

# CONSTRICTED PARTICLE SWARM OPTIMIZATION FOR DESIGN OF COLLINEAR ARRAY OF UNEQUAL LENGTH DIPOLE ANTENNAS

Banani Basu<sup>1</sup> and G. K. Mahanti<sup>2</sup>

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

E-mail: basu\_banani@yahoo.co.in<sup>1</sup>, gautammahanti@yahoo.com<sup>2</sup>

## Abstract

A method based on constricted Particle Swarm Optimization (CPSO) algorithm to design a non-uniformly spaced collinear array of thin dipole antennas of unequal height is proposed. This paper presents a method for computing the appropriate excitation and geometry of individual array elements to generate a pencil beam in the vertical plane with minimum Standing Wave Ratio (SWR) and fixed Side Lobe Level (SLL). Coupling effect between any two collinear center-fed thin dipole antennas having sinusoidal current distributions is analyzed using induced EMF method and minimized in terms of SWR. DRR of excitation distribution is fixed at a lower value for further mitigation of the coupling effect. Phase distribution for all the elements is kept at zero degree for broadside array. Optimization results show the effectiveness of the algorithm for the design of the array. Moreover method seems very conducive for estimating the mutual impedance between any two collinear center-fed thin dipole antennas having sinusoidal current distributions.

## Keywords:

Constricted Particle Swarm Optimization (CPSO), Collinear Dipole Array, Standing Wave Ratio (SWR), Induced EMF Method, Dynamic Range Ratio (DRR)

## 1. INTRODUCTION

Array pattern synthesis is achieved by appropriately computing the excitation and geometric configuration of its radiating elements. Many methods have been used to achieve specified radiation pattern for non-uniformly excited, non-uniformly spaced linear arrays [1-12]. The analysis of non-uniformly spaced linear arrays was proposed by Unz[1], who developed a matrix formulation to obtain the current distribution necessary to generate a desired radiation pattern [1]. Skolnik [2] employed dynamic programming to design a unequally spaced array. Mailloux and Cohen [3] utilized the statistical thinning of arrays with quantized element weights to improve side lobe level performance. Different global optimization algorithms such as Genetic Algorithm (GA), Simulated Annealing (SA) and pattern search algorithm were used to thin an array [4-8]. Non-uniformly spaced array was further synthesized by randomly positioning the array element along the desired direction. Harrington developed an iterative method to reduce the sidelobe level of uniformly excited linear arrays by employing unequal spacing [9]. His method reduces the sidelobe level effectively without increasing the beamwidth of the mainbeam as obtained by uniformly spaced linear array. Literature described in [10-12] proposed different conventional and soft computing techniques for synthesis of non-uniformly spaced array.

In article [10], the particle swarm optimization was applied for optimization of non-uniformly spaced antenna arrays and side lobe level was reduced. Neural Network (NN) and least mean square technique was used to design non-uniformly spaced

array [11,12]. However most of the works consider the minimization of the side lobe level without considering mutual coupling effect. In recent works driving point impedance matching has been derived with unequal spacing of elements [13,14]. King [15] presented a method for evaluating the real and imaginary components of mutual impedance between any two thin dipole antennas, with an emphasis on antennas having sinusoidal current distributions. In the proposed work, we synthesize a non-uniformly spaced array consists of radiators of unequal heights.

In our work, CPSO is used for the synthesis of pencil beam pattern with specified SLL, DRR and minimum SWR value by optimizing the excitation and geometry of the individual array element. Coupling effect is compensated by minimizing standing wave ratio along with fixing dynamic ranges of excitation current amplitude distributions to a lower value. Impedance matrix is calculated using induced EMF method [15, 16].

## 2. THEORETICAL FORMULATION

We consider an array of  $2N$  collinear wire dipoles oriented in the vertical direction. All the dipoles are assumed non-identical and have very thin radii. The radiation pattern in the vertical plane depends on the geometry of the array as well as on the excitation currents applied at the center of the dipoles. The geometry of the array is specified by the lengths  $l_n$  ( $n = 1, \dots, N$ ) of the dipoles and the inter element spacing  $d_{n,n-1}$  ( $n = 2, \dots, N$ ) between them. Array elements are placed symmetrically on each side of the origin. Excitation and geometry both are assumed symmetric with respect to the origin. Assuming sinusoidal current distribution of a very thin dipole antenna directed along Z-axis, the element pattern is given by Eq. (1).

$$Elepat(\theta) = \frac{\cos\left(\frac{kl_n \cos \theta}{2}\right) - \cos\left(\frac{kl_n}{2}\right)}{\sin \theta} \quad (1)$$

The far-field pattern [16]  $F(\theta)$  in the vertical plane considering the element pattern with symmetric amplitude distributions is given by Eq.(2)

$$F(\theta) = \sum_{n=1}^N 2I_n \cos[kp_n \cos \theta] \times Elepat(\theta) \quad (2)$$

Normalized power pattern in dB can be expressed as follows.

$$P(\theta) = 10 \log_{10} \left[ \frac{|F(\theta)|}{|F(\theta)_{\max}|} \right]^2 = 20 \log_{10} \left[ \frac{|F(\theta)|}{|F(\theta)_{\max}|} \right] \quad (3)$$



position according to its own experience, and according to the experience of the neighboring bees. In fact, the bee takes into account the best positions encountered by itself and by its neighbors. So, its movement is an attempt to balance exploration and exploitation. Finally, this behavior leads the bees to a point where the highest density of flowers is found. Unable to find any other place where the flower concentration is even higher, the bees go back to this point. The modeling of the above behavior results in the PSO method. In PSO terminology, every individual in the swarm is called “particle” or “agent”. The number  $S$  of the particles that compose the swarm is called “population size”. The experience indicates that a population size between 10 and 50 is optimal for many problems. Particles act in the same way like bees do, i.e., they move in the search space and update their velocity according to the best positions already found by themselves and by their neighbors, trying to find an even better position. Each particle is treated as point in an  $N$ -dimensional space. The position of the  $i$ -th particle ( $i = 1, \dots, S$ ) is represented as  $x_i = (x_{i1}, x_{i2}, \dots, x_{iN})$ , where  $x_{in}$  ( $n = 1, \dots, N$ ) are the position coordinates. Each coordinate  $x_{in}$  may be limited in the respective ( $n$ -th) dimension between an upper boundary  $U_n$  and a lower boundary  $L_n$ , so that  $L_n \leq x_{in} \leq U_n$  ( $n = 1, \dots, N$ ). The difference  $R_n = U_n - L_n$  between the two boundaries is called “dynamic range” of the  $n$ -th dimension. The performance of each particle is measured according to a predefined mathematical function  $J$  called “cost function”, which is related to the problem to be solved. The value of the cost function depends on the position coordinates, i.e.,  $J = J(x_i) = J(x_{i1}, x_{i2}, \dots, x_{iN})$ . Actually, the particle position is considered to be improved as the value of the cost function is minimized. The best previous position (pbest position) of the  $i$ -th particle is recorded and represented as  $p_i = (p_{i1}, p_{i2}, \dots, p_{iN})$ . The change of  $x_i$  is  $\Delta x_i = v_i \Delta t$ , where  $\Delta t$  is the time interval,  $v_i = (v_{i1}, v_{i2}, \dots, v_{iN})$  is the velocity of the  $i$ -th particle and  $v_{in}$  ( $n = 1, \dots, N$ ) are the velocity coordinates. The calculation of the velocity is explained below. Considering that  $\Delta t = 1$ , the position change becomes  $\Delta x_i = \Delta v_i$ . Thus, the new position of  $i$ -th particle after a time step is given by Eq. (8).

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (8)$$

Particle swarms have been studied in two types of neighborhood, namely “gbest” and “lbest”. In the gbest neighborhood, every particle is attracted to the best position found by any particle of the swarm. This position, called “gbest position” and represented as  $g = (g_1, g_2, \dots, g_N)$  and corresponds to the minimum cost  $J_{min} = J(g) = J(g_1, g_2, \dots, g_N)$  found so far by the swarm. The gbest neighborhood is equivalent to a fully connected social network. Every individual is able to compare the performances of every other member of the population, imitating the very best. In the lbest neighborhood, each ( $i$ -th) individual is affected by the best performance of its  $K_i$  immediate neighbors, i.e., the  $i$ -th individual is attracted to the best position found by its  $K_i$  neighbors. This position, called “lbest position” and represented as  $l_i = (l_{i1}, l_{i2}, \dots, l_{iN})$  and corresponds to the minimum value  $J_{min,i} = J(l_i) = J(l_{i1}, l_{i2}, \dots, l_{iN})$  of the cost function found so far by the  $K_i$  neighbors of the  $i$ -th particle. These  $K_i$  neighbors are not necessarily particles who are near the individual in the parameter space, but rather ones that are near it in a topological space. The optimal pattern of connectivity among individuals depends on the problem being

solved. Using the gbest neighborhood the swarm tends to converge more rapidly on optima, but it is more susceptible to convergence on local optima.

As mentioned above, individuals are influenced by their own previous behavior and by the successes of their neighbors. So, the particle’s velocity depends on its previous velocity and the distance between the particle’s position and the best position found by the particle so far and finally on the distance between the particle’s position and the best position found so far by the swarm (for gbest model) or by the particle’s neighborhood (for lbest model). According to the gbest model, the velocity of the  $i$ -th particle after a time step is given by

$$v_i(t+1) = \omega * v_i(t) + c_1 * rand(t) * (p_i(t) - x_i(t)) + c_2 * rand(t) * (g(t) - x_i(t)) \quad (9)$$

where  $\omega$  is a positive parameter called “inertia weight”,  $c_1$  and  $c_2$  are positive parameters called respectively “cognitive coefficient” and “social coefficient”, and  $rand(t)$  is a function that generates random numbers drawn from a uniform distribution between 0 and 1. According to the lbest model, the only change is to substitute  $l_i$  for  $g$ . Thus, Eq. (9) is modified as follows:

$$v_i(t+1) = \omega * v_i(t) + c_1 * rand(t) * (p_i(t) - x_i(t)) + c_2 * rand(t) * (l_i(t) - x_i(t)) \quad (10)$$

The weight  $\omega$  usually has fixed values between 0 and 1 and controls the impact of the previous values of velocity on the current velocity. A larger  $\omega$  facilitates global exploration, while a smaller  $\omega$  tends to facilitate local exploration to fine-tune the current search area. Suitable choices of  $\omega$  provide a balance between global and local exploration abilities and thus require less iteration to find the optimum [18]. A good approach is to decrease  $\omega$  linearly from 0.9 to 0.4 during the course of a simulation [19]. The same value of  $\omega$  is used for all dimensions of all particles in a given population. The coefficient  $c_1$  determines how much the particle is influenced by the memory of its best location, while  $c_2$  determines how much the particle is influenced by the swarm (for gbest model) or by its neighbors (for lbest model). It was suggested that the best choice for both  $c_1$  and  $c_2$  is 2 [20].

It is easy to realize that the changes in the velocity are stochastic and an undesirable effect is that the particle’s trajectory can expand into wider and wider cycles through the problem space, eventually approaching infinity. One method of solving the problem is to implement a maximum allowed velocity  $v_{max} = (v_{max,1}, v_{max,2}, \dots, v_{max,N})$ . So, for each ( $i$ -th) particle and each ( $n$ -th) dimension, if  $v_{in} > v_{max,n}$  then  $v_{in} = v_{max,n}$ , and also if  $v_{in} < -v_{max,n}$  then  $v_{in} = -v_{max,n}$ .  $v_{max}$  has the beneficial effect of preventing explosion and scales the exploration of the particle’s search. Unfortunately, the choice of a value for  $v_{max}$  depends on the problem. For example, the particle will be trapped if a step larger than  $v_{max}$  is required to escape a local optimum. However, in approaching an optimum it is better to take smaller steps. It was found that if  $\omega = 1$  it is better to set each coordinate  $v_{max,n}$  around 10–20% of the dynamic range  $R_n$  of the respective dimension, and if  $\omega < 1$  it is better to set  $v_{max,n} = R_n$  ( $n = 1, \dots, N$ ) [20].

A recent analysis of the PSO presents an alternative way of calculation of the velocity [21]. The modified methodology is referred as CPSO.

### 3.2 CONSTRICTED PARTICLE SWARM OPTIMIZATION

The CPSO used here is same as reported in the paper [21] and this is reproduced here entirely from [21]. This approach uses a new parameter  $k$  called the constriction factor for ensuring the convergence. According to the gbest model, the velocity of the  $i$ -th particle after a time step is calculated by

$$v_i(t+1) = k * \left\{ \begin{array}{l} v_i(t) + \phi_1 * rand(t) * (p_i(t) - x_i(t)) \\ + \phi_2 * rand(t) * (g(t) - x_i(t)) \end{array} \right\} \quad (11)$$

while, according to the lbest model, the velocity is given by

$$v_i(t+1) = k * \left\{ \begin{array}{l} v_i(t) + \phi_1 * rand(t) * (p_i(t) - x_i(t)) \\ + \phi_2 * rand(t) * (l_i(t) - x_i(t)) \end{array} \right\} \quad (12)$$

In the above equations, the parameter  $k$  is called “constriction coefficient” and is defined by the following expression.

$$k = \frac{2}{\left| 2 - \phi - \sqrt{\phi^2 - 4\phi} \right|} \quad (13)$$

where the parameter  $\phi$ , sometimes called “acceleration constant”, must be greater than 4 ( $\phi > 4$ ) and is calculated by the expression.

$$\phi = \phi_1 + \phi_2 \quad (14)$$

where the parameters  $\phi_1$  and  $\phi_2$  have the same meaning like  $c_1$  and  $c_2$ , respectively. A standard choice recommended for both  $\phi_1$  and  $\phi_2$  is 2.05 [20]. The use of the constriction coefficient was another attempt to eliminate the need for  $v_{max}$ , but most authors agree that it is still better to use  $v_{max}$  in order to keep the particles in bounds. Nevertheless, the above parameters  $\omega$ ,  $k$ , and  $v_{max}$  are not always able to confine the particles within the search space, i.e.  $L_n \leq x_{in} \leq U_n$  ( $n = 1, \dots, N$ ). To solve this problem, three different boundary conditions have been suggested:

- The absorbing walls: when a particle hits  $U_n$  or  $L_n$ ,  $v_{in}$  becomes zero and the particle is pulled back toward the search space, i.e., if  $x_{in} > U_n$  then  $x_{in} = U_n$  and  $v_{in} = 0$ , and also if  $x_{in} < L_n$  then  $x_{in} = L_n$  and  $v_{in} = 0$ . In that manner, the energy of the particles that try to escape the search space is considered to be absorbed by the boundary walls.
- The reflecting walls: when a particle hits  $U_n$  or  $L_n$ ,  $v_{in}$  is reversed (becomes  $-v_{in}$ ) and the particle is reflected back toward the search space.
- Invisible walls: the particles are allowed to move inside or outside the search space without any restriction, but the fitness function is not evaluated for those particles being outside the search space. Actually, this technique saves computational time because the cost function is calculated only for the particles inside the search space.

Using the theory described above, the algorithm was developed to synthesize the proposed array.

#### Initialization

**Step-1:** Initialize counters  $t$  (for time steps),  $n$  (for dimensions), and  $i$  (to count particles).

**Step-2:** Set random number seed.

**Step-3:** Set the values of  $N$ ,  $S$ ,  $K_i$ ,  $\phi_1$ ,  $\phi_2$ ,  $t_{max}$  (total number of iterations) and the values of  $L_n, U_n, v_{max,n}$  for  $n=1, \dots, N$ .

**Step-4:** Randomly initialize the particle positions  $x_i$  ( $i=1, \dots, S$ ) inside the search space, so that  $L_n \leq x_{in} \leq U_n$  ( $n=1, \dots, N$ ).

**Step-5:** Randomly initialize the particle velocities  $v_i$  ( $i=1, \dots, S$ ). If  $v_{in} > v_{max,n}$  then  $v_{in} = v_{max,n}$  and also if  $v_{in} < -v_{max,n}$  then  $v_{in} = -v_{max,n}$ .

**Step-6:** Evaluate the values of the fitness function  $J(x_i)$  ( $i = 1, \dots, S$ ) for all the particles.

**Step-7:** Set  $p_i = x_i$  and  $J(p_i) = J(x_i)$  for  $i = 1, \dots, S$  (the first position of each particle is considered as pbest position).

**Step-8:** Find the minimum value  $J_{min}$  among the  $J(p_i)$  ( $i = 1, \dots, S$ ). The position that corresponds to  $J_{min}$  is the gbest position, so that  $J_{min} = J(g)$ .

#### Optimization

**Step-1:** For each ( $i$ -th) particle, find randomly  $K_i$  particles, which are the neighbors of the  $i$ -th particle.

**Step-2:** Find the individual that gives the minimum cost value  $J_{min,i}$  among the  $K_i$  neighbors of each ( $i$ -th) particle. The position of this individual is the lbest position  $l_i$  in the neighborhood of the  $i$ -th particle, so that  $J_{min,i} = J(l_i)$ .

**Step-3:** Update the particle velocities  $v_i$  ( $i = 1, \dots, S$ ) using Eq.(12). If  $v_{in} > v_{max,n}$  then  $v_{in} = v_{max,n}$  and also if  $v_{in} < -v_{max,n}$  then  $v_{in} = -v_{max,n}$ .

**Step-4:** Update the particle positions  $x_i$  ( $i = 1, \dots, S$ ) using Eq.(8), and apply the absorbing walls condition.

**Step-5:** Evaluate the cost value  $J(x_i)$  ( $i = 1, \dots, S$ ) for all the particles.

**Step-6:** For each ( $i$ -th) particle, if  $J(x_i) < J(p_i)$  ( $i = 1, \dots, S$ ) then  $p_i = x_i$  (the new position becomes pbest position of the  $i$ -th particle).

**Step-7:** For each ( $i$ -th) particle, if  $J(p_i) < J(g)$  ( $i = 1, \dots, S$ ) then  $g = p_i$  (the pbest position with the minimum cost value in the swarm becomes gbest position).

**Step-8:** Increase the counter  $t$  by 1.

**Step-9:** If  $t < t_{max}$  and  $J(g)$  was improved then go to Step-2. If  $t < t_{max}$  and  $J(g)$  was not improved then go to Step-1 of initialization part (meaning that the lbest neighborhood must be reinitialized for each particle).

## 4. RESULTS AND DISCUSSION

We consider a collinear array of 20 dipole antennas of radius  $0.005\lambda$ . Array elements are randomly placed along z-axis. To generate a pencil beam in the vertical plane, all excitation current phases are kept fixed at 0 degree and excitation current amplitude, inter element spacing and antenna length of each element are varied in the range 0 to 1, 0.6 to 1.2 and 0.4 to 0.6 respectively. Desired DRR value of amplitude distribution is prefixed at 7.

Because of symmetry, only ten amplitudes, nine inter element distances and ten antenna height are to be optimized using CPSO. The algorithm is designed to generate a vector of

29 real values between zero and one. The first 10 values of the vector are mapped and scaled to desired amplitude weight (0 to 1) range and next nine values are mapped and scaled to desired intermediate spacing weight (0.6 to 1.2) range and last ten values are mapped and scaled to desired length weight (0.4 to 0.6) range. We consider inter element distances from center to center and place first element at a prefixed distance from the origin.

A swarm size of 32 particles ( $S = 32$ ) is used in the algorithm. The velocity is calculated by eqn. (12) according to the  $l_{best}$  model, where each particle is affected by three neighbors ( $K_i = 3$ , for  $i = 1, \dots, S$ ). The parameters  $\varphi_1$  and  $\varphi_2$  are chosen equal to 2.05, and thus  $\varphi = 4.10$ . Then, eqn. (13) yields  $k = 0.73$ . Finally, the values of  $k$ ,  $\varphi_1$  and  $\varphi_2$  are used in eqn. (12). The algorithm makes use of  $v_{max}$ , where  $v_{max,n} = 0.20Rn$  ( $n = 1, \dots, N$ ). Also, the absorbing walls condition is used in order to confine the particles within the search space.

Table.1 shows the desired and obtained results in absence of ground plane. There is a good agreement between the desired and synthesized results. For optimization purpose we consider 10 array elements and calculate the SWR values of individual elements. Maximum SWR value is minimized in each step. We also calculate the average SWR value and it is found to be 1.79. SLL value also closely meets the desired specification of  $SLL_d$  (-30 dB). DRR of excitation distribution is found to be 7.64.

Table.1. Desired and obtained result for the array

Design Parameters	Pencil Beam	
	Desired	Obtained
SLL (dB)	-30.00	-28.90
Average SWR	NA	1.79
DRR	7	7.64

Table.2. Current Amplitude, antenna height, inter element spacing and SWR for the array

n	Non-Uniformly spaced array			
	Amplitude $I_n$	Antenna Height	Spacing	SWR <sub>n</sub>
±1	0.9088	0.5078	0.4539 (from origin)	1.8015
±2	0.7331	0.53106	0.81051	2.3022
±3	0.7881	0.53294	0.80244	2.3069
±4	0.8344	0.44132	0.84389	1.0613
±5	0.6153	0.47256	0.80798	1.0971
±6	0.5311	0.50036	0.80462	1.4588
±7	0.5575	0.45666	0.80003	1.0935
±8	0.2874	0.54022	0.89568	2.8924
±9	0.2556	0.43392	0.90479	1.1489
±10	0.1189	0.51072	0.78424	2.7658

Parameters obtained from the simulation are shown in Table.2. It shows the length of the individual dipoles, inter-element distances and finally the SWRs at the feeding points of the dipoles. It must be noted that the first dipole is placed arbitrarily 0.4539 wavelength away from the origin ( $z=0$ ). In broadside case, the excitation phases are not subject to optimization because the dipoles of broadside arrays are always

in phase and thus the phase is kept at zero degree. The DRR of the excitation is found to be 7.64. Because of symmetry, remaining ten elements are also excited with the same parameters.

The optimized result shows good matching with desired specification.

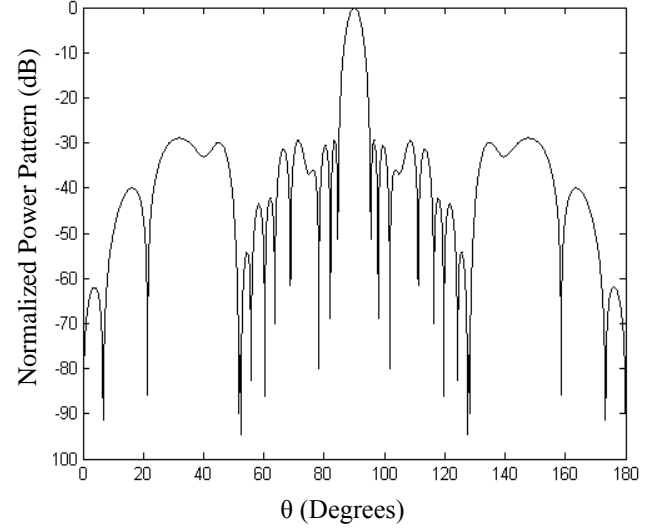


Fig.2. Normalized absolute power patterns in dB

Fig.2 shows the normalized absolute power patterns (pencil-beam) in dB for non-uniformly spaced collinear array antennas.

Average SWR value remains within the range from 1 to 2. The optimization process is capable of finding a structure where all the dipoles are closely matched to the feeding network. Introduction of constriction coefficient in PSO results in a quick convergence of the particles over time.

## 5. CONCLUSIONS

The use of constricted particle swarm optimization in the synthesis of non-uniformly spaced collinear array of unequal length is presented. An appropriate geometry and excitation distribution is chosen in order to satisfy the specified criterion. Phase is set at zero degree for all the elements. It is seen that perturbing the inter-element spacing significantly enhances array performance. The method used here remains limited for infinitely thin antennas having sinusoidal current distributions. The excitation and geometry both are symmetric in nature that greatly simplifies the feed network. Driving point impedance of each element is varied suitably by optimizing array geometry and excitation. Thus active impedances become matched with feed network and mutual coupling effect is compensated to the extent possible. There is a very good agreement between desired and obtained results using constricted PSO. The algorithm is capable of optimizing more complex geometries.

## REFERENCES

- [1] Unz H. 1960. Linear arrays with arbitrarily distributed elements. *IRE Trans. Antennas Propagation*. Vol.8. pp. 222–223.

- [2] Skolnik M. I., J. W. Sherman III, and G. Nemhauser. 1964. Dynamic programming applied to unequally spaced arrays. *IEEE Trans. Antennas Propagat.* Vol. 12. pp. 35–43.
- [3] Mailloux R. J., E. Cohen. 1991. Statistically thinned arrays with quantized element weights. *IEEE Trans. Antennas Propagat.* Vol. 39. pp. 436–447.
- [4] Haupt R. L. 1994. Thinned arrays using genetic algorithms. *IEEE Trans. Antennas Propagat.* Vol. 42. pp. 993–999.
- [5] Donelli M., S. Caorsi, F. DeNatale, M. Pastorino and A. Massa. 2004. Linear antenna synthesis with a hybrid genetic algorithm. *Progress In Electromagnetics Research*. PIER 49. pp. 1–22.
- [6] Mahanti G. K., N. Pathak, and P. Mahanti. 2007. Synthesis of thinned linear antenna arrays with fixed sidelobe level using real coded genetic algorithm. *Progress in Electromagnetic Research*. PIER 75. pp. 319–328.
- [7] C. A. Meijer, 1998, “Simulated annealing in the design of thinned arrays having Low sidelobe levels,” in *Proc. South African Symp. Communication and Signal Processing*, pp. 361–366.
- [8] Razavi C. A. and K. Forooghi. 2008. Thinned arrays using pattern search algorithms. *Progress In Electromagnetics Research*. PIER 78. pp. 61–71.
- [9] Harrington R. F. 1961. Sidelobe reduction by nonuniform element spacing. *IRE Trans. Antennas Propagat.* Vol. 9. pp. 187.
- [10] Lee K. C. and J.Y. Jhang. 2006. Application of particle swarm algorithm to the optimization of unequally spaced antenna arrays. *J. of Electromagn. Waves and Appl.* Vol.20. No. 14. pp. 2001–2012.
- [11] Ayestarán R. G., F. Las-Heras and J. A. Martínez. 2007. Nonuniform-antenna array synthesis using neural networks. *J. of Electromagn. Waves and Appl.* Vol. 21. No. 8. pp. 1001–1011.
- [12] Kazemi S. and H. R. Hassani. 2008. Performance improvement in amplitude synthesis of unequally spaced array using least mean square method. *Progress In Electromagnetics Research B*. Vol. 1. pp. 135–145.
- [13] Bray M. G., D. H. Werner, D. W. Boeringer and D. W. Machuga. 2002. Optimization of thinned aperiodic linear phased arrays using genetic algorithms to reduce grating lobes during scanning. *IEEE Trans. Antennas Propagat.* Vol. 50. No. 12. pp. 1732–1742.
- [14] M. G. Bray, D. H. Werner, D. W. Boeringer and D. W. Machuga, 2001, “Thinned aperiodic, linear phased array optimization for reduced grating lobes during scanning with input impedance bounds,” in *Proc. IEEE International Symposium on Antennas and Propagation Digest*, Vol. 3, pp. 688–691.
- [15] King H. E. 1957. Mutual impedance of unequal length antennas in echelon. *IRE Trans. Antennas Propagation*. Vol. AP-5. pp. 306-313.
- [16] Elliott R. S. 2003. *Antenna Theory and Design*, IEEE / Wiley Interscience, New York.
- [17] J. Kennedy and R. C. Eberhart, 1995, “Particle swarm optimization,” in *Proc. of the IEEE International Conference on Neural Networks*, Vol. IV, pp. 1942-1948.
- [18] Y. Shi, R. C. Eberhart, 1998, “Parameter selection in particle swarm optimization,” in *Proc. 7th annual conference on evolutionary programming*, pp. 591–600.
- [19] Y. Shi, R. C. Eberhart, 1999, “Empirical study of particle swarm optimization,” in *Proc. IEEE congress on evolutionary computation*, Vol. 3, pp 1945–1950.
- [20] R. C. Eberhart and Y. Shi, 2001, “Particle swarm optimization - developments, applications and resources,” in *Proc. IEEE congress on evolutionary computation*, Vol.1, pp 81–86.
- [21] Zaharis Z. D. 2008 Radiation pattern shaping of a mobile base station antenna array using a particle swarm optimization based technique. *Electrical Engineering (Archiv fur Elektrotechnik)*. Vol. 90. No. 4. pp. 301-311.

**INTENTIONALLY  
LEFT BLANK**