

# A Hybrid Bacterial Foraging Optimization Algorithm and a Radial Basic Function Network for Image Classification

Yasmina Teldja Amghar\* and Hadria Fizazi\*

## Abstract

Foraging is a biological process, where a bacterium moves to search for nutriments, and avoids harmful substances. This paper proposes a hybrid approach integrating the bacterial foraging optimization algorithm (BFOA) in a radial basis function neural network, applied to image classification, in order to improve the classification rate and the objective function value. At the beginning, the proposed approach is presented and described. Then its performance is studied with an accent on the variation of the number of bacteria in the population, the number of reproduction steps, the number of elimination-dispersal steps and the number of chemotactic steps of bacteria. By using various values of BFOA parameters, and after different tests, it is found that the proposed hybrid approach is very robust and efficient for several-image classification.

## Keywords

Bacterial Foraging Optimization Algorithm, Hybrid, Image Classification, Radial Basic Function

## 1. Introduction

For a long time, scientists have been fascinated by animals and biological organisms, and their behaviors in Nature. They have imitated them for solving various complex problems encountered in the real world.

Independently of classical optimization methods, like the gradient-based and the quasi-Newton methods [1], there are several widespread and well-established methods based on heuristic population as artificial immune system (AIS), ant colony algorithm (AC), and artificial neural network (ANN) [2].

Right from the time of its inception, methods based on ANNs acquired a growing attention from the industrial and academic areas because they treated optimization problems that the usual techniques were unable to solve [3-5]. The radial basis function (RBF) neural network is one of many types of ANNs that has been extensively used to the non-linear system optimization.

The training of the RBF neural network is not an easy task, as its performance is highly dependent upon factors such its architecture and connection weight value. Moreover, they constitute one of the major problems in the neural network. The selection of the hidden layer parameters of the RBF neural network, which means the determination of the number of hidden neurons, their position, center, and

\* This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.  
Manuscript received May 27, 2016; first revision December 27, 2016; accepted February 9, 2017.

**Corresponding Author:** Yasmina Teldja Amghar (amgharyasmina@yahoo.fr)

\* Dept. of Computer Science, Faculty of Mathematics and Computer Science, University of Sciences and Technology of Oran - Mohamed Boudiaf, Oran, Algeria (yasmina.amghar@univ-usto.dz, hadriafizazi@yahoo.fr)

width, is very important, and an efficient learning algorithm has a significant part to improve the neural network effectiveness.

In recent years, there has been a growing interest in optimizing the hidden neurons parameters of RBF neural network to solve the RBF problems using metaheuristic algorithms, and which are successfully employed in several search problems [4,6-8].

Following the similar direction of algorithms based on swarm, Passino [9,10] suggested a novel optimization approach and named it the “bacterial foraging optimization algorithm” (BFOA). This approach is a new member in the very large swarm intelligence domain [11], based on the behavior of *Escherichia coli* (*E. coli*) bacterium during foraging.

One of the attractive features of BFOA is that it has its individual local research method through the computational chemotactic steps and the reproduction with elimination-dispersal which helps in global research. The ability of BFOA to solve actual optimization problems has made it possible to design an optimization algorithm, deserving to invest this research field [10].

Regarding the current problems and since the choice of the RBF neural network parameters is problem dependent, we propose a hybrid approach, where the hidden layer parameters (number of hidden neurons, their position, center, and width) will be determined automatically through the BFOA, and the process can be stopped as soon as the network performance is good enough. Our contribution is beneficial since the algorithm will be adopted to train the neural network to evolve and improve the RBF neural network learning and accuracy, through the determination of the hidden layer parameters in a dynamic way.

In this work, the integration of the BFOA in the RBF neural network is evaluated for two reasons; firstly, to solve the problems of the classic neural network and secondly, to assess the potential of its inclusion in the production of good image classification results.

Given importance to the organization of this paper, Section 2 gives the description of the real *E. coli* bacterium. Section 3 provides the related works on the BFOA, while Section 4 offers the biological reason of the BFOA use and describes the algorithm itself in a complete way. The influence of the number of bacteria in the population, the number of reproduction steps, the number of elimination-dispersal steps and the number of chemotactic steps on the classification rate and the objective function value through several images’ classification are discussed in Section 5. The paper is finally recapitulated in Section 6.

## 2. Related Works

In 2000, Passino [9] explained the biological chemotactic behavior of the *E. coli*, how the bio-imitation of bacterial foraging can be used, and how to perform the stability analysis of swarms, thereby providing mathematical foundations for the study of social foraging.

In 2002, he proposed, in [10], a newcomer to the family of nature-inspired optimization algorithms after mimicking the environmental food foraging, reproduction and elimination-dispersal events behaviors of the bacterium, and named it “bacterial foraging optimization algorithm”. This algorithm is essentially a random search algorithm.

From the beginning, BFOA attracted the attention of researchers from diverse fields of knowledge

particularly due to its attractive structure and biological motivation. In 2007, the algorithm was described by Narendhar and Amudha [12] as a new evolutionary algorithm.

In 2009, Das et al. [13] discussed the theoretical foundations and the analysis of BFOA. Besides its hybridization with other optimization systems, they provided an account of most of the significant applications of BFOA until 2009. In the same year, they evaluated, in [14], the chemotaxis operation in the algorithm. The undertaken study provides important insights into its search mechanism. After that, they analyzed, in [15], the stability of the chemotactic dynamics in the algorithm, showing that the bacterial dynamics demonstrates an asymptotically constant behavior.

In 2010, the same authors gave an analysis of the reproduction operator and its stability [16], indicating that the reproduction event contributes to the fast convergence of the bacterial population near favorable environment, i.e., the optima.

In 2011, Sathya and Kayalvizhi [17] and Sanyal et al. [18] proposed, respectively, a paper about the application of the BFOA in the image segmentation and an adaptive BFOA in the same field. Both papers demonstrate that the algorithm proves its suitability for image segmentation.

In order to explore separately the local and global search proprieties of BFOA, researchers hybridized it with other algorithms, applied it to several real world problems and proved its effectiveness over many works, e.g., mathematical modeling for solving simultaneous equation problems [19], solving scheduling problems [12], mechanical problems [2], solar PV modeling [20], and numerical optimization [1]. More hybridization is observed in [18,21].

In 2015, Kaur and Kaur [8] demonstrated that the BFOA is able to train the ANN with high classification accuracy, thus, proved its efficiency as a training algorithm.

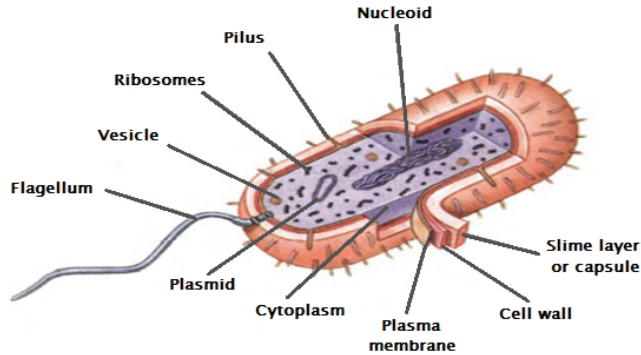
So these recent related works suggest that BFOA has attracted lot of attention from researchers that have analyzed each of its components, studied its effectiveness, and proposed after that several models, improvements and hybrid systems. Many successful applications in the engineering and other fields have been reported and are available in the literature. Most of the proposed works agree that using or integrating BFOA in a system produces suitable or better solutions for problems resolution.

### 3. *Escherichia coli*, the Real Bacterium

#### 3.1 Definition

Bacteria (singular, bacterium) are an organism composed of a single cell. In spite of their simple aspect, they obtain information about their milieu, guide themselves in this milieu, and use efficiently these data to survive. They constitute one of the simplest life forms on Earth [22].

Fig. 1 shows that the *E. coli* bacterium consists of a slime layer, a cell wall, a plasma membrane, and many other components. The pilus (plural, pili) is used for a gene displacement to other *E. coli* bacteria, and the flagellum (plural, flagella) is used for mobility. The bacterium has approximately a length of 2  $\mu\text{m}$  and a diameter of 1  $\mu\text{m}$ . The weight of *E. coli* bacterium is about 1 picogram, and is constituted of about 70% of water [23]. This bacterium is possibly the best understood micro-organism at the moment.

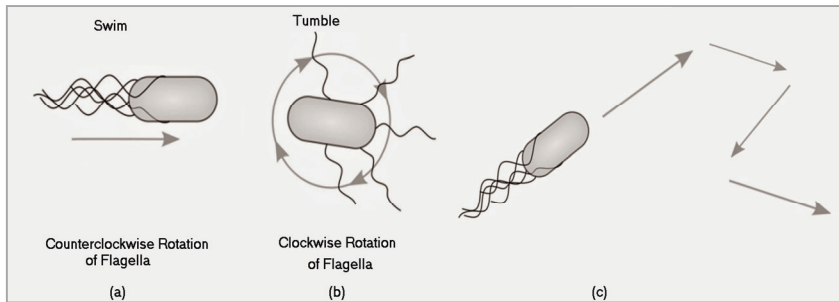


**Fig. 1.** *Escherichia coli* bacterium.

### 3.2 Chemotactic Behavior

Dahlquist et al. [24] assume that the bacterium movement is as follows:

- The bacterium's trajectory is a succession of straight lines followed by a direct turning (see Fig. 2(c)). This trajectory is typified by a length, a direction, and a velocity,
- The velocity is constant for all trajectories,
- A probability distribution governs the new direction after the bacterium turns, and the angle between consecutive trajectories as well,
- A probability distribution exponential governs the trajectory length,
- Angle and length probability distributions are independent from the precedent trajectory.



**Fig. 2.** (a) Swim, (b) tumble, and (c) chemotactic behavior of *E. coli*.

A bacterium's position and movement, in a  $n$ -dimensional space, are defined by  $X = \{x_1, x_2, \dots, x_n\}$ , with a radius  $r$  and  $n-1$  angles  $\Phi = \{\phi_1, \phi_2, \dots, \phi_{n-1}\}$  [22] :

$$x_1 = r \cdot \prod_{k=1}^{n-1} \cos(\phi_k), x_n = r \cdot \sin(\phi_{n-1}), x_i = r \cdot \sin(\phi_{i-1}) \cdot \prod_{k=i}^{n-1} \cos(\phi_k), i = 2, 3, \dots, n-1 \quad (1)$$

The Gaussian probability density distribution is used to define, in the  $(x_i, x_{i+1})$  plane, the new direction of  $\phi_i$ , where  $\phi_i$  is an angle calculated from  $x_i$  and taken for choosing between two possible directions [22].

A bacterium is composed of six rigid rotating flagella governing its motion ability and its foraging behavior [9].



The bacterial chemotactic behaviors are:

- If the bacterium alternates between runs (swims) and tumbles in a neutral milieu → it is the search.
- If it swims out of harmful substances (or up nutrient gradient) or it climbs down harmful substances gradient or up nutrient gradient (swim longer) → it is the search for increasingly advantageous environments.
- If it swims up harmful substances gradient (or down nutrient gradient) → it is the avoidance of disadvantageous environments.

### 3.3 Forage Theory

In order to seek for and obtain nutrients, the animals search for increasing the quantity

$$E/T \quad (2)$$

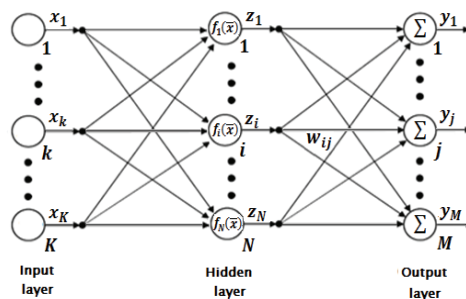
where  $E$  represents the acquired energy, and  $T$  the time spent to forage, or, they increase energy consumption average rate. Through the evolution theory, animals with a poor forager strategy die, thus there will be a real forager optimization [9].

Foraging can be done individually or in a group. When an animal performing social foraging, it requires communication capacities, and can obtain gains because it can basically exploit the group perception capacities. And in a group, individuals acquire protection from enemies (predators), and can also ally against a big prey. They can forage with a cooperative intelligence type. As an individual and as a group, it is clear that the species having large cognitive capacities will have less problems to succeed in forage [9]. To solve optimization problems, search and forage bacteria have been used recently [5].

## 4. Hybrid Approach Consisting of Radial Basis Function Network and Bacterial Foraging Optimization Algorithm

### 4.1 Radial Basic Function Neural Network

ANN is a powerful artificial tool suitable for solving combinatorial problems such as prediction and classification, and it can be defined as a parallel system composed of many treating cells simply interrelated. Various categories of ANNs exist, however, only the RBF neural network will be discussed whose architecture is given in Fig. 3.



**Fig. 3.** Architecture of radial basis function neural network.

The first layer ( $X$ ) is a  $K$ -dimensional input layer composed of a set of neurons that collects information and transmits them to the neurons of the following layer. The second layer ( $Z$ ) is a hidden layer containing a non-linear transformation in order to separate the data linearly for the next layer. The neuron output of the hidden layer is given by the Eq. (3). In the third layer ( $Y$ ), the outcomes of the hidden layer will be weighed and summed.

$$z_i = f_i(\bar{x}) = \exp\left(-\|\bar{x} - \bar{\mu}_i\|^2 / 2\sigma_i^2\right), \bar{x} \in R^n, \bar{\mu}_i \in R^n, i = 1, \dots, N \quad (3)$$

$$y_j = \sum_{i=0, \dots, N} w_{ij} z_i, z_0 = 1 \quad (4)$$

In the hidden layer, every neuron  $i$  has a vector  $\bar{\mu}_i$ , determining the neuron's position in the input space and a standard deviation  $\sigma_i$ , defining the neuron's width. Every neuron is also connected (with a connection having a weight value  $w_{ij}$ ) to all the neurons of the output layer, and its activation function (mentioned  $f_i(\bar{x})$  in Fig. 3) represents a Gaussian function [4].

The advantage of RBF neural network lies in the fact that it does not require any complex mathematical formulations or quantitative correlation between inputs and outputs. However, training a neural network, regardless the chosen training method, is a continuous optimization problem thus a complex task, as the performance of RBF neural network is highly dependent upon the factors such its architecture and connection weight value. To have a high performance in the network, the selection of appropriate parameters is very important and an efficient learning algorithm has a significant role to improve its effectiveness. The selection of the parameters means to determine the number of hidden neurons, their position, center, and width.

According to their 2009 work, Tinos and Murta Jr [7], demonstrated that the choice of the RBF neural network parameters is problem dependent, e.g., while the RBF neural network with hidden neurons parameters presented best performance in one problem, the choice of others parameters was beneficial for another problem, which can be viewed as a search problem too.

In [8], the authors put the number of bacteria equal to the number of connection weights, i.e., the number of neurons in the hidden layer, which made of this system a fixed hybrid network, because of this chosen and predefined number (of bacteria) at the beginning, i.e., before training. In this way the neural network is static because it is not capable of evolving, adapt itself or eliminate elements that are not useful to attain the best configuration.

In order to solve this problem, the idea is to substitute the hidden layer by a dynamic structure, namely the BFOA. The algorithm will be employed to train the RBF neural network and evolve the hybrid approach learning and effectiveness, through the determination of the hidden layer parameters, automatically and in a dynamic way. The process can be stopped as soon as the hybrid approach performance is good enough since positioning of hidden neurons and training of connection weights are performed in parallel. Thus, the network can generalize well and have a relatively small size, which helps to avoid a high computational complexity.

The hybrid approach can be described as a RBF network with a modified version of the hidden layer and as the method for managing and constructing this latter.

## 4.2 Bacterial Foraging Optimization Algorithm

In the real world, animals will be killed or transformed to better ones after several generations if they have a poor foraging method [10]. Several researchers have used this process as an optimization means [10,20].

### 4.2.1 Description of the BFOA technique

The mobility of the real bacteria is attained by using a set of rigid flagella allowing a bacterium to swim. All flagella rotate in similar direction, creating energy (power) against the cell and thus pushing the bacterium [13,25].

During the *E. coli* movements, the bacterium has two distinct behaviors: if the flagella turn clockwise, it changes direction and tumbles; else it swims in the same direction [25].

Fig. 2(a) and (b) describes the bacterium's movements in nutrient milieu.

A bacterium increases in length and divides into two, creating, thus a perfect copy of itself if it is under an adequate temperature, and if it obtains enough food. Passino [9] was inspired by this phenomenon to represent, in BFOA, the reproduction event. The chemotactic development may be damaged if the environment is attacked or changes suddenly. Bacteria, then, can be displacing to different places or new ones can emerge in the swarm. In a real group of bacteria, this phenomenon represents the elimination-dispersal event [9].

A bacterium calculates the nutriment concentration, at the first position, and then takes a way (direction) randomly by tumbling. After that it swims for a constant distance and calculates the concentration another time. At the next position, if the result is superior, the bacterium moves in the same direction. Moreover, to locate a new direction, the bacterium tumbles and then swims in this direction. This process is repeated a number of times, as long as the bacterium is still alive.

The bacterium splits into two units if it has collected an excellent health at the end of its life. This helps the next bacteria generation to begin from a good position during reproduction steps. To produce the next bacteria generation, a half of the population (the best bacteria) divides, and the other half (the worst bacteria) dies. This process is also repeated a number of times.

### 4.2.2 Description of the BFO algorithm

When there aren't any information about measurements or the gradient  $\nabla J(\theta)$ , the main aim is the minimization of

$$J(\theta), \theta \in R^p \quad (5)$$

Assuming that  $\theta$  is a  $p$ -dimensional vector variable of real numbers, which is the bacterium's position, and  $J(\theta)$  is an attractant-repellent profile, indicating the location of the nutrients and the harmful substances, so  $J > 0$ ,  $J = 0$ , and  $J < 0$  indicates respectively that, there are harmful substances, the milieu is neutral, and there are nutrients [9].

We consider for swarming the minimization of

$$J(i, j, k, l) + J_{cc}(\theta^i(j, k, l)) \quad (6)$$

so that the bacterium will attempt to locate nutrients, avoid harmful substances, and attempt to move in the direction of the other bacteria, but not nearby them. The  $J_{cc}^i(\theta^i(j, k, l))$  function distorts the search space to show its need for swarm [9].

A chemotactic step can be defined as a tumble following a tumble or a run following a tumble. Let be

- $j$  - The index for the chemotactic step,
- $k$  - The index for the reproduction step,
- $l$  - The index of the elimination-dispersal event,
- $p$  - The dimension of the search space, is also the number of elements of  $\theta$  [2],
- $S$  - The global number of bacteria in the population [16],
- $S_r$  - The number of bacteria in the reproduction steps, with  $S_r = S/2$  [2],
- $N_c$  - The bacteria longevity as calculated by the number of chemotactic steps,
- $N_s$  - The swimming length is also the maximum number of steps,
- $N_{re}$  - The number of reproduction steps,
- $N_{ed}$  - The number of elimination-dispersal events,
- $P_{ed}$  - The elimination-dispersal probability,
- $C(i), i = 1, 2, \dots, S$  - The step size taken during every tumble or swim,
- $\theta^i$  - The random swim direction [26],
- $d_{attract}$  - The depth of the attractant produced by the bacterium,
- $w_{attract}$  - The width of the attractant, i.e., the diffusion rate measure of the chemical signal,
- $h_{repel}$  - The height of the repellent,
- $w_{repel}$  - The width of the repellent.

Let

$$P(j, k, l) = \{\theta^i(j, k, l) | i = 1, 2, \dots, S\} \quad (7)$$

indicates the position of every bacterium of the bacteria  $S$  in the population.  $\theta^i(j, k, l)$  symbolizes the bacterium  $i$  at the chemotactic step  $j$ , reproduction step  $k$ , and elimination-dispersal event  $l$ . To simplify notations, we refer to the bacterium's position  $i$  as  $\theta^i$  [13].  $J(i, j, k, l)$  represents the objective function value (cost) at the bacterium's position  $i$  [9].

The BFOA imitates the main steps used during the forage of the real bacterium *E. coli*: swarming, chemotaxis, reproduction, and elimination-dispersal.

**(i) Swarming:** When a bacterium finds the shortest way towards food, it should inform the other bacteria to take this way. In *E. coli* swarm, the cell-to-cell indicator can be modeled by

$$\begin{aligned} J_{cc}(\theta, P(j, k, l)) &= \sum_{i=1}^S J_{cc}(\theta, \theta^i(j, k, l)) \\ &= \sum_{i=1}^S [-d_{attract} \exp(-w_{attract} \sum_{m=1}^p (\theta_m - \theta_m^i)^2)] \\ &\quad + \sum_{i=1}^S [-h_{repel} \exp(-w_{repel} \sum_{m=1}^p (\theta_m - \theta_m^i)^2)] \end{aligned} \quad (8)$$

where  $J_{cc}(\theta, P(j, k, l))$  is the objective function value to be auditioned to the actual value to represent an objective function changing in time [11].

**(ii) Chemotaxis:** This process reproduces the behavior of an *E. coli* bacterium via the swim and the tumble. The bacterium chooses between the two alternatives during its entire lifetime [16]. Its movement can be defined by

$$\theta^i(j+1, k, l) = \theta^i(j, k, l) + C(i) \frac{\Delta(i)}{\sqrt{\Delta^T(i) \Delta(i)}} \quad (9)$$

where “ $C$ ” represents the “run length unit”, and  $\Delta$  a vector in a random direction with components belonging to the interval  $[-1, 1]$ .

**(iii) Reproduction:** The half of the bacteria population having a bad health (a high objective function value) dies, and the other half having a good health divides into two cells. The new cells are positioned at the same place as their parents. This maintains the number of bacteria in the population constant.

**(iv) Elimination-dispersal:** In the real-world, the consumption of food or the high increase in temperature can stop the chemotactic development, but at the same time, it helps it too because the dispersal event can position bacteria close to good nutrients sources, limiting the stagnation behavior. In BFOA, this process is reproduced by randomly eliminating some bacteria with a little probability, and the new bacteria are randomly placed in the search space.

#### 4.2.3 Bacteria Foraging Optimization Algorithm

A significant factor in the BFOA is its ability to constantly and randomly generate new results during the exploration. This is done by the continual selection of the best bacteria for reproduction [27].

The pseudo-code as well as the flowchart (see Fig. 4) of the BFO algorithm is presented below.

##### BFO Algorithm [2,13,23]

**Step 1:** Initialize variables  $p, S, S_r, N_c, N_s, N_{ed}, N_{re}, P_{ed}, C(i)$  ( $i = 1, 2, \dots, S$ ),  $\theta^i$ .

**Step 2:** Start the elimination-dispersal loop, and increase its index:  $l = l + 1$ .

**Step 3:** Start the reproduction loop, and increase its index:  $k = k + 1$ .

**Step 4:** Start the chemotactic loop, and increase its index:  $j = j + 1$ .

[a] For  $i = 1, 2, \dots, S$ , take for the bacterium  $i$ , a chemotactic step as follows,

[b] Calculate objective function value  $J(i, j, k, l)$ , and let

$$J(i, j, k, l) = J(i, j, k, l) + J_{cc}(\theta^i(j, k, l), P(j, k, l))$$

i.e., add on the cell-to-cell attractant-repellent profile to imitate the swarm behavior, and  $J_{cc}$  is defined in (8).

[c] Let  $J_{last} = J(i, j, k, l)$  to save this value since a better cost can be found through a run.

[d] **Tumble:** Produce a random vector  $\Delta(i) \in R^p$  where every element  $\Delta_m(i)$ ,  $m = 1, 2, \dots, P$  is a random number in  $[-1, 1]$ .

[e] **Move:** The bacterium's position  $i$  is updated by effectuating a step size  $C(i)$  in the tumble direction for this bacterium. Let

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

[f] Calculate the objective function value  $J(i, j+1, k, l)$  by [b] with  $\theta^i(j+1, k, l)$  instead of  $\theta^i(j, k, l)$  and let

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

[g] **Swim:**

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

(ii) While  $m < N_s$ , do

- Let  $m = m + 1$ .
- If  $J(i, j+1, k, l) < J_{last}$ , then  $J_{last} = J(i, j+1, k, l)$ , and use  $\theta^i(j+1, k, l)$  as given in [e] to calculate the new  $J(i, j+1, k, l)$  as in [f].
- Else, let  $m = N_s$ .

[h] Go to the bacterium  $(i+1)$ . If  $i < S$ , go to [b] to process the next bacterium. Else, go to the next step.

**Step 5:** If  $j < N_c$ , go the step 4. And keep on chemotaxis as bacteria are still alive. Else, go to the next step.

**Step 6: Reproduction:**

[a] For the given  $k$  and  $l$ , and for every  $i = 1, 2, \dots, S$ , let

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

where  $J_{health}^i$  is the quantity of nutrients obtained during the life of the bacterium  $i$ . Bacteria and their chemotactic variables  $C(i)$  will be sorted in an order of growing value  $J_{health}^i$ .

[b] The  $S_r$  bacteria having the highest  $J_{health}^i$  values die and the rest divides in two, and the new bacteria are positioned at the same place as their parents.

**Step 7:** If  $k < N_{re}$ , go to step 4. In this case, the number of defined reproduction steps is not attained, so begin the next iteration of the chemotactic loop.

**Step 8: Elimination-dispersal:** For  $i = 1, 2, \dots, S$  with a  $P_{ed}$  probability, remove and disperse every bacterium, thus, keeping the number of bacteria constant. For that, if a bacterium is removed, then another one will be dispersed to a random position on the search space. If  $l < N_{ed}$ , go to step 3, else end.

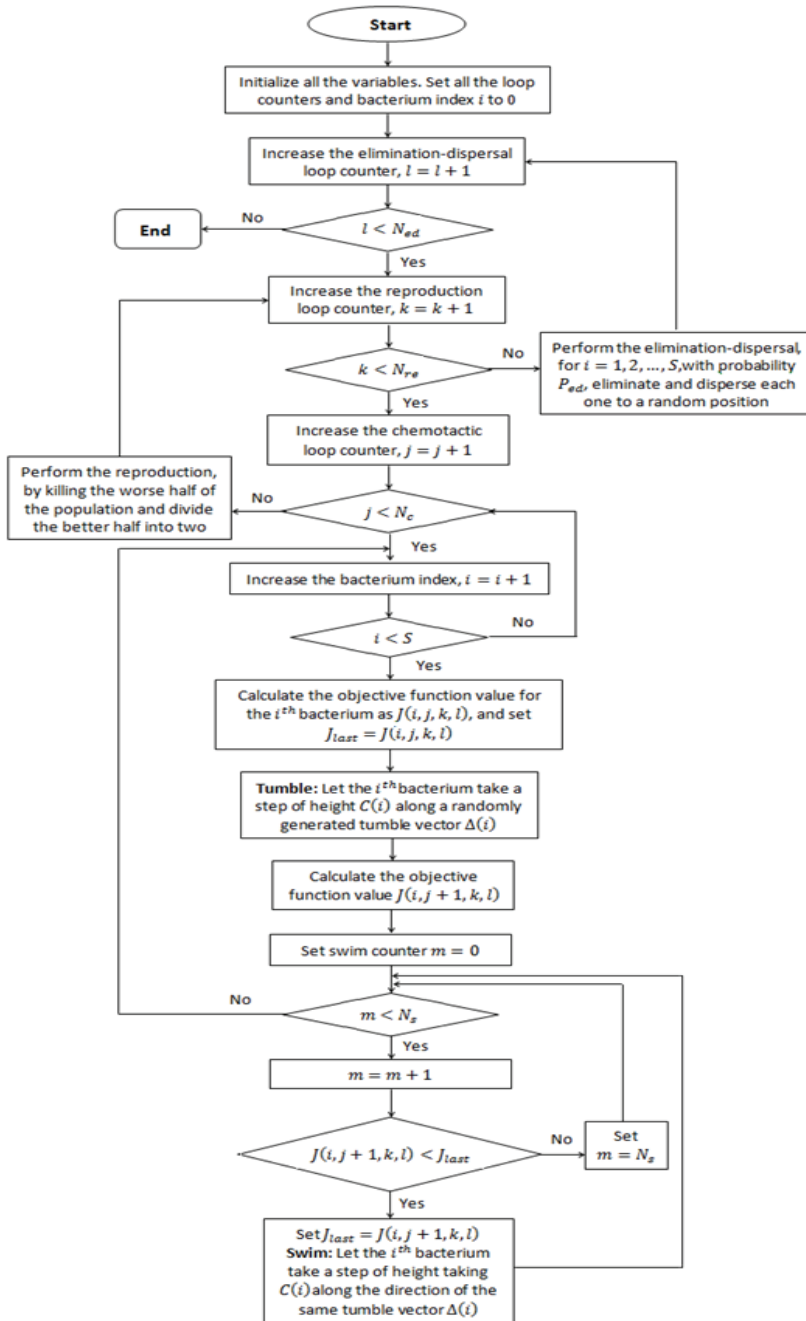


Fig. 4. Flowchart of the bacterial foraging optimization algorithm (BFOA).

## 5. Experimental Study

To overcome the problems of classic RBF neural network, we have integrated the BFOA structure in its hidden layer. The hybrid network will be composed of:



- An input layer representing the test image where every neuron processes a pixel of the image,
- A hidden layer substituted by a population of bacteria that we seek to optimize by performing several tests, and where every bacterium will act as a neuron,
- An output layer representing the classified image where the number of neurons is equal to the number of classes composing the image.

While having a classic RBF neural network but with a substituted hidden layer, we get the hybrid RBF-BFOA approach (neural network).

We devote this section to the effectiveness analysis of the hybrid approach including the BFOA in the RBF neural network, by means of several tests image classification. The effectiveness assessment is based on two criterions. First, the best *classification rate* reached after the predefined number of iterations, and second, the *objective function value J* (see Eq. (8)).

### 5.1 Test Images

In order to assess the performance of the hybrid approach, we test several images shown in Fig. 5. These images are 450×450 in size, and in gray levels.



**Fig. 5.** Test images: (a) Satellite, (b) Boat, and (c) Lena.

### 5.2 Parameters

Two nearby bacteria (neurons) repel each other by local food consumption that is close to each bacterium (neuron). Two bacteria can never be at the same position [2,9]. To represent this phenomenon, let

$$h_{repel} = d_{attract} \tag{10}$$

As mentioned above, some parameters must be initialized before the beginning of the algorithm. The BFOA parameters for the optimal hybrid approach are given in Table 1.

**Table 1.** BFOA initialization parameters

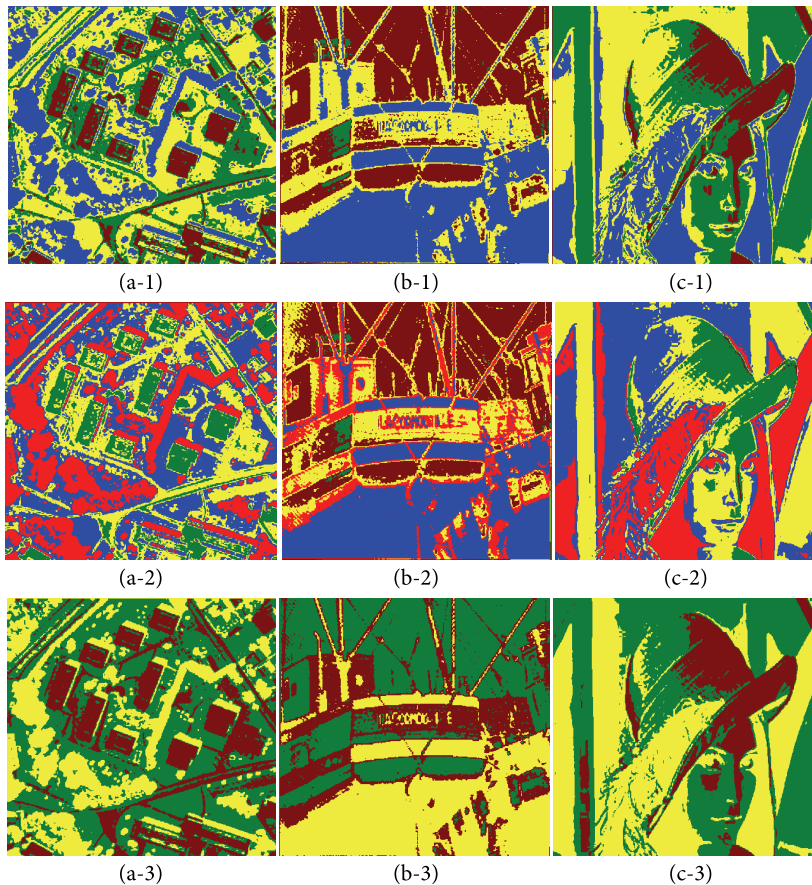
$C(i)$	$N_c$	$N_s$	$N_{ed}$	$N_{re}$	$P_{ed}$	$d_{attract}$	$w_{attract}$	$h_{repel}$	$w_{repel}$
0.05	9	12	4	16	0.25	0.1	0.2	0.1	10

## 5.3 Experimental Results and Discussion

### 5.3.1 Influence of the number of bacteria in the population

The hybrid approach was tested several times with five numbers of bacteria in the population,  $S \in \{8, 10, 12, 14, 16\}$ . The results of the different tests are given in Table 2.

At first, the bacteria number in the population is varied in steps from 8 to 16. Results given in Table 2 show that the classification rate is influenced by the number of bacteria  $S$ . When  $S$  value is small, the hybrid approach is extremely dependent on the first bacterium's position. When the value is increased, the BFOA will cover better the optimization domain, and at least one bacterium is expected to be closer to the optimum point. So the hybrid approach has better probabilities to yield the maximum classification rate. Nevertheless, choosing  $S$  to large causes an increase in computational complexity even if the classification rate also increases (in most cases). Results also show the performance of the hybrid approach with respect to the objective function value. Since BFOA depends less on the number of bacteria than on their first position, the minimum objective function values are similar for large  $S$  values.



**Fig. 6.** Classified test images. (a-1), (b-1), and (c-1) for  $S=1$ . (a-2), (b-2), and (c-2) for  $S=12$ . (a-3), (b-3), and (c-3) for  $S=16$ .

Fig. 6 shows that with  $S$  value equal to 12, we obtained best results and the several image classes are distinct, contrary to  $S$  value equal to 16 where the hybrid approach does not classify images very well.

**Table 2.** Influence of the number of bacteria in the population on classification rate and objective function value

	<i>S</i>	Classification rate (%)	<i>J</i>
Satellite	8	71.78	0.905
	10	74.34	0.818
	12	76.98	0.802
	14	75.56	0.805
	16	73.11	0.803
Boat	8	62.59	0.807
	10	77.88	0.779
	12	78.33	0.765
	14	77.97	0.766
	16	78.07	0.767
Lena	8	73.32	0.915
	10	78.04	0.897
	12	78.26	0.881
	14	77.90	0.885
	16	75.09	0.887

5.3.2 Influence of the number of reproduction steps

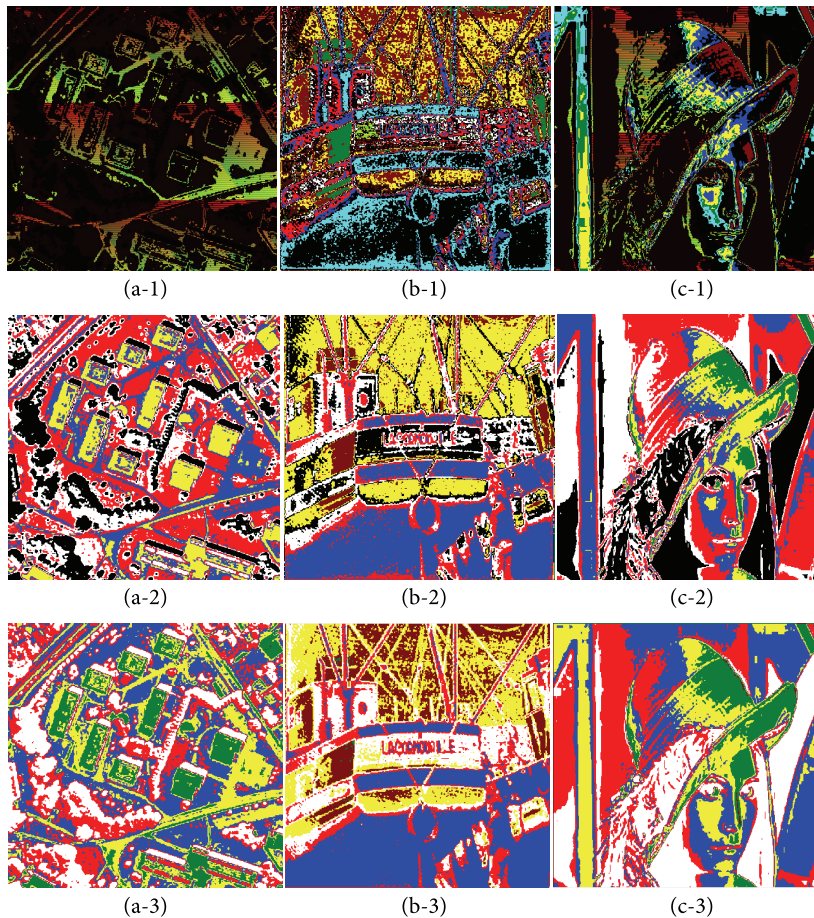
Considering  $N_{re} \in \{14,16,17,18\}$  instead of  $N_{re} = 16$ ,  $S = 12$ ,  $S_r = 6$ , and the other parameters which are presented in Table 1, the results of the different tests are given in Table 3.

Once  $S$  is fixed, we started varying the number of reproduction steps  $N_{re}$  from 14 to 18. This number represents how the BFOA concentrates on favorable areas in the space search and ignores the unfavorable ones, having higher  $J$  values. Results of Table 3 show that the classification rate depends on  $N_{re}$  value. They also demonstrate that the BFOA drops areas if small  $J$  values are not attained. Though the computational complexity increases with the increase in number of reproduction steps, the chance of getting optimum result also increases. If the  $N_{re}$  value is increased, results with inferior values of  $J$  are found, until some threshold (more than  $N_{re}=16$ ).

**Table 3.** Influence of the number of reproduction steps on classification rate and objective function value

	$N_{re}$	Classification rate (%)	<i>J</i>
Satellite	14	72.19	0.867
	16	76.98	0.802
	17	73.09	0.872
	18	70.81	0.892
Boat	14	77.56	0.787
	16	78.33	0.765
	17	78.01	0.807
	18	74.66	0.855
Lena	14	69.45	0.896
	16	78.26	0.881
	17	77.98	0.911
	18	77.98	0.894

Fig. 7 shows that best result images are obtained with a number of reproduction steps equal to 16, so for both classification rate and objective function,  $N_{re} = 16$  gives considerably a good result for image classifications. Hence  $N_{re}$  is fixed at 16.



**Fig. 7.** Classified test images. (a-1), (b-1), and (c-1) for  $N_{re} = 14$ . (a-2), (b-2), and (c-2) for  $N_{re} = 16$ . (a-3), (b-3), and (c-3) for  $N_{re} = 17$ .

### 5.3.3 Influence of the number of elimination-dispersal events

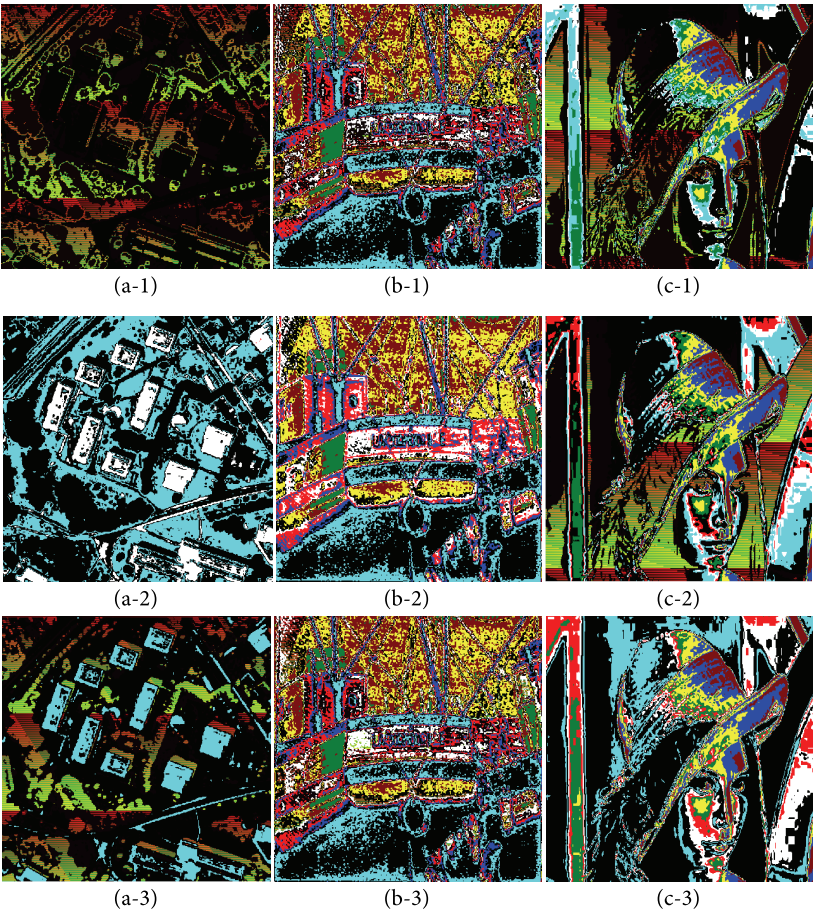
Considering  $N_{ed} \in \{1,2,4\}$  instead of  $N_{ed} = 4$ ,  $S = 12$ ,  $S_r = 6$ , and the other parameters which are presented in Table 1, the results of the three tests are given in Table 4.

Now as  $N_{re}$  is fixed, we can vary the number of elimination-dispersal events  $N_{ed}$ . This number represents how the BFOA attempts to locate a good area in the search space. The results obtained show that if the  $N_{ed}$  value is increased, the algorithm will be capable of finding results with higher classification rate and lower objective function value. The probable reasons are that the selection behavior of bacteria tends to enhance results by maximizing the classification rate and decrease the objective function value through the improvement of the hybrid approach architecture, and then recovering successful foraging strategies, removing bacteria with poor foraging strategies, or poor cell-



to-cell communication, thus maintaining the best bacteria population in a good environment.

Fig. 8 shows that the enhancement in the image quality demonstrates that the hybrid approach, through the BFOA, is in a continual adaptation and improvement, and with  $N_{ed} = 4$  we obtained the best result images.



**Fig. 8.** Classified test images. (a-1), (b-1), and (c-1) for  $N_{ed} = 1$ . (a-2), (b-2), and (c-2) for  $N_{ed} = 2$ . (a-3), (b-3), and (c-3) for  $N_{ed} = 4$ .

**Table 4.** Influence of the number of elimination-dispersal events on classification rate and objective function value

	$N_{ed}$	Classification rate	$J$
Satellite	1	74.75	0.873
	2	76.87	0.856
	4	76.98	0.802
Boat	1	73.98	0.854
	2	77.26	0.838
	4	78.33	0.765
Lena	1	67.54	0.965
	2	74.67	0.901
	4	78.26	0.881

### 5.3.4 Influence of the number of chemotactic steps

Considering  $N_c \in \{7,8,9,10\}$  instead of  $N_c = 9$ ,  $S = 12$ ,  $S_r = 6$ , and the other parameters which are presented in Table 1, the results of the different tests are given in Table 5.

Now as  $N_{ed}$  is fixed, we can vary the number of chemotactic steps  $N_c$ . This number provides the longevity of every bacterium in the population. A small value will restrict the covered search space by every bacterium, thereby limiting the probabilities of the BFOA to discover a local minimum.

**Table 5.** Influence of the number of chemotactic steps on classification rate and objective function value

	$N_c$	Classification rate	$J$
Satellite	7	65.95	0.911
	8	71.02	0.935
	9	76.98	0.802
	10	69.56	0.896
Boat	7	77.57	0.790
	8	77.96	0.767
	9	78.33	0.765
	10	72.01	0.784
Lena	7	71.44	0.899
	8	73.09	0.901
	9	78.26	0.881
	10	70.04	0.893

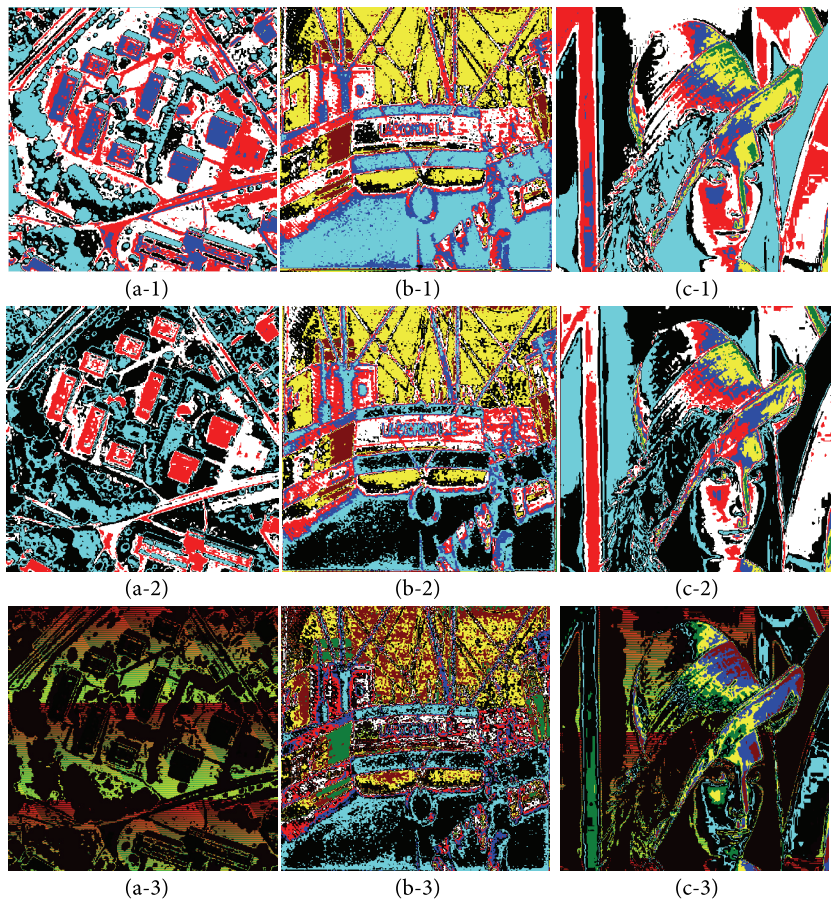
Results of Table 5 show that if the  $N_c$  value is increased, the bacterium finds a new promising region during the exploitation state, so the classification rate have better chance to be improved and the objective function value (cost) will be adapted in the last chemotactic generation into a smaller value. However, if a too large  $N_c$  value is used, the computational complexity will increase, and the quality of the result image will unfortunately decrease (see Fig. 9(a-3), (b-3) and (c-3)).

The variation of the classification rate and the objective function value according to various studied parameters, in the classification of test images, is illustrated by the following graphs.

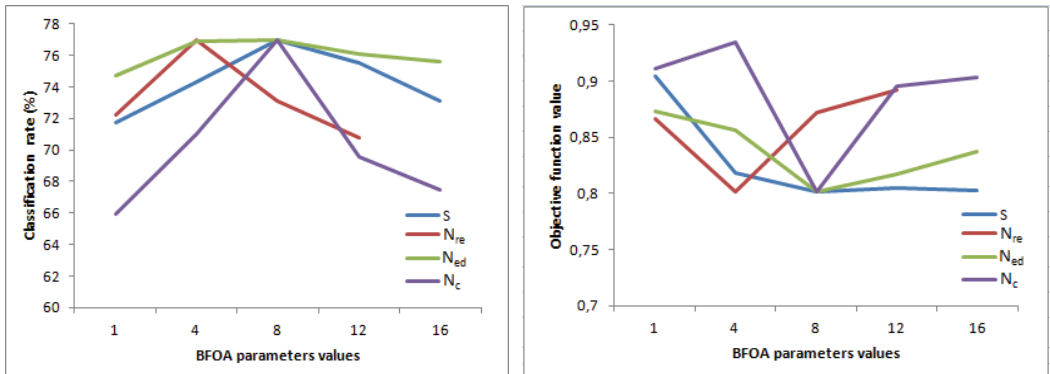
An interesting property of the BFOA is its capability to reproduce efficiently and continuously several hidden neurons (bacteria), through reproduction and elimination-dispersal phenomena. By integrating the algorithm in the hidden layer of the RBF neural network, one can observe, in experiments presented in Figs. 6–9, that the proposed approach can search for the best configuration of the hidden layer, and produces eventually good results. Further, the results obtained after some training achieve better reported results than those reported at the beginning of tests. A possible reason for this behavior is the appropriate flexibility of the approach.

All the simulations of the images' treatment prove that each BFOA parameter is important and influence results at its level. One observes from the obtained images and the literature on the RBF neural network that the number of neurons (i.e., number of bacteria) in the hidden layer determines basically the outcomes. Nevertheless, graphics results (Figs. 10–12) demonstrate that the first role in the image classification performance evaluated by the classification rate and the objective function value is

not played by only one parameter but by all parameters, since each BFOA parameter deeply affects the search results.

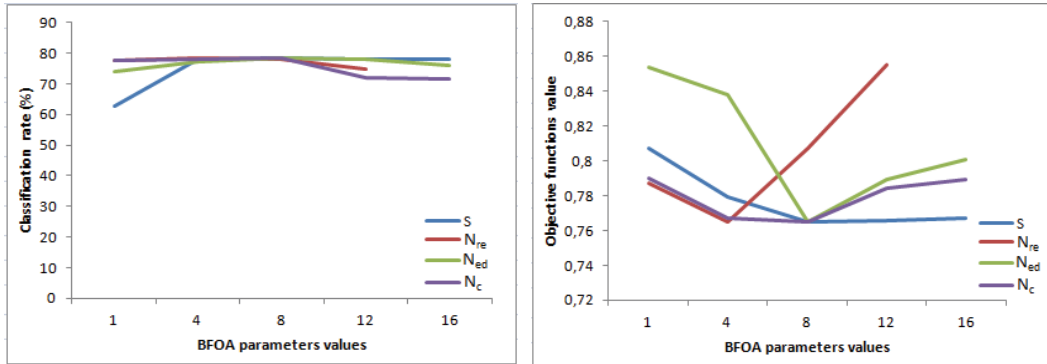


**Fig. 9.** Classified test images. (a-1), (b-1), and (c-1) for  $N_c=8$ . (a-2), (b-2), and (c-2) for  $N_c=9$ . (a-3), (b-3), and (c-3) for  $N_c=10$ .

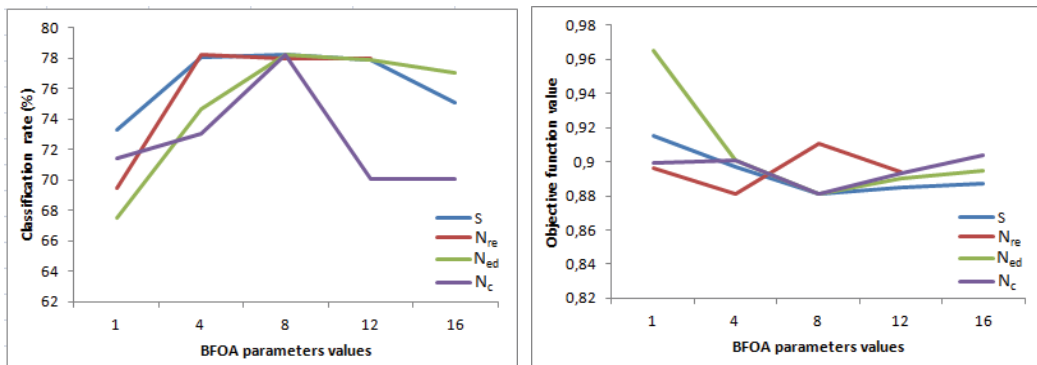


**Fig. 10.** Changes in classification rate and objective function value according to BFOA parameters' variation, in the Satellite image.





**Fig. 11.** Changes in classification rate and objective function value according to BFOA parameters' variation, in the Boat image



**Fig. 12.** Changes in classification rate and objective function value according to BFOA parameters' variation, in the Lena image.

## 6. Conclusions

This paper treated the performance analysis of a new approach hybridizing the RBF neural network with the BFOA applied to image classification. Moreover, the basic goal was to maximize the classification rate while minimizing the objective function value. Various tests were carried out by using the hybrid RBF-BFOA approach and its performance was discussed. As the empirical and digital results show, the suggested hybrid approach produced good classifications. The obtained results clearly show the effectiveness of the approach, and its dependence towards the type and the complexity of the images to be classified, and also illustrate that the proposed approach is very robust and can be clearly prolonged for other optimization problems.

Considering the number of BFOA parameters and since the image classification is a stochastic problem, there are too many variables which can influence the hybrid approach performance. It is a challenge for a researcher to find the best set of BFOA parameters for which the proposed approach performance is suitable. However, this study and experiments are encouraging us to find different improvements, modifications of the BFOA, or of the entire hybrid approach, thus the new model will be able to be used as one more important tool on computations and other optimization applications.

## References

- [1] K. Tang, Z. Li, L. Luo, and B. Liu, "Multi-strategy adaptive particle swarm optimization for numerical optimization," *Engineering Applications of Artificial Intelligence*, vol. 37, pp. 9-19, 2015.
- [2] R. E. Precup, A. L. Borza, M. B. Radac, and E. M. Petriu, "Performance analysis of torque motor systems with PID controllers tuned by bacterial foraging optimization algorithms," in *Proceedings of 2014 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*, Ottawa, ON, 2014, pp. 141-146.
- [3] O. D. Orman and L. Arslan, "A comparative study on closed set speaker identification using RBF network and modular networks," in *Proceedings of 14th Turkish Symposium on Artificial Intelligence and Neural Networks (TAINN)*, Izmir, Turkey, 2000, pp. 291-296.
- [4] J. Holmström, "Growing Neural Gas, Experiments with GNG, GNG with Utility and Supervised GNG", M.S. thesis, Uppsala University, Sweden, 2002.
- [5] D. H. Kim, A. Abraham, and J. H. Cho, "A hybrid genetic algorithm and bacterial foraging approach for global optimization and robust tuning of PID Controller with Disturbance Rejection," *Information Sciences*, vol. 177, no. 18, pp. 3918-3937, 2007.
- [6] Y. T. Amghar and H. Fizazi, "Contributions of bio-inspired methods for satellite images classification," in *Proceedings of International Congress on Telecommunication and Application*, Bejaia, Algeria, 2012.
- [7] R. Tinos and L. O. Murta Jr, "Use of the q-Gaussian function in radial basis function networks," *Foundations of Computational Intelligence*, vol. 5, pp 127-145, 2009.
- [8] R. Kaur and B. Kaur, "Bacterial foraging optimization algorithm for evolving artificial neural networks," *International Journal of Applied Information Systems (IJAIS)*, vol. 8, no. 5, pp. 16-19, 2015.
- [9] K. M. Passino, "Distributed optimization and control using only a germ of intelligence," in *Proceedings of the 2000 IEEE International Symposium on Intelligent Control*, Rio Patras, Greece, 2000, pp. 5-13.
- [10] K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," *IEEE Control System Magazine*, vol. 22, no. 2, pp. 52-67, 2002.
- [11] T. Datta, I. S. Misra, B. B. Mangaraj, and S. Imtiaj, "Improved adaptive bacteria foraging algorithm optimization of antenna array for faster convergence," *Progress In Electromagnetics Research C*, vol. 1, pp. 143-157, 2008.
- [12] S. Narendhar, and T. Amudha, "A hybrid bacterial foraging algorithm for solving job shop scheduling problems," *International Journal of Programming Languages and Applications (IJPLA)*, vol. 2, no. 4, pp. 1-11, 2012.
- [13] S. Das, A. Biswas, S. Dasgupta, and A. Abraham, "Bacterial foraging optimization algorithm theoretical foundations, analysis, and applications," *Foundations of Computational Intelligence*, vol. 3, pp. 23-55, 2009.
- [14] 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.
- [15] S. Das, S. Dasgupta, and A. Biswas, "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, 2009.
- [16] A. Biswas, S. Das, A. Abraham, and S. Dasgupta, "Analysis of the reproduction operator in an artificial bacterial foraging system," *Applied Mathematics and Computation*, vol. 215, no. 9, pp. 3343-3355, 2010.
- [17] P. D. Sathya and R. Kayalvizhi, "Image segmentation using minimum cross entropy and bacterial foraging optimization algorithm," in *Proceedings of the International Conference on Emerging Trends in Electrical and Computer Technology (ICETECT)*, Chunkankadai, India, 2011, pp. 500-506.
- [18] N. Sanyal, A. Chatterjee, and S. Munshi, "An adaptive bacterial foraging algorithm for fuzzy entropy based image segmentation," *Expert Systems with Applications*, vol. 38, no. 12, pp. 15489-15498, 2011.
- [19] G. Mahapatra and S. Banerjee "A study of bacterial foraging optimization algorithm and its applications to solve simultaneous equations," *International Journal of Computer Applications*, vol. 72, no. 5, pp. 1-6, 2013.

- [20] N. Rajasekar, N. K. Kumar, and R. Venugopalan, "Bacterial foraging algorithm based solar PV parameter estimation," *Solar Energy*, vol. 97, pp. 255-265, 2013.
- [21] S. M. Abd-Elazim and E. S. Ali, "A hybrid particle swarm optimization and bacterial foraging for optimal power system stabilizers design," *Electrical Power and Energy Systems*, vol. 46, pp. 334-341, 2013.
- [22] W. Kou, L. Chen, F. Sun, and L. Yang, "Application of bacterial colony chemotaxis optimization algorithm and RBF neural network in thermal NDT/E for the identification of defect parameters," *Applied Mathematical Modelling*, vol. 35, no. 3, pp. 1483-1491, 2011.
- [23] K. M. Passino, "Bacterial foraging optimization," *International Journal of Swarm Intelligence Research*, vol. 1, no. 1, pp. 1-16, 2010.
- [24] F. W. Dahlquist, R. A. Elwell, and P. S. Lovely, "Studies of bacterial chemotaxis in defined concentration gradients: a model for chemotaxis toward L-serine," *Journal of Supramolecular Structure*, vol. 4, no. 3, pp. 329-342, 1976.
- [25] N. K. Jhankal and D. Adhyaru, "Bacterial foraging optimization algorithm: a derivative free technique," in *Proceedings of Nirma University International Conference on Engineering (NUiCONE)*, Ahmedabad, India, 2011, pp. 1-4.
- [26] J. B. Edward, N. Rajasekar, K. Sathiyasekar, N. Senthilnathan, and R. Sarjila, "An enhanced bacterial foraging algorithm approach for optimal power flow problem including FACTS devices considering system loadability," *ISA Transaction*, vol. 52, no. 5, pp. 622-628, 2013.
- [27] M. El-Abd, "Performance assessment of foraging algorithms vs. evolutionary algorithms," *Information Science*, vol. 182, no. 1, pp. 243-263, 2012.



#### **Yasmina Teldja Amghar**

She received, in 2006, her Engineering degree in Computer Science from the University of Sciences and Technology of Oran-Mohamed Boudiaf and her M.S. degree, in 2009, in Modeling and Simulation from the same university. Since 2009, she is at the University of Sciences and Technology of Oran as a PhD candidate. Her research interests include image processing, classification algorithms development for remotely sensed data and analysis of spatial data.



#### **Hadria Fizazi**

She received, in 1981, her Engineering degree in Electrical Engineering from the University of Sciences and Technology of Oran-Mohamed Boudiaf, her "Doctor Engineer" degree, in 1987, in Industrial Computer Science and Automation from the University of Lille 1 in France, and her Ph.D. degree, in 2005, in Computer Science from the University of Sciences and Technology of Oran. She is currently a professor at the Faculty of Mathematics and Computer Science at the University of Sciences and Technology of Oran. Her research interests cover image processing and application of pattern recognition to remotely sensed data.