchers from diverse fields of knowledge especially due to its biological motivation and graceful structure. Researchers are trying to hybridize BFOA with different other algorithms in order to explore its local and global search properties separately.

2. GENETIC ALGORITHM
Genetic algorithms (GAs) are general purpose optimization algorithms, inspired from phenomena found in living nature. They are stochastic search techniques based on the mechanism of natural selection and natural genetics. GA starts with an initial set of random solutions called population. Each individual in the population is called chromosome, representing a solution to the problem. A chromosome is a string structure, typically a concatenated list of binary digits representing a coding of control parameters of a given problem. The chromosomes evolve through successive iterations, called generations. During each generation, the chromosomes are evaluated, using some measure of fitness. To create the next generation, new chromosomes called offspring are formed by either (a) merging two chromosomes from the current generation using a crossover operator or (b) modifying a chromosome using a mutation operator. A new generation is formed by selecting, according to the fitness value, some of the parents and offspring and rejecting others, so as to keep the population size constant. Suitable chromosomes having higher probabilities are being selected.

The selection rule used in our approach is a roulette-wheel selection. After several generations, the algorithms converge to the best chromosome, which hopefully represents the optimal or near optimal solution to the problem.

3. BACTERIA FORAGING OPTIMIZATION: A BRIEF OVERVIEW

The survival of species in any natural evolutionary process depend upon their fitness criteria, which relies upon their food searching and motile behavior. The law of evolution supports those species who have better food searching ability and either eliminates or reshapes those with poor search ability. The genes of those species who are stronger gets propagated in the evolution chain since they posses ability to reproduce even better species in future generations. So a clear understanding and modeling of foraging behavior in any of the evolutionary species, leads to its suitable application in any non-linear system optimization algorithm. The foraging strategy of *E. coli* bacteria present in the human intestine can be explained by four processes namely Chemotaxis, Swarming, Reproduction, Elimination and Dispersal [1][28].

a) Chemotaxis: The characteristics of movement of bacteria in search of food can be defined in two ways, i.e. swimming and tumbling together known as chemotaxis. A bacterium is said to be 'swimming' if it moves in a predefined direction, and 'tumbling' if moving in an altogether different direction. Mathematically, tumble of any bacterium can be represented by a unit length of random direction $\phi(j)$ multiplied by step length of that bacterium *C(i)*. In case of Swimming this random length is predefined.

b) Swarming: For the bacteria to reach at the richest food location (i.e. for the algorithm to converge at the solution point), it is desired that the optimum bacterium till a point of time in the search period should try to attract other bacteria so that together they converge at the solution point more rapidly. To achieve this, a penalty function based upon the relative distances of each bacterium from the fittest bacterium till that search duration, is added to the original cost function. Finally, when all the bacteria have merged into the solution point this penalty function becomes zero. The effect of Swarming is to make the bacteria congregate into groups and move as concentric patterns with high bacterial density.

c) Reproduction: The original set of bacteria, after getting evolved through several chemotactic stages reach the reproduction stage. Here, the best set of bacteria (chosen out of all the chemotactic stages) gets divided into two groups. The healthier half replaces the other half of bacteria, which gets eliminated, owing to their poorer foraging abilities. This makes the population of bacteria constant in the evolution process. The survival and elimination behavior of any bacterium is better known as its 'motile behavior'.

d) Elimination and Dispersal: In the evolution process a sudden unforeseen event can occur, which may drastically alter the smooth process of evolution and cause the elimination of the set of bacteria and/or disperse them to a new environment. Most ironically, instead of disturbing the usual chemotactic growth of the set of bacteria, this unknown event may place a newer set of bacteria nearer to the food location. From a broad perspective, elimination and dispersal are parts of the population-level long-distance motile behavior. In its application to optimization it helps in reducing the behavior of *stagnation*, (i.e. being trapped in a premature solution point or local optima) often seen in such parallel search algorithms. This section is based on the work in [11]. The detailed mathematical derivations as well as theoretical aspect of this new concept are presented in [10]-[11].

4. PROBLEM STATEMENT

The main goal of the GA based BF algorithm is to find the minimum of a function $J(\theta)$ where $\theta \in R^b$ (i.e. θ is a b- dimensional vector of a real numbers), and we do not have measurements or an analytical description of the gradient $\nabla J(\theta)$.

Test function [29]:

$$f_1(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$$

5. GA BASED BF OPTIMIZATION ALGORITHM

In GA based BF optimization algorithm there are no cell-to-cell attraction and repellent effects. New individuals are generated via mutating al the dimensions from the eliminated bacteria. The brief pseudo-code of GA based BFA has been provided below.

[Step 1] **Initialization:** Parameters Setting.

- p : Dimension of the search space.
- S : The number of bacteria in the population.
- Nc: Chemotactic steps.
- Ns: Swimming length.
- Nre: The number of reproduction steps.
- Ned: The number of elimination-dispersal events.
- Ped: Elimination-dispersal with probability.
- C(i) (i=1,2,…..S) : The size of the step taken in the random direction specified by the tumble.
- P(j,k,l):P(j,k,l)={θ^i(j,k,l) | i=1,2,…..S}.
- Generate random vector $\phi(j)$ which elements lie in [-1,1].

[Step 2] **Elimination Dispersal loop**: *l=l+1*

[Step 3] **Reproduction loop**: *k=k+1*

[Step 4] **Chemotaxis loop**: *j=j+1*

[4.1] Take a chemotactic step for every bacterium (i).

[4.2] Compute fitness function: $J(i, j, k, l)$, then let $J_{last}=J(i,j,k,l)$.

[4.3] For i=1,2,…S , take the tumbling/ swimming decision

- Tumble: Generate a random vector $\Delta \in R^b$ with each element $\Delta_m(i)m$=1,2….p , a random number on [0,1].
- Move: let

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

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

[4.5] Swimming loop: Let m=0 (counter for swim length). While m<Ns, $m=m+1$. If $J(i,j,k,l)<J_{last}$.

Let $J_{last}=J(I,j,k,l)$, let

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i)\phi(j),$$

Compute fitness function $J(I,j,k,l)$, else let m=Ns.

[4.6] Go to next Bacterium.

[Step 5] Apply crossover and mutation

[Step 6] If j < Nc , go to step 4. In this case, continue chemotaxis, since the life of the bacteria is not over.

[Step 7] **Reproduction:** Computer the health of the bacterium i: $J^{i}_{health} = \sum_{j=1}^{Nc+1} J(i, j, k, l)$ Sort bacteria and chemotactic parameters $C(i)$ in order of ascending cost J_{health}. Bacteria with the highest J_{health} values die, the remaining bacteria reproduce.
[Step 8] If (k < Nre), go to Step 3.
[Step 9] **Elimination-dispersal**: Eliminate and disperse bacteria with probability Ped.
[Step 10] If (l < Ned), go to Step 2.

6. RESULTS USING TEST FUNCTION

This section illustrates results using above test function. The initial conditions of objective values, parameter values, Chemotactic Steps (CS)=1000, total number of chemotactic reaction of bacteria, step sizes, basic unit for movement of bacteria the number of critical reaction (N)=1e-1000, the number of bacteria (S)=10, generations (G)=600, probability of mutation (Mu)=0.9, and probability of crossover (Cr)=0.1. Figure 2 shows the nutrients obtained by bacteria during life. Figure 3 shows the health of the bacteria i.e. J_{health} .

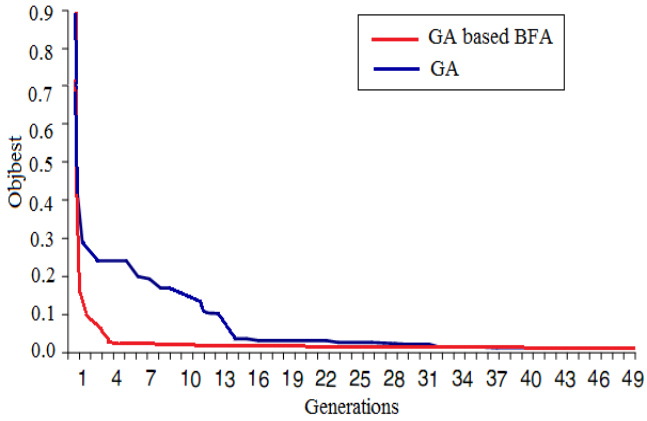

Fig:1 Comparison between GA and GA based BFA

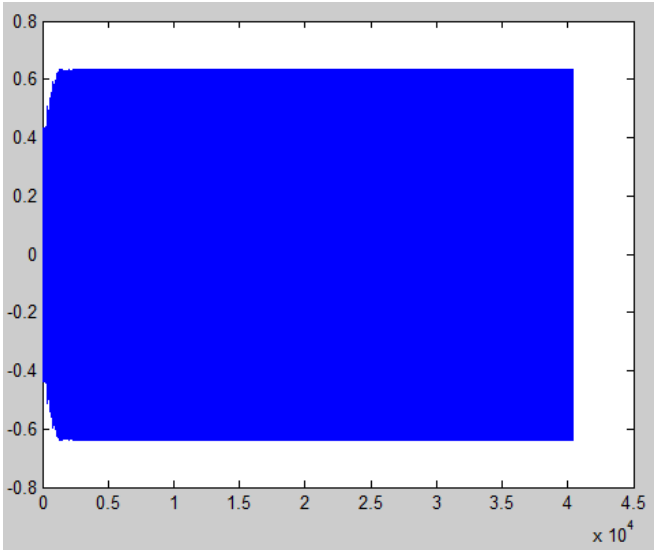

Fig.2: Nutrients obtain during life

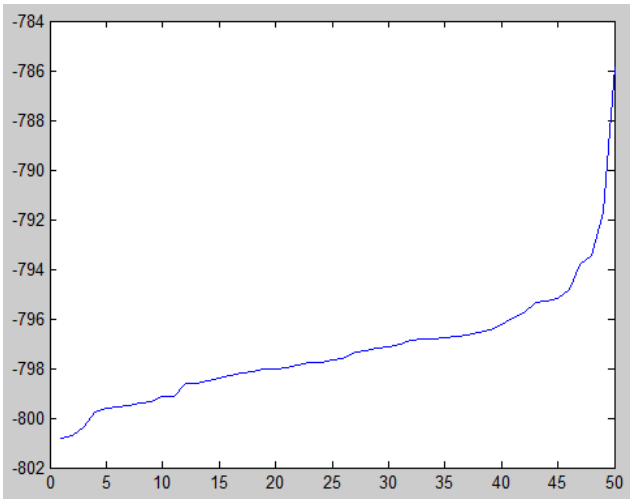

Fig.3: Health of Bacteria i.e. J_{health}

7. REFRENCES

[1] K. M. Passino, Biomimicry of bacterial foraging, IEEE Contr. Syst. Mag., pp. 5267, Jun. 2002.

[2] D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning. London, U.K.: Addison-Wesley, 1989.

[3] J. Kennedy, R. C. Eberhart, and Y. H. Shi, Swarm Intelligence. London, U.K.: Morgan Kaufmann, 2001.

[4] Sood, Y. R., Evolutionary programming based optimal power flow and its validation for deregulated power system analysis, Electrical Power and Energy Systems 29 (2007) 65–75.

[5] J. Alcock, Animal Behavior, An Evolutionary Approach, Sinauer Associates, Sunderland, Massachusetts, 1998.

[6] P. Angelov, A fuzzy controller with evolving structure, Information Sciences 161 (1–2) (2004) 21–35.

[7] J. Arabas, Z. Michalewicz, J. Mulawka, GAVaPS – A genetic algorithm with varying population size, in: Proceedings IEEE International Conference on Evolutionary Computation, Orlando, 1994, pp. 73–78.

[8] W.J. Bell, Searching Behavior, The Behavioral Ecology of Finding Resources, Chapman and Hall, London, England, 1991.

[9] M. Dotoli, G. Maione, D. Naso, E.B. Turchiano, Genetic identification of dynamical systems with static nonlinearities, in: Proceedings IEEE SMC Mountain Workshop Soft Computing Industrial Applications, Blacksburg, VA, 2001, pp. 65–70.

[10] P.J. Fleming, R.C. Purshouse, Evolutionary algorithms in control system engineering: A survey, Control Engineering Practice 10(2002) 1223–1241.

[11] C.M. Fonseca, P.J. Fleming, Multiobjective optimization and multiple constraint handling with evolutionary algorithms – Part I: A unified formulation; – Part II: Application example, IEEE Transactions on Systems Man and Cybernetics Part A – Systems and Humans 28 (1) (1998) 26–47.

[12] G.J. Gray, D.J. Murray-Smith, Y. Li, K.C. Sharman, T. Weinbrenner, Nonlinear model structure identification

using genetic programming, Control Engineering Practice 6 (1998) 1341–1352.

[13] D. Grunbaum, Schooling as a strategy for taxis in a noisy environment, Evolutionary Ecology 12 (1998) 503–522.

[14] K. Kristinnson, G.A. Dumont, System identification and control using genetic algorithms, IEEE Transactions on Systems Man and Cybernetics 22 (1992) 1033–1046.

[15] C.L. Lin, H.W. Su, Intelligent control theory in guidance and control system design: An overview, Proceedings of the National Science Council, Republic of China – Part A: Physical Science and Engineering 24 (1) (2000) 15–30.

[16] B. Maione, D. Naso, B. Turchiano, GARA: A genetic algorithm with resolution adaptation for solving system identification problems, in: Proceedings European Control Conference, Porto, Portugal, 2001, pp. 3570–3575.

[17] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, Springer-Verlag, New York, 1999.

[18] R. Tanese, Distributed genetic algorithm, in: Proceedings of International Conference on Genetic Algorithms, 1989, pp. 434–439.

[19] S. Tsutsui, D.E. Goldberg, Simplex crossover and linkage identification: Single-stage evolution vs. multi-stage evolution, in: Proceedings IEEE International Conference on Evolutionary Computation, 2002, pp. 974–979.

[20] H. Yoshida, K. Kawata, Y. Fukuyama, A particle swarm optimization for reactive power and voltage control considering voltage security assessment, IEEE Transactions on Power Systems 15 (2000) 1232–1239.

[21] D. P. Acharya, G. Panda, S. Mishra, and Y. V. S. Lakhshmi, Bacteria foaging based independent component analysis, in Proc. Int. Conf. Comput. Intell. Multimedia Applicat. Piscataway, NJ: IEEE Press, 2007, pp. 527—531.

[22] A. Chatterjee and F. Matsuno, Bacterial foraging techniques for solving EKFBased SLAM problems, in Proc. Int. Control Conf., Glasgow, U.K., Aug. 30—Sep. 1, 2006.

[23] R. P. Anwal, Generalized Functions: Theory and Technique. 2nd ed. Boston, MA: Birkhuser, 1998.

[24] R. Fletcher, Practical Methods of Optimization. 2nd ed. Chichester, U.K.: Wiley, 1987.

[25] K. Price, R. Storn, and J. Lampinen, Differential Evolution A Practical Approach to Global Optimization. Berlin, Germany: Springer-Verlag, 2005.

[26] H.-G. Beyer, Theory of Evolution Strategies, in Proc. Natural Computing Ser.. Berlin, Germany, Springer-Verlag, 2001.

[27] L. J. Fogel, A. J. Owens, and M. J. Walsh, Artificial Intelligence through Simulated Evolution. New York: Wiley, 1966.

[28] H.-G. Beyer, Toward a theory of evolution strategies: Self-adaptation, Evol. Comput., vol. 3, no. 3, pp. 311—347, 1996.

[29] Dong Hwa Kim, Ajith Abraham, Jae Hoon Cho–A hybrid genetic algorithm and bacterial foraging approach for global optimization, Information Sciences 177 (2007) 3918–3937.