Multibeam Antennas Array Pattern Synthesis using Hybrid Particle Swarm Optimiser with Breeding and Subpopulations Algorithm

Hichem CHAKER
Telecom laboratory
BP 285, 13400 Ghazaouet
Tlemcen ALGERIA

S.M. MERIAH
Telecom laboratory
BP 285, 13400 Ghazaouet
Tlemcen ALGERIA

F.T. BENDIMERAD
Telecom laboratory
BP 285, 13400 Ghazaouet
Tlemcen ALGERIA

ABSTRACT

In this paper a new effective optimization algorithm called hybrid particle swarm optimizer with breeding and subpopulation is presented. This algorithm is essentially, as PSO and GA, a population-based heuristic search technique, now in use for the optimization of electromagnetic structures, modeled on the concepts of natural selection and evolution (GA) but also based on cultural and social rules derived from the analysis of the swarm intelligence and from the interaction among particles (PSO). The optimized design of multibeam antennas arrays is reported with numerical results.

General Terms

Hybrid model, global approach, genetic algorithm, particle swarm optimizer, multibeam pattern, antenna arrays.

Keywords

Hybrid Particle Swarm Optimizer, Synthesis, Multibeam Linear Antenna Arrays.

1. INTRODUCTION

In recent years several global optimization algorithms have been developed for the optimization of every kind of electromagnetic problems. Global search methods present two competing goals, exploration and exploitation: exploration is important to ensure that every part of the solution domain is searched enough to provide a reliable estimate of the global optimum; exploitation, instead, is also important to concentrate the search effort around the best solutions found so far by searching their neighbourhoods to reach better solutions [1]. Advantages of evolutionary computation are the capability to find a global optimum, without being trapped in local optima, and the possibility to face nonlinear and discontinuous problems, with a great number of variables. On the other hand, these algorithms have strong stochastic bases, thus they require a great number of iterations to get significant results. To solve the antenna array pattern synthesis problems, among a number of optimization procedures, the artificial intelligence techniques such as genetic, simulated annealing and tabu search algorithms owing to their simplicity, flexibility and accuracy have received much attention. Genetic algorithm (GA) is a search technique based on an abstract model of Darwinian evolution. Simulated annealing (SA) technique is essentially a local search, in which a move to an inferior solution is allowed with a probability, according to some Boltzmann-type distribution, that decreases as the process progresses. Tabu search (TS) algorithm has been developed to be an effective and efficient scheme for combinatorial optimization that combines a hill-climbing search strategy based on a set of elementary moves and a heuristics to avoid to stops at sub-optimal points and the occurrence of cycles. Recently, particle swarm optimization algorithm (PSO) is proposed for solving global numerical optimization problem. The search techniques mentioned above are the probabilistic search techniques that are simple and easily be implemented without any gradient calculation. This study uses an electromagnetic optimization technique, hybrid particle swarm optimiser with breeding and subpopulation [2].

In this article, an efficient method for the pattern synthesis of the linear multibeam antenna arrays is presented. Multibeam pattern is achieved by finding both magnitudes and phases excitation of each array element. The proposed method is based on the hybrid particle swarm optimiser with breeding and subpopulations algorithm, and the linear antenna array synthesis was modelled as a mono-objective optimization problem. To verify the validity of the technique, several illustrative examples are simulated, and multibeam patterns are demonstrated.

2. PROBLEM FORMULATION

An array can form multiple narrow beams towards different directions. For example, suppose it is desired to form two or three beams towards steering angles. The design of a linear array antenna is based on finding both magnitudes and phases excitation that can generate the desired patterns. We consider a linear array of 2N isotropic antenna elements, which are assumed, uncoupled, symmetrically and equally spaced with half wavelength. Its array pattern can be described as follows [3]:

$$F(\theta) = 2\sum_{k=1}^{N} a_k \cos\left(\frac{2\pi}{\lambda} d_k \sin(\theta) + \delta_k\right) \tag{1}$$

N = element number;

λ = wavelength;

δ_k = phases of the elements (-180°≤θ≤180°);

a_k = amplitude of the elements;

d_k = distance between position of i^{th} element and the array center;

θ = scanning angle.

In order to generate a beam pattern fulfilling some constraints, SLL lower than a fixed threshold or reproducing a desired shape, an array configuration must be synthesized. First of all, it is necessary to define the objective function that measures the difference between desired and synthesized beam pattern. Let us define a function called fitness function as follows:

$$Fitness = Max - \int_0^{\pi} \left| F_d(\theta) - F(\theta) \right| d\theta \qquad (2)$$

3. HYBRID PARTICLE SWARM OPTIMIZER WITH BREEDING AND SUBPOPULATIONS

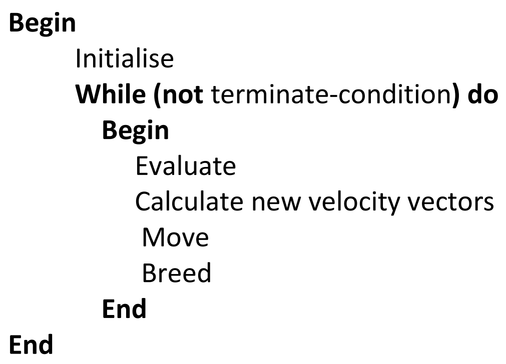
Both Eberhart and Angeline conclude that hybrid models of the standard GA and the PSO could lead to further advances. We present such a hybrid model. The model incorporates one major aspect of the standard GA into the PSO, the reproduction. In the following work, we will refer to the used reproduction and recombination of genes only as “breeding”. Breeding is one of the core elements that make the standard GA a powerful algorithm. Hence our hypothesis was that a PSO hybrid with breeding has the potential to reach a better solution than the standard PSO. In addition to breeding we introduce a hybrid with both breeding and subpopulations. Subpopulations have previously been introduced to standard GA models mainly to prevent premature convergence to suboptimal points [4]. Our motivation for this extension was that the PSO models, including the hybrid PSO with breeding, also reach suboptimal solutions. Breeding between particles in different subpopulations was also added as an interaction mechanism between subpopulations. The traditional PSO model, described by [5], consists of a number of particles moving around in the search space, each representing a possible solution to a numerical problem. Each particle has a position vector $Xi=(x_{i1},\ldots,x_{id},\ldots,x_{iD})$, a velocity vector $Vi=(v_{i1},\ldots,v_{id},\ldots,v_{iD})$, the position $Pi=(p_{i1},\ldots,p_{id},\ldots,p_{iD})$ and fitness of the best point encountered by the particle, and the index (g) of the best particle in the swarm. At each iteration the velocity of each particle is updated according to their best encountered position and the best position encountered by any particle, in the following way:

$$v_{id} = w \times v_{id} + c_1 \times rand\ () \times (p_{id} - x_{id}) + c_2 \times rand\ () \times (p_{gd} - x_{id}) \qquad (3)$$

w is the inertia weight described in [6]-[7] and P_{gd} is the best position known for all particles. C_1 and C_2 are random values different for each particle and for each dimension. If the velocity is higher than a certain limit, called V_{max}, this limit will be used as the new velocity for this particle in this dimension, thus keeping the particles within the search space. The position of each particle is updated at each iteration. This is done by adding the velocity vector to the position vector;

$$x_{id} = x_{id} + v_{id} \qquad (4)$$

The particles have no neighborhood restrictions, meaning that each particle can affect all other particles. This neighborhood is of type star (fully connected network), which has been shown to be a good neighborhood type in [8]. Figure 1 shows the structure illustration of the hybrid model.



```
Begin
    Initialise
    While (not terminate-condition) do
      Begin
          Evaluate
          Calculate new velocity vectors
          Move
          Breed
      End
End
```

Fig 1: The structure of the hybrid model

The breeding is done by first determining which of the particles that should breed. This is done by iterating through all the particles and with probability p_b (breeding probability = 0.6), mark a given particle for breeding. Note that the fitness is not used when selecting particles for breeding. From the pool of marked particles we now select two random particles for breeding. This is done until the pool of marked particles is empty. The parent particles are replaced by their offspring particles, thereby keeping the population size fixed. The position of the offspring is found for each dimension by arithmetic crossover on the position of the parents:

$$Child_1(x_i) = p_i * parent_1(x_i) + (1 - p_i) * parent_2(x_i) \qquad (5)$$

$$Child_2(x_i) = p_i * parent_2(x_i) + (1 - p_i) * parent_1(x_i) \qquad (6)$$

Where p_i is a uniformly distributed random value between 0 and 1. The velocity vectors of the offspring are calculated as the sum of the velocity vectors of the parents normalized to the original length of each parent velocity vector.

$$Child_1(\vec{v}) = \frac{parent_1(\vec{v}) + parent_2(\vec{v})}{\left| parent_1(\vec{v}) + parent_2(\vec{v}) \right|} \left| parent_1 \right| \qquad (7)$$

$$Child_2(\vec{v}) = \frac{parent_1(\vec{v}) + parent_2(\vec{v})}{\left| parent_1(\vec{v}) + parent_2(\vec{v}) \right|} \left| parent_2 \right| \qquad (8)$$

The arithmetic crossover of positions in the search space is one of the most commonly used crossover methods with standard real valued GAs, placing the offspring within the hypercube spanned by the parent particles. The main motivation behind the crossover is that offspring particles benefit from both parents. In theory this allows good examination of the search space between particles. Having two particles on different suboptimal peaks breed could result in an escape from a local optimum, and thus aid in achieving a better one. We used the same idea for the crossover of the velocity vector. Adding the velocity vectors of the parents results in the velocity vector of the offspring. Thus each parent affects the direction of each offspring velocity vector equally.

In order to control that the offspring velocity was not getting too fast or too slow, the offspring velocity vector is normalized to the length of the velocity vector of one of the parent particles. The starting position of a new offspring particle is used as the initial value for this particle's best found optimum ($\vec{p}_i$). The motivation for introducing subpopulations is to restrict the gene flow (keeping the diversity) and thereby attempt to evade suboptimal convergence. The subpopulation hybrid PSO model is an extension of the just described breeding hybrid PSO model. In this new model the particles are divided into a number of subpopulations. The purpose of the subpopulations is that each subpopulation has its own unique best known optimum. The velocity vector of a particle is updated as before except that the best known position ($\vec{p}_g$ in the formula) now refers to the best known position within the subpopulation that the particle belongs to. In terms of the neighborhood topology suggested by Kennedy in [8], each subpopulation has its own star neighborhood. The only interaction between subpopulations is if parents from different subpopulations breed. Breeding is now possible both within a subpopulation but also between different subpopulations. An extra parameter called probability of same subpopulation breeding p_{sb} determines whether a given particle selected for breeding is to breed within the same subpopulation (probability $p_{sb} = 0.6$), or with a particle from another subpopulation (probability 1-p_{sb}). Replacing each parent with an offspring particle ensures a constant subpopulation size.

4. RESULTS

In order to illustrate the capabilities of the hybrid evolutionary algorithm for solving the array configuration for desired pattern synthesis by varying the amplitude and phase of the elements feed, we introduce the case of an array with 10 isotropic elements with λ/2 spacing, which is supposed to generate two beams steered towards the two angles 70° and 130° , figure 2 shows the output pattern, the relative amplitudes of the two beams were equal to unity, after 940 iterations maximum side lobes level of -29.13 dB was achieved amplitude and phase distributions in degree are shown table 1.

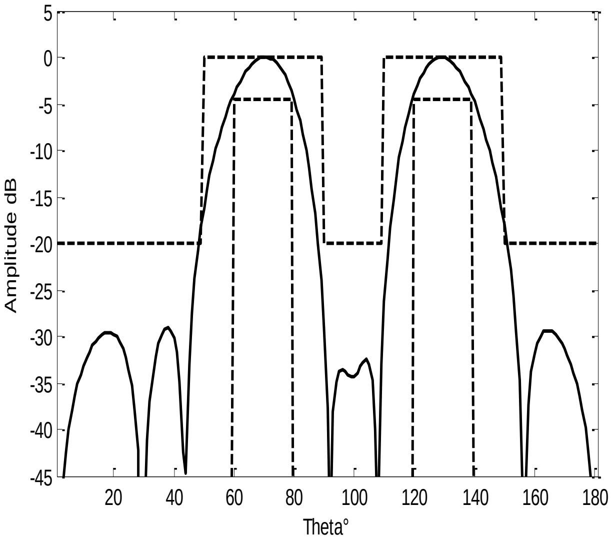


Fig 2: Multi-beam arrays with maximum sidelobes level equal to -29.13dB.

Table 1. Amplitude and phase distributions

Element N°	Magnitude	Phase (Degree)
1	0.1034	26.1784
2	0.3094	111.0851
3	0.5454	-64.5265
4	0.7122	-31.5929
5	0.8548	170.157
6	0.8548	-170.157
7	0.7122	31.5929
8	0.5454	64.5265
9	0.3094	-111.085
10	0.1034	- 26.1784

In the second example, we consider an array of 20 isotropic elements spaced 0.5 λ apart in order to generate three beams towards the steering angles 60°, 90°, 110° with amplitude-phase synthesis. Because of symmetry, here only ten phases and ten amplitudes are to be optimized.

Acceptable side lobe level should be equal to or less than the desired value -20dB. Figure 3 shows normalized absolute power pattern in dB the maximum side lobes level reach -20.84 dB, there is a very good agreement between desired and obtained results. The optimized excitation magnitudes and phases elements are presented in the table 2. For design specifications of amplitude-phase synthesis, the hybrid algorithm is run for 960 generations.

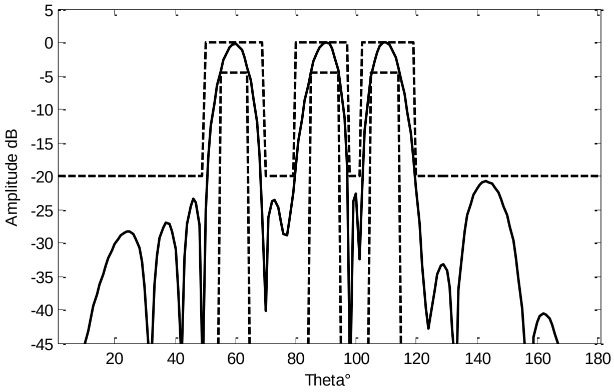


Fig 3: Multi-beam arrays with maximum sidelobes level equal to - 20.84dB.

In order to evaluate the performance of the proposed algorithm, we compare the numerical results calculated by the hybrid model, and the neural network [9]. We show the comparison of the far-field patterns among the hybrid model results, and the neural network simulated results in [9]. The hybrid model side lobe level is -20 dB, these results remain comparable to the neural network, and an improvement in the side lobe level is obtained.

With the same array as the last section, and the same type of synthesis, we present synthesis results of multibeam array as indicated in the figures 4, 5 and 6. Figure 4 shows normalized absolute power pattern in dB for multibeam array by amplitude-phase synthesis.

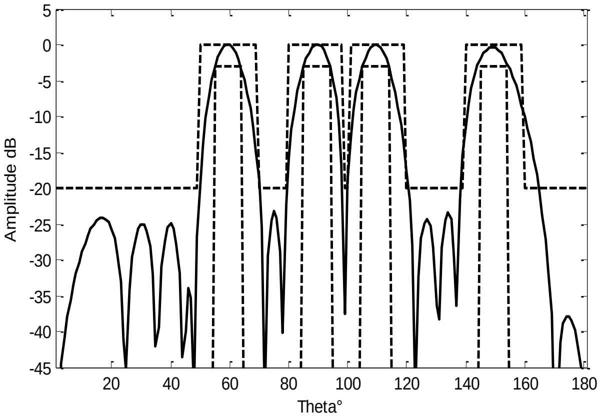


Fig 4: Multi-beam arrays with maximum sidelobes level equal to - 23.21dB.

For design specifications of amplitude-phase synthesis, the hybrid algorithm is run for 738 generations. Side lobes level obtained for desired pattern is -23.21dB. Simulated results are shown in table 2.

After 1200 iterations, an optimum multibeam pattern is obtained and plotted in figure 5, the optimized excitation magnitudes and phases of the elements from number one to twenty are shown in table 2. The optimized result indicates that a 20 elements symmetrically linear array is capable to realize the design goal with less number of antenna elements.

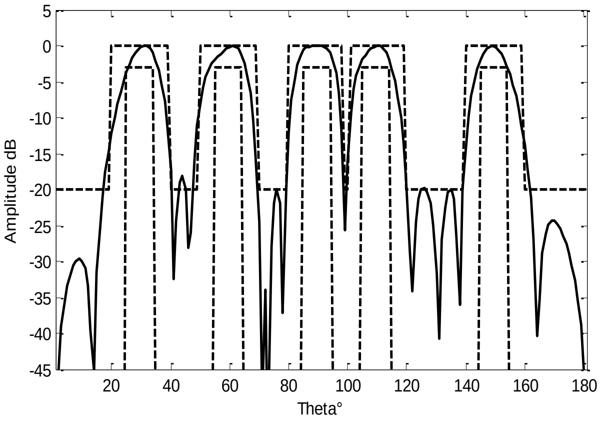


Fig 5: Multi-beam arrays with maximum sidelobes level equal to - 18 dB.

After 1000 iterations, the fitness value reached to it maximum, and the optimization process ended due to meeting the design goal. Figure 6 shows the normalized absolute power pattern for the array with six beams. The element excitation required to achieve this desired pattern is presented in table 2.

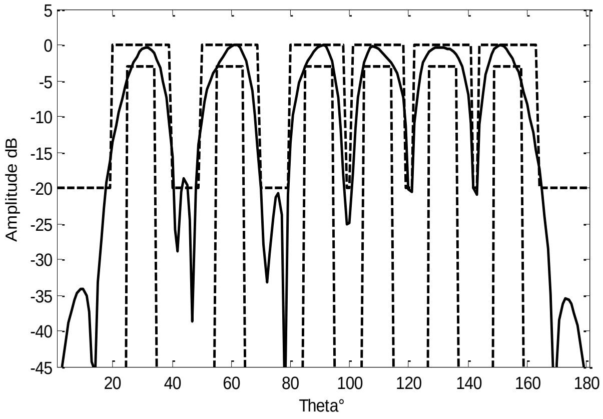


Fig 6: Multi-beam arrays with maximum sidelobes level equal to - 18.74 dB.

The desired patterns are achieved, which again demonstrate the validity of the hybrid model.

5. CONCLUSION

The proposed method, based on hybrid particle swarm optimizer with breeding and subpopulation, allows the successful synthesis of multibeam pattern antenna arrays. The implementation procedure is described, and a linear antenna arrays are discussed to demonstrate the validity of the hybrid model. Optimized results show that the desired patterns are successfully obtained. The hybrid model is easy to implement and converges to the desired goal quickly. This study shows that the hybrid optimization performance model is quite excellent than other global optimization methods.

Table 2. Amplitude and phase distributions

Element N°	**Figure 3**		**Figure 4**		**Figure 5**		**Figure 6**	
	Amplitude	**Phase (Degree)**	**Amplitude**	**Phase (Degree)**	**Amplitude**	**Phase (Degree)**	**Amplitude**	**Phase (Degree)**
1	0.0187	14.3182	0.1006	-61.6503	0.2099	-101.0182	0.2943	-88.3386
2	0.1291	-68.4169	0.2094	108.0255	0.2777	112.7867	0.2986	85.6973
3	0.0723	-148.8201	0.2119	-51.0334	0.3835	-79.6068	0.4032	-84.5514
4	0.4628	25.8519	0.2074	88.8027	0.5819	100.7203	0.5905	81.0907
5	0.5793	13.0233	0.3661	-147.7257	0.4763	-64.7041	0.4962	-56.6942
6	0.1048	54.7404	0.6384	20.9703	0.1094	-41.9863	0.3927	43.2583
7	0.5509	119.6450	0.3477	-28.6307	0.2738	33.6441	0.7684	122.4296
8	0.3262	48.2029	0.7013	-12.6452	0.7132	17.2632	0.5303	-80.4375
9	0.9916	-40.1529	0.8981	104.2153	0.8468	50.4317	0.9469	6.0963
10	0.6673	-41.3446	0.3524	5.5462	0.7477	-102.8975	0.7611	-79.0911
11	0.6673	41.3446	0.3524	-5.5462	0.7477	102.8975	0.7611	79.0911
12	0.9916	40.1529	0.8981	-104.2153	0.8468	-50.4317	0.9469	-6.0963
13	0.3262	-48.2029	0.7013	12.6452	0.7132	-17.2632	0.5303	80.4375
14	0.5509	-119.6450	0.3477	28.6307	0.2738	-33.6441	0.7684	-122.4296
15	0.1048	-54.7404	0.6384	-20.9703	0.1094	41.9863	0.3927	-43.2583
16	0.5793	-13.0233	0.3661	147.7257	0.4763	64.7041	0.4962	56.6942
17	0.4628	-25.8519	0.2074	-88.8027	0.3835	-100.7203	0.5905	-81.0907
18	0.0723	148.8201	0.2119	51.0334	0.3835	79.6068	0.4032	84.5514
19	0.1291	68.4169	0.2094	-108.0255	0.2777	-112.7867	0.2986	-85.6973
20	0.0187	-14.3182	0.1006	61.6503	0.2099	101.0182	0.2943	88.3386

6. REFERENCES

[1] Torn, A. and Zilinskas, A. 1989. Global Optimization, lecture notes in computer science. New York springer-verlag, vol. 350.

[2] Lovbjerg, M., Rasmussen, T.K., and T. Krink. 2001. Hybrid Particle Swarm Optimizer with Breeding and Subpopulations, In Proceedings of the Genetic and Evolutionary Computation Conference, San Francisco.

[3] Akdagli, A and Guney, K. 2003. Shaped-beam Synthesis of Equally and Unequally Spaced Linear Antenna Arrays Using A Modified Tabu Search Algorithm, Microwave and Optical Technology Letters, Vol. 36, No. 1.

[4] Spears, W.M. 1994. Simple subpopulation schemes, proceeding of the evolutionary programming conference, pp.296-30.

[5] Kennedy, J. and Eberhart, R. C. 1995. Particle swarm optimization, IEEE proceeding of international conference on neural networks, vol.4, pp. 1942-1948.

[6] Y. Shi, R. C. Eberhart, A modified particle swarm optimiser, *IEEE international conference on evolutionary computation, Alaska, May 1998.*

[7] Shi, Y. and Eberhart, R. C. 1998. Parameter selection in particle swarm optimization, lecture notes in computer science. Evolutionary programming VII, 591-600. Springer.

[8] Kennedy, J. 1999. Small words and mega-minds: Effects of neighbourhood topology on particle swarm performance, proceedings of the congress of evolutionary computation, vol. 3, pp. 1931-1938. IEEE.

[9] Ghayoula, R., Traii, M., and Gharsallah, A. 2006. Application of the Neural Network to the Synthesis of Multibeam Antennas Arrays, Transaction on engineering, computing and technology. Vol. 14. August.