Sonar False Alarm Rate Suppression using Classification Methods Based on Interior Search Algorithm

Sajjad Ravakhah$^{1}$, Mohammad Khishe$^{2}$, Majid Aghababaie$^{3*}$, Esmail Hashemzadeh$^{4}$

$^{1}$Department of Electrical Engineering, Iran University of Science and Technology, Tehran, Iran.
$^{2,4}$Department of Electrical Engineering, Imam Khomeini Marine University, Nowshahr, Iran.
$^{3*}$corresponding author, Department of Electrical Engineering, Imam Khomeini Marine University, Nowshahr, Iran.

Summary
False alarm rate reduction is one of the challenging issues in the sonar systems. This paper uses classification technique to identify real targets from false alarms. For this purpose, Radial Basis Function Networks (RBFNs) are utilized. Taking into account the use of gradient descent and recursive methods in the classic RBFNs, low Classification accuracy, slow convergence rate, and getting stuck in local optima, are the main drawbacks of RBFNs. In order to overcome these shortcomings, this paper suggests the use of the newly proposed Interior Search Algorithm (ISA) for training the RBFN. In order to measure the performance, ISA is compared with five well-known benchmark algorithms named PSO, ACO, GA, DE, and BBO in terms of entrapment in local minima, classification rate, and convergence speed. The results show that ISA is significantly better than the other well-known benchmark meta-heuristic algorithms in identifying real targets from false alarms.

Key words:
Sonar, Classification, Interior Search Algorithm, Radial Basis Function Networks.

1. Introduction
Classification of underwater targets from the acoustic backscattered signals includes discrimination between target and non-target objects as well as the description of background clutter [1]. There are a lot of components that complicate this procedure such as: non-repeatability and alteration of the target signature with aspect angles, range and grazing angle [2], challenging natural and man-made clutter [3], effects of latitude and longitude [4], highly variable and reverberant working environment [5,6], dependence on the water's temperature, the salinity, the depth [7] and the lack of any pre-knowledge about the form and the geometry of the non-target [8]. Considering mentioned complexities, three main classification schemes have been proposed in recent years: a) Methods based on oceanography [9,10], sonar modeling and engineering [11,12] and also statistical processing [13,14], b) signal processing [15,16] and feature extraction methods [17], and c) development of new classifiers [18-20].

In the first group, researchers attempt to consider environmental circumstances [1-4], multi-path effects [21], sonar specifications [10], sound propagation models [12], topographic effects [9], seabed's scattering models [10], and non-stationary clutter's resources [8] then to calculate the accurate statistical model (statistical distribution) for real targets and non-target echoes [13,14]. In these methods, discrimination was performed by calculating the distribution's parameters. For example, mean and variance of the normal distribution, the shape parameter of the K-distribution, shape and scale parameters of the gamma distribution and lambda parameter of the Poisson distribution.

In the second category, researchers attempt to utilize different signal processing techniques and feature extraction methodologies such as various filters [22], Discrete Wavelet Transform (DWT) [23], Mel-Frequency Cepstral Coefficient (MFCC) [24], Zero Crossing Rate (ZCR), entropy and dynamism features, low-frequency features [25] and etc to extract the best feature for reaching the best performance of the used classifier.

In the last one, scientists propose new classifier to classify sonar dataset effectively. In recent years, many efforts have been done to propose effective classifier in this field [26-30]. Recently, using of Artificial Neural Networks (ANNs) is taken account into consideration for their outstanding achievements [31-34]. High accuracy, versatility, the inherently parallel structure which is very useful in hardware implementation and then real-time processing are some of the distinguished features of ANNs in the sonar target identification which encourage us to use assumed classifier.

RBFNs are one of the most suitable ANNs for industrial application. Using these networks, complex problems can be solved. Generally speaking, RBFNs are used to target identification, function approximation, and time series prediction [35-37]. In spite of their applications, the unique ability of RBFNs is learning [38]. Learning is the fundamental part of all ANNs that can be separated into supervised [39] and unsupervised [40]. Generally speaking, the learning of the RBFNs is a crucial point for them. Many derivative-based methods have been used to train.
RBFNs such as Gradient Descent (GD) [40], Kalman Filter (KF) [41] and Decoupling Kalman Filter (DKF) [42] and also Back Propagation (BP) [43]. As well as derivative-based methods, stochastic methods such as Genetic Algorithm (GA) [44], modified Biogeography-Based Optimization (BBO) [45, 46], Stochastic Fractal Search (SFS) [47], Particle Swarm Optimization (PSO) [48], and Gray Wolf Optimizer (GWO) [49] have been used in training ANNs.

The ultimate purpose of the training process in RBFNs is regulating the best combination of network’s parameters for the sake of the least amount of error. To satisfy aforementioned condition, this paper suggests the use of the recently proposed meta-heuristic algorithm named Interior Search Algorithm (ISA) [50]. The main advantage of ISA is low setting parameters and low complexity in comparison to other meta-heuristic algorithms.

The rest of the paper is structured as follow: Section 2 presents RBFNs. The ISA algorithm is described in section 3. The method of applying ISA as a trainer for RBFN is presented in Section 4. Section 5 discusses the results. Finally, Section 6 provides conclusion and suggests some directions for future work.

2. Radial Basis Function Networks

RBFNs are one of the Feed-Forward Neural Networks (FFNNs) which are composed of three layers (an input layer, hidden layer, and output layer). The general block diagram of a typical RBFN is shown in Fig.1. In RBFNs, outputs of the input layer are manipulated by calculating the distance between inputs and centers of the hidden layer. The outputs of the second layer (hidden layer) are calculated by multiplying the outputs of the input layer and related connection weight. Each neuron of the hidden layer has a center. So, the general description of a typical RBFN is given by equation (1) [47]:

\[ \hat{y}_j = \sum_{i=1}^{I} w_{ij} \phi(\|x - c_i\|) + \beta_j. \]  

(1)

In this paper, Euclidean distance is considered as the classic distance and Gaussian basis function is considered as RBF function as shown by the equation (2):

\[ \phi(t) = \exp(-\alpha_i \|x - c_i\|^2). \]  

(2)

In equations (1) and (2), i is defined as \( i \in \{1, 2, 3, ..., I\} \) where I is the number of hidden neurons, \( w_{ij} \) shows the connection weight from ith neuron in the hidden layer to jth neuron in the output layer, \( \phi \) indicates Gaussian basis function, \( \alpha_i \) shows variance parameter for ith hidden neuron, x is input vector, \( c_i \) is the center vector for neuron i, \( \beta \) shows the bias of jth neuron in the output layer and \( \hat{y} \) is the output of the RBFN.

Fig.2 shows an RBFN with three layer, where the number of inputs (x) is m. In this figure, the number of hidden neurons is \( I \) where the output of each neuron is calculated in terms of the Euclidean distance between the inputs and center vectors. The hidden neuron is included an activation function named RBF Gaussian Basis Function. Outputs of hidden layers transfer to the output layer through weights ( \( w_1, ..., w_2 \) ). The output of the RBFN is a linear combination of the outputs of the hidden layer and bias parameter \( \beta \). Finally, \( \hat{y} \) is calculated as RBF’s output.

![Fig.1: The general description of a typical RBFN.](image1)

![Fig.2: An RBFN with one hidden layer.](image2)
Where $y$ is the desired output and $y$ shows the calculated output. The final aim of RBF training method is minimizing the RMSE.

### 3. Interior Search Algorithm

This section describes Interior Search Algorithm (ISA), which is used to train an RBFN in the next section. ISA is inspired by the architectural processes, which is suggested by Gandomi [50]. This algorithm utilizes the concept of architectural design and decoration. Considering Fitness function, ISA uses two main searching operators named composition and mirror group for solving global optimization problems. In the first stage (composition group), the composition of the searching agents is changed to obtain a more beautiful vision. In the second stage (mirror group), mirrors are placed between these searching agents and the best searching agent to obtain better vision. A typical Scheme of this stage is shown in Fig.3.

![Fig.3: A typical Scheme of mirror search.](image)

Generally speaking, ISA is described as follow:

- **Step 1**: The Positions of searching agents are stochastically generated. These stochastically positions are located between Upper Bounds (UB) and Lower Bounds (LB).
- **Step 2**: Calculate the fitness functions of the searching agents.
- **Step 3**: Obtain the best searching agents. In this special problem (RBFN training), the best searching agent has the minimum fitness function i.e. SSE. This searching agent ($x^i_{gb}$) is the best agent at jth iteration.
- **Step 4**: Other searching agents are stochastically divided into two main groups, composition group and, mirror group. For this purpose, a parameter ($\alpha$) is suggested as equation (4).

$$f \begin{cases} \eta \leq a & \text{then mirror group} \\ \eta \geq a & \text{then composition group} \end{cases}$$

Where $\eta$ is a random value between 0 and 1. It is worth mentioning that $\alpha$ Must be carefully adjusted because it is the only parameter of the algorithm and it balances the swapping behavior between exploration and exploitation phases.

- **Step 5**: In the first stage (composition), the composition of each searching agent is stochastically changed within a bounded search space as follow [50]:

$$x_i^j = LB_i + (UB_i - LB_i) \times r_2,$$  \hspace{1cm} (5)

Where $x_i^j$ is the ith searching agent in the jth iteration, LB$_i$ and UB$_i$ are lower and upper bounds, respectively, and r$_2$ is a random value between 0 and 1.

- **Step 6**: For the second stage (mirror group), a mirror is stochastically located between each searching agent and the global best (best searching agent). The location of a mirror for the ith searching agent at jth iteration is as follows:

$$x_{m,i}^j = r_2 x_{i,j}^{j-1} + (1 - r_2) x_{gb}^i.$$  \hspace{1cm} (6)

Where $r_2$ is a random value between 0 and 1. The location of the virtual position of the searching agents depends on the mirror location, as shown in equation (7).

$$x_i^j = 2 x_{m,i}^j - x_{i,j}^{j-1}.$$  \hspace{1cm} (7)

- **Step 7**: It is useful for the best searching agent (global best) to slightly change its location using the random walk as follows:

$$x_{gb}^j = x_{gb}^{j-1} + r_n \times \lambda.$$  \hspace{1cm} (8)

Where $r_n$ is a vector of random numbers, and $\lambda$ is a scale factor depending on the size of the search space as follows [50]:

$$\lambda = 0.01 \times (UB - LB).$$  \hspace{1cm} (9)

This random vector works as a local search operator because it discovers around the best searching agent.

- **Step 8**: In the next step, the fitness functions of the searching agent and virtual agent’s location are calculated. Then update their positions if their fitness values are improved this step can be expressed as:

$$x_i^j = \begin{cases} x_i^j & f(x_i^j) < f(x_i^{j-1}) \\ x_i^{j-1} & \text{else} \end{cases}.$$  \hspace{1cm} (10)

- **Step 9**: Repeat the algorithm from step 2, if any of the stop criteria is not satisfied. The pseudo code of ISA is shown as follows.

```plaintext
While
    For I = 1 to n
        If $x_{gb}^i$
            $x_{gb}^i = x_{gb}^{j-1} + r_n \times \lambda$
        Else if $\eta_i < \alpha$
            $x_{m,i}^j = r_2 x_{i,j}^{j-1} + (1 - r_2) x_{gb}^i$
        Else
            $x_i^j = x_i^{j-1}$
    End For
End While
```
4. The Training of RBFN Using ISA

Generally speaking, there are three methods to present the parameters of RBFN: 1) vector-based, 2) matrix-based and 3) binary state [1]. In the vector-based method, only one vector presents each searching agent. All weights, biases, and related centers should be obvious to train an RBFN. In the matrix-based presentation, one matrix presents each searching agent. In binary state presentation, each searching agent is shown in the form of a string of binary bits. Any of these presentations have special disadvantages and advantages that can be helpful within a specific problem.

In this paper, the vector-based presentation is utilized to present the RBFN, because its structure is not very complex. Also, the ANN toolbox of MATLAB software is not utilized in order to reduce the time of running RBFN. As previously stated, training RBFN can be obtained by selecting optimum values for the following parameters:

- W: Weights between the output layer and hidden layer.
- a: Emission parameters of the Gaussian Basis function of the hidden layer.
- c: Centers of the hidden layer.
- β: Bias parameters of neurons in output layer.

The number of neurons in the hidden layer is crucial parameter that should be carefully specified. Using more neurons than the normal number leads to over-fitting network, which caused to increase structural complexity and algorithm’s running time. According to [47] and studies that have been carried out, 4 neurons are chosen in the hidden layer. ISA’s searching agents are consist of weight ( ), bias vectors ( ), center vectors (c) and emission parameters ( ). A typical searching agent of ISA can be presented in the vector that is shown in equation (11):

\[
P_i = \left[ w_ia\beta \right]
\]

As previously stated, the ultimate aim of the training methods is tuning the special parameters of the RBFN. Each training iteration should calculate the fitness value of all searching agents. In this paper, searching agent’s fitness values are calculated by SSE as follows:

\[
f = E_{SSE}
\]

5. Setting Parameters and experimental result

In order to evaluate the performance of ISA in training RBFN, as well as ISA, the RBFN are taught by some well-known benchmark algorithms such as Biogeography Based Optimization (BBO), Ant Colony Optimization (ACO), Differential Evolution (DE), Particle Swarm Optimization (PSO), and Genetic Algorithm (GA). The initial values and essential parameters of these algorithms are presented in Table 1. In the next section, sonar dataset will be completely explained and then the designed RBFN will be evaluated on that dataset.

5.1 Sonar dataset

This paper uses Sonar dataset that is extracted from Gorman and Sejnowski marine experiment available in references [51, 52]. In this experiment, there are two types of echo: the first relates to the metallic cylinder (real target) and the second relates to a rock as the same size as the cylinder (false alarm).

In the Gorman and Sejnowski experiment, a metal cylinder with a length of 5 feet and a rock with the same size located on the sandy seabed and a wide-band linear FM chirp pulse (ka=55.6) has been transmitted to the real target and false target.

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>BBO</td>
<td>The probability of correcting the habitants</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>The probability range for migrating into for each gene</td>
<td>[0, 1]</td>
</tr>
<tr>
<td></td>
<td>Step size for the probability numerical integral</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Maximum migration into (I) and migrating out of (E) coefficient</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Mutation probability</td>
<td>0.005</td>
</tr>
<tr>
<td></td>
<td>Population size</td>
<td>200</td>
</tr>
<tr>
<td>PSO</td>
<td>Layout</td>
<td>Full connection</td>
</tr>
<tr>
<td></td>
<td>Cognitive constant (C1)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Social constant (C2)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Local constant (W)</td>
<td>0.3</td>
</tr>
</tbody>
</table>
Population size 200

**GA**
- Type: Real coded
- Selection: Roulette wheel
- Recombination: Single-point (1)
- Mutation: Uniform (0.01)
- Population size 200

**ACO**
- Primary pheromone (τ₀): 0.000001
- Pheromone updating constant (Q): 20
- Pheromone constant (q₀): 1
- Decreasing rate of the overall pheromone (P_g): 0.9
- Decreasing rate of local pheromone (P_t): 0.5
- Pheromone sensitivity (α): 1
- Observable sensitivity (β): 5
- Population size 200

**DE**
- Weighting factor (F): 0.5
- Crossover constant (CR): 0.5

**ISA**
- Random value [0.1, 0.3]

Backscattered echoes have been accumulated in the distance of 10 meters. Based on the SNR of received echo, of 1,200 collected backscattered echo has been selected 208 echoes that their SNR between 4dB to 15dB. From this 208 backscattered echoes, 111 ones are of metal cylinders and 97 echoes are related to false alarm (rocks). Fig.5 shows typical backscattered echoes from the rock and metal cylinder.

The pre-processing method for obtaining the spectrum envelope is shown in Fig.6. Fig.6a indicates a typical apertures and Fig.6b show a set of sampling apertures that are applied on two-sided spectrogram of the Fourier transform of the backscattered echoes. Spectral envelope is obtained from the accumulation of all aperture’s effects.

5.2 Sonar False Alarm Suppression

After pre-processing of the backscattered echoes, in this section, the normalized dataset got exerted on the RBFN, which is trained by various meta-heuristic algorithms. The designed RBFN is applied on sonar dataset and the performance of the newly proposed RBFN is evaluated in terms of the convergence speed and false alarm rate suppression. Each designed RBFN is executed 10 times and then the average suppression rate (Classification rate) is presented in Table 2. The typical results for convergence curve are shown in Fig.7.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>ISA</th>
<th>PSO</th>
<th>BBO</th>
<th>ACO</th>
<th>DE</th>
<th>GA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Suppression Rate</td>
<td>93.17%</td>
<td>79.78%</td>
<td>91.45%</td>
<td>72.25%</td>
<td>85.39%</td>
<td>88.45%</td>
</tr>
</tbody>
</table>

According to the Table 2 and Fig.7, ISA with %93.17 has the best performance among the benchmark algorithms and ACO with %72.25 has the weakest performance. Regarding the high-dimension and the extra local minima of the sonar dataset, the possibility of falling into local minima is too much for an algorithm such as ACO. Whereas ISA with stochastic nature, having two powerful searching group (composition group and mirror group), and just one setting parameter, has better performance than other benchmark meta-heuristic algorithms.

6. Conclusion

In this paper, a newly proposed meta-heuristic algorithm known as ISA is firstly used to train an RBFN. To evaluate the performance of designed RBFN, sonar dataset have
been used and then obtained result are compared with BBO, PSO, ACO, DE, and GA. Results indicated that the ISA due to a simple structure and the capability to discover the search space, is able to provide much better results in terms of convergence speed and false alarm suppression’s rate in compare to benchmark algorithms. Due to the simple structure of Multi-Layer Perceptron Neural Network (MLP NN), it can be used as a neural network in future works instead of RBFN.

References


Q. Zhang and B. Li, “A Low-Cost GPS/INS Integration based on UKF and BP Neural Network,” Fifth International Conference on Intelligent Control and Information Processing (ICICIP), pp.100-107, 2014.


Sajjad Ravakhah was born on 1991, in Isfahan, Iran. He received the B.Sc. degree in Robotic Engineering from the Shahrood University of Technology, Shahrood, Iran in 2014 and M.Sc. degree in Electrical and Electronic from the University of Science and Technology, Tehran, Iran in 2016. His research interests are Array Signal Processing and Distributed Detection.

Mohammad Khishe received his B.Sc. degree in Imam Khomeini Maritime Sciences University, Noshahr, Iran, M.Sc. degrees in Islamic Azad University, Qazvin Branch in 2007 and 2011, respectively. He is currently a Ph. D. student in Electronic Engineering at Iran University of Science and Technology and a faculty member of Electronic and Communication Engineering at Imam Khomeini Maritime Sciences University, Noshahr, Iran. His research interests include neural networks, meta-heuristic algorithms and digital design.
Majid Aghababaee was born on 1971, in Tehran, Iran. He received the B.Sc. and M.Sc. degrees in Electrical and Electronic Engineering from the Sharif University of Technology, Tehran, Iran in 1992 and 1995 respectively, Ph.D. degree in Electrical and Electronic Engineering from Iran University of Science and Technology, Tehran, Iran in 2011. Since 2000, he is a faculty member of Electrical and Electronic Engineering at Imam Khomeini Maritime Sciences University, Noshahr, Iran. His research interests are Digital Signal Processing, Digital Image Processing, Sonar and Radar Signal Processing and Power Electronics.

Smaeel Hashemzadeh was born on 1981, in Babol, Iran. He received the B.Sc. degree in Electrical and Electronic Engineering from the Imam Khomeini Naval University, Noshahr, Iran in 2006 and M.Sc. degree in Telecommunications Engineering from the Islamic Azad University, Bushehr, Iran in 2011. His research interests are Array Signal Processing, Sonar and Radar Signal Processing and Distributed Detection.