Pattern Synthesis of Sparse Phased Array Antenna

Using Genetic Algorithms

Rongcang Han

Department of Physics, Linyi Normal University, Linyi 276005, China

E-mail: hanrongcang@lytu.edu.cn

Abstract

Sparsely sampled irregular arrays and random arrays have been used or proposed in several fields such as radar, sonar, and ultrasound imaging. One method of pattern synthesis for sparse phased array antenna using genetic algorithms is introduced. We start with an introduction to genetic algorithms and then consider the problem of finding the best amplitude layout of elements in sparse arrays. The optimization criteria are then reviewed: creation of beam patterns with low main lobe width and low side lobes.

Keywords: Genetic Algorithms, Sparse Array Antenna, Phased Array, Pattern Synthesis

1. Introduction

Making an array sparse means turning off some elements in a uniformly spaced or periodic array to create a desired amplitude density across the array aperture. An element connected to the feed network is "on", and an element connected to a matched or dummy load is "off". Making an array sparse to produce low side lobes is much simpler than the more general problem of non-uniform spacing the elements. Non-uniform spacing has an infinite number of possibilities for placement of the elements. Making an array sparse has 2^Q possible combinations, where Q is the number of array elements. If the array is symmetric, then the number of possibilities is substantially smaller. Making an array sparse may also be thought of as quantized amplitude taper where the amplitude at each element is represented by one bit.

Traditional optimization techniques searching for the best solutions usually use gradient methods or random searching. Gradient methods are efficient, but have disadvantages of getting stuck in local minima, requiring gradients calculations, working only on continuous parameters. Random-search methods don't require gradient calculations, but tend to be slow, and susceptible to getting stuck in local minima. Genetic algorithm optimizer is robust global search method. Its research is based on probability, having advantage of avoiding getting stuck in local minima(Haup,1994,p993).

2. Genetic Algorithms

This section is a quick overview of genetic algorithms, much more detail on genetic algorithms is found in the book of Holland (Holland, 1992). Gene is the basic building blocks of genetic algorithms. A gene is binary encoding of a parameter. In computer algorithm, a chromosome is an array of genes, a number of chromosomes make up one population. Each chromosome has an associated fitness function, assigning a relative merit to that chromosome. The algorithm begins with a large list of random chromosomes. Fitness functions are evaluated for each chromosome. The chromosomes are ranked from the most-fit to the least-fit, according to their respective fitness functions. Unacceptable chromosomes are discarded, leaving a superior species-subset of an original list, which is the process of selection. Genes that survive become parents, by crossing over some of their genetic material to produce two new offspring. The parents reproduce enough to offset the discarded chromosomes. Thus, the total number of chromosomes remains constant after every iteration. Mutations cause small random changes in a chromosome. Fitness functions are evaluated for the offspring and mutated chromosome, and the process is repeated. The algorithm stops after a set number of iterations, or when an acceptable solution is obtained. Figure1 is a flow chart of genetic algorithms. (Diógenes Marcano and Filinto Durán, 2000, p13).

A sparse array has discrete parameters. One bit represents the element state as "on" = 1 or "off "= 0. For example, an eight element array may be represented by 10110101, where elements 2 and 5 are turned "off." Assuming the linear array is symmetric about its center allows the 2N element array to be represented by a gene with N bits. Our six-element

array example can then be represented by the gene 101. The fitness associated with this gene is the maximum relative side lobe level (rsll) of its associated far-field pattern. The function in this paper is the relative far-field pattern of an array of point sources. Its output to be minimized is the maximum rsll. The parameters affecting the output are whether an antenna element is on or off.

3. Method

Suppose we consider an array of antenna elements uniformly spaced in a straight line along the z axis. The far-field radiation pattern produced by such an array may be expressed as

$$E(\theta,\varphi) = \sum_{n=0}^{N-1} I_n EP_n(\theta,\varphi) e^{j(nkd\cos\theta + \beta_n)} \tag{1}$$

where I_n are the current amplitudes of element excitations, $\beta_n = -nkd\cos\theta_0$, θ_0 is the angle that the main beam of the antenna directed to. $EP_n(\theta,\varphi)$ are the individual array element patterns, k=$2\pi/\lambda$ is the free-space wave number, d is the separation distance between elements.

Now the main process of antenna synthesis in genetic algorithms is given below.

A. Establish decision variable and constraint condition, then encoding them.

$$I_n = \begin{cases} 1 & on \\ 0 & off \end{cases} \tag{2}$$

B. Create optimization pattern:

$$\min E_{mt} = \left[\frac{1}{Q} \sum_{i=1}^{Q} |e_i|^2 \right]^{\frac{1}{2}} \tag{3}$$

Where

$$e_i = \frac{T_i - F_i}{T_i}, i = 1, 2, \cdots, Q \tag{4}$$

T_i is the level of the desired radiation pattern at the point Q, and F_i is the level of the pattern generated by genetic algorithms.

C. Define fitness function:

$$F_m = \frac{1}{1 + E_{mt}^{\alpha}} \tag{5}$$

where $\alpha \in (0,1]$. Figure2 shows the behavior of the fitness as a function of the error. We can see that the algorithm favors individuals with a high fitness. (Daniel, 2005, p358).

4. Numerical Illustration

Here, genetic algorithms is used to minimize the maximum side lobe level for an array with aperture L= 40λ (total number of elements is 81) with 35 active elements lying on a grid with $\lambda/2$ element distance. In this problem, our goal is to get minimum side lobe level. A best layout of elements is given below by genetic algorithms. The symmetric amplitude of the array can be used to making scanning beam. The pattern of this array is given in figure3. The pattern when the array scanning is shown in figure4. The layout of the elements is shown by a string of bits below.

5. Conclusions

This paper introduced the use of genetic algorithms for sparse linear arrays to obtain scanning beam. The beauty of the genetic algorithm is that it can optimize a large number of discrete parameters. The genetic algorithm intelligently searches for the best layout of element amplitude intelligently searches for the best layout of element amplitude that produces low side lobes.

Many additional extensions are possible, including thinning circular arrays, planar arrays with directional elements, scanning planar arrays, etc.

References

Daniel W. Boeringer and Douglas H. Werner. (2005). A Simultaneous Parameter Adaptation Scheme for Genetic Algorithms with Application to Phased Array Synthesis. *IEEE Transactions on antennas and Propagation*, Vol. 53, No. 1, 356-371.

Diógenes Marcano and Filinto Durán. (2000). Synthesis of Antenna Array Using Genetic Algorithms. *IEEE, Antennas*

and Propagation Magazine*, Vol. 42, No. 3, 12~20.

Holland J.H. (1992). *Adaptation in Nature and Artificial Systems. MIT Press*, 1992.

Randy L, Haupt. (1994). Arrays Using Genetic Algorithm. *IEEE Transactions on Antennas and Propagation*, Vol. 42, No. 7, 993~999.
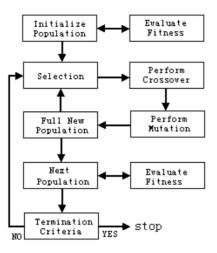
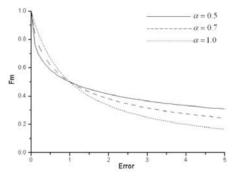

Figure 1. A flow chart of genetic algorithms

Figure 2. A plot of fitness function versus error

1 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 1 1 0 1 1 0 1 1 1 1 1 1 1 1, 1,

1 1 1 1 1 1 1 1 0 1 1 0 1 1 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 1
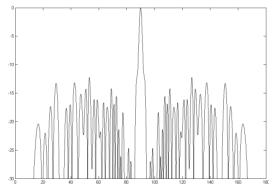
Sidelobe Peak: -12.4 [dB]

Figure 3. Result of non-scanning pattern of genetic algorithms
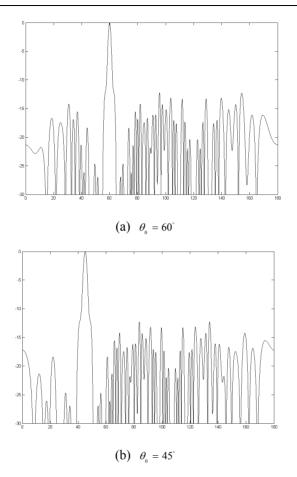
(a) $\theta_0 = 60°$

(b) $\theta_0 = 45°$

Figure 4. Radiation patterns of the phased array with different scanning angles