IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE)
e-ISSN: 2278-1676,p-ISSN: 2320-3331, Volume 9, Issue 3 Ver. I (May – Jun. 2014), PP 91-97
www.iosrjournals.org

Pattern Synthesis for Linear Antenna Array Using Characteristics Evolution Optimization

Pranav Wani, Om Prakash Acharya
B.Tech Student, Department of Mechanical Engineering, IIT Roorkee, Roorkee, 247667 India
PhD Candidate, Department of Electronics Engineering, IIT Roorkee, Roorkee, 247667 India

Abstract: *Linear antenna array design is one of the most important electromagnetic optimization problems of current research interests. In this paper a new method of optimization, Characteristics Evolution Optimization, has been introduced. The method is based on the binary digit system. Since the algorithm works on binary digits and in a parallel fashion, the method can perform well in parallel processing environment. For the purpose of illustration, a 16- element array has been considered. The performance of the introduced technique has been compared with some of the existing techniques, IWO (Invasive Weed Optimization), DE (Differential Evolution), and PSO (Particle Swarm Optimization). Various variants of Invasive Weed Optimization have also been considered. It has been observed that the proposed method (Characteristics Evolution optimization) performs satisfactorily comparing with other existing methods. It has been observed that the introduced method, in certain instances, outperforms other optimization techniques.*

Keywords: *Characteristics Evolution Optimization, Linear Array, Pattern Synthesis*

I. Introduction

Antenna Arrays play an important role in detecting and processing signals arriving from different directions. The role of antenna array synthesis is to determine the physical layout of the array, and the amplitude and phase excitation that produces a radiation pattern that is closest to the desired radiation pattern. The shape of the desired pattern can vary widely depending upon the application. Some applications require a low sidelobe level, while other applications require a interference reduction using null control. However the global synthesis of antenna arrays that generate a desired radiation pattern are a highly non-linear optimization problem and hence analytical methods are not applicable anymore. For this purpose several optimization techniques have been developed that work with non-linear optimization problems. Many methods are bio-inspired. These methods have proven to be highly successful. Some of these are Genetic Algorithm, and Particle Swarm Optimization. In GA, a sample of possible solutions is assumed, then mutation, crossover, and selection are employed based on the concept of survival of fittest. In PSO each possible solution is represented as a particle in a swarm with a certain position and velocity vector. The position and velocity of each particle are updated by some fitness function.

In this paper a new method of optimization has been introduced. The paper has been arranged as mentioned below.

At the beginning of the paper, the linear array design synthesis problem is mentioned. In later section the new method has been explained. In next section a design problem has been solved using this method and also with PSO, IWO and some modifications of IWO, and DE (Differential Evolution Algorithm). Their results have been compared in the last section.

II. Formulation Of The Design Problem

In the synthesis, one begins by specifying the desired array pattern. Since the discussion here is restricted to linear arrays, the desired pattern must be a function of θ alone and not Φ, i.e. in the form of *a(θ)* or *F(θ).*[3] But the class of such functions is large. It includes sum patterns, difference patterns, etc.

2.1 Sum And Difference Pattern

Many applications of linear arrays involve the need to produce sum and difference patterns with the main beam of the sum pattern pointing at an angle θ, with the twin main beams of the difference pattern straddling θ, and with both patterns exhibiting a symmetrical sidelobe structure. Consider a array with 2N elements in it, equispaced by an amount *d*, the array factor can be written in the form

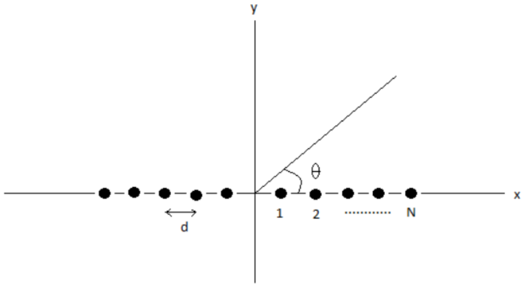


Fig 2.1: *Linear Antenna Array. The array elements are placed d distance apart from each other. 2N elements have been used. Distance between the elements or excitation of the elements can be controlled to get the desired array pattern.*

$$a_a(\theta) = \sum_{n=-N}^{-1} \frac{I_n}{I_1} e^{j[\frac{2n+1}{2}]kd(\cos\theta-\cos\theta_0)} + \sum_{n=1}^{N} \frac{I_n}{I_1} e^{j[\frac{2n-1}{2}]kd(\cos\theta-\cos\theta_0)}$$

Under the conditions, all the current amplitudes can be taken as pure real. For the sum pattern $I_n = I_{-n}$. Thus for the sum pattern the above eqn becomes

$$S(\theta) = 2\sum_{n=1}^{N} \frac{I_n}{I_1} \cos[(2n-1)(\frac{\Pi d}{\lambda})(\cos\Theta - cos\Theta_0)]$$

For the difference pattern, $I_n = -I_{-n}$. For this condition the array factor becomes

$$D(\Theta) = 2j\sum_{n=1}^{N} \frac{I_n}{I_1} \sin[(2n-1)(\frac{\Pi d}{\lambda})(\cos\theta - \cos\theta_0)]$$

An array with 2N+1 elements is not suitable for the creation of a difference pattern because of the presence of the central element. However it can be used to produce a sum pattern. The pattern is given by

$$S(\theta) = 1 + 2\sum_{n=1}^{N} \frac{I_n}{I_0} \cos[2n(\frac{\Pi d}{\lambda})(\cos\theta - \cos\theta_0)]$$

III. Description Of The Proposed Algorithm

The algorithm (Characteristics Evolution Optimization) operates on the binary representation of the numbers. In this algorithm the less evolved organisms try to adopt the characteristic of the highly evolved organisms. They try to adopt in them some of the significant characteristics of the evolved organisms. At the same time, organisms modify their own characteristics. This adoption and modification leads to continuous evolution. In this process of evolution several diverse groups are formed with significant differences among them. Even in humans when people try to follow leaders, they try to adopt some of the most significant characteristics of their leaders. In this process several groups are formed with significant difference in them in terms of thinking and values. These groups then evolve independently for some time. After some time, when one group is seen to succeed much more than the other groups, the remaining groups start merging with the successful group. Then the merged group evolves to obtain higher success. The proposed algorithm (CEO) tries to adopt this procedure to obtain optimum solution. The step wise explanation of the algorithm has been given below.

3.1 Initialization

The problem in hand is of linear antenna array pattern synthesis. A sum pattern with 2N elements has been considered. The number of variables is thus N. To initialize, a population size NP is considered, with each particle being initialized in a N-dimensional space. The particles are initialized with numbers ranging from 0 to R, where R is the predefined range.

Each particle contains N numbers. These are then converted into their binary form. The binary representation of each number in predefined bits will be called parts from hence forward. The bit length to represent the numbers can be chosen by the user. Larger bit length tends to provide more accuracy. Decimal numbers can be represented in binary form by shifting the decimal point to the right to appropriate steps so that the number on the left of the decimal point can be represented in the allocated bit length.

For example to represent 2.765 in binary system using five bits would become 11011. Once the particles are initialized, their fitness is calculated according to the optimization function at hand. The particles are then arranged in accordance with their fitness value.

3.2 Segregation into Groups

The entire population is equally segregated into groups. For the purpose of explanation, it is considered that there are four groups. Thus the number of particles in each group would be NP/4. These particles are then arranged according to their fitness values within their respective groups.

3.3 Adoption of Characteristics

In every group, the group members try to adopt the characteristics of their corresponding leader. The leader is the one with the best fitness value. In binary representation, each bit is a characteristic. The importance of the characteristics increases going left and the importance of the characteristics decreases going right.

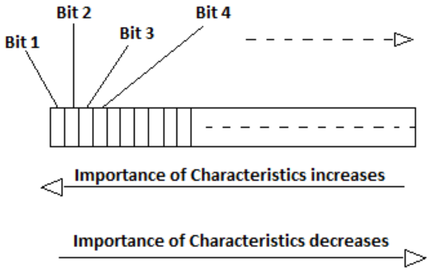

Bit 1
Bit 2
Bit 3
Bit 4
Importance of Characteristics increases
Importance of Characteristics decreases

Fig 3.1: ***A particle contains N parts (N being the dimension). The figure shows a particular part. Each part is composed of several bits or characteristics. Their importance increases as we move left.***

Every bit is a characteristic. The left bits are important characteristics. The length of bits is represented by bit_length. Every particle has N parts, and each part has bit_length number of characteristics.

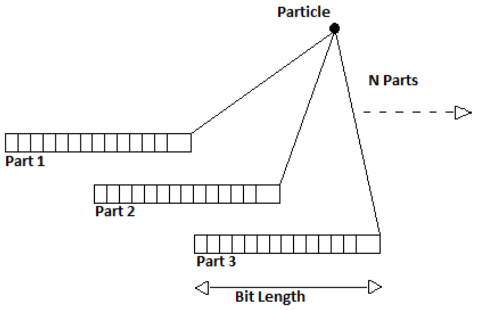


Fig 3.2: ***Figure shows the composition of a particle. Every particle has N parts, each part has bit_length characteristics. Characteristics are allowed to take either 1 or 0.***

During adoption, every characteristic in any part is given a particular adoption probability number. This number tells the probability of that particular characteristic to get adopted. During adoption, in a group, every particle adopts the characteristics of the best particle of that group. The adoption of characteristics takes place according to the assigned adoption probability number. Thus there can be bit_length number of adoption probability numbers.

For example, if the third characteristic of the second part of the best particle in a particular group is 1, then the chance of this 1 getting transmitted to the third characteristic of second part of some other particle of that same group is given by third adoption probability number.

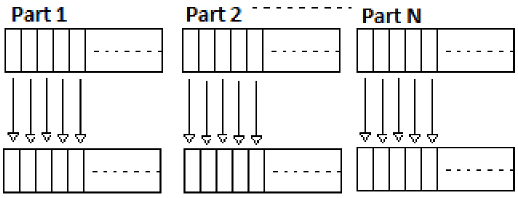


Fig 3.3: ***Figure shows the method of adoption. Every characteristic in every part tries to adopt the characteristics from the corresponding characteristic position and part of other particle.***

The adoption probability number is given in decreasing order from left to right i.e. from most important characteristic to the lest characteristic. The more important characteristics are given higher adoption probability number to make the algorithm converge. This way the other particles move closer to the best particle.

The less important characteristic are given less adoption number since we do not want other particles to rapidly come at exactly the same position as that of the best particle in the N-dimensional space. It is required that other particles search wider space. Thus as the particles try to imitate the main characteristics of the best particle of that group, they have the freedom to span the nearby space using the less important characteristics.

3.4 Evolution of Characteristics

As the particles try to adopt the characteristics from the best particle, they themselves try to evolve and search the entire space by changing their own characteristics. Every characteristic is given a evolution probability number. This number tells the probability of the characteristics to get altered, i.e. from 1 to 0 and from 0 to 1.

The evolution probability number is given in an increasing order from left to right, i.e. from the most important characteristics to the least important characteristics. This is done to avoid extreme deviation from their position while preserving their rights to explore the region around them.

Thus every particle evolves itself according to the evolution probability number.

3.5 Competitive Selection

For a particular group, there are now three sets of particle position that have been formed. One is the original, the other is the one formed after particles adopt the characteristics of the best particle of the group, and the third is after the particles evolve their own characteristics.

Thus for every particle, there are three positions available in N-dimensional space. To competitively select, for every particle, the best position out of three is selected.

There was another was of competitive selection. This was to arrange all the particle positions of a particular group in accordance with their fitness value, and then to select best 1/3rd positions from them. But this method faces the problem of pre-mature convergence. This arises because all the particles closer to the current best position would be given preference.

3.6 Merging of the Groups

For some time all the different groups continue to evolve independently. For the purpose of merging a point of confidence is defined. The best fitness values of all the groups are recorded. Whenever any group crosses the point of confidence, all the groups are merged together into a single group. This directs all the resources to search the space around the position of the leading group. All the particles search the space around them with all the previously mentioned rules.

IV. Exprimental Results

Here the proposed algorithm (CEO) has been used for linear antenna design. The results have been compared with that of Differential Evolution algorithm (DE), Invasive Weed Optimization (IWO), variants of IWO, and Particle Swarm Optimization (PSO). The parametric setup used for the proposed algorithm has been mentioned below.

Bit_length has been taken 40 and 50. Initial range of spread of the particles is taken ranging from 0 to 10. Number of groups is taken to be

4. Each group has been assigned 50 particles. The population size after recombination of the group is taken to be 75. The point of confidence is taken at 5. Thus the four groups merge together when the error comes down below 5. The adoption probability number of the characteristics is taken to vary from 40% for the leftmost i.e. most important characteristics to 20% for the rightmost characteristics. The evolution probability number of the characteristics has been taken to be, starting from leftmost bit 1%, 2%, 3%, 4%, 10% for the next 4 bits or characteristic, 20% for the next 6 characteristics, and 40% for the remaining bits or characteristics. For the purpose of competitive selection, all the particle positions of a particular group have been arranged in order of their fitness value, and best NP have been selected.

The problem statement is to design a linear array with 16 elements. The maximum sidelobe level is required to be at -30dB. The function used to determine error value is **abs(max_SL-desired_max_SL)**, where abs is absolute value, desired_max_SL is -30dB. Angle scanned for sidelobe ranges from 0 to 77 degrees and from 103 to 180 degrees. Sum pattern is formed using the antenna.

The parametric setup used for PSO, DE, IWO [9] and it's variants is given below. For the Differential Evolution algorithm, the crossover constant is taken to be 0.5, and the mutation factor is taken to be 0.2. Number of population is taken to be 400. For Particle Swarm Optimization (PSO), w_max and w_min are taken to be 0.9 and 0.4 respectively, and Vmax=0.05*pi. For the Invasive Weed Optimization (IWO), the number of agents is taken to be 10 times the dimension. S_initial=1 and S_final=0.00000001. n=3. Maximum number of seeds is taken to be 5 and maximum number of population is 20 times the dimension.

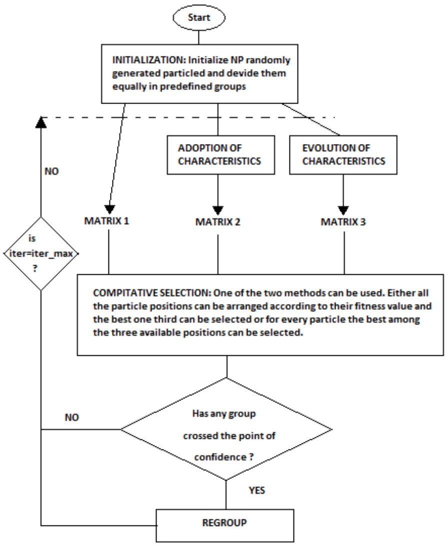

Start
INITIALIZATION: Initialize NP randomly generated particled and devide them equally in predefined groups
ADOPTION OF CHARACTERISTICS
EVOLUTION OF CHARACTERISTICS
NO
MATRIX 1
MATRIX 2
MATRIX 3
is iter=iter_max ?
COMPITATIVE SELECTION: One of the two methods can be used. Either all the particle positions can be arranged according to their fitness value and the best one third can be selected or for every particle the best among the three available positions can be selected.
NO
Has any group crossed the point of confidence ?
YES
REGROUP

Fig 3.4: ***Flow chart of Characteristics Evolution Optimization***

The variants of the IWO used in this example are briefly described below. Modified IWO [1] uses a |cos(iter)| term in calculating the standard deviation, to allow for the fast convergence of the weeds present in location of the global optimum solution without having to wait for the standard deviation to decrease with iterations. MIWO [2] uses a modified formula for calculating standard deviation, which is based not only on the iteration but also on the fitness value of corresponding weed. So the standard deviation is different for every weed. This gives opportunity to the far away weeds to get closer to the global optimum solution, and prevent the close weeds to get trapped. DIWO [2] merges the MIWO with the differential evolution. It adopts the concept of mutation and crossover from DE algorithm and applies it to MIWO.[2]

For the purpose of plotting the error plots and finding beam widths, every algorithm has been played 20 times. The plots shown below are the average of the 20 values. The lines in the plots have been artificially or externally smoothened for better presentation.

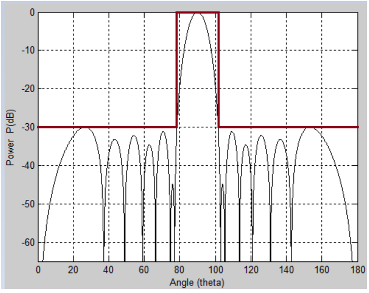


Fig 4.1: ***Beampattern of 16-elements array with -30dB sidelobe level***

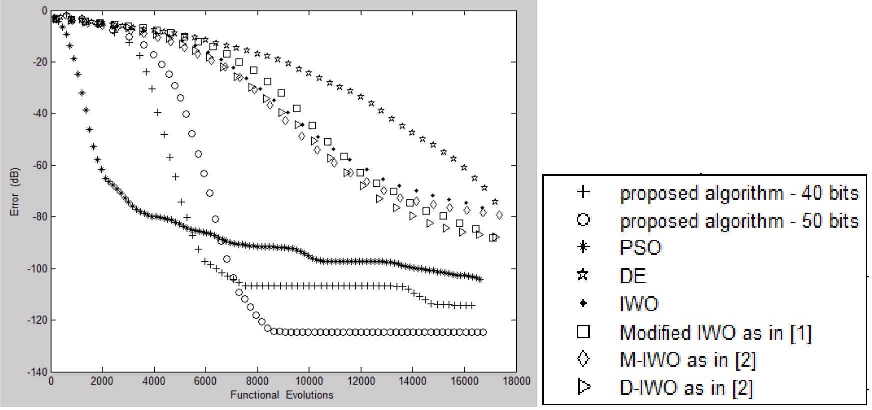
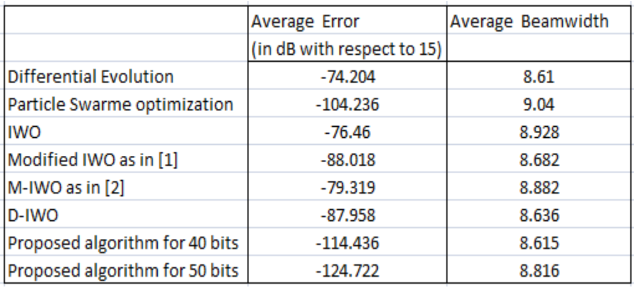


Fig 4.2: ***Error plots for pattern synthesis of 16-element array using different algorithms. Error has been expressed in dB with respect to 15.***



	Average Error (in dB with respect to 15)	Average Beamwidth
Differential Evolution	-74.204	8.61
Particle Swarme optimization	-104.236	9.04
IWO	-76.46	8.928
Modified IWO as in [1]	-88.018	8.682
M-IWO as in [2]	-79.319	8.882
D-IWO	-87.958	8.636
Proposed algorithm for 40 bits	-114.436	8.615
Proposed algorithm for 50 bits	-124.722	8.816

Fig 4.3: ***Table listing the average error and beamwidth obtained by each algorithm after approximately 17000 functional evaluations.***

The performance of the algorithm vastly depends upon the point of confidence. It is the point which decides the condition of merging of groups. Larger the point, sooner the groups will merge, but this may cause the particles to get stuck in the local optimum solution. The error plots for different values of point of confidence (c_point) are given below.

The performance also depends on the available bit_length of a part. The larger the bit_length better is the performance, but simultaneously it takes more time to converge, since training a longer sequence of bits is difficult. For good performance, initially the bit_length should be small to allow faster convergence, and at latter stages the bit_length should be large.

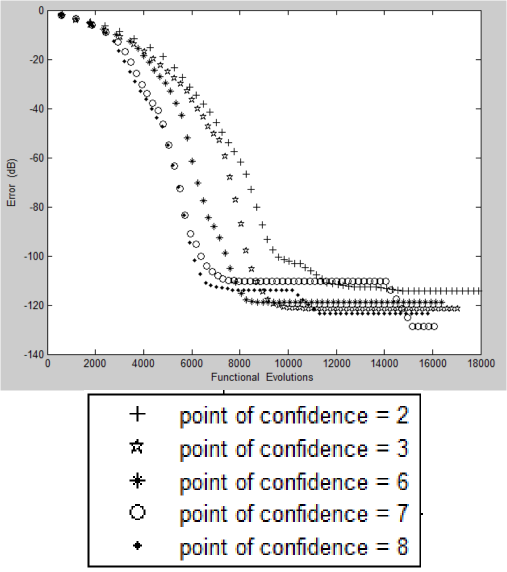


Fig 4.4: ***Error plots for pattern synthesis of 16-element array using the proposed algorithm for different points of confidence.***

V. Conclusion

In this paper a novel method of optimization was proposed (Characteristics Evolution Optimization). The method works on the binary representation of numbers. Since the method works in a parallel fashion, and only with 0 and 1, the method can perform well in parallel processing environment. As an example a 16-element linear array was used. The results show that the proposed algorithms, under certain instances, works well with respect to other algorithms. The performance of the algorithm was compared to PSO, DE, IWO and few variants of IWO.

Further modifications and research are required to make this process even better. With it's present form, the method finds difficulty under certain instances. It is expected that with further research the method can perform very well in most of the optimization problems.

References

[1]. Aniruddha Basak, Siddharth Pal, Swagatam Das, Ajith Abraham, Vaclav Snasel, "*A modified invasive weed optimization algorithm for the time-modulated linear antenna array synthesis*", IEEE 2010

[2]. Siddharth Pal, Aniruddha Basak, Swagatam Das, Ajith Abraham, Ivan Zelinka, "*Concentric Circular Antenna Array Synthesis Using a Differential Invasive Weed Optimization Algorithm*", IEEE 2010 International conference on soft computing and pattern recognition.

[3]. Constantine A. Balanis and Panayiotis I. Ioannides, *Introduction to Smart Antennas*, 1st edition: Morgan & Claypool, 2007

[4]. Siddharth Pal, Aniruddha Basak, Swagatam Das, Ajith Abraham, "*Linear antenna array synthesis using invasive weed optimization algorithm*", 2009 International conference on soft computing and pattern recognition.

[5]. Herve Lebret, Stephen Boyd, "*Antenna array pattern synthesis via convex optimization*", IEEE transactions on signal processing, 1997

[6]. Ritwik Giri, Ariba Chowdhury, Arnob Ghosh, Swagatam Das, Ajith Abraham, Vaclav Snasel, "*A modified invasive weed optimization algorithm for training of feed-forward neural networks*", IEEE 2010

[7]. Mohamadreza Ahmadi, Hamed Mojallali, "*Chaotic invasive weed optimization algorithm with applications to parameter estimation of chaotic systems*", 2012 Elsevier

[8]. Shaya Karimkashi, Ahmed A. Kishk, "*Invasive weed optimization and it's features in electromagnetic*", IEEE transactions in antennas and propagation, 2010

[9]. Iqroop Kaur, Sanjeev Kumar, "*Linear Antenna Array Optimization Techniques*", International journal of engineering trends and technologies, 2011