A Study of Bacterial Foraging Optimization Algorithm and its Applications to Solve Simultaneous Equations

Gautam Mahapatra
Dept. of Comp Sc
Asutosh College
University of Calcutta, Kolkata, India

Soumya Banerjee
Dept. of Comp Sc & Engg
BIT Mesra Ext. Center Deoghar
Jharkand, India

ABSTRACT
For the solution of a set equation (linear or non-linear) with n number (n > 1) of variables we need at least n number of different relations (called as rank). Our present work is showing how the bio-inspired Bacteria Foraging Optimization Algorithm (BFOA), which is mimicry of the life-cycle of common type of bacteria like E.Coli, can be used to solve such system of equation with rank less than or equal to n. The BFOA simulates efficient nutrient foraging technique called as Chemotaxis to maximize the intake energy per unit time spend, the reproduction for evolution and the elimination-dispersal for environmental changes like any kind of natural calamities that are observed in the Bacterial system. As a sample tests we have used a numbers of system of linear equations with rank equal to the number of variables and a system of non-linear equations used in the derivation process of 4th order Runge-Kutta method for the ordinary differential equation solution, and experimental results are showing the applicability of the BFOA and in case of Runge-Kutta method we present an alternative form of the recursive equation.

General Terms
Evolutionary computing, System of linear and non-linear equations, Runge-Kutta method, Taylor method, Simulation Algorithm.

Keywords
BFOA, chemotaxis, tumble, swim, swarm, reproduction, elimination and dispersion, Total Square Errors, BFOASimulation.

1. INTRODUCTION
In 2002, Prof. K.M. Passino [14] proposed a new nature inspired computation technique after mimicking the food foraging, evolutionary reproduction and environmental elimination-dispersal behaviors of common Escherichia Coli (E.Coli) bacteria named as called as Bacteria Foraging Optimization Algorithm (BFOA). This is a meta-heuristic type algorithm, because it provides a general framework and set of guidelines for creating solution of a problem rather than providing the detailed structure and analysis. To simulate the natural and evolutionary properties this algorithm adds randomness and iterations respectively. This statistical bio-inspired optimization technique is a relatively new member in the swarm intelligence and researchers are considering this as new state-of-the art of the optimization paradigm [2]. A number of improvements and models have already been developed by the researchers and also a number of successful applications in the engineering and other fields are available in the literatures and also some theoretical analysis have been studied for the effectiveness of this optimization algorithm.

As an evolutionary computational technique, BFOA is also an iteration based optimization tool. As initialization all artificial or simulated bacteria are placed at random positions in multidimensional search space of problem and then measure the costs or fitness of these solutions and using these fitness find the global optimum solution among these solution positions. After this implement certain algorithm dependent operations on these individual bacteria to generate newer solutions. These processes are iterated until predefined objective is attended or maximum number of iterations passed.

Generally system of linear equations can be solved using several popular mathematical theory based techniques like exponential time complexity based Crammers rule, Matrix-inversion, numerical computation based techniques with polynomial time complexity like Gauss-elimination, Gauss-Jordon Elimination, LU Factorization and iteration based Gauss-Jacobi iteration, Gauss-Seidel iteration etc. methods [13][23][25]. In this work we have successfully used the BFOA to solve a system of linear equations, even for non-converging conditions required for iteration based techniques; also for some cases we obtain multiple solutions which is not possible for these popular techniques. It has been successfully implemented for system of linear and non-linear equations with rank less than or equal to the number of unknowns, but these are not applicable to other popular methods.

2. BFOA FUNDAMENTALS
The BFOA is a non-gradient, bio-inspired self-organizing natural and newly developed efficient optimization technique. In this technique the foraging, evolutionary reproduction and natural birth-death based elimination and dispersion strategies of common E.Coli bacteria survives in the complex human intestine system is mimicked. In complex and impossible problem domain, BFOA searches optimum living fuels i.e. energy intake per unit time, which is considered as the fitness, and it is collected by the bacterium using foraging behaves called as chemotaxis, and due to limited life span they survive through evolution based reproduction by the fittest bacterium and to provide variation in the bacteria society and hence to globalize the search space to avoid trapping into local and premature solution the natural calamities dependent birth-death based elimination-dispersal technique is used. Also, inter-communication based social swarming is considered for faster solution searches.

2.1 Chemotaxis
The random moving patterns that the bacteria generate in the presence of chemical attractants and repellants are called chemotaxis. For E.Coli, this process was simulated by two different moving modes ‘tumble ’and ‘run or move ’, and bacterium alternates between these modes until divided into

two. In tumble bacterium randomly searches a direction of moving and in run it moves a number of small fixed length steps $(C(i))$ in the selected tumble direction $(\emptyset(i))$ or better nutrients collection. Mathematically this can be expressed as

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i)\emptyset(i),$$
$$\forall i = 1,2,\dots,S$$

where S is size of the colony j,k and l are respectively the Chemotaxis, reproduction and elimination-dispersal step indices respectively and

$\theta^i(j+1,k,l)\ or\ \theta^i(j,k,l) \in D_1 \times D_2 \times \dots \times D_N$

$(in\ short\ it\ is\ represented\ \theta^i)$

be position or solution vector for i^{th} bacterium, $(D_i^{'}s\ \ are\ domain\ of\ \mathbb{R}^N Search\ Space)$.

If $J(\theta^i(j,k,l))$ be the cost or fitness function then bacterium uses run if $J(\theta^i(j+1,k,l))$ is better than $J(\theta^i(j,k,l))$, otherwise it enters into the next tumbling step.

For tumble

$$\emptyset^i(m) = \frac{\Delta(m)}{\sqrt{\Delta^T(m)\Delta(m)}}, \forall m = 1,2,\dots,N$$

$\Delta(m)$ is a random number in $[0,1]$

and $\emptyset(i) = \{\emptyset^i(m)\}_{m=1}^N$.

2.2 Reproduction

As bacteria are not immortal and like to grow population for better social structure they uses rule of evolution and when appropriate conditions appear then individual will reproduce themselves after certain number of chemotaxis steps. For this purpose health of the bacteria, which is sum of fitness in each chemotaxis including initialization step is considered $J^i_{healt\,h} = \sum_{j=1}^{N_c} J(\theta^i(j,k,l))$ and to keep constant population bacterium with better health capture the position of bacterium with poor heath, for this purpose individual reproduce one of its identical clone.

2.3 Elimination-Dispersal

In the evolutionary process, elimination and dispersal events occur such that bacteria in a region are eliminated or a group is dispersed from current location and may reappear in the other regions due to environmental changes or some natural calamities. They have the effect of possibly destroying chemotactic progresses, but they also have the effect of assisting the chemotaxis, since dispersal may place bacteria near good food sources. From evolutionary point of view, elimination and dispersal was used to guarantee diversity of the individuals and to strengthen the ability of global optimization. In BFOA, bacteria are eliminated with a probability P_{ed}, and to keep population size constant, if a bacterium is eliminated, simply disperse one new bacterium to a random location of the search space.

2.4 Swarming

Prof. Passino had experimented for an interesting group behavior of the E.Coli bacteria and then he was successful to explain this swarming behavior using the following mathematical model. He observed that when a group of E.Coli bacteria is placed in the center of a semisolid agar with a single nutrient chemo-effecter, 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, and this cell-to-cell signaling attractant and a replant based network group can be modeled:

$$J^i_{cc}\left(\theta^i_{best}(j,k,l),\theta_{best}\right) = Penalty(i)$$
$$= \sum_{i=1}^{S}(-d_{attract}\ e^{-w_{attract}\ \sum_{j=1}^{N}(\theta^{'j}_{best} - \theta^{ij}_{best})^2}) +$$
$$\sum_{i=1}^{S}(-h_{repellant}\ e^{-w_{repellant}\ \sum_{j=1}^{N}(\theta^{'j}_{best} - \theta^{ij}_{best})^2})$$

Where $J^i_{cc}\left(\theta^i_{best},\theta_{best}\right)$ is the cell-to-cell cost or penalty $(Penalty(i))$ for i^{th} bacteria in the colony, $\theta^{'j}_{best}$ is the j^{th} component of the current N-dimensional global best solution vector $\theta^{'}_{best}$ and θ^{ij}_{best} is the j^{th} component of N-dimensional best solution vector attended by the i^{th} bacteria in the colony. Other parameters are attractant and repellant dependent constant, $d_{attract}$ - is the depth of the attractant released by the bacterial cell and $w_{attract}$ is the measure of the width of the attract signal. Generally $h_{repellant}$ is the height of the repellant effect and $w_{repellant}$ is a measure of the width of the repellant.

3. BFOA Algorithm

Following pseudo code structure is representing the step-by-step details of the simulation of BFOA used in our study.

Algorithm BFOASimulation

(* S-Size of colony or population, N_c - Number of Chemotaxis Steps, N_s-Number of Swimming Steps, N_{re}-Number of reproduction steps, N_{ed}-Number of elimination and dispersal steps, P_{ed}-Probability of elimination and dispersal, C_i- Constant step size for Chemotaxis, $d_{attract}$ & $w_{attract}$ - Chemotactic Attraction parameters for swarming, $h_{repellant}$ & $w_{repellant}$ - Chemotactic Repulsion parameters for swarming, N-Dimension of solution vector or problem or number of optimizing parameters, h^i - is the health of the i^{th} bacterium, $J(\theta)$ - is the objective or fitness function*)

Step 1: [Initialization of Simulation Parameters]

1.1 $read\ S, N_c, N_s, N_{re}, N_{ed}, P_{ed}, C_i, d_{attract}$, $w_{attract}, h_{repellant}, w_{repellant}$

1.2 $for\ i = 1\ to\ S\ do$
begin
Randomly generate an N-dimensional position vector $\theta^i \in D_1 \times D_2 \times \dots \times D_N$.
Calculate fitness value: $J^i = J(\theta^i)$.
Initialize the i^{th} bacterium with $J^i_{best} = J^i$, $\theta^i_{best} = \theta^i$ and $h^i = 0$
end

1.3 Find the initial global best solution vector $\theta_{best} = \theta^i$ and fitness $J_{best} = J^i$ from current bacteria colony

1.4 Initialize Elimination-dispersal loop index $l = 1$

Step 2: [Perform all elimination-dispersals]
$repeat\ Step\ 3\ thru\ Step\ 10\ while\ l \le N_{ed}\ do$

Step 3: [Perform all reproductions]
$k = 1$
$repeat\ Step\ 4\ thru\ Step\ 8\ while\ k \le N_{re}\ do$

Step 4: [Perform all chemotaxis steps]
$j = 1$
$repeat\ Step\ 5\ thru\ Step\ 7\ while\ j \le N_c\ do$

Step 5: [For each bacterium in the colony do the chemotaxis]
$for\ i = 1\ to\ S\ do$
begin

5.1 Tumble

Generate a random number $\Delta(m)$ in [0,1]

$$\emptyset^i(m) = \frac{\Delta(m)}{\sqrt{\Delta^T(m)\Delta(m)}}, \forall m = 1,2,\dots,N$$

$$\emptyset(i) = \{\emptyset^i(m)\}_{m=1}^{N}$$

5.2 Move

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

5.3 Compute fitness

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

if(J^i is better than J_{best}) then

set $J_{best} = J^i$

$\theta_{best} = \theta^i(j+1,k,l)$

endif

if(J^i is better than J_{best}^i then

set $J_{best}^i = J^i$

$\theta_{best}^i = \theta^i(j+1,k,l)$

$h^i = h^i + J^i$

endif

5.4 Swim

$p = 0$

$while\left(p \le N_s \wedge \left(J^i\ is\ better\ than\ J_{best}^i\right)\right) do$

begin

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

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

if(J^i is better than J_{best}) then

set $J_{best} = J^i$

$\theta_{best} = \theta^i(j+1,k,l)$

endif

if(J^i is better than J_{best}^i then

set $J_{best}^i = J^i$

$\theta_{best}^i = \theta^i(j+1,k,l)$

$h^i = h^i + J^i, p = p+1$

endif

end

Step 6: [Update the chemotaxis step counter]

$j = j + 1$

Step 7: [Reproduction]

7.1 Arrange bacteria in descending order of health

$Sort(Colony)$

7.2 Reproduce

Replace all $S/2$ number of relatively unhealthy bacteria appearing at the end half of the sorted list by the copy of bacteria with better health appearing towards the beginning of the list.

Step 8: [Update reproduction Counter]

$k = k + 1$

Step 9: [Perform eliminate and dispersal of bacteria colony]

For each bacterium $i = 1\ to\ S$ in the colony with probability P_{ed}, eliminate and disperse. For this purpose generate a random probability (r) and check whether it is greater or equal to the given elimination probability P_{ed} or not, if successful then eliminate the bacterium and place it at a random position in the search space $\theta^i \in D_1 \times D_2 \times \dots \times D_N$, calculate fitness value $J^i = J(\theta^i)$ and then initialize the i^{th} bacterium with $J_{best}^i = J^i$, $\theta_{best}^i = \theta^i$ and $h^i = 0$, if lucky also update the global best $\theta_{best} = \theta^i$ and fitness $J_{best} = J^i$.

Step 10: [Update eliminate – dispersal counter]

$l = l + 1$

Step 11: [Finished]

Return solution vector θ_{best} and fitness J_{best}.

4. SYSTEM OF EQUATIONS WITH BFOA COST FUNCTION DESIGN

As application of BFOA we have studied solution of different system of linear and non-linear equations. In our study we have used two cases – first case rank is equal to the number of unknowns or variables and for second case rank is less than the number unknowns.

4.1 System of Linear Equations with Rank Equal to Number of Unknowns

Systems of Linear Equations are used to model different system behaviors in the field of Science, Social Science and Engineering applications. Finding solution to these set of equations through the evolutionary process of bio-mimic BFOA is a new and developing research area of interest. It is a set of non-homogeneous equations, and to obtain a solution, the system should be non-singular and have a point in space where it coincides.

A system of linear equations has the form:

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1N}x_N = d_1$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2N}x_N = d_2$$

$$\dots\ \dots$$

$$a_{N1}x_1 + a_{N2}x_2 + \dots + a_{NN}x_N = d_N$$

Subject to $x_i^L \le D_i \le x_i^U\ or\ x_i^L \le x_i \le x_i^U, \forall i = 1,2,\dots,N$, where x_i^L and x_i^U are lower and upper limit for i^{th} component of the N-dimensional solution vector

$X = (x_1, x_2, \dots, x_N) \in D_1 \times D_2 \times \dots \times D_N$.

This set of equations can be represented in matrix-vector form as $AX = b$, where A is the coefficient matrix with determinant value non-zero and X is the solution vector and b is the constant vector and can be solved using different popular matrix methods, numeric algorithm methods and iterative methods available in [13][23][25].

This system of linear equations can be solved using BFOA, if we can formulate a proper objective or fitness function, and a trial solution will be correct if it can satisfy this objective function. In our method we have considered sum of square errors as the objective or fitness or cost function and this is defined as follow:

$X = (x_1, x_2, \dots, x_N)$ - will be a solution for the above system of linear equations if we have

$\sum_{i=1}^{N} |f_i(x_1, x_2, \dots, x_N) - c_i| = 0,$

where $f_i(x_1, x_2, \dots, x_N) - c_i = a_{i1}x_1 + a_{i2}x_2 + \dots + a_{iN}x_N - d_i = \delta_i\ (error), \forall i = 1,2,\dots,N$

So the objective function is:

minimize $J(\theta(j,k,l)) = \sum_{i=1}^{N} \delta_i^2$,

where δ_i^2 is the square error and $X = (x_1, x_2, \dots, x_N) = \theta(j,k,l)$ is the solution or position vector of the bacteria.

4.2 System of Non-Linear Equations with Rank Less Than Number of Unknowns

Derivation of Runge-Kutta Method: To study the dynamic behavior of any system the gradient or variation relations of the system are expressed as a set of differential equation of equal or different orders. Most of the simple form is the first

order form called as Ordinary Differential Equations (ODE). The solutions of these ODEs describe the functional behavior of the system and for this purpose different calculus oriented methods like integrating factor etc are used, and to find the numerical values a number of numerical optimization techniques like Euler's method, Taylor's series method, Runge-Kutta method etc used. The Runge-Kutta method is very much efficient, as it avoids all kind of differentiation required in Taylor Series method. In the derivation of the RK4 method we get eleven independent algebraic relations for thirteen different coefficients and these are forming a system of equations with rank less than number unknowns. To find the value of unknowns Runge-Kutta had assigned fixed values for two such variables and then solve the system of equation using numeric algorithm [13], finally we got the most popular recursive equation for evaluation.

ODE is of the form: $\frac{dy}{dx} = f(x, y)$

with initial condition $x = x_0, y = y_0$ the solution will be of the form $y = g(x)$. Different complex forms of $f(x, y)$ are possible and to get $g(x)$ different techniques are used and for numeric solution different approximate numeric algorithms are used. Most popular and accurate numeric method is Runge-Kutta method. This method is derived from Taylor Series method after approximating it upto 4th order term in the series.

Runge-Kutta assumed the recursive solution of the following form:

$$y_{k+1} = y_k + w_1 k_1 + w_2 k_2 + w_3 k_3 + w_4 k_4$$

where

$$k_1 = hf(x_k, y_k)$$

$$k_2 = hf(x_k + a_1 h, y_k + b_1 k_1)$$

$$k_3 = hf(x_k + a_2 h, y_k + b_2 k_1 + b_3 k_2)$$

$$k_4 = hf(x_k + a_3 h, y_k + b_4 k_1 + b_5 k_2 + b_6 k_3).$$

After coefficient comparison with the Taylor's series method following equation are generated to determine the values of $w_i's$ and $k_i's$.

$$b_1 = a_1$$
$$b_2 + b_3 = a_2$$
$$b_4 + b_5 + b_6 = a_3$$
$$w_1 + w_2 + w_3 + w_4 = 1$$
$$w_2 a_1 + w_3 a_2 + w_4 a_3 = \frac{1}{2}$$
$$w_2 a_1^2 + w_3 a_2^2 + w_4 a_3^2 = \frac{1}{3}$$
$$w_2 a_1^3 + w_3 a_2^3 + w_4 a_3^3 = \frac{1}{4}$$
$$w_3 a_1 b_3 + w_4 (a_1 b_5 + a_2 b_6) = \frac{1}{6}$$
$$w_3 a_1 a_2 b_3 + w_4 a_3 (a_1 b_5 + a_2 b_6) = \frac{1}{8}$$
$$w_3 a_1^2 b_3 + w_4 (a_1^2 b_5 + a_2^2 b_6) = \frac{1}{12}$$
$$w_4 a_1 b_3 b_6 = \frac{1}{24}$$

For this system of equations we have thirteen unknowns and eleven relations, so the rank is less than number of unknowns. Runga-Kutta equalized these after assuming $a_1 = \frac{1}{2}, b_1 = 0$, then solved these equations and finally derived the recursive relation:

$$y_{k+1} = y_k + h\left(\frac{k_1 + 2k_2 + 2k_3 + k_4}{6}\right)$$

with the solutions

$$a_2 = \frac{1}{2},\ a_3 = 1,\ b_2 = b_3 = \frac{1}{2}, b_4 = b_5 = 0, b_6 = 1, w_1 = w_4 = \frac{1}{6}, w_2 = w_3 = \frac{1}{3}$$

For BOFA technique we use these thirteen constants as $N = 13$ -dimensional position vector:

$$\theta = (a_1, a_2, a_3, b_1, b_2, \dots, b_6, w_1, w_2, w_3, w_4)$$

and difference of left hand sides with the right hand sides of these equations for certain position vector is the respective error δ_i and sum of these square error as objective function:

minimize $J(\theta(j,k,l)) = \sum_{i=1}^{N} \delta_i^2$,

5. RELATED WORKS

5.1 The Analysis of BFOA

Most important step in BFOA is chemotaxis. In 2009, S. Dasgupta and S. Das et al. [19] made an analysis of the chemotaxis operation in BOFA. The analysis undertaken provides important insights into the search mechanism of BOFA.

The analysis points out that the chemotaxis usually results in sustained oscillation, especially on flat fitness landscapes, when a bacterium cell is close to the optima. Therefore, it is necessary to bound on the chemotactic step-height parameter that avoids limit cycles and guarantees convergence of the bacterial dynamics into an optimum. Two simple schemes for adapting the chemotactic step-height have been subsequently proposed.

In the same year, A. Abraham and A. Biswas [1] provided a simple mathematical analysis of the reproduction step used in BFOA. The analysis focuses on the reproduction in a simple two-bacterial system working on a one dimensional fitness landscape and is showing that the reproduction event contributes to the quick convergence of the bacterial population near optima.

In 2009, S.Das and S.Dasgupta et al. [20] provided an analysis on the stability of chemotactic dynamics using the concept of Lyapunov stability theorems and show that the bacterial dynamics exhibits an asymptotically stable behavior with respect to the single optimum for certain constraint on step size.

5.2 Improved BFOAs

BFOA use function to model the cell-to-cell signaling via an attractant and a repellant. However, its value does not depend on the nutrient concentration at position θ. In 2002, Y. Liu and K. M. Passino [28] used a new function to represent the environment-dependent cell-to-cell signaling –

$$J_{ar}\left(\theta^i(j,k,l)\right) = e^{(M - J\left(\theta^i(j,k,l)\right))J_{cc}\left(\theta^i(j,k,l)\right)}$$

where M is a tunable parameter. Then, for swarming, we need to minimize the new objective or cost function will be:

$$J\left(\theta^i(j,k,l)\right) + J_{ar}(\theta^i(j,k,l)).$$

By performing social foraging with chemical-attractant-induced swarming, E. Coli have better chance in locating the optimal point in a noisy environment.

H. Chen et al. [6]-[10] have introduced the idea of exploration and exploitation for they have used adaptive strategy called as the producer-scrounger foraging which dynamically determine the chemotactic step size for the whole bacteria colony during a run and self-adaptive foraging for single bacteria for area concentrated search.

Tripathy et al. [17] proposed an improved BFOA using two approaches (i) in order to speed up the convergence, the average value is replaced by the minimum value of all the chemotactic cost functions for deciding the bacterium health; (ii) for swarming, the distances of all the bacteria in a new chemotactic step are evaluated from globally optimum bacterium to these points and not the distance of each bacterium from the rest of the others.

Considering BFOA lacks in adaptation according to the operating condition, S. Mishra [21] presented a new algorithm Fuzzy Bacterial Foraging (FBF). FBF uses variable run length in the chemotaxis step in place of the original constant through a Takagi-Sugeno type fuzzy inference scheme. The results are showing that FBF has superior performance than GA when applied to the harmonic estimation problem.

W. J. Tang et al. [27] proposed a dynamic bacterial foraging algorithm (DBFA) which aims at optimization in dynamic environments. The DBFA adopts a selection scheme which enables the bacteria to flexibly adapt to the changing environment. Compared with BFA, DBFA is able to provide satisfactory performance, and can react to most of the environmental changes in time.

W. Korani [26], H. Shen et al.[11] and S. S. Patnaik et al.[24] proposed an improved BFOA using hybridization with PSO, namely BF-PSO. The BF-PSO combines both algorithm BF and PSO. The aims are to make use of PSO ability to exchange social information and BF ability in finding a new solution by elimination and dispersal. In BFOA, a unit length direction of tumble behavior is randomly generated and this random direction may lead to delay in reaching the global solution, for this reason idea of PSO for global best position is used to find the best position of bacteria. More hybridization is observed in GA-BF and these are proposed by D. H. Kim and A. Abraham et al. [5] and its application by N. Kushwaha et al.[18].

H. Chen et al. [8] proposed a multi-colony BFOA called as MC-BFO, which integrates the cell to-cell communication strategies of multi-colony bacterial community with the chemotaxis behavior of single cell and for this purpose it combines two algorithms BFOA and the Bacterial Chemotaxis Algorithm (BCA), which are both inspired by the bacterial chemotactic behavior of E. Coli.

M. S. Li et al. [16] have shown how BFOA requires less time in the more realistic model of bacterial foraging in variable population environment and this idea breaks the fixed population concept of most evolutionary algorithms (EA).

5.3 Significant Applications of BFOA

BFOA was used for solving highly non-linear and non-convex problems like optimal current harmonic mitigation by non-linear Active Power Filter (APF), power distribution of RFID based sensor network using MCBFOA [8] and Adaptive BFOA [10]. Modified BFOA used for active filter design by S. Mishra et al.[22] and L. Ulagammai et al. [15] used BFOA as learning method for Wavelet-based Artificial Neural Network (WANN) and identify the inherent non-linear characteristics of power system loads. BFOA can be used general optimization problems like Job Shop Scheduling problem [3].

5.4 Solving System of Equations using Evolutionary Computing Techniques

We took inspiration from similar approaches undertaken using Genetic Algorithm in the year 2011 by Ikotun Abiodun M et al. [12], they have used squared multiple correlation coefficient as fitness function for the simultaneous solution of a system of linear equations with rank equal to the number of unknowns.

6. SIMULATION AND RESULTS

For simulation we use following parameter values in most of the simulations:

$S = 20,\ N_c = 200,\ N_s = 5,\ N_{re} = 8,\ N_{ed} = 4, P_{ed} = 0.25, C_i = 0.01, d_{attract} = 0.1, w_{attract} = 0.2, h_{repellant} = 0.1, w_{repellant}$ =10

Table I. Results of Linear Equation Solution

Set	BFOA	Gen	GE	GS
$x + 2y + 3z = 14$ $x + y + z = 6$ $3x + 2y + z = 10$	2.6667 -1.3333 4.6667 & 4.1131 -4.2258 6.1130	2	2.6667 -1.3333 4.6667	NaN NaN NaN
$2x + 4y + z = 5$ $4x + 4y + 3z = 8$ $4x + 8y + z = 9$	0.4980 0.7510 1.0012	4	0.5000 0.7500 1.0000	NaN NaN NaN
$10x + y + z = 12$ $2x + 10y + z = 13$ $2x + 2y + 10z = 14$	1.0002 1.0004 0.9996	5	1.0000 1.0000 1.0000	1.000 1.000 1.000
$x + 2y + 3z = 6$ $2x + 4y + z = 7$ $3x + 2y + 9z = 14$	0.9871 1.0042 1.0034	7	1.6667 1.0000 1.3333	NaN NaN NaN
$2x + y + 3z = 13$ $x + 5y + z = 14$ $3x + y + 4z = 17$	0.9844 2.0022 3.0110	1	1.0000 2.0000 3.0000	NaN NaN NaN
$2x + 4y + 8z = 44$ $2x + 6y + 10z = 66$ $6x + 4y + 10z = 84$	13.3339 17.6704 -6.6689	8	13.3333 17.6667 -6.6667	NaN NaN NaN
$4w + 3x + 2y + z = 10$ $3w + 2x + y + 4z = 9$ $2w + x + 4y + 3z = 14$ $3w + 2x + y + 10z = 10$	6.9879 -7.1538 1.6689 0.1678	1	7.0000 -7.1667 1.6667 0.1667	NaN NaN NaN NaN

We had simulated individual bacterium and corresponding colony using object oriented technology in Microsoft Visual Studio.net (Visual C#) which is used as implementation programming language and then a number of linear equation with and without converging conditions as object oriented problem environment in each cases. We get successful solutions and compared with other standard non-iterative and iterative methods, results are very much satisfactory and these are shown in the Table I. For the first set of linear equations we get two solutions. Here number generation column is representing the number reproductions steps used to generate the required solutions. In last column of the table we are presenting the results from Gauss-Seidel iteration method, most of the cases results are 'NaN' (not a number) because set of equations are not diagonally dominant and cannot be

arranged in diagonal dominant form, this indicates that the iteration method is not applicable for these sets of equations.

One more experiment had been done for eleven non-linear equations with thirteen unknowns used by Runge-Kutta in the derivation of the recurrence relation for the fourth order convergence of the solution of ordinary first order differential equation. We get different values for these thirteen variables which were found by Runge-Kutta, this is shown in Table II, so we get a new form of the recurrence relation with equal performance, even in some cases it outperforms.

Table II. Results of Linear Equation Solution

Parameters	RK Solution	BFOA Solution
a_1	0	0.197491
a_2	$\frac{1}{2}$	0.665014
a_3	1	0.813665
b_1	$\frac{1}{2}$	0.201191
b_2	$\frac{1}{2}$	-0.737769
b_3	$\frac{1}{2}$	1.409750
b_4	0	1.005670
b_5	0	-0.864207
b_6	1	0.667915
w_1	$\frac{1}{6}$	0.109450
w_2	$\frac{1}{3}$	0.277182
w_3	$\frac{1}{3}$	0.362196
w_4	$\frac{1}{6}$	0.248217

As an experiment we had taken the differential equation:

$$\frac{dy}{dx} = f(x, y) = xy$$

where the output function is growing exponentially. The Graphs in Fig. 1 and Fig. 3 are representing the comparison of the values obtained from library function, conventional RK4 method and BFOA based method. This comparison is showing the effectiveness of the use of BFOA. The Fig. 2 is representing the convergence of the BFOA method applied for the determination of coefficients and constants used in the derivation RK4 method mentioned earlier.

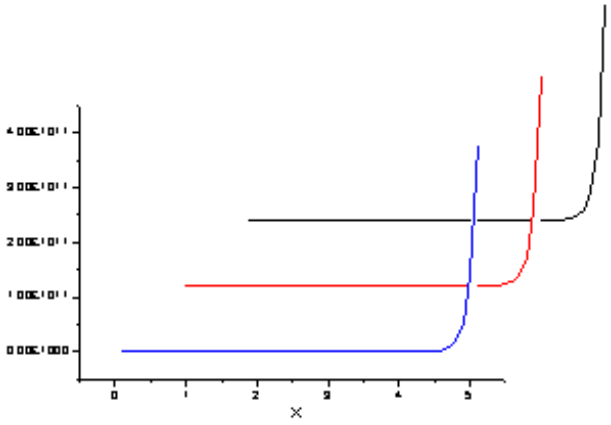

Fig 1: 3D view RK4 comparison graph

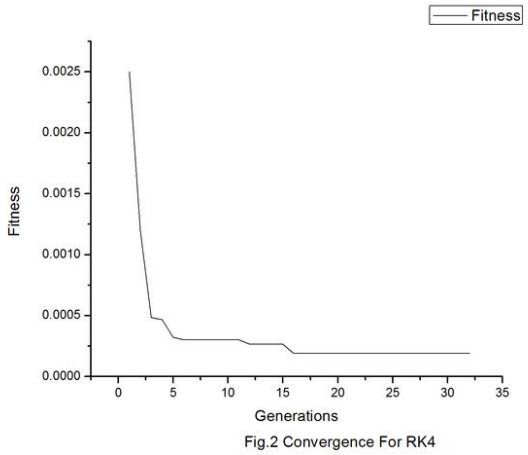

Fig.2 Convergence For RK4

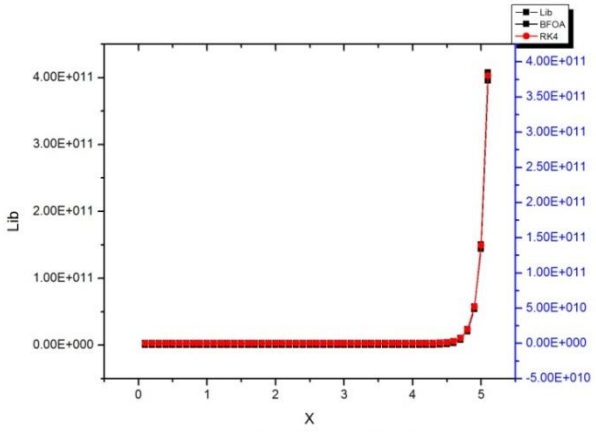

Fig. 3: Sample Results of RK4

7. CONCLUSION

This study, the simulation and experiments are encouraging us to find different improvements, modifications of BFOA, and important applications so that it can be used as one more important tool like GA in the field of AI based computations and other optimization paradigm. Like GA we may use BFOA as problem solver both linear and non-linear fields with more power than the conventional methods.

8. REFERENCES

[1] A.Abraham, A. Biswas, and S.Dasgupta et al., Analysis of reproduction operator in bacterial foraging optimization algorithm, In CEC 2008: IEEE World Congress on Computational Intelligence, IEEE Press, pages 1476-1483, Hong Kong, June, 2008.

[2] Badamchizadeh, M.A., Nikdel,A., Kouzehgar, M., 'Comparison of Genetic algorithm and particle swarm optimization for data fusion method based on kalman filter' International Journal of Artificial Intelligence,5(10), pp 67-78, 2010.

[3] C. Wu, N. Zhang, J. Jiang, J. Yang, and Y. Liang, Improved bacterial foraging algorithms and their applications to job shop scheduling problems, B. Beliezynski et al. (Eds.): ICANNGA 2007, Part-I, LNCS 4431, pp. 562-569, 2007.

[4] D.H. Kim and C.H. Cho, Bacterial foraging based neural network fuzzy learning, In IICAI 2005: Proceedings of the 2nd Indian International Conference on Artificial Intelligence, IEEE Press, pp. 2030-2036, Pune, India , December, 2005.

[5] D.H. Kim, A. Abraham, and J.H. Cho, A hybrid genetic algorithm and bacterial foraging approach for global optimization, Information Science 177 (18), pp. 3918-3937, 2007.

[6] H. Chen, Y. Zhu, and K. Hu, Adaptive bacterial foraging optimization, Abstract and Applied Analysis, vol. 2011, Article ID 108269, 27 pages,2011.

[7] H. Chen, Y. Zhu, and K. Hu, Self-Adaptation in bacterial foraging optimization algorithm, In ICISKE 2008: Proceeding of 3rd International Conference on Intelligent System and Knowledge Engineering, IEEE Press, pp. 1026-1031,2008.

[8] H. Chen, Y. Zhu, and K. Hu, Multi-colony bacteria foraging optimization with cell-to-cell communication for RFID network planning, Applied Soft Computing 10(2010), pp. 539-547,2009.

[9] H. Chen, Y. Zhu, and K. Hu, Cooperative bacteria foraging optimization, Discrete Dynamics in Nature and Society, vol. 2009, Article ID 815247, 17 pages, 2009.

[10] H. Chen, Y. Zhu, K. Hu, and T. Ku, Dynamic RFID network optimization using a self-adaptive bacterial foraging algorithm, International Journal of Artificial Intelligence, vol. 7, no. A11, pp. 219-231, 2011.

[11] H. Shen, Y.Zhu, X. Zhou, H. Guo, and C.Chang, Bacterial foraging optimization algorithm with particle swarm optimization strategy for global numerical optimization, In GEC 2009: Proceedings of Global Evolutionary Computing, ACM Press, pp. 497-504, Shanghai, China, June, 2009.

[12] Ikotun Abiodun M, Lawal Olawale N, and Adelokun adebowale P, The effectiveness of genetic algorithm in solving simultaneous equations, International Journal of Computer Applications (0975-8887), vol. 14, pp. 38-41, February, 2011.

[13] J. H. Mathews, Numerical Methods for Mathematics, Science and Engineering, 2nd Edition, PHI Prentice-Hall India, 2005.

[14] K. M. Passino, Biomimicry of Bacterial Foraging for distributed optimization and control, IEEE Control System Magazine, vol. 22, no. 2, pp. 52-67, 2002.

[15] L. Ulagammai, P. Vankatesh, and P.S. Kannan et al., Application of bacteria foraging technique trained and artificial and wavelet neural networks in load forecasting, Neurocomputing, 70(16-18), pp. 2659-2667, 2007.

[16] M.S. Li, T.Y. Ji, W.J. Tang, Q.W. Wu, and J.R. Saunders, Bacterial foraging algorithm with varying population, BioSystems 100 (2010), pp. 185-197, 2010.

[17] M. Tripathy, S. Mishra, I.L. Lai, and Q.P. Zhang, Transmission loss reduction based on FACTS and bacterial foraging algorithm, In PPSN 2006: Proceeding of the 9th International Conference on Parallel Problem Solving from Nature, vol. 4193 of lecture notes in Computer Science, pp. 222-231, September, 2006.

[18] N. Kushwaha, V.S. Bisht and G. Shah, Genetic algorithm based bacterial foraging approach for optimization, National Conference on Future Aspects of AI in Industrial Automation (NCFAAIIA 2012), Proceedings published by International Journal of Computer Applications (IJCA), (0975-8887), vol. 17, pp. 11-14, September, 2012.

[19] S. Dasgupta, S. Das, A. Abraham, and A. Biswas, Adaptive computational chemotaxis in bacterial foraging optimization: an analysis, IEEE Transactions on Evolutionary Computing, vol. 13, no. 4, pp. 919-941, 2009.

[20] S. Das, S. Dasgupta, and A. Biswas et al., On stability of the chemotactic dynamics in bacterial foraging, IEEE Transactions on System, Man and Cybernetics Part A: Systems and Humans, vol. 39, no. 3, pp. 670-679, May, 2009.

[21] S. Mishra, A hybrid least square-fuzzy bacterial foraging strategy for harmonic estimation, IEEE Transactions on Evolutionary Computation, vol. 9, no. 1, pp. 61-73, 2005.

[22] S. Mishra and C. N. Bhende, Bacterial foraging technique based optimized active power filter for load compensation, IEEE Transactions on Power Delivery, vol. 22, no. 1, pp. 457-465, 2007.

[23] S. Gilbert, 2007, Linear Algebra and its Applications. Pacific Grove: Brooks Cole

[24] S. S. Patnaik and A.K. Panda, Particle swarm optimization and bacterial foraging optimization techniques for optimal current harmonic mitigation by employing active power filter, Applied Computational Intelligence and Soft Computing, vol. 2012, Article ID 897127, 10 pages, 2012.

[25] W. H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, Numerical Recipes The art of scientific computing, 3rd Edition, Cambridge University Press, 2007.

[26] W. Korani, Bacterial foraging oriented by particle swarm optimization strategy for PID tuning, In GECCO 2008: Proceedings of the Genetic and Evolutionary Computation Conference, ACM Press, pp. 1823-1826, Atlanta, GA,USA, July, 2008.

[27] W.J. Tang, Q.H. Wu, and J.R. Saunders, Bacterial foraging algorithm for dynamic environments, In CEC 2006: IEEE Congress on Evolutionary Computation, IEEE Press, pp. 1324-1330,BC, Canada, July, 2006.

[28] Y. Liu and K.M. Passino, Biomimicry of social foraging bacteria for distributed optimization: Models, principles, and emergent behaviors, Journal of Optimization Theory and Applications. 115(3): 603-628, December, 2002.