Published in IET Microwaves, Antennas & Propagation
Received on 23rd October 2010
Revised on 23rd January 2011
doi: 10.1049/iet-map.2010.0525

ISSN 1751-8725

Hybrid differential evolution and enhanced particle swarm optimisation technique for design of reconfigurable phased antenna arrays

H.M. Elragal   M.A. Mangoud   M.T. Alsharaa

Department of Electrical and Electronics Engineering, University of Bahrain, P. O. Box 32038, Isa Town, Kingdom of Bahrain
E-mail: mangoud@eng.uob.bh

Abstract: This study introduces a new design method for reconfigurable phased arrays using hybrid differential evolution (DE) and enhanced particle swarm optimisation (EPSO) technique. The proposed technique combines DE and enhanced version of standard PSO with improved mechanism that updates velocities and global best solution. In the hybrid algorithm, DE and EPSO are executed in parallel with frequent information sharing to enhance the newly generated population. To demonstrate the effectiveness of the proposed algorithm over each separate algorithm, examples for designing reconfigurable linear and circular antenna arrays with prescribed null directions are presented. Null steering is achieved by position perturbation of array elements in arbitrary directions with minimum sidelobe level change constraint. Another objective is to minimise the number of mobilised elements by introducing elements selection criteria. Simulation results show that the global search ability of the proposed algorithm is improved when compared with DE and EPSO separately.

1 Introduction

Reconfigurable antenna arrays that are capable of radiating with multiple power patterns using a single-power dividing network are desirable for many applications. These arrays have the potential of supporting operations such as beamforming, null steering, interference suppression and adaptive matching in a single geometry. Owing to the increasing pollution of the modern electromagnetic environments, a significant attention has been given to nulling steering techniques in reconfigurable arrays. Null steering techniques include controlling the feeding elements of the array with amplitude control, phase control, combination of both amplitude and phase control or element position control [1–4]. Null steering by adjusting complex weights of feeding excitations is the most efficient method because it has greater degrees of freedom for the solution space [5]. However, it is also the most expensive control method considering the cost of both a phase shifter and a variable attenuator for each array element. For reconfigurable array with a single feeding network, the most suitable technique is position perturbations of array elements. This technique needs servomotor for each element to make elements mobilised. It frees the phase shifters to be used solely for steering the main beam toward the direction of the desired signal. However, when the number of elements in the array increases, the complexity of the mechanism for position control of the elements will be higher. Also, the process will require longer computational time to find the new position perturbations. Hence, it is preferred to reduce the number of mobilised elements by perturbing only the position of selected elements depending on their contribution in the null forming. The selection process is introduced in [6] by eliminating the elements with less contribution to control the nulls in the antenna pattern. However, the solution is obtained for linear antenna arrays using a semi-analytical technique. Linear array has excellent directivity and it can form the narrowest main lobe in a given direction, but it does not work equally well in all azimuthal directions. In contrast, circular array structure owing to its symmetry can be electronically rotated in azimuthal plane of the array without a significant change of the beam shape. Recently, evolutionary algorithms have been widely used as a powerful technique to design phased antenna arrays [7]. Evolutionary algorithms such as genetic algorithms (GAs), simulated annealing, particle swarm optimisation (PSO) and differential evolution (DE) have been widely used in pattern synthesis applications [8]. Several methods for improving global optimisation performance are previously proposed. These methods can be classified into three categories. First, it is performing the same optimisation scheme successively with different initialisations. Although this method increases the chance of finding the optimum solution, it has high computational cost. Second, it is improving the optimisation algorithm itself by introducing a modified version with enhanced searching capability [9]. Finally, it is to introduce hybrid techniques of different evolutionary algorithms with collaborative usage of population. A hybrid technique that includes the essence and merits of GA and gradient-descent methods was introduced in [10]. The idea is to embed a gradient-descent algorithm into the evolution concept of the GA in order to provide a structured random search. Later on, improved genetic algorithm-extremum disturbed simple particle swarm optimisation (IGA-EDSPSO) hybrid technique has been presented in [11]. The algorithm is executed such that best solutions are exchanged between improved version of GA and a simplified version of PSO in an iterative way. Other robust hybrid methods applying combination of more than two different meta-heuristic optimisers have been developed in [12]. Recently, the idea of hybridisation DE and PSO has been emerged to inherit advantages of both techniques [13–15]. In [13], a combination of DE and PSO has been presented in sequential procedures. In [14], the mutation operation of DE is embedded in the PSO algorithm to improve its diversity exploration. The mutation operators are activated if velocity value of PSO is near to zero or violated from the boundaries. Another hybrid version of differential evolution named as Hybrid DE is introduced in [15]. Hybrid DE starts with the classical DE algorithm and then switch to PSO when the DE procedure starts to slow down. In this paper, a hybrid DE and enhanced version of PSO (EPSO) is proposed to increase the search space. The main idea is to execute the two techniques simultaneously. The evolved generations are obtained by merging and interchanging individuals produced from both techniques. The proposed evolutionary algorithm is developed and then applied for optimising linear and circular configurations by position perturbations of minimum number of elements. The position perturbation will be performed either in axial or vertical or both directions simultaneously for linear array configuration. However, for circular configurations the optimisation will be performed for one of the three cases: angular, radial or arbitrary direction that is combination of both angular and radial directions. The rest of this paper is organised as follows. Section 2 describes the classical DE algorithm, EPSO scheme and the proposed hybrid DE/EPSO algorithm. Section 3 presents null steering techniques using position perturbation for linear and circular antenna, in addition the objective function formulation of the design problem is presented. In Section 4, numerical examples are provided and discussed to show the capability of the proposed hybrid DE/EPSO. Finally, this paper is concluded in Section 5.

2 Hybrid DE/EPSO optimisation technique

2.1 Differential evolution

DE [16] is a parallel direct search meta-heuristic approach for global optimisation. The initial vector population is chosen randomly. Each population vector is set as a target vector. Mutation operation is performed by adding the weighted difference between two population vectors to a third one. The parameters of mutated vector are then mixed with the parameters of the target vector, to yield the so-called trial vector. Parameter mixing is often referred to as ‘crossover’. If the trial vector yields a lower-cost function value than the target vector, then the trial vector replaces the target vector in the following generation. Each population vector has to serve once as the target vector, so that competitions take place in one generation. DE algorithm can be summarised as follows:

Step 1: Initialise N-dimensional population vectors with P individuals, all randomly chosen from lower and upper bound interval [lb, ub].

Step 2: For each target vector $\boldsymbol{x}^i(k)$ for individual i at time k, where $i \in \{1, 2, \ldots, P\}$, select another three random individuals ($\boldsymbol{x}^{r1}$, $\boldsymbol{x}^{r2}$ and $\boldsymbol{x}^{r3}$) from the population. $r1$, $r2$ and $r3$ are random integers with mutually different indices $\in\{1, 2, \ldots, P\}$. One of these selected individuals ($\boldsymbol{x}^{r1}$) will serve as the base vector, and the other two ($\boldsymbol{x}^{r2}$ and $\boldsymbol{x}^{r3}$) will produce the differentiation vector ($\boldsymbol{x}^{r2} - \boldsymbol{x}^{r3}$).

Step 3: Create a mutant vector $\boldsymbol{x}_{\text{mut}}(k+1) = \boldsymbol{x}^{r1} + F(\boldsymbol{x}^{r2} - \boldsymbol{x}^{r3})$, where F is a constant of differentiation that is a randomly selected real number from the [0, 2] set.

Step 4: Crossover is introduced between target vector $\boldsymbol{x}^i(k)$ and mutant vector $\boldsymbol{x}_{\text{mut}}(k+1)$ in order to increase the diversity of the perturbed vector and create a trial vector as

$$x_{\text{trial}}^{i,j}(k+1) = \begin{cases} x_{\text{mut}}^{i,j}(k+1), & \text{if } \text{rand}(j) \le C_r \\ x^{i,j}(k), & \text{Otherwise} \end{cases} \tag{1}$$

where $x^{i,j}(k)$ is the jth element in the $\boldsymbol{x}^i(k)$ target vector, rand(j) is a random number taken from the interval [0, 1], $j \in \{1, 2, \ldots, N\}$ and C_r is preset crossover probability.

Step 5: When all the target vectors $\boldsymbol{x}^i(k)$, $i \in\{1, 2, \ldots, P\}$, have been processed this way, evaluate the objective function with the new trail vectors. If the objective function value of any trial vector is less than that of the corresponding target vector, replace the target vector with the new trial vector.

Step 6: Repeat from Step 2 until convergence criterion is met.

2.2 Enhanced particle swarm optimisation

PSO is a swarm intelligence method for global optimisation [17]. For classical PSO, each individual of the population adjusts its trajectory towards its own previous best position, and towards the previous best position attained by any member of its topological neighbourhood. EPSO algorithm [9] is similar to classical PSO with improved global search ability. This is accomplished by introducing an updating formula for global best particle position and adding two new terms in the velocity updating formula of classical PSO. EPSO algorithm can be summarised as follows:

Step 1: The positions, $\boldsymbol{x}^i(k)$ and velocities, $\boldsymbol{v}^i(k)$ of the initial population of particles are randomly generated for the ith particle at time k, where i is the current particle number in the swarm, $i \in \{1, 2, \ldots, P\}$ and P is the swarm size.

Step 2: According to fitness function values for each particle, locate particle with the best position value $\boldsymbol{p}^i(k)$ over the current swarm, and also update the global best position $\boldsymbol{p}^g(k)$ for the current and all the previous swarm moves.

Step 3: Update the position of the global best particle with zero velocity according to the following equation

$$\boldsymbol{p}^g(k+1) = [1 + (\lambda \cdot U)]\, \boldsymbol{p}^g(k) \tag{2}$$

where U is a Gaussian random number with zero mean and unit variance, λ is a convergence acceleration parameter.

Step 4: Update velocities of all particles at time $k+1$ as follows

$$\begin{aligned} \boldsymbol{v}^i(k+1) = {} & w \cdot \boldsymbol{v}^i(k) + c_1 \cdot r_1 \cdot (\boldsymbol{p}^i(k) - \boldsymbol{x}^i(k)) \\ & + c_2 \cdot r_2 \cdot (\boldsymbol{p}^g(k) - \boldsymbol{x}^i(k)) + c_3 \cdot r_3 \cdot (\boldsymbol{p}^i_{fdr}(k) - \boldsymbol{x}^i(k)) \\ & + c_4 \cdot r_4 \cdot (\boldsymbol{p}^g_{fdr}(k) - \boldsymbol{x}^i(k)) \end{aligned} \tag{3}$$

where, r_1, r_2, r_3 and r_4 are uniformly distributed random variables in [0, 1], w is an inertia factor, c_1 is a self-confidence factor, c_2 is a swarm confidence factor, c_3 and c_4 are acceleration constants, $\boldsymbol{p}^i_{fdr}(k)$ and $\boldsymbol{p}^g_{fdr}(k)$ are local and global candidate positions that are selected by locating the individual with minimum fitness-to-distance ratio (FDR) over all particles in the swarm.

The local and global FDR for each particle at time k is defined as

$$\mathrm{FDR}^i_{\mathrm{local}} = \frac{\mathrm{fitness}(\boldsymbol{p}^g(k)) - \mathrm{fitness}(\boldsymbol{x}^i(k))}{\mathrm{dist}(\boldsymbol{p}^i(k), \boldsymbol{x}^i(k))} \tag{4}$$

$$\mathrm{FDR}^i_{\mathrm{global}} = \frac{\mathrm{fitness}(\boldsymbol{p}^g(k)) - \mathrm{fitness}(\boldsymbol{x}^i(k))}{\mathrm{dist}(\boldsymbol{p}^g(k), \boldsymbol{x}^i(k))} \tag{5}$$

where fitness($\cdot$) is the cost function to be minimised and dist($\boldsymbol{p}^g(k)$, $\boldsymbol{x}^i(k)$) is a measure related to the distance between the particle global best position and all other particles on the swarm defined as

$$\mathrm{dist}(\boldsymbol{p}^z(k), \boldsymbol{x}^i(k)) = \left|\sum_{j=1}^{N}\sqrt{p^{z,j}(k)^2 - x^{i,j}(k)^2}\right| \tag{6}$$

where $\boldsymbol{p}^z(k)$ is either the local best $\boldsymbol{p}^i(k)$ or global best $\boldsymbol{p}^g(k)$ position vectors, $\boldsymbol{p}^{z,j}(k)$ is the jth component of the $\boldsymbol{p}^z(k)$ vector. $x^{i,j}(k)$ is the jth component of the ith particle position vector $\boldsymbol{x}^i(k)$ that is represented as $\boldsymbol{x}^i(k) = (x^{i,1}, x^{i,2}, \ldots, x^{i,N})$ in the N-dimensional search space (particle size).

Step 5: The position of each particle is updated using its velocity vector at time $k+1$ as

$$\boldsymbol{x}^i(k+1) = \boldsymbol{x}^i(k) + \boldsymbol{v}^i(k+1) \tag{7}$$

Step 6: Repeat from Step 2 until convergence criterion is met.

2.3 Hybrid DE/EPSO algorithm

In this section, the hybrid technique of the two robust meta-heuristic global optimisers DE and EPSO is presented. The hybridisation purpose is to include the essence and merits of both techniques. Generally, a hybrid DE/EPSO procedure is aimed to give a fairly accurate optimum solution by increasing the solution space. For each generated population from both optimisers that are simultaneously executed, a new generation is produced to form DE/EPSO population. By interchanging individuals between the two parallel optimisers, better solutions could be achieved.

Also, it decreases the chance of trapping in local minima. The proposed hybridisation works as follows:

Step 1: Initialise N-dimensional population of vectors with P individuals and apply DE to create a new generation.

Step 2: Initialise another N-dimensional population of vectors with P individuals and apply EPSO to create new generation.

Step 3: Merge the two new generations ($2P$ individuals) and locate the individual with the minimum fitness function value (best individual).

Step 4: Randomly select P individuals from the merged generation and insure that the selected individuals contain the best individual located in Step 3. If not, sort the selected individuals according to their fitness function values and replace the individual with the highest fitness function value by the best individual.

Step 5: Apply EPSO to the generation that contains the best individual that is produced in Step 4. This is to ensure that the EPSO will perform properly since the global best solution is needed in the updating equations.

Step 6: Apply DE to the remaining P individuals from the merged $2P$ population of Step 3.

Step 7: Go to Step 3 until a convergence criterion is met.

3 Unequally spaced antenna arrays with linear and circular geometries

In order to show the capability and flexibility of the proposed hybrid DE/EPSO to design reconfigurable antennas that steer a single null or multiple nulls in the imposed directions by controlling the position of selected elements. This section presents generalised array factor expressions for both linear and circular configurations when their elements are perturbed in arbitrary directions.

3.1 Array factor of linear antenna array with position perturbation

Consider initially a linear antenna array with $2N$ elements that are placed symmetrically along x-axis with inter-element spacing of d_{o} as shown in Fig. 1. The array factor in the azimuthal plane (x–y-plane) can be written as [18]

$$AF(\phi) = \sum_{n=1}^{2N} I_n \mathrm{e}^{jd_nk(\cos(\phi)-\cos(\phi_{\mathrm{o}}))} \tag{8}$$

where I_n is the nth element complex excitation, d_n is the nth element position from centre of the array that is defined as $d_n = d_{\mathrm{o}}(n\text{–}0.5)$, k is the wave number, ϕ is the angle measured from the array line direction and ϕ_{o} is the direction of the main beam steering angle.

Defining v_n as the position vector of the nth element with distance $|v_n|$ and angle β_n are defined as

$$|v_n| = (x_n^2 + h_n^2)^{1/2} \quad \text{and} \quad \beta_n = \tan^{-1}(h_n/x_n) \tag{9}$$

where x_n is the perturbation in axial direction and h_n is the perturbation in the normal direction with respect to the antenna array line. Owing to the array symmetry, the perturbation vector, $\boldsymbol{v}$, can be defined for only half of the elements number as $\boldsymbol{v} = [v_1, v_2, \ldots, v_N]$. The array factor of the geometry after position perturbation of the elements can be derived by including the contribution of perturbation

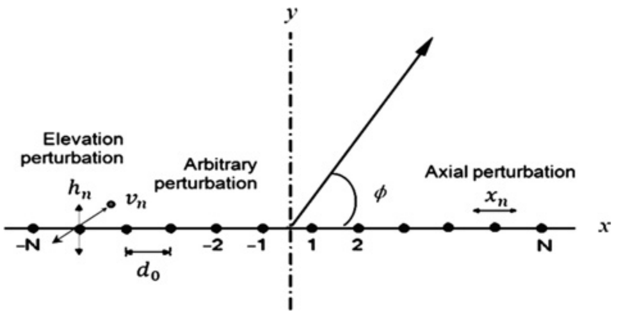

Fig. 1 *Geometry of linear array with illustration of elements perturbation directions*

doi: 10.1049/iet-map.2010.0525

vector $\boldsymbol{v}$ to (8) and it can be expressed as

$$\mathrm{AF}(\phi, \boldsymbol{v}) = 2\sum_{n=1}^{N}[I_n \cos[d_n(k\cos(\phi) - k\cos)(\phi_0) + |v_n|(k\cos(\beta_n - \phi) - k\cos(\phi_0))] \qquad (10)$$

3.2 Array factor of circular antenna array with position perturbation

Consider a uniformly spaced circular antenna array that is placed on the x–y-plane and consists of $2N$ isotropic elements equally spaced by d_0 around the circumference of a circle of radius r as shown in Fig. 2. The array factor pattern of this array can be described as in [18]

$$\mathrm{AF}(\phi) = \sum_{n=1}^{2N} I_n \mathrm{e}^{\mathrm{j}rk[\cos(\phi-\phi_n)-\cos(\phi_0-\phi_n)]} \qquad (11)$$

where I_n is the amplitude excitation of the nth element, $r = (2Nd_0)/2\pi$, k is the wave number, ϕ_n is the angular position of the nth element on the x–y-plane, where $\phi_n = 2\pi(n - 1/2N)$ for $n \in \{1, 2, \ldots, 2N\}$, ϕ_0 is the direction of the main beam. Each element in the array is subjected to one of three types of perturbation. Radial perturbation (Δr_n) along the radius r, angular perturbation ($\Delta\phi_n$) along the circumference of the circle or arbitrary perturbation that can be represented in by ($\Delta r_n e^{j\Delta\phi_n}$). The arbitrary perturbation can also be presented by the displacement vector $\boldsymbol{v}$ with respect to element's initial positions, where $v_n = |v_n|\mathrm{e}^{\mathrm{j}\beta_n}$, where β_n in the range between $\{-\pi, \pi\}$. Noting that the array has even number $2N$ with equal inter-elements spacing d_0 and $\phi_{n+N} = \phi_n + \pi$, then the overall position vector $\boldsymbol{v}$ can be reduced to N dimensions vector as $\boldsymbol{v} = [v_1, v_2, \ldots, v_N]$. The array factor patterns in this case can be written as

$$\mathrm{AF}(\phi, v) = 2\sum_{n=1}^{N} I_n \cos\{k(r + |v_n|)[\cos(\phi - (\phi_n + \beta_n)) - \cos(\phi_0 - (\phi_n + \beta_n))]\} \qquad (12)$$

3.3 Null steering technique by position control of selected elements

Null steering technique is presented in this section for linear- and circular-phased array configurations. The technique optimises displacement vectors and number of perturbed elements using the proposed hybrid DE/EPSO. In this optimisation process, two constrains are included while achieving the nulls in the pattern: minimum change of both average side-lobe level (SLL) and beamwidth of the main beam compared with initial equally spaced pattern. Considering the above constrains, the objective function is defined as follows

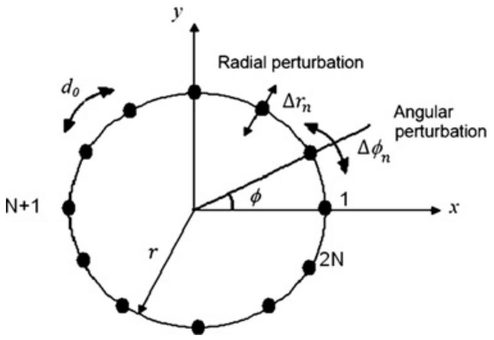

Fig. 2 *Geometry of circular array with illustration of the different types of elements position perturbation*

$$F(\boldsymbol{v}) = a\sum_{m=1}^{M}|\mathrm{AF}(\phi_m, \boldsymbol{v})|^2 + b\sum_i \frac{1}{\Delta\phi_i}\int_{\phi_{1i}}^{\phi_{2i}}|\mathrm{AF}(\phi, \boldsymbol{v})|d\phi \qquad (13)$$

where $|\mathrm{AF}(\phi, \boldsymbol{v})|$ is the array factor for linear- or circular-phased array that are described in (10) and (12), $\boldsymbol{v}$ is the perturbation position vector that needs to be optimised, ϕ_m is the angle of the prescribed null locations, M is the total number of nulls, $[\phi_{1i}, \phi_{2i}]$ are the ith spatial region boundaries of the array pattern excluding the main beam in which average peak SLL need to be constrained, $\Delta\phi_i = \phi_{1i} - \phi_{2i}$, a and b are weighting coefficients. In (13), the first term represents the fitness function that need to be optimised for nulls control purpose, whereas the second term is used for constraining the average SLL. The null steering technique is developed by optimising (13) for a given prescribed nulls to obtain the array geometry. The problem is then extended to iteratively minimise number of mobilised elements as follows:

Step 1: Optimise the position vector ($\boldsymbol{v}$) for N elements noting that the array has $2N$ elements with symmetric configuration.

Step 2: Eliminate two elements with lowest $|v_n|$ by setting them to zero.

Step 3: Optimise $\boldsymbol{v}$ for $N-2$ elements and record the resultant $\boldsymbol{v}$ and the corresponding $F(\boldsymbol{v})$.

Step 4: Eliminate one more element with the lowest $|v_n|$.

Step 5: Optimise $\boldsymbol{v}$ and then record the resultant $\boldsymbol{v}$ and the corresponding $F(\boldsymbol{v})$.

Step 6: Go to Step 4 if there is no significant decreasing change (approximately 20%) between $F(\boldsymbol{v})$ values for current and previous iterations. Otherwise, retrieve previous iteration and stop eliminating elements.

4 Numerical results

The goal of this section is to show the capability of DE/EPSO technique for optimising reconfigurable antennas to achieve accurate null steering. Designs of reconfigurable linear and circular array configurations with 20 isotropic elements are demonstrated. The objective is to steer two and five prescribed nulls for both configurations by position perturbation of minimum number of arrays elements. DE, EPSO and the proposed hybrid technique are implemented to optimise $F(\boldsymbol{v})$ to obtain the position vector $\boldsymbol{v}$ for each geometry starting with uniformly spaced configuration. The produced pattern is required to maintain deep nulls, while having the beamwidth of the main beam unchanged.

The DE algorithm uses the following parameters: population size = 20, number of generations = 100 000, constant of differentiation F in range of [0, 2] and probability of crossover $C_r = 0.9$. Meanwhile, EPSO algorithm uses the following parameters: swarm size = 20, number of generations = 100 000, inertia weight factor $w = 0.5$, acceleration constants $c_1 = c_2 = 2$, $c_3 = 0.5$, $c_4 = 0.55$. First, null position control is implemented for

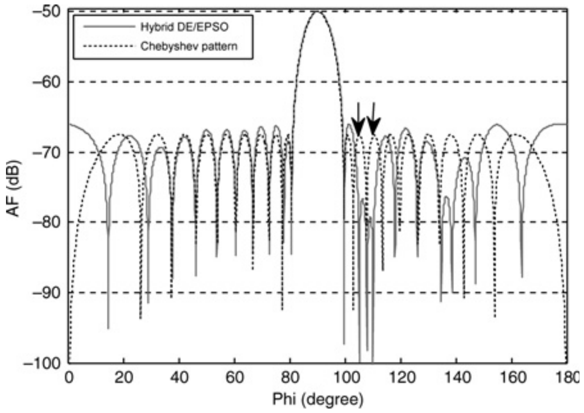

Fig. 3 *Initial and the three algorithms perturbed pattern for linear array with imposed nulls at 105 and 110°*

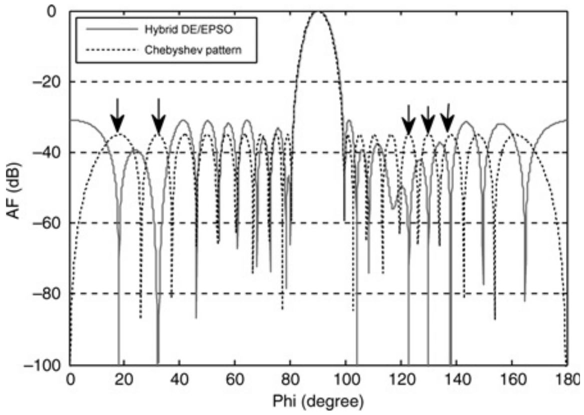

Fig. 4 *Initial and the three algorithms perturbed pattern for linear array with imposed nulls at 18, 32, 123, 130 and 138°*

linear antenna array placed on x-axis. The objective is to reconfigure the initial array pattern to have two nulls at azimuthal angles $\phi = 105°$ and $110°$. Also, the problem is extended to impose five nulls at $\phi = 18°$, $32°$, $123°$, $130°$ and $138°$. Initial pattern is set by assigning the excitation coefficients to produce −35 dB Chebyshev pattern with main beam directed toward the broadside ($\phi = 90°$). As defined in Section 3, each nth element is perturbed by a distance $|v_n|$ and angle β_n for one of three perturbation cases axial, elevation and arbitrary as shown in Fig. 1. The optimisation parameters are set as $0 \leq |v_n| \leq 0.08\lambda$ and $-\pi \leq \beta_n \leq \pi$. Figs. 3 and 4 depict the initial and perturbed pattern of the linear array with arbitrary perturbation using hybrid DE/EPSO for two and five imposed nulls, respectively. Tables 1 and 2 compare different perturbations and optimisation algorithms according to the resultant number of perturbed elements, peak SLL and the corresponding value of the worst case null depth. It is obvious that arbitrary perturbation using hybrid DE/EPSO algorithm gives the best solution. Table 3 presents perturbation vector values and locations of the selected mobilised elements for the best solution. As can be seen only six elements is to be position controlled to achieve a worst case null depth of −225 dB for two nulls case; also, ten elements need to be perturbed to obtain −212 dB for five nulls case. A comparison can be made between the hybrid method and another semi-analytical optimisation method [6]. Where the same 2 null example was performed resulting in reducing the initial level to ≃−72 dB by controlling 12 selected elements. Another difference is that the hybrid technique yields peak SLL of −32 dB besides having non-symmetrical nulls. However in [6], a higher SSL of −28 dB was observed with symmetrical null occurs at the other side of the main beam. Fig. 5 shows the convergence curve for the three

Table 1 Comparison between position perturbation types and optimisation algorithms for linear array and two nulls case

Null	Perturbation	Algorithm	No. of perturbed elements	SLL, dB	Null depth, dB
105°, 110°	axial	DE	10	−30	−86
		EPSO	10	−30.5	−42.2
		HYBRID	10	−30.5	−127
	elevation	DE	10	−30	−335
		EPSO	14	−30	−239
		HYBRID	8	−30	−296
	arbitrary	DE	6	−29	−80.5
		EPSO	6	−30.5	−125
		HYBRID	6	−32	−225

Table 2 Comparison between position perturbation types and optimisation algorithms for linear array and five nulls case

Null	Perturbation	Algorithm	No. of perturbed elements	SLL, dB	Null depth, Db
18°, 32°, 123°, 130°, 138°	axial	DE	16	−31	−43.2
		EPSO	16	−31	−47.2
		HYBRID	16	−31	−48.7
	elevation	DE	16	−31	−36.6
		EPSO	16	−31	−41.8
		HYBRID	16	−31	−41.8
	arbitrary	DE	16	−31.1	−39.3
		EPSO	12	−31.2	−212
		HYBRID	10	−31.1	−212

Table 3 Values of the position vector for arbitrary perturbation of linear array using hybrid DE/EPSO

Element location	Nulls 105°, 110°		Nulls 18°, 32°, 123°, 130°, 138°	
	$\|v_n\|$	β_n	$\|v_n\|$	β_n
1,11	0	0	0	0
2,12	0	0	0	0
3,13	0	0	0	0
4,14	0	0	0.0137	2.9216
5,15	0	0	0	0
6,16	0	0	0.0261	2.7572
7,17	0.0332	4.1457	0	0
8,18	0	0	0.0292	2.5362
9,19	00181	3.3935	0.0659	3.269
10,20	0.0678	0.9771	0.0646	1.3362

optimisation techniques that are used to achieve the imposed two nulls by arbitrary perturbing the elements of the considered linear antenna array. As shown EPSO, DE and hybrid DE/EPSO achieve convergence after 3000, 6000 and 8000 iterations, respectively, that are equivalent to 150, 300 and 400 generations. This figure also indicates that the proposed hybrid DE/EPSO has the minimum value of fitness function. It should be noted here that the fitness function is a weighted sum function of the null depth, SLL change and the number of perturbed elements.

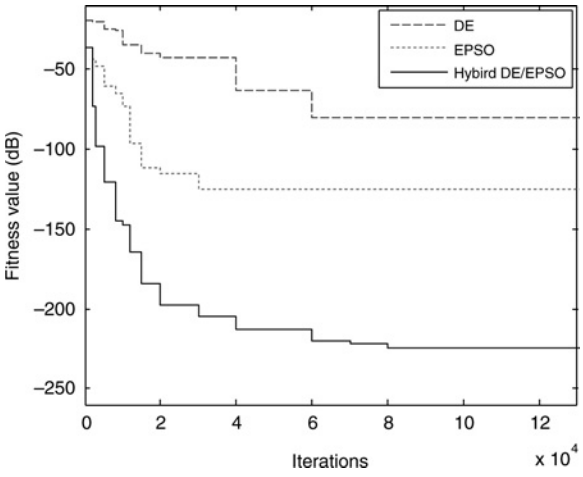

Fig. 5 *Convergence curves for the three algorithms used to design perturbed linear array (two nulls – arbitrary perturbation case)*

Second, hybrid DE/EPSO is applied to design reconfigurable circular array to form multiple nulls in the azimuthal radiation pattern. The initial circular array pattern is assumed to be similar to the Chebyshev pattern of linear antenna array. The initial array geometry consists of 36 uniformly excited elements with 0.25λ inter-element arc spacing. The complex excitation coefficients of this circular array are obtained from [19]. These excitation coefficients produce a -33 dB Chebyshev-like pattern. The positions of the array elements are perturbed in one of three directions, angular, radial or arbitrary. The objective is to steer two

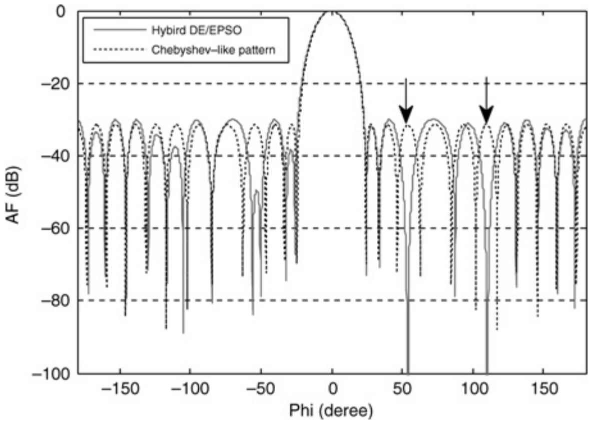

Fig. 6 *Initial and the three algorithms perturbed pattern for circular array with imposed nulls at 54, 110°*

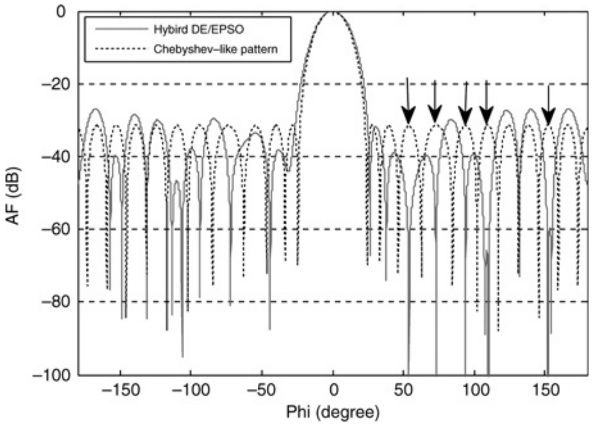

Fig. 7 *Initial and the three algorithms perturbed pattern for circular array with imposed nulls at 54, 73, 94, 110, 152°*

Table 4 Comparison between position perturbation types and optimisation algorithms for circular array and two nulls case

Angle	Perturbation type	Algorithm	No. perturbed element	SLL, dB	Null depth, dB
54°, 110°	angular	DE	36	−30	−33.3
		EPSO	36	−30	−34.9
		HYBRID	36	−30	−34.4
	radial	DE	28	−30	−166.5
		EPSO	28	−30	−208
		HYBRID	18	−30	−324.9
	arbitrary	DE	36	−30	−34
		EPSO	18	−30	−221.7
		HYBRID	18	−30	−332

Table 5 Comparison between position perturbation types and optimisation algorithms for circular array and five nulls case

Angle	Perturbation type	Algorithm	No. perturbed element	SLL, dB	Null depth, dB
54°, 73°, 94°, 110°, 152°	angular	DE	36	−27	−32.3
		EPSO	36	−27	−34.5
		HYBRID	36	−27	−34.3
	radial	DE	36	−27	−39.5
		EPSO	26	−27	−222
		HYBRID	26	−27	−223.4
	arbitrary	DE	24	−27	−212.2
		EPSO	24	−27	−201.7
		HYBRID	22	−27	−178

nulls at 54° and 110° and five nulls at 54°, 73°, 94°, 110° and 152°. The optimising parameters was set to $-0.1\lambda \leq \Delta r_n \leq 0.1\lambda$ for radial perturbation and $-2.5^\circ \leq \Delta\phi_n \leq 2.5^\circ$ for angular perturbation. Figs. 6 and 7 show the initial and the perturbed pattern by imposing two and five nulls respectively. As can be seen, nulls are perfectly formed at the prescribed values. Tables 4 and 5 compare angular, radial or arbitrary perturbation for different optimisation techniques. Similar to linear array examples, arbitrary perturbation using hybrid DE/EPSO method leads to the best solution. The superiority of hybrid DE/EPSO technique is proved by achieving the two preset nulls by perturbing only 18 out of the 36 elements with values not greater than −332 dB. Also, 22 elements are perturbed to form five nulls with worst case depth of −178 dB. Table 6 provides further details for the best solution by presenting the Δr_n and $\Delta\phi_n$ values and the locations of mobilised elements. It should be noted that the number of selected elements could be further reduced if either the null depth or SLL optimisation are relaxed. Similar to the linear array case, the convergence curves for imposing two nulls using arbitrary perturbation of circular array are presented in Fig. 8. In this particular example, it is noted that DE converges too fast and trapped in a local minimum solution. However, the proposed hybrid DE/EPSO achieves the minimum value compared with DE and EPSO.

Table 6 Values of the position vector for arbitrary perturbation of circular array using hybrid DE/EPSO

Element's location	Nulls 54°, 110°		Nulls 54°, 73°, 94°, 110°, 152°	
	Δr_n (λ)	$\Delta\phi_n$	Δr_n (λ)	$\Delta\phi_n$
1,19	−0.02	0.412	0	0
2,18,20,36	0	0	0	0
3,17,21,35	0.0046	0.272	0.0169	−0.097
4,16,22,34	0	0	0.023	−0.0657
5,15,23,33	0	−0.0495	−0.028	−0.186
6,14,24,32	−0.02	−0.042	−0.053	−0.2211
7,13,25,31	0	0	−0.025	−0.277
8,12,26,30	0	0	0	0
9,11,27,29	0.007	−0.168	0	0
10,28	0	0	−0.005	0.261

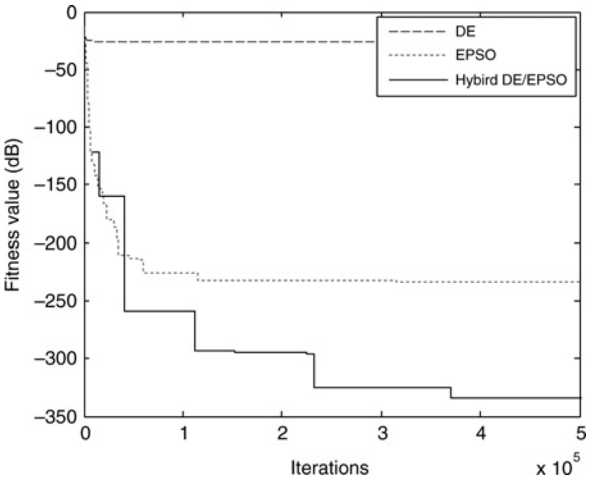

Fig. 8 *Convergence curves for the three algorithms used to design perturbed circular array (two nulls – arbitrary perturbation case)*

5 Conclusion

In this paper, a novel hybrid global optimisation algorithm DE/EPSO combining DE and EPSO has been introduced. The proposed technique has been applied for interference suppression by position-only control using minimum number of mobilised elements for linear- and circular-phased arrays. Null steering is performed by perturbing the positions of selected elements while freezing the positions of those elements that have insignificant contributions. The resultant patterns are optimised to impose non-symmetric deep nulls and be as close as possible to the initial equiripple Chebyshev patterns. The numerical results show that the hybrid DE/EPSO outperforms the classical DE and the EPSO in obtaining the desired patterns with minimum number of perturbed elements. Different types of position perturbations in either one or two dimensions have been presented and compared. It can be concluded that the hybrid DE/EPSO is very effective algorithm with good accuracy and it is promising to be applied for solving global optimisation problems and designing reconfigurable antenna arrays.

6 References

1 Steyskal, H., Shore, R.A., Haupt, R.L.: 'Methods for null control and their effects on the radiation pattern', *IEEE Trans. Antennas Propag.*, 1986, **AP-34**, pp. 404–409

2 Chung, Y.C., Haupt, R.L.: 'Amplitude and phase adaptive nulling with a genetic algorithm', *J. Electromagn. Waves Appl.*, 2000, **14**, pp. 631–649

3 Yang, S., Gan, Y.B., Qing, A.: 'Antenna-array pattern nulling using a differential evolution algorithm', *Int. J. RF Microw. CAE*, 2004, **14**, (1), pp. 57–63

4 Güney, K., Onay, M.: 'Amplitude-only pattern nulling of linear antenna arrays with use of bees algorithm', *Prog. Electromagn. Res.*, 2007, **70**, pp. 21–36

doi: 10.1049/iet-map.2010.0525

5 Karaboga, D., Güney, K., Akdagli, A.: 'Antenna array pattern nulling by controlling both amplitude and phase using modified touring ant colony optimisation algorithm', *Int. J. Electron.*, 2004, **91**, (4), pp. 241–251

6 Hejres, J.A.: 'Null steering in phased arrays by controlling the positions of selected elements', *IEEE Trans. Antennas Propag.*, 2004, **52**, (11), pp. 2891–2895

7 Panduro, M.A., Brizuela, C.A.: 'A comparative analysis of the performance of GA, PSO and DE for circular antenna arrays'. Int. Symp. on Antennas and Propagation Society, 2009, APSURSI'09. IEEE, 1–5 June 2009, pp. 1–4

8 Hoorfar, A.: 'Evolutionary programming in electromagnetic optimisation: a review', *IEEE Trans. Antennas Propag.*, 2007, **55**, (3), pp. 523–537

9 Mangoud, M.A.-A., Elragal, H.M.: 'Antenna array pattern synthesis and wide null control using enhanced particle swarm optimization', *Prog. Electromagn. Res. B*, 2009, **PIERB 17**, pp. 1–14

10 Donelli, M., Caorsi, S., De Natale, F., Pastorina, M., Massa, A.: 'Linear antenna synthesis with a hybrid genetic algorithm', *Prog. Electromagn. Res.*, 2004, **PIER 49**, pp. 1–22

11 Zhang, S., Gong, S.-X., Guan, Y., Zhang, P.-F., Gong, Q.: 'A novel IGA-EDSPSO hybrid algorithm for the synthesis of sparse arrays', *Prog. Electromagn. Res.*, 2009, **PIER 89**, pp. 121–134

12 Oldenhuis, R.P.S.: 'Trajectory optimization of a mission to the solar bow shock and minor planets'. Thesis Report, Delft University of Technology, 2010

13 Vinkó, T., Izzo, D., Bombardelli, C.: 'Benchmarking different global optimisation techniques for preliminary space trajectory design'. 58th Int. Astronautical Congress, Hyderabad, India, 2007

14 Khamsawang, S., Wannakarn, P., Jiriwibhakorn, S.: 'Hybrid PSO-DE for solving the economic dispatch problem with generator constraints'. The Second Int. Conf. on Computer and Automation Engineering (ICCAE), 2010

15 Ali, M., Pant, M., Abraham, A.: 'Inserting information sharing mechanism of PSO to improve the convergence of DE'. VIII Int. Conf. on Computer Information Systems and Industrial Management (CISIM 2009), Coimbatore, India, 8–11 December 2009

16 Storn, R., Price, K.: 'Differential evolution-A simple and efficient heuristic for global optimisation over continuous spaces', *J. Glob. Optim.*, 1997, **11**, (4), pp. 341–359

17 Kennedy, J., Eberhart, R.: 'Particle swarm optimisation'. Proc. IEEE Int. Conf. Neural Networks, Piscataway, NJ, USA, 1995, pp. 1942–1948

18 Balanis, C.A.: 'Antenna theory, analysis and design' (John Wiley & Sons, 1997, 2nd edn.)

19 Goto, N., Tsunoda, Y.: 'Side-lobe reduction of circular array with a constant excitation amplitude', *IEEE Trans. Antennas Propag.*, 1977, **25**, (6), pp. 896–898

doi: 10.1049/iet-map.2010.0525