2012 International Conference on Information and Network Technology (ICINT 2012)
IPCSIT vol. 37 (2012) © (2012) IACSIT Press, Singapore

Array Signal Processing & Optimization using Algorithms in Nature

Sumanta Bose +, Prabu. K and D. Sriram Kumar

Department of Electronics & Communication Engineering, National Institute of Technology, Trichy, India

Abstract. Computer Science algorithms and optimization techniques have been deeply inspired from biological models in nature since its inception. Several computational methods are derived from the mathematical model of the high-level design principles of biological systems. Array signal processing is one such application that has emerged as a promising technology for the fourth generation (4G) wireless communication network. It is widely used in smart antenna technology for estimating direction-of-arrival (DOA) and beam-forming. In the recent past it has evolved massively delivering enhanced performance by utilizing algorithms in nature: Artificial Neural Network (ANN), Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Bacteria Foraging optimization (BFO) and Ant Colony Optimization (ACO). This paper discusses the algorithms and analyses the performance of the stated optimization techniques.

Keywords: algorithms in nature, array signal processing, ANN, GA, PSO, BFO, ACO.

1. Introduction

The classical problem in array signal processing is to determine the location of an energy radiating planar source relative to the location of the array. We are interested in estimating the direction-of-arrival of a signal in the presence of noise and interfering signals. The antenna array filters data collected over a spatial spectrum much in the same manner that an FIR filter processes temporally sampled data. One such popular technique is the MUltiple SIgnal Classification (MUSIC) algorithm. However, new generation array signal processing techniques employ iterative evolutionary algorithms inspired from nature to estimate the location of energy radiators.

This paper explains and compares such algorithms. The rest of the paper is distributed as such: Section 2 lays the mathematical formulation; Section 3 explains the different algorithms; Section 4 analyses them on the basis of performance and Section 5 concludes the paper.

2. Mathematical Formulation

To facilitate the discussion, mathematical notation must be developed for referring to the array processing system parameters. Consider an array of $\boldsymbol{m}$ sensors having inter-element spacing Δ which receives signals generated by $\boldsymbol{d}$ narrowband sources with known centre frequency ω and locations $\theta_1, \theta_2 \cdots \theta_d$. Since the signals are narrowband, the propagation delay across the array is much smaller than the reciprocal of the signal bandwidth $\boldsymbol{\lambda}$, and it follows that by using a complex envelop representation, the array output can be expressed as $\boldsymbol{z}(t) = \sum_{i=1}^{d} \{a(\theta_i)s_i(t) + \boldsymbol{n}(t)\}$. Here, $\boldsymbol{z}(t)$ is the vector of the signals received by the array sensors; $s_i(t)$ is the signal emitted by the i^{th} source as received at the reference sensor of the array; $a(\theta_i)$ is the steering vector of the array toward direction θ_i; and $\boldsymbol{n}(t)$ is the noise vector. We may also express the system as $\boldsymbol{z}(t) = \boldsymbol{A}(\Theta)\boldsymbol{s}(t) + \boldsymbol{n}(t)$; where $\boldsymbol{A}(\Theta) \stackrel{\text{def}}{=} [a(\theta_1) \cdots a(\theta_d)]$ is called the steering matrix. The received signal spatial correlation matrix $\boldsymbol{\mathcal{R}}_{zz}$ can be expressed as $\boldsymbol{\mathcal{R}}_{zz} = \boldsymbol{A}(\Theta)\boldsymbol{\mathcal{R}}_{ss}\boldsymbol{A}(\Theta)^{\mathrm{H}} + \boldsymbol{\mathcal{R}}_{nn}$; where $\boldsymbol{\mathcal{R}}_{ss}$ and $\boldsymbol{\mathcal{R}}_{nn}$ are the spatial correlation matrix of the signal-of-interest and noise respectively and $(\cdot)^{\mathrm{H}}$ denotes the conjugate transpose.

+ Corresponding author. Tel.: +91 9655441077
E-mail address: sumantabose@gmail.com

3. Application of Algorithms-in-Nature to Array-Signal-Processing

3.1. Artificial neural network (ANN):

Artificial Neural Network is widely used for DOA estimation [1, 2] where the estimation problem is mapped onto a Lyapunov energy function for the Hopfield network to obtain the optimum estimates. Considering the mathematical model discussed in Section 2, the antenna array can be thought to perform a mapping $\boldsymbol{G}: \boldsymbol{\mathcal{R}}_{zz}^{\boldsymbol{d}} \rightarrow \boldsymbol{C}^{\boldsymbol{m}}$ from the space of DOAs $\{\Theta = [\theta_1, \theta_2, \cdots, \theta_d]\}$ to $\{\boldsymbol{z}(t) = [z_1(t)\ z_2(t) \cdots z_m(t)]\}$. The Radial Basis Function Neural Network (RBFNN) is used to perform the inverse mapping $\boldsymbol{F}: \boldsymbol{C}^{\boldsymbol{m}} \rightarrow \boldsymbol{\mathcal{R}}_{zz}^{\boldsymbol{d}}$.

The RBFNN has 3 layers of nodes: input, hidden & output layer. Inputs are applied to the input layer; outputs are produced at the output layer. The RBFNN performs an input/output mapping trained with examples. The hidden layer of the RBFNN transforms the input data from an input space of some dimensionality to a new space of possibly higher dimensionality. The rationale behind this transformation is based on Cover's theorem.

For the array with $\boldsymbol{m}$ sensors and $\boldsymbol{d}$ impinging signals, the network consists of $\boldsymbol{m} \times \boldsymbol{m}$ input units and $\boldsymbol{d}$ output units. Let the hidden layer has $\boldsymbol{n}$ neurons. This network is trained to perform an input/output mapping by varying the synaptic weight parameters $\boldsymbol{\mathcal{W}}$, leading to the linear relation $\Theta = \boldsymbol{\mathcal{W}}^{\mathrm{T}}\Phi$; where Θ represents the desired response and Φ represents the learning data matrix. The weight is adjusted by the network according to the Delta Rule or RLS to obtain the best estimate of $\boldsymbol{F}$, thereby estimating the DOAs of the signals.

After the training of the RBFNN is complete, the trained neural network can operate in the performance mode. In this mode the network is generalized i.e. it responds to unknown inputs signals (drawn from the same distribution as the training inputs) to accurately estimate the DOA of the signals. Zooghby et. al. in [1] have summarized the mathematical steps of the RBFNN approach.

3.2. Genetic algorithm (GA):

Genetic Algorithms are adaptive search algorithms based on the mechanism of natural selection where fitter individuals have higher chances to survive in a competing environment. The first step of GA is to generate an initial population of possible solutions containing signal parameters/ chromosomes (DOA in our case). The next step is to evaluate each individual's fitness, and rank the population in their fitness order. The core of GA consists of two genetic operators: crossover and mutation. Crossover is the method of exchanging genetic information between parents to generate a higher quality offspring for the next generation. Mutation, on the other hand is the method that randomly alters the population state in order to diversify the population and force the algorithm to search in previously unexplored areas of the search space. Relatively "fitter" individuals survive and "unfit" individuals are discarded in successive generations to yield fitter solutions that approach the optimal solution to the problem.

GA based solution for the DOA problem is an optimization problem, for which the Steered Response Power with Phase Transform filter (SRP-PHAT) expression is used [3]. An n variable fitness function in GA optimization is defined as f: $\mathbb{R}^n \rightarrow \mathbb{R}$ where $\mathbb{R}^n$ is an n dimensional space. The search space of the GA is defined as $\mathbb{S} \stackrel{\text{def}}{=} \mathbb{R}^n \cap \mathbb{F}$ where $\mathbb{F}$ is the feasible search space. The fitness function to be optimized is spatial likelihood function (SLF) described by SRP-PHAT. Detailed GA implementation for DOA estimation are discussed in [3, 4].

3.3. Particle swarm optimization (PSO):

Particle Swarm Optimization (PSO) is an optimizer engine developed by Kennedy and Eberhart that emulates the behaviour of a swarm of bees in their chase of the biggest concentration of flowers in a field. Initially, each individual or particle randomly travels over the field. Each individual records its best position (p_{best}), and the best position of any bee in the swarm or group (g_{best}). With this knowledge, the particle decides where to move next and how fast: towards p_{best}, towards g_{best} or, more likely, towards a weighted position between p_{best} and g_{best}. Finally through iterations the swarm gathers around the position with the biggest concentration of flowers (final g_{best}).

For PSO based DOA estimation [5], the solution to the problem is a point of the $\boldsymbol{d}$ -dimensional space where each coordinate is given by the direction of arrival of each signal and can vary between 0º and 180º. A point in space is a solution if a certain fitness function for that point takes values below a certain threshold, or if the number of movements allowed for the particles has taken its maximum value.

3.4. Bacterial foraging optimization (BFO):

The Bacterial Foraging optimization (BFO) technique, as introduced by Passino, is inspired from the swarm behaviour of E. coli bacteria. The idea of BFO is based on the fact that natural selection tends to eliminate bacteria with poor foraging strategies, and tends to favour those having better foraging strategies, which results in a healthier population with superior foraging strategies in subsequent generations.

The group of bacteria forage for food by trying to move upward along the food-concentration gradient. They tumble randomly, swim for a fixed distance (exhausting 1 chemotactic step) and continue in that direction if the food-concentration gradient is +ve; else they randomly change direction. This continues until the bacteria's lifetime. At the end of their lifetimes, half the bacteria that are in the better concentration region reproduce to generate the next generation, whereas the other half die, keeping the population constant. This process is iteratively carried out until the desired variable is optimized i.e. the entire population is above a threshold concentration.

The BFO is governed by four processes, viz., chemotaxis, swarming, reproduction, and elimination & dispersal. For BFO based DOA estimation [6], an exact ML function is defined as: $\mathrm{f_{EML}}(\theta) \stackrel{\mathrm{def}}{=} \log\left|\mathrm{A}(\theta)\mathrm{S}(\theta)\mathrm{A^H}(\theta) + \sigma^2(\theta)\mathrm{I}\right|$ where $\mathrm{S}(\theta) = \{\mathrm{A}'(\theta)(\mathcal{R}_{\mathrm{zz}} - \sigma^2\mathrm{I})\mathrm{A'^H}(\theta)\}$ and $\sigma^2(\theta) = (1/\boldsymbol{m} - \boldsymbol{d})\,\mathrm{Tr}\left\{\mathrm{P_A^\perp}(\theta)\mathcal{R}_{\mathrm{zz}}\right\}$. Here A' is the pseudo-inverse of A and $\mathrm{P_A^\perp}$ is the orthogonal projection onto the null spaces of $\mathrm{A^H}$. Now this $\mathrm{f_{EML}}$ is regarded as the fitness of a bacteria and BFO is simulated to estimate DOA.

3.5. Ant colony optimization (ACO):

Ant colony optimization (ACO), as developed by Dorigo, is a metaheuristic swarm intelligence algorithm inspired from the foraging behaviour of ants. The principle behind ACO is based on pheromone deposition by ants while travelling to-and-from food sites and its subsequent decay. A path with large level of pheromone signifies one of two possibilities: either the path is long and a lot of ants traversed it because a large food source was found, or the path is short and a moderate number of ants recently traversed the path.

ACO was originally used to solve discrete combinatorial optimization problems and its application in the continuous domain was extended later. It has thereafter been found that ACO could be adapted to solve continuous optimization problem and obtain the approximate solutions within a reasonable number of iteration.

For the problem of DOA estimation [7, 8], the search space is sampled and L solution vectors are uniformly distributed in the search range $[\theta_l, \theta_h]$. For each solution Θ_l, its fitness value $f(\Theta_l)$ is computed and is associated with a weight, associated with its Gaussian Function. Thereafter the solutions are sorted in descending order of the $f(\Theta_l)$ value and stored in an archive.

Different Gaussian function that composes the Gaussian kernel $G^i(\Theta)$ is chosen with respective probability $\wp_l$ and sampled. The Gaussian kernel $G^i(\Theta)$ defined as a weighted sum of several 1D Gaussian functions is used for updating the solutions. If the fitness value of the new solution Φ is greater than the worst fitness values, i.e. if $f(\Phi) > f(\Theta_1)$, then the last solution Θ_L is replaced with the new solution Φ. The solution archive is sorted again in descending order of fitness values. This process represents one ant travelling through the solution space. The replacements of the solution is similar to pheromone update of a potential pathway through the solution space. This is continued for a fixed number of iterations or till the solutions approaches the estimate to determine the DOA.

4. Simulation Results and Performance Analysis:

4.1. Artificial neural network (ANN):

The RBFNN has the ability to interpolate data into higher dimensions. It does not require an initial estimate of the direction of the sources. The DOA estimation is done through a nonlinear mapping from the space of sensor output to that of the angles theta, and not by any matrix inversion or decomposition. Furthermore, the parallel implementation of the RBFNN makes it a real-time approach with wide scope of VLSI implementation. Although a large learning set is used in learning, the learning itself can be carried out offline. This makes the network generalized by training and testing data sets derived from different signal conditions mainly with the effect of noise added to the data used for testing.

Computer simulations in [2, 9] show that RBFNN has a better performance in terms of estimation errors compared to the standard MUSIC algorithm. It yielded better performance in the sense that the network produced actual output very close to the desired DOA. The main advantage of the RBFNN is the substantial reduction in the CPU time needed to estimate the DOA.

4.2. Genetic algorithm (GA):

For a signal with DOA of $\theta = 29°$ & $\varphi = 39°$ impinging on an array with 8 sensors with SNR of 30dB, the plot of the SLF values [3] as a function of θ & φ is shown in Figure 1, which gives a clear indication of the DOA. The smaller peaks are due to reverberation and noise.

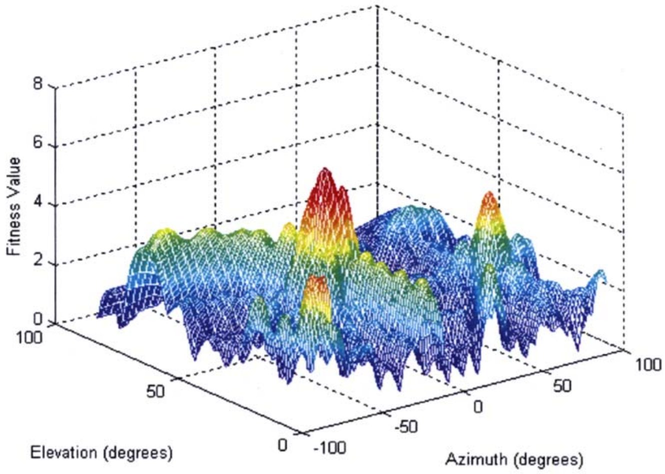

Fig. 1: SLF plot of SRL-PHAT based GA as a function of θ & φ

The GA for DOA estimation succeeded in giving reliable results and converged globally when it was used to compute exact solutions for the ML function [10].

At low SNR values, MUSIC might fail to resolve the two sources by showing only one peak in its spectrum, while GA-based ML method can differentiate it, as it is statistically consistent and efficient than MUSIC at lower SNR values. However at higher SNR values, both behave similarly. Another factor is that the computational time of GA-based ML method depends on the number of generations required to obtain a satisfactory fitness value, which increases with the number of sources. [10] shows that the computing time of GA-based ML method is far less than that of a fine grid MUSIC.

On the whole, the statistical performance of GA-based ML method is superior to MUSIC in the threshold region and at lower SNR values.

4.3. Particle swarm optimization (PSO):

The PSO method for DOA estimation can not only be applied for the cases where MUSIC algorithm is useful, but also for the cases where MUSIC is not useful, for example, when the number of signal sources is greater than or equal to the number of the elements of the array. It can simultaneously estimate the DOAs and powers of signals accurately.

Computer simulations show that the DOA RMSE increases with SNR value almost in the same way for MUSIC and PSO methods, indicating similar performances. For $d < m$ the PSO continues to perform while the MUSIC fails.

For the model described in Section 2, the performance of PSO for DOA estimation has been deduced in [12]. The fitness function are in decreasing order of magnitude for increasing d. It can be deduced that the number of iterations needed to reach a certain fitness value or angular error grows with $\boldsymbol{d}$. It is also robust against noise, and the error decreases with an increase in $\boldsymbol{m}$.

It is noteworthy that for couple of equally spaced signals, detection is significantly better for those signals whose angle of incidence is around angles near 90^{o}. It is noteworthy that the initial guess of direction does not influence the algorithm's convergence. Moreover its computational complexity is also modest. A comparison is illustrated in Section 4.4.

4.4. Bacterial foraging optimization (BFO):

Datta et. al. in [11] compared the BFO and PSO for array signal processing applications. The two methods were compared by placing them in the same platform. It was observed that PSO converged faster than BFO and had a lower average cost function. PSO performed better in null-steering, but sidelobe suppression was better with BFO.

In another work [6], the BFO and PSO based EML method were compared with the MUSIC algorithm. For an 8 sensor UCA with signals impinging from $\theta = 130^{\mathrm{o}}$ & $\theta = 140^{\mathrm{o}}$ with the following specifications: number of snapshots=*20,* dimension of search space=2, number of bacterium=10, chemotactic steps=20, swim steps=4, reproductive steps=4, elimination & dispersal steps=2, probability of elimination=0.25, run-length during each tumble=0.005, the DOA estimation RMSE values were obtained using PSO-EML, BFO-EML, and MUSIC as a function of SNR. Figure 2. shows the plot obtained.

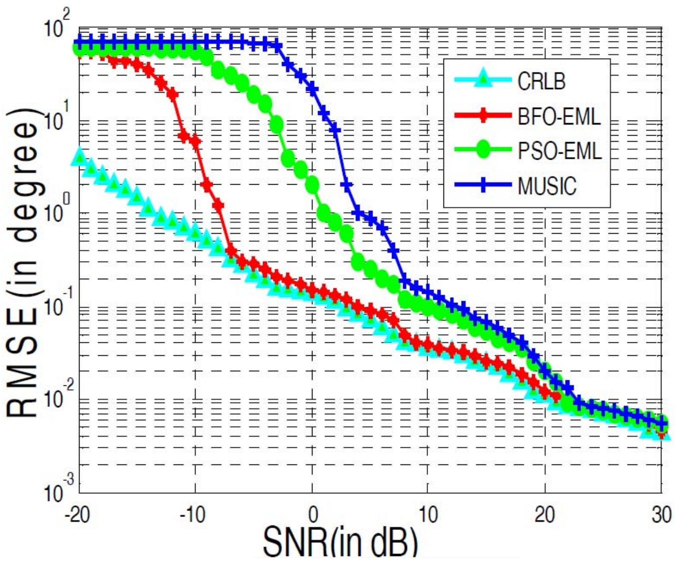

Fig. 2: RMSE for different algorithms as a function of SNR

It is clear that BFO-EML yields significantly superior performance over PSO-EML as a whole, by demonstrating lower DOA estimation RMSE and higher resolution probabilities.

4.5. Ant colony optimization (ACO):

ACO has been applied to MUSIC in [7] and to Multidimensional MUSIC (MD-MUSIC) in [8]. Simulations show that ACO-MUSIC maintains the high accuracy of 2-D MUSIC algorithm. In fact, the RMSE of the two methods roughly maintains similar level. Yang et. al. in [7] have defined a kind of Gauss kernel function to dynamically sample the continuous solution space, thereby reducing the computational cost by using ACO without compromising on the excellent performance of the original MUSIC. The ACO-MD-MUSIC [8] uses a kind of continuous pheromone probability distribution for the sampling process. The conventional MD-MUSIC algorithm is impractical due to its prohibitive computational burden incurred by the multidimensional grid search. However, its ACO implementation (ACO-MD-MUSIC) can reach its global maximum after reasonable number of iterations. It provides performance similar to the MD-MUSIC, while its computational cost is only $(1/13)^{\mathrm{th}}$ of MD-MUSIC algorithm, hence it is more suitable for real-time engineering applications.

Consider a 12 sensor ULA with each sensor $\lambda/2$ apart, on which 2 uncorrelated narrow-band signals impinge from -3^{o} & 3^{o}. Performing ACO-MD-MUSIC with the following parameters: number of solution vectors=100, balance factor between the iteration-best and the best-so-far=0.1, pheromone evaporation rate=0.01, maximum allowable error=0.001, SNR=10dB, $[\theta_l, \theta_h]$=$[-60^o, 60^o]$. On performing 100 Monte Carlo experiments, the spatial spectrum as shown in Figure 3 is produced.

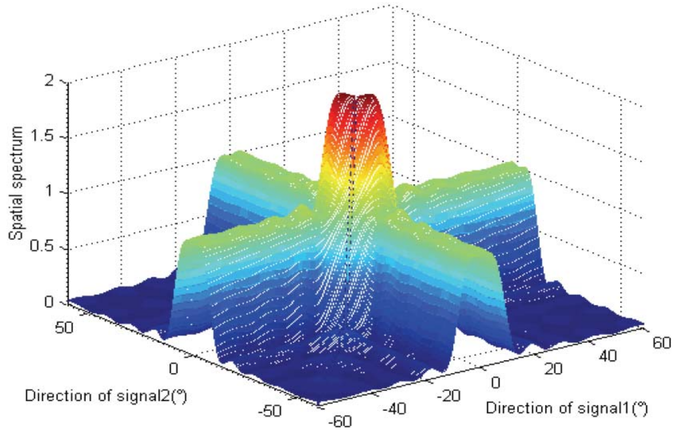

Fig. 3: The ACO-MD-MUSIC spatial spectrum

5. Conclusion

Algorithms in nature takes a more bottom-up, decentralised approach compared to the conventional spatial search algorithm for DOA estimation. It involves the method of specifying a set of learning rules and iteratively applying those rules for evolutionary development and result optimization. Clearly it yields significantly superior performance over traditional unilateral algorithmic approach by demonstrating better quantitative results of array signal processing parameters.

6. References

[1] Ahmed H. El Zooghby, Christos G. Christodoulou and Michael Georgiopoulos, "A Neural Network-Based Smart Antenna for Multiple Source Tracking," *IEEE Trans. on Antennas and Propagation, Vol. 48, No. 5, May* 2000

[2] Titus Lo, Henry Leung, and John Litva, "Radial Basis Function Neural Network for Direction-of-Arrivals Estimation," *IEEE Signal Processing Letters, Vol. 1, No. 2, February* 1994

[3] Ehtsham Elahi, Ahsan Naeem, Irfan Elahi, "A Genetic Algorithm based Solution to Three Dimensional Direction of Arrival (DOA) problem," *Proc. of the 12th IEEE International Multitopic Conference, December 23-24,2008*

[4] P. Karamalis, A. Marousis, A. Kanatas, P. Constantinou, "Direction of Arrival Estimation Using Genetic Algorithms," *IEEE VTS 53rd Vehicular Technology Conference, 2001. VTC 2001 Spring*

[5] Falih M. Mousa, "Particle Swarm Optimization Algorithm for Smart Antenna System," *Journal of Mobile Communication*, Year: 2011, Volume: 5, Issue: 1, Page No.: 6-10

[6] A. D. Hanumantharao, T. Panigrahi, U. K. Sahoo, G. Panda, B. Suresh, "Exact Maximum Likelihood Direction Of Arrival Estimation Using Bacteria Foraging Optimization," *International Conf. on Emerging Technologies -2011*

[7] Yiping Yang, Yunshan Hou and Xianxing Liu, "Two Dimensional DOA Estimation by Ant Colony Optimization," *IEEE International Conference on Intelligent System Design and Engineering Application,* 2010.

[8] Yameng Jiao, Jianguo Huang, Yunshan Hou, "Multidimensional MUSIC DOA Estimation Using Ant Colony Optimization Algorithm," *IEEE* 10*th International Conference on Signal Processing (ICSP),* 2010.

[9] A. H. El Zooghby, C. G. Christodoulou and M. Georgiopoulos, "Performance of Radial-Basis Function Networks for Direction of Arrival Estimation with Antenna Arrays," *IEEE Trans. on Antennas & Prop., Vol.* 45*, No.* 11.

[10] M. Li and Y. Lu, "Genetic Algorithm Based Maximum Likelihood DOA Estimation," *IEEE Radar Conf. 2002*

[11] T. Datta and I. S. Misra, "A Comparative Study of Optimization Techniques in Adaptive Antenna Array Processing: The Bacteria-Foraging Algorithm and Particle-Swarm Optimization," *IEEE Antennas and Propagation Magazine, Volume: 51 , Issue: 6, 2009*

[12] B. Errasti, D. Escot, D. Poyatos and I. Montiel, "Performance analysis of the Particle Swarm Optimization algorithm when applied to direction of arrival estimation," *IEEE International Conference on Electromagnetics in Advanced Applications, 2009*