Abstract- Bacterial Foraging Optimization (BFO) has been widely accepted as a global optimization technique. This technique is proposed by K.M. Passino in 2002 to handle complex problems of the real world. In this work, we aim to classify the satellite image using the theory of Bacterial Foraging Optimization. One key step in BFO is the computational Chemotaxis, where a bacterium takes steps over the foraging landscape in order to reach regions with high nutrient content. The Chemotactic movement of a bacterium may be viewed as a guided random walk. In this paper we design a new algorithm which is based on Bacterial Foraging Optimization which is used to classify the satellite image. Firstly we use a swarm data clustering method based upon flower pollination by artificial bees (FPAB) to cluster the satellite image pixels. Those clusters will be further classified using BFO. This new method shows an improved highly accurate results for the classification of satellite image when the proposed algorithm is used.

Keywords – Bacterial Foraging Optimization, Chemotactic, Satellite Image Classification, FPAB.

I. INTRODUCTION

Bacterial Foraging Optimization is a new corner to the family of nature-inspired optimization technique. Bacterial Foraging Optimization is a novel optimization based on the social foraging behavior of E.coli bacteria. In the recent years, bacterial foraging behaviors (i.e. bacterial chemotaxis) as a rich source of potential engineering applications and computational models. A few models have been developed to mimic bacterial foraging behaviors and they have been applied for practical problems [1], [10]. BFO is a population-based numerical optimization algorithm and has been applied successfully to some engineering problems, such as optimal control [4], harmonic estimation [8], transmission loss reduction [12] and machine learning [5].

Parminder Singh, Associate Professor, Department of Computer Science and Engineering, Guru Nanak Dev Engineering College, Ludhiana, India. (e-mail: parminder2u@gmail.com).

Navdeep Kaur, Assistant Professor, Department of Computer Science and Engineering, Khalsa college of Engineering and Technology, Amritsar, India. (e-mail:navdeepkjohal@gmail.com).

Loveleen Kaur, Assistant Professor, Department of computer science and Engineering, Khalsa College of Engineering and Technology, Amritsar, India (e-mail:loveleenchhina@gmail.com).

II. BACTERIAL FORAGING OPTIMIZATION ALGORITHM

Bacterial Foraging Optimization is inspired by the social foraging behavior of Escherichia coli. The underlying biology behind the foraging strategy of E.coli is emulated in an extraordinary manner and used as a simple optimization algorithm. The motile bacteria such as Escherichia coli propel themselves by rotating their flagella. They rotate their flagella counter clockwise to move forward rotate also called as “swimming” (or “runs”). But to move the bacteria in random direction i.e. “tumble” they rotate their flagella clockwise and then swims again. Tumbling just changes the direction of movement of bacteria. The bacteria first of all tumble in random direction to search for food. As the bacteria find food in a particular direction it then swims toward the food in that direction. An alternation between “swim” and “tumble” enables the bacteria to search for food in random directions [1], [10].

The original Bacterial Foraging Optimization system consists of three principal mechanisms, namely: Chemotaxis, Reproduction and Elimination-dispersal. These are described as follows [1], [10].

A. Chemotaxis

In the original BFO, a unit walk with random direction represents a “tumble” and a unit walk with the same direction in the last step indicates a “run.” Suppose $\theta^i (j, k, l)$ represents the bacterium at $j$th chemotactic, $k$th reproductive, and $l$th elimination-dispersal step. $C(i)$ is the chemotactic step size during each run or tumble (i.e., run-length unit). Then in each computational chemotactic step, the movement of the $i$th bacterium can be represented as:

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

Where $\Delta(i)$ is the direction vector of the $j$th chemotactic step. When the bacterial movement is run, $\Delta(i)$ is the same with the last chemotactic step; otherwise, $\Delta(i)$ is a random vector whose elements lie in $[-1, 1]$. With the activity of run or tumble taken at each step of the chemotaxis process, a step fitness, denoted as $J(i, j, k, l)$, will be evaluated.
B. Reproduction

The health status of each bacterium is calculated as the sum of the step fitness during its life, that is, \( \sum_{j=1}^{N_c} J(i, j, k, l) \), where \( N_c \) is the maximum step in a chemotaxis process. All bacteria are sorted in reverse order according to health status. In the reproduction step, only the first half of population survives and a surviving bacterium splits into two identical ones, which are then placed in the same locations. Thus, the population of bacteria keeps constant.

C. Elimination and Dispersal

The chemotaxis provides a basis for local search, and the reproduction process speeds up the convergence which has been simulated by the classical BFO. While to a large extent, only chemotaxis and reproduction are not enough for global optimia searching. Since bacteria may get stuck around the initial positions or local optima, it is possible for the diversity of BFO to change either gradually or suddenly to eliminate the accidents of being trapped into the local optima. In BFO, the dispersion event happens after a certain number of reproduction processes. Then some bacteria are chosen, according to a preset probability \( Ped \), to be killed and moved to another position within the environment.

The original BFO algorithm is briefly outlined step by step as follows:

**Step (i)** Initialize parameters \( n, S, N_c, N_s, N_r, N_e, N_d, Ped \),

\( C(i) \) \( (i = 1, 2, \ldots, S) \), \( \theta \), where

- \( n \): dimension of the search space,
- \( S \): the number of bacteria in the colony,
- \( N_c \): chemotactic steps,
- \( N_s \): swim steps,
- \( N_r \): reproductive steps,
- \( N_d \): elimination and dispersal steps,
- \( Ped \): probability of elimination,
- \( C(i) \): the run-length unit (i.e., the size of the step taken in each run or tumble).
- \( \theta \): position of \( i \)th bacterium.

**Step (ii)** Elimination-dispersal loop: \( l = 1 \) + 1.

**Step (iii)** Reproduction loop: \( k = k + 1 \).

**Step (iv)** Chemotaxis loop: \( j = j + 1 \).

**Substep (a)**. For \( i = 1, 2, \ldots, S \), take a chemotactic step for bacterium \( i \) as follows.

**Substep (b)** Compute fitness function, \( J(i, j, k, l) \).

**Substep (c)** Let \( J_{\text{last}} = J(i, j, k, l) \) to save this value since we may find better value via a run.

**Substep (d)** Tumble. Generate a random vector \( \Delta(i) \in \mathbb{R}^D \) with each element \( \Delta m(i), m = 1, 2, \ldots, n \), a random number on \([-1, 1]\).

**Substep (e)** Move. Let

\begin{equation}
\theta'(j + 1, k, l) = \theta'(j, k, l) + C(i) \frac{\Delta(i)}{\sqrt{\Delta'(i) \Delta(i)}}
\end{equation}

This results in a step of size \( C(i) \) in the direction of the tumble for bacterium \( i \).

**Substep (f)** Compute \( J(i, j + 1, k, l) \) with \( \theta'(j + 1, k, l) \).

**Substep (g)** Swimming.

(i) Let \( m = 0 \) (counter for swim length).

(ii) While \( m < N_s \) (if has not climbed down too long), the following hold.

- Let \( m = m + 1 \).
- If \( J(i, j + 1, k, l) < J_{\text{last}} \), let \( J_{\text{last}} = J(i, j + 1, k, l) \); then another step of size \( C(i) \) in this same direction will be taken as \( (2) \) and use the new generated \( \theta'(j + 1, k, l) \) to compute the new \( J(i, j + 1, k, l) \).
- Else let \( m = N_s \).

**Substep (h)** Go to next bacterium \( (i + 1) \). If \( i \neq S \), go to Substep(iv)(b) to process the next bacterium.

**Step (v)** If \( j < N_c \), go to Step (iii). In this case, continue chemotaxis since the life of the bacteria is not over.

**Step (vi)** Reproduction.

**Substep (a)**. For the given \( k \) and \( l \), and for each \( i = 1, 2, \ldots, S \), let

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

be the health of the bacteria. Sort bacteria in order of ascending values \( J_{\text{health}} \).

**Substep (b)**. The \( Sr \) bacteria with the highest \( J_{\text{health}} \) values die and the other \( Sr \) bacteria with the best values split and the
copies that are made are placed at the same location as their parent.

Step (vii) If $k < N_{re}$, go to Step (ii). In this case the number of specified reproduction steps is not reached and start the next generation in the chemotactic loop.

Step (viii) Elimination-dispersal: for $i = 1, 2, \ldots, S$, with probability $P_{ed}$, eliminate and disperse each bacterium, which results in keeping the number of bacteria in the population constant. To do this, if a bacterium is eliminated, simply disperse one to a random location on the optimization domain. If $l < N_{ed}$, then go to Step (ii); otherwise end.

III. FLOWER POLLINATION BY ARTIFICIAL BEES

FPAB is a swarm clustering method based on flowers pollination by artificial bees. It is based on the fact in nature that each species of plants has better growth in special region and agglomeration of these species of plants is observable. In FPAB, Each individual artificial bee is behaviorally a simple agent with a one-slot memory. The memory is used to save the growth of carrying pollen in its source. The bees will pick up the pollen of flower with lowest growth and pollinate the pollen where it will grow better. Each pollen grows in proportion to its neighbor flowers and after some iteration the natural selection will select the flower with best growth of one species to survive and will sear others. Using these simple stochastic behaviors and limited memory, the artificial bees can perform complicated task as clustering. Initially, in FPAB, bees move the pollens and pollinate them. Each pollen will grow in proportion to its garden flowers. Better growing will occur in better conditions. After some iteration, natural selection reduces the pollens and flowers and the gardens of the same type of flowers will be formed. The prototypes of each garden are taken as the initial cluster centers for Fuzzy C Means algorithm which is used to reduce obvious misclassification errors. In the next stage, the prototypes of gardens are assumed as a single flower and FPAB is applied to them again [3].

IV. REMOTE SENSING IMAGE CLASSIFICATION

Remote Sensing is the most generally accepted meaning refers to instrument-based techniques employed in the acquisition and measurement of spatially organized data/information on spectral, spatial and physical properties. Remote sensing with multi spectral satellite imagery is based on the concept for capturing the geo-spatial information of different features/objects constitutes the land cover. A multi-spectral remote sensing system operates in a limited number of bands and measures radiations in series of discrete spectral bands. The spectral response is represented by the discrete digital number (DN). Spectral signatures of an object may be used for identification much like a fingerprint [9].

Image classification is conducted in two main techniques: Supervised and Unsupervised classification. A supervised classification requires the manual identification of known surface features within the imagery and then using a statistical package to determine the spectral signature of the identified feature [6]. To recognize spectrally similar areas, numerical information in all spectral bands for the pixels comprising these areas is used. For each pixel in the image a comparison is made with these signatures and defined in the class it most closely "resembles". When spectral classes, based on numerical information, are grouped first and are then matched by the analyst to information classes, then it is termed as unsupervised classification. Clustering algorithms are used to determine the statistical structures in the data for example K-Means approach [7].

V. PROPOSED WORK

Bacterial Foraging Optimization is generally used to find the optimal solution of a problem. In this work, a satellite image has been classified by used of Bacterial Foraging Optimization based algorithm. Satellite image classification is a clustering problem that requires each class to be extracted as a cluster. The original BFO algorithm inherently does not have the property of clustering. Hence to extract features from the satellite image, a swarm clustering method i.e FPAB has been used to cluster the image pixels. The FPAB algorithm for clustering the image pixels has been explicitly discussed in [2].

In this research work, a 7-band satellite image of the study area of Alwar city (Rajasthan) has been considered for clustering and classification by making use of FPAB and BFO techniques respectively. This image of size 472 X 546 contains 2.5 lakh pixels. Firstly, these pixels are clustered into homogeneous groups using FPAB. Subsequently these clusters are classified using Bacterial Foraging Optimization. The inherent flexibility of the Bacterial Foraging Optimization scheme has been exploited by implementation of its bacterial chemotaxis mechanism in particular, for problem solution to the satellite image classification case. The BFO parameters of the proposed algorithm include the two main steps of bacterial chemotaxis representation and fitness function evaluation. These steps are defined as follows:

A. Bacterial Chemotaxis Representation

As a first step, the standard deviation of each of the final clusters which are obtained by artificial honey bees is computed. The standard deviation of each cluster will indicate the position of the bacteria corresponding to that cluster.

Position of each bacteria = Standard deviation of each cluster.

The number of bacteria is equal to the number of clusters under consideration.
No. of Bacteria = No. of clusters.

For each land cover feature of the study area, training pixels provided by experts help to classify the clusters. The training pixels, provided by the experts for one land cover feature are treated as one food source where the bacterium has to tumble. The Standard deviation of each food source is computed. The Standard deviation of each food source will form the position of food source.

Food = Pixels provided by the experts.

No. of food sources = No. of land cover features to be extracted.

Position of Food = Standard deviation of pixels provided by experts.

Each Bacteria has two slots of memory for remembering two things: the position of fittest food source and the fitness value of the fittest food source. The bacteria will randomly tumble to each food source and find the fittest food as shown in figure 1.

Memory (Bacteria) = fittest food source.

Fitness (memory (bacteria)) = fitness value of fittest food source.

C. Proposed Algorithm

The following Bacterial Foraging Optimization based algorithm has been proposed to classify the satellite image.

Algorithm: Bacterial Foraging Optimization Satellite Image Classification.

1. Get the Multispectral satellite image.
2. Cluster the image into homogeneous clusters (using FPAB Algorithm) and compute the standard deviation of each cluster.
3. The standard deviation of one cluster will form the position of one bacteria.
4. Consider each land cover feature - Water, Urban, Rocky, Barren and Vegetation - having training pixels (produced by experts) as one food source.
5. The standard deviation of each food source will form the position of food source.
6. (a) for j=1 to No. of bacteria
   (i) memory (j) = 0.
   (ii) fitness (memory(j)) = ∞.
   (b) for i=1 to No. of food source.
   (i) tumble jth bacteria to ith food source.
   (ii) calculate the fitness function θ(i).
   (iii) if fitness(memory(j)) > θ(i).
       memory (j) = i.
       Fitness (memory (j)) = θ(i).
   End loop.
(c) tumble jth bacteria to memory (j).
   This is the fittest food source for jth bacteria.
   End loop.

Figure 2: Algorithm for BFO Based Satellite Image Classification.

In this algorithm, firstly homogeneous clusters of pixels are produced using FPAB algorithm and the standard deviation of each cluster is computed. The Standard deviation of one cluster will form the position of one bacterium. In our image, we want to classify 5 features - water, vegetation, urban, rocky and barren. Each of these features is represented and the pixels provided by the experts are considered as one food source. The Standard deviation of each food source will form the position of food source. In this work, the number of bacteria is considered equal to the number of clusters where the food is considered as the pixels provided by the experts. The fittest food is to be found by calculating the minimum value of the fitness function θ(i).

Figure 1: Bacterial Representation

B. Fitness function

In each generation, each bacterium is evaluated, and a value of goodness or fitness is returned by a fitness function. This evolution is driven by the fitness function θ(j). Let j=1 to the no. of bacteria and i=1 to the no. of food source denote the bacteria and the food sources respectively.

The fitness function is calculated as:

Fitness function θ(j) = min (avg (position of bacteria – position of food)).
Initially memory of bacteria is set to 0 and the fitness inside memory of each bacterium is set to $\infty$.

In each iteration, one bacterium will tumble to each food source one by one. At each food source, bacteria will check if the new food source is fitter as compared to the one in memory, if it is so, bacteria will update its memory to new food source. Thus this algorithm will classify the satellite image in one iteration till the minimum value of fitness function is found.

VI. RESULTS AND DISCUSSION

The objective of this research work is to use the proposed Bacterial Foraging Optimization algorithm for satellite image classification. This algorithm has been implemented in Matlab 7.0 [11]. It has been applied to validate the multi-spectral, multi resolution and multi-sensor image of Alwar area in Rajasthan. The images have been taken from D.T.R.L. lab, D.R.D.O. India [13]. The satellite image for 7 different bands is taken (Fig 3). These bands are Red, Green, Near Infra Red (NIR), Middle Infra Red (MIR), Radarsat-1 (RS1), Radarsat-2 (RS2), and Digital Elevation Model (DEM).

![Seven Band Image of Alwar (Rajasthan)](image)

![Training set of Barren region.](figure4)

![Bacterial Foraging Optimization based classification of satellite image of Alwar region.](figure5)

Firstly, the clusters of image pixels are obtained using FPAB algorithm. Each pixel of satellite image is treated as an object of FPAB algorithm. So total number of objects (n) in FPAB algorithm is equal to total number of pixels in satellite image. In the first iteration of FPAB, honey bees will pick and drop one pixel only. In second iteration, honey bees will pick and drop cluster of pixels obtained in the first iteration. The clusters obtained by FPAB algorithm are further classified by using BFO.

During classification, firstly the average of standard deviation of pixels (provided by experts) is obtained which forms the position of one food source. The pixels provided by experts are some of the pixels of the image, for which an expert is sure about that these pixels are of particular land cover region. After that the bacteria will tumble to each food source and find the fittest food source in the end. The training pixels (provided by experts) used in this study have been stored in the Microsoft Excel sheets, which shows one row representing one pixel and 7-columns representing 7-bands of the image. The training pixels originally present in the barren region as shown in figure 4. Decision attribute in fig.4 is describing the class of each pixel.

After applying the proposed algorithm to the 7-band of Alwar Image, the classified image is obtained in figure 5. The yellow, black, blue, green, red color represents rocky, barren, water, vegetation, urban region respectively.
VII. ACCURACY ASSESSMENT

Accuracy Assessment is one of the key concepts for the classification process. The main motive of this classification is to determine how effectively pixels were grouped into the correct feature classes in the area under investigation. Classification accuracy of our proposed algorithm is expressed using classification error matrix. Error matrices compare, on a category-by-category basis, the relationship between known reference data (ground truth) and the corresponding results of an automated classification.

We took 150 vegetation pixels, 190 urban pixels, 200 rocky pixels, 70 water pixels, 170 barren pixels from the training set and the error matrix obtained is shown figure 6. The error matrix’s interpretation along column suggests how many pixels are classified correctly by algorithm. For e.g. in the first column, out of total 150 vegetation pixels, 138 pixels were correctly classified into vegetation by the proposed algorithm, 10 were misclassified as rocky and 2 were misclassified as barren .All water pixels are correctly classified. Producer’s Accuracies result from dividing the number of correctly classified pixels in each category (on major diagonal ) by the number of training set pixels used for that category (the column total). The producer accuracy for urban, rocky and barren are 48.94%, 91% and 62.35% respectively whereas in original work [2] these were 47.36%, 90.5% and 61.75%. So the urban, rocky and barren areas are classified better with the proposed algorithm.

The Kappa coefficient of the Alwar image can be calculated by applying following formula to the Error Matrix [6]:

\[
k = \frac{N \sum_{i=1}^{r} x_{ii} - \sum_{i=1}^{r} (x_{i+} \cdot x_{+i})}{N^2 - \sum_{i=1}^{r} (x_{i+} \cdot x_{+i})}
\]

\(r\) = number of rows in the error matrix (\(r=5\) in our case)
\(x_{ii}\) = the number of observations in row i and column i (on the major diagonal)
\(x_{i+}\) = total of observations in row i (shown as marginal total to right of the matrix)
\(x_{+i}\) = total of observations in column i (shown as marginal total at bottom of the matrix)
\(N\) = total number of observations included in matrix. (\(N=780\) in our case).

The Kappa (K) coefficient of the Alwar image is 0.68745 which indicates that an observed classification is 68.75 percent better than one resulting from chance.

VIII. CONCLUSION AND FUTURE WORK

In this paper we have presented a FPAB/BFO based algorithm for the classification of satellite image. The results obtained are better than the results based on FPAB/BBO. The Kappa coefficient is used to measurement of correctness of image classification. This algorithm produces homogeneous clusters using FPAB algorithm and these clusters are further classified using the technique of Bacterial Chemotaxis.

For the further research work may focus on some modification of the proposed algorithm and developing some new algorithms for the improvement to develop an efficient classifier. The future scope of the research includes improvement of Kappa coefficient further by using different algorithms. In this work, bacterial chemotaxis step of bacterial foraging optimization has been used particularly, the future scope includes applying other two steps i.e. bacteria reproduction and elimination-dispersal steps to further refine the results. Further, an improvement in the fitness function can be proposed.

<table>
<thead>
<tr>
<th>Vegetation</th>
<th>Urban</th>
<th>Rocky</th>
<th>Water</th>
<th>Barren</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vegetation</td>
<td>138</td>
<td>10</td>
<td>0</td>
<td>2</td>
<td>150</td>
</tr>
<tr>
<td>Urban</td>
<td>0</td>
<td>93</td>
<td>1</td>
<td>36</td>
<td>130</td>
</tr>
<tr>
<td>Rocky</td>
<td>10</td>
<td>0</td>
<td>182</td>
<td>0</td>
<td>218</td>
</tr>
<tr>
<td>Water</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>70</td>
<td>70</td>
</tr>
<tr>
<td>Barren</td>
<td>2</td>
<td>87</td>
<td>17</td>
<td>0</td>
<td>106</td>
</tr>
<tr>
<td>Total</td>
<td>150</td>
<td>190</td>
<td>200</td>
<td>70</td>
<td>780</td>
</tr>
</tbody>
</table>

Figure 6: Error Matrix of FPAB/BFO based satellite image classification of Alwar region.
REFERENCES


AUTHORS PROFILE

Parminder Singh is Associate professor in department of Computer Science and Engineering at Guru Nanak Dev Engineering College, Ludhiana (India). He is having about 12 years academics experience. He has guided about 17 post graduate students for their dissertation work. He has published about 23 research papers in journals and conferences. Presently he is working for the development of a Text-to-Speech synthesis system for Punjabi language.

Navdeep Kaur has done B-Tech (Hons.) in Computer Science & Engineering & scored 81% marks from Punjab Technical University, Jalandhar (India) in 2005 and M-Tech in Computer Science & Engineering from Guru Nanak Dev Engineering College, Ludhiana of India in 2009. She is currently working as an Assistant Professor in computer science of Khalsa College of Engineering & Technology of India.

Loveleen Kaur has done B-Tech in Computer Science & Engineering from Punjab Technical University, Jalandhar (India) in 2007 and pursuing M-Tech in Computer Science & Engineering from Guru Nanak Dev Engineering College, Ludhiana of India. She is currently working as an Assistant Professor in computer science of Khalsa College of Engineering & Technology of India.