

Research Article

Shaped Beam Pattern Synthesis of Antenna Arrays Using Composite Differential Evolution with Eigenvector-Based Crossover Operator

Sotirios K. Goudos

Radiocommunications Laboratory, Section of Applied and Environmental Physics, Department of Physics, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece

Correspondence should be addressed to Sotirios K. Goudos; sgoudo@physics.auth.gr

Received 6 May 2015; Revised 5 July 2015; Accepted 12 July 2015

Academic Editor: Kerim Guney

Copyright © 2015 Sotirios K. Goudos. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper addresses the problem of designing shaped beam patterns with arbitrary arrays subject to constraints. The constraints could include the sidelobe level suppression in specified angular intervals, the mainlobe halfpower beamwidth, and the predefined number of elements. In this paper, we propose a new Differential Evolution algorithm, which combines Composite DE with an eigenvector-based crossover operator (CODE-EIG). This operator utilizes eigenvectors of covariance matrix of individual solutions, which makes the crossover rotationally invariant. We apply this novel design method to shaped beam pattern synthesis for linear and conformal arrays. We compare this algorithm with other popular algorithms and DE variants. The results show CODE-EIG outperforms the other DE algorithms in terms of statistical results and convergence speed.

1. Introduction

Array synthesis is a classic and challenging optimization problem, which has been extensively studied using several analytical or stochastic methods [1–4]. Common optimization goals in array synthesis are the sidelobe level suppression and the matching of the main lobe to a desired pattern. Thus, the optimization problem is usually to find a set of element excitations and/or positions that closely match a desired pattern. The desired pattern shape can vary widely depending on the application. The shaped beam pattern synthesis problem has received wide attention over the years, and several methods or techniques have been reported in the literature [5–11]. Among others these methods include Tabu search [5], ant colony optimization [11], and linear programming [7]. Additionally, evolutionary algorithms (EAs) like Genetic Algorithms (GAs), Particle Swarm Optimization (PSO), and Differential Evolution (DE) [12], which is a population-based stochastic global optimization algorithm, have been applied to a variety of array design problems [13–21]. Several DE variants or strategies exist. One of the DE advantages is that

very few control parameters have to be adjusted in each algorithm run.

Composite DE (CODE) [22] is an adaptive DE variant, which combines three different trial vector generation strategies with three preset control parameter settings. The above combination is performed in a random way in order to generate trial vectors. The main advantage of CODE is that it has a simple structure and thus it is very easy to be implemented in any programming language. In this way significant computational costs spent on searching using a trial-and-error procedure can be avoided. The DE performance depends on control parameters, mutation strategies, and crossover operators. Most of the DE strategies or variants use the binomial crossover operator, which has been found to produce better results than the exponential crossover operator [23]. The authors in [24] propose an alternative crossover operator, namely, the eigenvector-based crossover. This operator utilizes the eigenvector information of the covariance matrix of the population to rotate the coordinate system. Additionally, this crossover operator can be applied to any DE variant. In this paper, we incorporate this scheme

to the CODE algorithm and therefore introduce a new algorithm, the Composite DE with Eigenvector-Based Crossover (CODE-EIG).

In this paper, we apply CODE-EIG to the shaped beam pattern synthesis problem. More specifically, we utilize the CODE-EIG algorithm for designing different linear arrays cases and a conformal array case. Additionally, in order to evaluate the new algorithm, we compare the algorithm's performance with other DE strategies, PSO, and GAs.

This paper is organized as follows: A brief description of the CODE and the CODE-EIG algorithm is given in Section 2. Section 3 describes the problem formulation and the objective functions and presents the numerical results for different array design cases. Finally, the conclusion is given in Section 4.

2. The Composite Differential Evolution (CODE) Algorithm

A population in DE consists of NP vectors $\bar{x}_{G,i}$, $i = 1, 2, \dots, NP$, where G is the generation number. The population is initialized randomly from a uniform distribution. Each D -dimensional vector represents a possible solution, which is expressed as

$$\bar{x}_{G,i} = (x_{G,1i}, x_{G,2i}, \dots, x_{G,ji}, \dots, x_{G,Di}). \quad (1)$$

The population is initialized as follows:

$$x_{0,ji} = \text{rand}_{j|0,1}(x_{j,U} - x_{j,L}) + x_{j,L} \quad j = 1, 2, \dots, D, \quad (2)$$

where $x_{j,L}$ and $x_{j,U}$ are D -dimensional vectors of the lower and upper bounds, respectively, and $\text{rand}_{j|0,1}$ is a uniformly distributed random number within $[0, 1)$.

The initial population evolves in each generation with the use of three operators: mutation, crossover, and selection. Depending on the form of these operators, several DE variants or strategies exist in the literature [12, 25]. The choice of the best DE strategy depends on problem type [23]. In CODE the following three strategies are used for trial vector generation. These include DE/rand/1bin, DE/rand-to-best/2/bin, DE/rand/2/bin, and DE/current-to-rand/1 [26]. In these strategies, a mutant vector $\bar{v}_{G+1,i}$ for each target vector $\bar{x}_{G,i}$ is computed by

DE/rand/1/bin

$$\bar{v}_{G+1,i} = \bar{x}_{G,r_1} + F(\bar{x}_{G,r_2} - \bar{x}_{G,r_3}), \quad r_1 \neq r_2 \neq r_3. \quad (3a)$$

DE/rand/2/bin

$$\bar{v}_{G+1,i} = \bar{x}_{G,r_1} + F(\bar{x}_{G,r_2} - \bar{x}_{G,r_3}) + F(\bar{x}_{G,r_4} - \bar{x}_{G,r_5}), \quad (3b)$$

$$r_1 \neq r_2 \neq r_3 \neq r_4 \neq r_5.$$

DE/current-to-rand/1

$$\bar{v}_{G+1,i} = \bar{x}_{G,i} + \text{rand}_{j|0,1}(\bar{x}_{G,r_1} - \bar{x}_{G,i}) + F(\bar{x}_{G,r_2} - \bar{x}_{G,r_3}), \quad (3c)$$

$$r_1 \neq r_2 \neq r_3,$$

where $r_1, r_2, r_3, r_4,$ and r_5 are randomly chosen indices from the population, which are different from index i , F is a mutation control parameter, and $\text{rand}_{j|0,1}$ is a randomly generated number from a uniform distribution within the interval $[0, 1)$. The basic idea of CODE [22] is to generate a trial vector for each of the three above-mentioned strategies. The control parameters settings are randomly chosen. Therefore, CODE maintains a strategy candidate pool, consisting of the three strategies given in (3a), (3b), and (3c). A parameter candidate pool is also used with the settings given below:

$$(F = 1.0, CR = 0.1),$$

$$(F = 1.0, CR = 0.9), \quad (4)$$

$$(F = 0.8, CR = 0.2).$$

These are strategies and settings that have been widely used. Therefore, their properties have been extensively studied. The authors in [22] have modified the DE/rand/2/bin strategy by using a random number from the interval $[0, 1)$ instead of the first F mutation control parameter. The implementation of the CODE is quite simple. The best trial vector of the three generated is chosen. Then it is compared with the old vector and it replaces it if its fitness function value is smaller.

After mutation in each generation for every vector of the population, the binary crossover operator is applied to generate a trial vector $\bar{u}_{G+1,i} = (u_{G+1,1i}, u_{G+1,2i}, \dots, u_{G+1,ji}, \dots, u_{G+1,Di})$ whose coordinates are given by

$$u_{G+1,ji} = \begin{cases} v_{G+1,ji} & \text{if } \text{rand}_{j|0,1} \leq CR \text{ or } j = \text{rn}(i), \\ x_{G+1,ji} & \text{if } \text{rand}_{j|0,1} > CR, \quad j \neq \text{rn}(i), \end{cases} \quad (5)$$

where $j = 1, 2, \dots, D$, $\text{rand}_{j|0,1}$ is a number from a uniform random distribution within the interval $[0, 1)$, $\text{rn}(i)$ a randomly chosen index from $(1, 2, \dots, D)$, and CR the crossover constant from the interval $[0, 1]$.

In CODE, the crossover operator is not applied for the DE/current-to-rand/1 strategy. CODE uses a greedy selection operator, which for minimization problems is defined by

$$\bar{x}_{G+1,i} = \begin{cases} \bar{u}_{G+1,i}, & \text{if } f(\bar{u}_{G+1,i}) < f(\bar{x}_{G,i}), \\ \bar{x}_{G,i}, & \text{otherwise,} \end{cases} \quad (6)$$

where $f(\bar{u}_{G+1,i})$, $f(\bar{x}_{G,i})$ are the fitness values of the trial and the old vector, respectively. Therefore, the newly found trial vector $\bar{u}_{G+1,i}$ replaces the old vector $\bar{x}_{G,i}$ only when it produces a lower objective function value than the old one. Otherwise, the old vector remains in the next generation. The stopping criterion for the CODE or DE in general is usually the generation number or the number of objective function evaluations.

2.1. CODE with Eigenvector-Based Crossover Operator (CODE-EIG). The authors in [24] propose an eigenvector-based crossover operator, which utilizes eigenvectors of covariance matrix of individual solutions. Thus, the crossover is rotationally invariant. To avoid losing diversity of population,

the offspring can be stochastically born from the parents with either the standard coordinate system or the rotated coordinate system. The authors in [24] also introduce a new parameter to control the probability of selecting one of the coordinate systems. They have shown that this scheme can increase the population diversity and prevent premature convergence. Additionally, another significant advantage of this operator is that it can be applied to any crossover strategy with minimal changes. Therefore, it can enhance any existing DE variant. The main idea is to exchange the information between the target vector and the mutant vector in the eigenvector basis instead of the natural basis. We consider the same population of NP D -dimensional vectors defined in the previous section. The covariance between i th and j th dimension of the population in the G th generation is given by [24]

$$\text{cov}(i, j) = \frac{\sum_{k=1}^{\text{NP}} (x_{G,ik} - \mu_{G,i})(x_{G,jk} - \mu_{G,j})}{\text{NP} - 1}, \quad (7)$$

where $\mu_{G,i}, \mu_{G,j}$ are the mean values of the variables in the i th and j th dimension, respectively.

To compute the eigenvector basis we need to factorize the covariance matrix $\mathbf{C}_G = (c_{ij}, c_{ij} = \text{cov}(i, j))$ into a canonical form

$$\mathbf{C}_G = \mathbf{Q}_G \Lambda_G \mathbf{Q}_G^{-1}, \quad (8)$$

where \mathbf{Q}_G is the square matrix ($D \times D$) whose i th column is the eigenvector $\bar{q}_{G,i}$ of \mathbf{C}_G and Λ_G is the diagonal matrix whose diagonal elements are the corresponding eigenvalues. The factorization of a matrix into a canonical form is called eigendecomposition. The authors in [24] use Jacobi's method [27] for eigendecomposition. When the eigenvector basis is found, the i th target vectors $\bar{x}_{G,i}$ can be expressed by $(\mathbf{Q}_G \cdot \bar{x}_{G,i})$; the i th mutant vectors $\bar{v}_{G+1,i}$ can be expressed by $(\mathbf{Q}_G \cdot \bar{v}_{G+1,i})$. Then, a predefined crossover operator, such as binomial crossover, will exchange some of the elements of the mutant vector with some of the elements of its target vector to form a trial vector. The trial vector is then given by [24]

$$\bar{u}_{G+1,i} = \begin{cases} \mathbf{Q}_G^* \cdot \text{xover}(\mathbf{Q}_G \cdot \bar{x}_{G,i}, \mathbf{Q}_G \cdot \bar{v}_{G+1,i}), & \text{if } \text{rand} \leq P, \\ \text{xover}(\bar{x}_{G,i}, \bar{v}_{G+1,i}), & \text{otherwise,} \end{cases} \quad (9)$$

where \mathbf{Q}_G^* is the conjugate transpose of the eigenvector basis \mathbf{Q}_G and $\text{xover}(\bar{x}, \bar{y})$ is a crossover operator on two vectors \bar{x} and \bar{y} , where P is a new control parameter introduced in [24] which is called eigenvector ratio between 0 and 1 that determines the ratio of the eigenvector-based crossover operator and the other crossover operator. The main feature of this approach is that no matter how the crossover operator exchanges the elements in the eigenvector basis, the crossover behavior will become rotationally invariant in the natural basis.

Therefore, to create the CODE-EIG variant we use (3a), (3b), and (3c) to generate the mutant vector and (9) to create the trial vector. Additionally, this new variant requires the setting of the eigenvector ratio P . It must be noted that when $P = 1$ then only the eigenvector crossover operator is used, while when $P = 0$ then only the binary crossover operator is used.

The CODE-EIG algorithm is outlined below:

- (1) Initialize the CODE-EIG parameters. That is, set the maximum number of generations G_{max} , the population size number NP, and the eigenvector ratio P .
- (2) Initialize a uniformly distributed random population of NP individuals. Set $G = 1$.
- (3) Evaluate objective function value for every vector of the population.
- (4) For each vector of the population, generate three mutant vectors $\bar{v}_{G+1,i,1}, \bar{v}_{G+1,i,2}, \bar{v}_{G+1,i,3}$, each with one of the three strategies given in (3a), (3b), and (3c) and with a control parameter setting randomly selected from the parameter candidate pool.
- (5) Apply the eigenvector-based crossover operator according to (9) and create three trial vectors $\bar{u}_{G+1,i,1}, \bar{u}_{G+1,i,2}, \bar{u}_{G+1,i,3}$.
- (6) Evaluate objective function values for the trial vectors.
- (7) Select the best of the trial vectors.
- (8) Apply the selection operator according to (6).
- (9) Keep the best solution so far.
- (10) If the maximum number of generations G_{max} is reached, then exit; otherwise increment generation number and go to step 4 for the next generation.

The CODE-EIG flowchart is depicted in Figure 1.

3. Numerical Results

In this paper, we compare the proposed CODE-EIG with three other DE algorithms, PSO, and GAs on two different array design cases. These algorithms are the CODE algorithm [26], the popular DE/rand/1/bin strategy (binomial crossover), the DE/rand/1/eig strategy (eigenvector-based mixed with binomial crossover), the Constriction Factor PSO [28], and a real coded GA. In the DE/rand/1/eig and DE/rand/1/bin strategies, F is set to 0.5 and CR to 0.9. In the PSO, the cognitive learning factor c_1 and the social learning factor c_2 are both set to 2.05. For the GA the crossover probability is set to 0.9 and the mutation probability is set to 0.1. The eigenvector ratio is set to 0.5 for both CODE-EIG and DE/rand/1/eig. For the cases where the algorithms are compared, the population size and the number of generations are set equal for all algorithms. The stopping criterion for all algorithms is the generation number. The same initial conditions are used for all algorithms. All algorithms are compiled using the same compiler (Microsoft Visual C++ 2010) in a PC with Intel Core i5-2410M, at 2.30 GHz with 4 GB RAM running Windows 7.

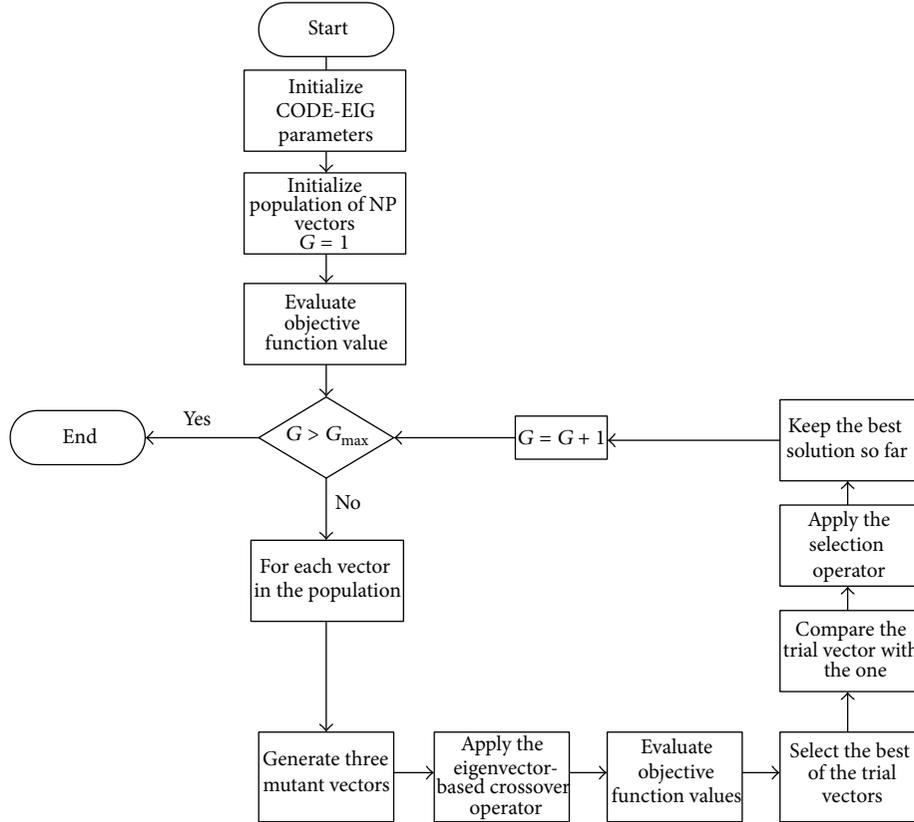


FIGURE 1: CODE-EIG flowchart.

3.1. *Linear Array Synthesis.* We consider an N -element linear array of equally spaced isotropic elements. The array factor is expressed as

$$AF(u, \bar{I}, \bar{\varphi}) = \sum_{n=1}^N I_n e^{j(2\pi/\lambda)ndu + \varphi_n}, \quad (10)$$

where λ is the wavelength, I_n and φ_n are the amplitude and the phase of the n th element, respectively, \bar{I} and $\bar{\varphi}$ are the corresponding vectors, d is the distance between two adjacent elements, $u = \sin(\vartheta)$ is the direction cosine, and ϑ is the steering angle measured from broadside of the array.

For a symmetrically excited array (10) becomes

$$AF(u, \bar{I}, \bar{\varphi}) = 2 \sum_{n=1}^M I_n \cos \left[\frac{2\pi}{\lambda} dnu + \varphi_n \right] + I_0, \quad (11)$$

where $M = \lfloor N/2 \rfloor$ is the largest integer less than or equal to $N/2$. For an even number of elements $I_0 = 0$ and for an odd number of elements we set $I_0 = 1$. We assume that $d = 0.5\lambda$ for all cases.

The shaped beam design cases are chosen such as the desired pattern is a cosecant beam and outside the main beam the peak sidelobe level (PSLL) should not exceed predefined

values [5, 7, 11]. We consider case 1, where the desired pattern is given by the following expression:

$$AF_{\text{desired}}(u) = \begin{cases} \text{cosecant}(u), & \text{for } 0.1 \leq u \leq 0.5, \\ 0, & \text{elsewhere,} \end{cases} \quad (12)$$

$$PSLL_d(u) \leq -30 \text{ dB},$$

where $PSLL_d$ is the desired peak sidelobe level.

One may observe that the $PSLL_d$ is set to -30 dB instead of -25 dB in [5, 11]. The second case uses the same cosecant main beam in the sector of $u \in [0.1, 0.5]$ as in (12), while the $PSLL_d$ is given by [5, 7, 11]

$$PSLL_d(u) = \begin{cases} -35 \text{ dB}, & \text{for } -1 \leq u \leq -0.35, \\ -25 \text{ dB}, & \text{for } -0.35 \leq u \leq 0, \\ -40 \text{ dB}, & \text{for } 0.6 \leq u \leq 1. \end{cases} \quad (13)$$

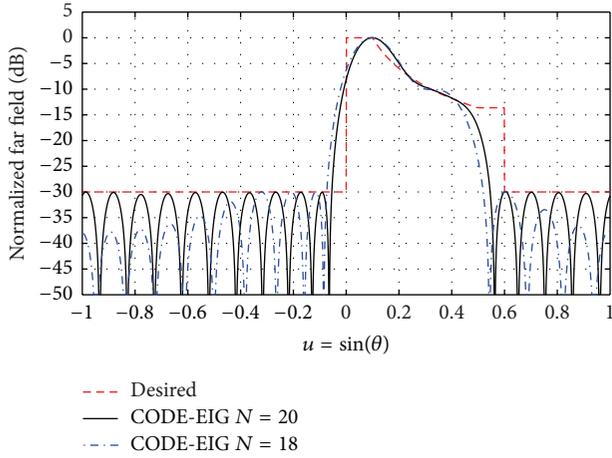


FIGURE 2: Case 1. Radiation patterns obtained for $N = 20$ and $N = 18$ symmetrically excited isotropic elements.

The requirement of matching a desired shaped beam pattern can be expressed as follows:

$$F(\bar{x}) = \sum_{k=1}^K w(u_k) [|AF_{\text{desired}}(u_k)| - |AF_{\text{calc}}(u_k, \bar{x})|]^2, \quad (14)$$

$$w(u) = \begin{cases} 500, & \text{for } 0.1 < u \leq 0.5, \\ 1000, & \text{for } u = 0.1, \\ 300, & \text{if } \text{PSLL}_c(u) > \text{PSLL}_d(u), \\ 1, & \text{otherwise,} \end{cases}$$

where PSLL_c is the calculated peak sidelobe level and \bar{x} is the vector of the unknown variables (the array amplitudes and phases). We have found the above weight values based on a trial-and-error procedure. The population size is set to 100 and the maximum number of generations is set to 1000 for all linear array design cases.

The first example case is that of a symmetrically excited array of 20 and 18 elements. Thus, the total number of unknowns is 20 and 18, respectively. The desired pattern is that of case 1. In this case, we compare the algorithms performance for $N = 20$ elements. The best value, the worst value, the mean, and the standard deviation of the last generation computed by each algorithm are presented here. Figure 2 shows the best radiation patterns obtained by CODE-EIG for this case. The obtained patterns are in good agreement with the desired one. The 20-element array obtained a PSLL value of -30 dB, while the 18-element array obtained a PLL value of -29.96 dB. Table 1 holds the optimum element amplitudes and phases derived for the first two arrays. The convergence rate plot for this case for $N = 20$ is shown in Figure 3. The CODE-EIG strategy converges faster than the other DE algorithms. The DE/rand/1/eig strategy converges faster than the CODE and the DE/rand/1/bin strategy. PSO and GA converge slower than the DE algorithms. Table 2 presents the statistical results for all algorithms. Both CODE-EIG and

TABLE 1: Case 1. Element amplitudes and phases $N = 20$ and $N = 18$ symmetrically excited isotropic elements.

n	I_n	φ_n (deg)	I_n	φ_n (deg)
1	0.909	163.906	0.997	165.307
2	0.762	133.396	0.851	137.129
3	0.535	109.236	0.618	113.545
4	0.348	99.892	0.393	101.486
5	0.288	100.513	0.290	105.145
6	0.279	88.565	0.288	97.232
7	0.226	67.017	0.249	71.910
8	0.144	46.497	0.151	51.300
9	0.073	41.120	0.083	75.256
10	0.132	63.919		

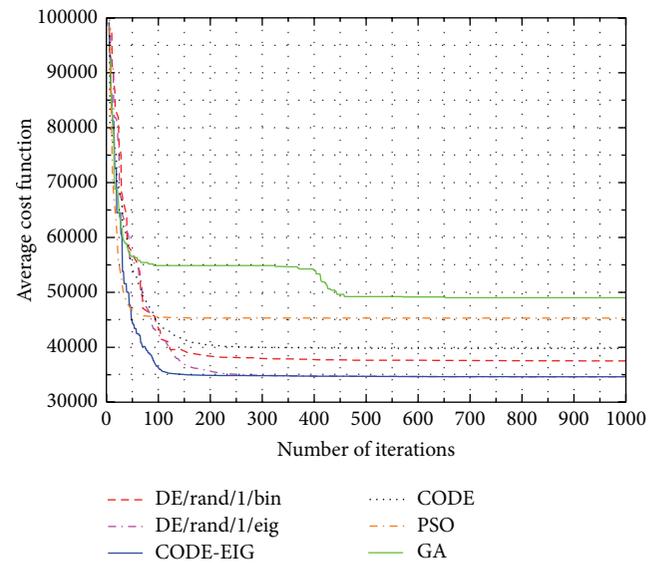


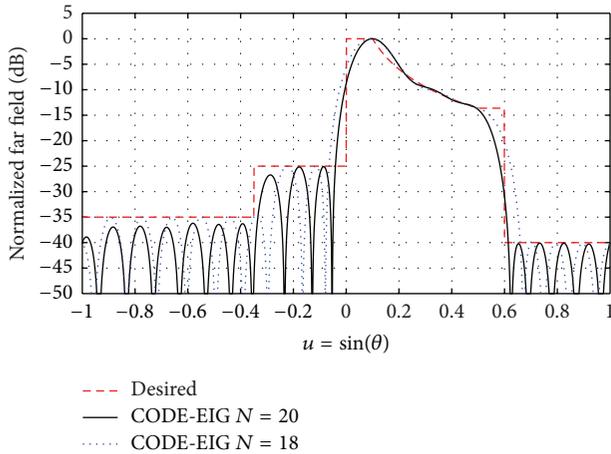
FIGURE 3: Case 1. Convergence rate plot for the 20-element array design case.

DE/rand/1/eig strategy produce about the same best result. However, CODE-EIG is better than DE/rand/1/eig in terms of mean and standard deviation values. We also notice that the DE/rand/1/eig results are better than the CODE and DE/rand/1/bin strategy. The PSO and GA obtained results present higher mean and standard deviation values. The DE/rand/1/bin strategy is the fastest among all the algorithms in terms of computational time. The PSO and GA are also faster than the other DE algorithms.

The second example case is again a symmetrically excited array of 20 and 18 elements. However, the desired pattern is that of case 2. Again, we compare the algorithms performance for $N = 20$ elements. Figure 4 presents the patterns of the best designs obtained for both $N = 20$ and $N = 18$ elements. We notice that the derived patterns closely match the desired one, while the 20-element array is a better approximation than the 18-element one. The maximum SLL of the 20-element case is -25.09 dB while for the 18-element case the SLL value is -24.94 dB. Table 3 presents the optimum amplitudes and phases obtained by the CODE-EIG strategy for both arrays.

TABLE 2: Comparative results for symmetrically excited design Case 1 for $N = 20$. The smaller values are in bold.

Algorithm	Best	Worst	Mean	St. dev.	CPU time (sec)
CODE-EIG	34619.53	37476.33	36030.69	1121.19	239.63
CODE	37416.86	39859.79	38629.17	1190.15	243.36
DE/rand/1/eig	34619.57	37497.31	36866.06	1405.14	217.44
DE/rand/1/bin	37406.04	39856.17	38370.45	1168.07	193.36
PSO	37534.00	54611.18	43211.50	5502.29	206.31
GA	36993.32	48823.57	43758.32	3804.47	202.42

FIGURE 4: Case 2. Radiation patterns obtained for $N = 20$ and $N = 18$ symmetrically excited isotropic elements.TABLE 3: Case 2. Element amplitudes and phases $N = 20$ and $N = 18$ symmetrically excited isotropic elements.

n	I_n	φ_n (deg)	I_n	φ_n (deg)
1	1.000	162.224	0.999	162.997
2	0.811	129.582	0.765	131.770
3	0.535	107.654	0.473	114.275
4	0.360	105.240	0.358	114.863
5	0.340	99.504	0.351	105.281
6	0.314	78.941	0.286	86.582
7	0.239	59.991	0.182	82.091
8	0.169	59.714	0.137	83.111
9	0.142	59.948	0.109	83.064
10	0.111	55.830		

Figure 5 shows the convergence rate plots for the 20-element case. It is obvious that CODE-EIG converges faster than the other algorithms. CODE is the second faster algorithm for this case. All DE algorithms converge at similar final values. PSO and GA converge slower than the other algorithms and a higher final value. The mean values and the standard deviation values are larger for PSO and GA. Table 4 presents the statistical results of the objective function values for both arrays. CODE-EIG clearly outperforms the other algorithms. It has obtained the lower best, mean, and standard deviation values. The DE/rand/1/eig strategy has produced the best worst value and produces better results than the CODE

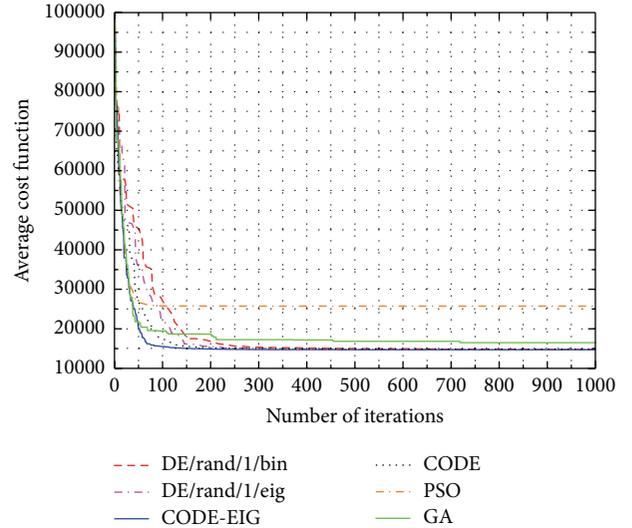


FIGURE 5: Case 2. Convergence rate plot for the 20-element array design case.

and the DE/rand/1/bin. Regarding computational time, again the DE/rand/1/bin is faster than the other algorithms. The CODE-EIG is faster than CODE and DE/rand/1/eig in this case.

Next, the examples that follow design the same two cases but with asymmetrically excited arrays. Therefore, the total number of unknowns is $2N$ for these cases. The third example is that of an asymmetrically excited array with $N = 16, 17,$ and 18 elements. The desired pattern is that of case 1. The best obtained patterns by CODE-EIG are depicted in Figure 6. We notice that the three patterns approximate the desired pattern. The peak SLL values are -30.20 dB, -30 dB, and -29.99 dB for $N = 18, 17,$ and 16 , respectively. The element amplitude and phases for the three patterns are presented in Table 5.

The fourth example presents the design case 2 as the desired pattern obtained by an asymmetrically excited linear array of 20 and 18 isotropic elements. The author has not been able to find in the literature an example that uses this design case asymmetrically excited. The best obtained patterns by CODE-EIG are illustrated in Figure 7. We notice that the 20-element array matches better the desired pattern than the 18-element one. The peak SLL values are -25.01 dB and -24.99 dB for $N = 20$ and $N = 18$ elements, respectively.

TABLE 4: Comparative results for symmetrical excited design Case 2 for $N = 20$. The smaller values are in bold.

Algorithm	Best	Worst	Mean	St. dev.	Mean CPU time (sec)
CODE-EIG	9385.09	14856.80	11553.94	2638.87	212.69
CODE	9502.48	22680.59	15026.70	3014.12	300.09
DE/rand/1/eig	9493.09	14718.99	12051.00	2665.16	226.53
DE/rand/1/bin	14716.18	22681.45	18394.28	3762.84	184.83
PSO	10348.90	28280.86	21872.56	5090.96	188.25
GA	15993.97	28415.97	19632.79	4315.33	190.61

TABLE 5: Case 1. Element amplitudes and phases for $N = 18, 17$, and 15 asymmetrically excited isotropic elements.

n	I_n	φ_n (deg)	I_n	φ_n (deg)	I_n	φ_n (deg)
1	0.136	179.895	0.057	0.774	0.139	179.976
2	0.268	180.000	0	177.597	0.120	179.946
3	0.342	149.887	0.108	179.997	0.167	179.975
4	0.370	137.275	0.104	179.914	0.336	179.994
5	0.455	136.661	0.153	179.702	0.572	161.028
6	0.625	134.342	0.277	179.926	0.853	139.058
7	0.868	114.851	0.549	174.629	0.999	107.863
8	0.990	91.554	0.839	144.046	0.972	81.897
9	0.973	66.089	0.985	116.748	0.871	60.612
10	0.833	41.468	1.000	91.827	0.630	42.012
11	0.560	10.548	0.905	67.285	0.430	37.924
12	0.291	0	0.674	47.620	0.411	39.555
13	0.142	2.974	0.472	39.502	0.386	27.793
14	0.108	0.884	0.355	36.326	0.311	0.003
15	0.065	0	0.354	28.654	0.161	0.004
16	0	0.630	0.241	0.021	0.036	2.615
17	0.048	177.224	0.139	0.099		
18	0.070	135.981				

TABLE 6: Case 2. Element amplitudes and phases for $N = 20$ and 18 asymmetrically excited isotropic elements.

n	I_n	φ_n (deg)	I_n	φ_n (deg)
1	0.155	179.993	0.132	179.993
2	0.297	167.574	0.182	179.93
3	0.356	133.836	0.246	177.114
4	0.227	104.702	0.392	174.602
5	0.081	179.853	0.552	155.925
6	0.460	179.979	0.643	134.614
7	0.809	146.989	0.670	117.929
8	1.000	114.094	0.766	107.713
9	0.980	86.945	0.926	90.237
10	0.844	68.696	0.980	63.686
11	0.757	56.667	0.807	31.301
12	0.698	40.979	0.463	0
13	0.595	20.449	0.120	0.454
14	0.413	5.048	0.226	69.332
15	0.282	0.173	0.341	44.786
16	0.244	3.129	0.305	20.131
17	0.196	0	0.194	0.009
18	0.109	0.070	0.108	0.022
19	0.070	24.731		
20	0.094	25.046		

Table 6 holds the corresponding element amplitudes and phases for both arrays.

3.2. Conformal Array Synthesis. The final design case presents a conformal cylindrical array design. This case has been solved in [4] using simulated annealing. In [7] another design is obtained using linear programming. We assume a cylindrical array consisting of MXN elements in the ϕ and z directions, respectively. For this case $M = N = 8$. The radius R_0 of the cylindrical surface is 5λ , while the cylinder height is $h_0 = 3.5\lambda$. The angular region of the array extends from $-\phi_0$ to ϕ_0 , where $\phi_0 = 20^\circ$. The array elements are uniformly spaced by 0.5λ .

The far-field radiation pattern of the conformal cylindrical array is given by

$$\text{FF}(\vartheta, \phi, \bar{I}, \bar{\varphi}) = \sum_{m=1}^M \sum_{n=1}^N I_{mn} \times \text{el}(\vartheta, \phi) \times e^{j[\varphi_{mn} + ((2\pi/\lambda)(x_{mn} \sin \vartheta \cos \phi + y_{mn} \sin \vartheta \sin \phi + z_{mn} \cos \vartheta))]} \quad (15)$$

where m, n are the element indices in the ϕ and z directions, respectively, and $\text{el}(\vartheta, \phi)$ is the element pattern given by [7]

$$\text{el}(\vartheta, \phi) = \begin{cases} \sin \vartheta \cos \phi, & \text{for } |\phi - \phi_m| \leq \frac{\pi}{2}, \\ 0, & \text{otherwise,} \end{cases} \quad (16)$$

where ϕ_m is the element azimuthal position.

The positions of the array elements are

$$\begin{aligned} x_{mn} &= R_0 \cos \phi_m, \\ y_{mn} &= R_0 \sin \phi_m, \\ z_{mn} &= -\frac{h_0}{2} + n \times 0.5\lambda. \end{aligned} \quad (17)$$

The desired far-field pattern, FF_d , has a cosecant square dependency in the $\phi = 0^\circ$ plane for an elevation angle ranging between $\vartheta_{\text{start}}^{\text{cos}}$ and $\vartheta_{\text{end}}^{\text{cos}}$. The setting of the half-power beamwidth (HPBW) in the $\vartheta = 90^\circ$ plane to a desired value, $\text{HPBW}_d^{\vartheta=90^\circ}$, within a tolerance percentage is also a

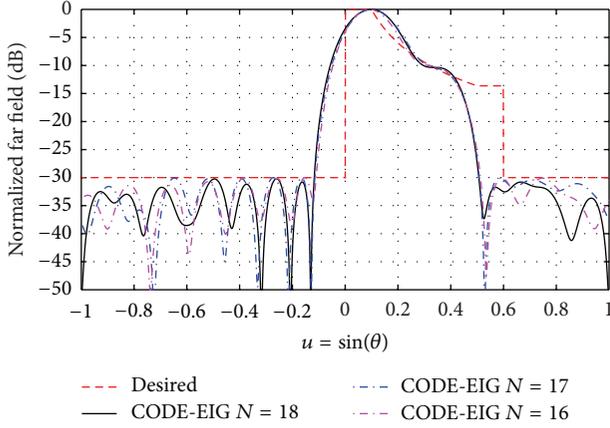


FIGURE 6: Case 1. Radiation patterns obtained for $N = 18, 17$, and 16 asymmetrically excited isotropic elements.

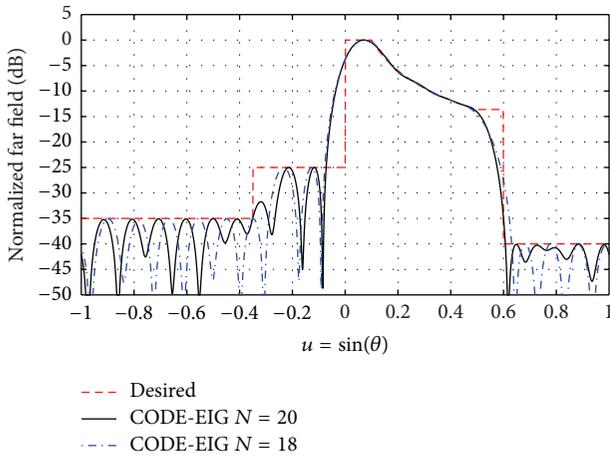


FIGURE 7: Case 6. Radiation patterns obtained for $N = 20$ and 18 asymmetrically excited isotropic elements.

requirement. Additionally, the peak sidelobe levels outside the shaped beam region must remain below a predefined desired value, PSLL_d .

For this case, we express the objective function as

$$\begin{aligned}
 F(\bar{I}, \bar{\varphi}) &= \frac{1}{K_1} \\
 &\cdot \sum_{k=1}^{K_1} \left[\left| \text{FF}_d(\vartheta_k, 0, \bar{I}, \bar{\varphi}) \right| - \left| \text{FF}_c(\vartheta_k, 0, \bar{I}, \bar{\varphi}) \right| \right]^2 + \Xi \\
 &\cdot \max \{0, \text{PSLL}_d - \text{PSLL}_c\} + \Xi \\
 &\cdot \max \left\{ 0, \frac{\left| \text{HPBW}_c^{\vartheta=90^\circ} - \text{HPBW}_d^{\vartheta=90^\circ} \right|}{\text{HPBW}_d^{\vartheta=90^\circ}} - T_{\text{HPBW}} \right\},
 \end{aligned} \tag{18}$$

where FF_c is the calculated far-field pattern in the $\phi = 0^\circ$ plane, K_1 is the number of pattern samples taken for the $\phi = 0^\circ$ plane, PSLL_c is the calculated peak sidelobe level in a set of ϕ -cuts, $\text{HPBW}_c^{\vartheta=90^\circ}$ is the calculated half-power

TABLE 7: Excitation coefficients for the best conformal design obtained by CODE-EIG. The phase values are in degrees.

m	1,8		2,7		3,6		4,5	
n	I_{mn}	φ_{mn}	I_{mn}	φ_{mn}	I_{mn}	φ_{mn}	I_{mn}	φ_{mn}
1	0.27	179.14	0.15	168.91	0.33	96.05	0.28	104.38
2	0.43	-61.11	0.39	163.30	0.61	147.49	0.74	130.96
3	0.57	-116.48	0.51	-161.67	1.06	-157.97	1.08	155.87
4	0.81	-20.42	1.04	-106.93	1.31	-133.54	1.37	-137.05
5	0.24	49.47	0.69	-53.72	0.87	-50.12	1.05	-108.07
6	0.31	-34.05	0.23	-6.12	0.63	-123.44	0.58	-83.90
7	0.26	50.05	0.53	-76.22	0.57	-33.05	0.64	-92.52
8	0.08	37.82	0.25	30.14	0.36	-23.14	0.47	-46.99

TABLE 8: Comparative results for symmetrically excited design Case 1 for $N = 20$ using different eigenvector ratio settings. The smaller values are in bold.

P	Best	Worst	Mean	St. dev.
0.5	34619.53	37476.33	36030.69	1121.19
0.2	34647.83	37471.51	36779.67	1368.40
0.4	34655.49	37502.64	36573.75	1308.63
0.8	34641.35	37528.71	36510.02	1294.57
0.9	34652.01	37527.25	36806.05	1387.19

beamwidth in the $\vartheta = 90^\circ$ plane, and T_{HPBW} is the desired beamwidth tolerance.

The design parameters we have selected for this case are $\vartheta_{\text{start}}^{\text{cos}} = 100^\circ$, $\vartheta_{\text{end}}^{\text{cos}} = 140^\circ$, $\text{HPBW}_d^{\vartheta=90^\circ} = 16^\circ$, $T_{\text{HPBW}} = 0.01$, and $\text{PSLL}_d = -27$ dB (instead of -25 dB in [4, 7]). We set the population size to 100 vectors and the maximum number of iterations is set to 3000. Figures 8(a), 8(b), and 8(c) depict the best obtained radiation pattern with CODE-EIG for the $\phi = 0^\circ$, $\vartheta = 90^\circ$, and a 3D pattern, respectively. One may notice from the $\phi = 0^\circ$ plane pattern that the pattern found has PSLL value of -27.01 dB, which is about 3 dB below the one from [4], while it is about 0.4 dB higher than the one from [7]. Moreover, the $\vartheta = 90^\circ$ plane pattern has a HPBW about 16° and the PSLL value in this pattern is below the corresponding values from [4, 7]. Table 7 holds the amplitudes and phases of the best obtained pattern found by CODE-EIG.

3.3. CODE-EIG Tuning Parameter Sensitivity. The main control parameter of CODE-EIG is the eigenvector ratio P . In order to explore the CODE-EIG sensitivity to control parameter selection we evaluated the first example using different eigenvector ratio settings. The number of elements of the pattern is $N = 20$. We set the number of generations to 1000 and the population size to 100. We run the algorithm for 50 independent trials for each case. The best value, the worst value, the mean, and the standard deviation at the last generation are presented here. Table 8 shows the results. CODE-EIG performance is relatively robust to different P values, as long as P is 0.5 or 0.8. Higher or lower eigenvector ratio values result in slightly worse statistical values. This conclusion is in agreement with [24], where the authors show the eigenvector ratio value of 0.5 produces better results.

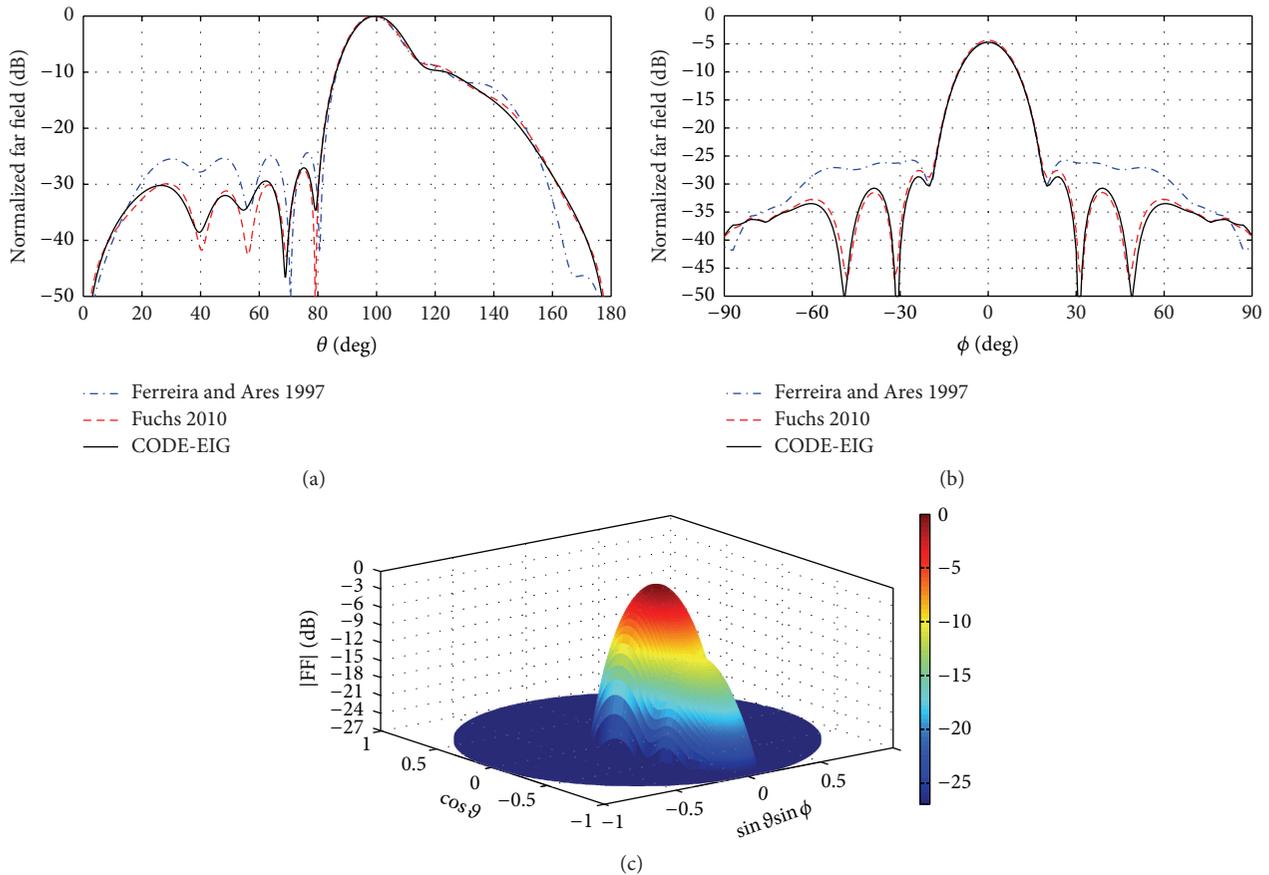


FIGURE 8: Conformal array radiation far-field patterns in (a) $\phi = 0^\circ$ plane, (b) $\theta = 90^\circ$ plane, and (c) 3D pattern.

4. Conclusion

We have presented a novel design approach for shaped beam pattern synthesis with multiple constraints based on Differential Evolution. We have introduced a new DE algorithm, namely, the CODE-EIG, which extends the Composite DE with the eigenvector-based crossover operator. The CODE-EIG algorithm is a powerful and robust global optimizer, which selects the appropriate strategy for trial vector generation and control parameters from a predefined candidate pool. An additional control parameter that is included in CODE-EIG is the eigenvector ratio. Thus, CODE-EIG requires only the adjustment of three parameters: the population size, the number of iterations, and the eigenvector ratio.

One of the DE advantages is that very few control parameters have to be adjusted in each algorithm run. However, the control parameters involved in DE are highly dependent on the optimization problem. Moreover, the selection of the appropriate strategy for trial vector generation requires additional computational time using a trial-and-error search procedure. Therefore, it is not always an easy task to tune these parameters. The CODE-EIG algorithm self-adapts these parameters. Although the CODE-EIG algorithm requires more computational time than the DE/rand/1/bin

strategy, significant time is saved from using the trial-and-error search procedure.

We have designed shaped beam patterns of linear and conformal arrays, which have proven the effectiveness of this approach. The numerical results show that compared with the other DE strategies the CODE-EIG converges faster and produces in average better results. However, as with all evolutionary algorithms there is no assurance for locating the global optimum. In our future work, we plan to study possible modifications to CODE-EIG and apply the algorithm to additional antenna design problems.

Conflict of Interests

The author declares that there is no conflict of interests regarding the publication of this paper.

References

- [1] T. Isernia, F. J. Ares Pena, O. M. Bucci, M. D'Urso, J. E. Gómez, and J. A. Rodríguez, "A hybrid approach for the optimal synthesis of pencil beams through array antennas," *IEEE Transactions on Antennas and Propagation*, vol. 52, no. 11, pp. 2912–2918, 2004.

- [2] N. G. Gómez, J. J. Rodríguez, K. L. Melde, and K. M. McNeill, "Design of low-sidelobe linear arrays with high aperture efficiency and interference nulls," *IEEE Antennas and Wireless Propagation Letters*, vol. 8, pp. 607–610, 2009.
- [3] J. W. Hooker and R. K. Arora, "Optimal thinning levels in linear arrays," *IEEE Antennas and Wireless Propagation Letters*, vol. 9, pp. 771–774, 2010.
- [4] J. A. Ferreira and F. Ares, "Pattern synthesis of conformal arrays by the simulated annealing technique," *Electronics Letters*, vol. 33, no. 14, pp. 1187–1189, 1997.
- [5] A. Akdagli and K. Guney, "Shaped-beam pattern synthesis of equally and unequally spaced linear antenna arrays using a modified tabu search algorithm," *Microwave and Optical Technology Letters*, vol. 36, no. 1, pp. 16–20, 2003.
- [6] M. J. Buckley, "Synthesis of shaped beam antenna patterns using implicitly constrained current elements," *IEEE Transactions on Antennas and Propagation*, vol. 44, no. 2, pp. 192–197, 1996.
- [7] B. Fuchs, "Shaped beam synthesis of arbitrary arrays via linear programming," *IEEE Antennas and Wireless Propagation Letters*, vol. 9, pp. 481–484, 2010.
- [8] T. Isernia, O. M. Bucci, and N. Fiorentino, "Shaped beam antenna synthesis problems: feasibility criteria and new strategies," *Journal of Electromagnetic Waves and Applications*, vol. 12, no. 1, pp. 103–138, 1998.
- [9] S. L. Ho and S. Yang, "Multiobjective synthesis of antenna arrays using a vector tabu search algorithm," *IEEE Antennas and Wireless Propagation Letters*, vol. 8, pp. 947–950, 2009.
- [10] S. L. Ho and S. Yang, "Multiobjective optimization of inverse problems using a vector cross entropy method," *IEEE Transactions on Magnetics*, vol. 48, no. 2, pp. 247–250, 2012.
- [11] A. A. Akdagli, K. Guney, and D. Karaboga, "Touring ant colony optimization algorithm for shaped-beam pattern synthesis of linear antenna," *Electromagnetics*, vol. 26, no. 8, pp. 615–628, 2006.
- [12] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.
- [13] M. M. Khodier and C. G. Christodoulou, "Linear array geometry synthesis with minimum sidelobe level and null control using particle swarm optimization," *IEEE Transactions on Antennas and Propagation*, vol. 53, no. 8, pp. 2674–2679, 2005.
- [14] D. G. Kurup, M. Himdi, and A. Rydberg, "Synthesis of uniform amplitude unequally spaced antenna arrays using the differential evolution algorithm," *IEEE Transactions on Antennas and Propagation*, vol. 51, no. 9, pp. 2210–2217, 2003.
- [15] R. L. Haupt, "Thinned arrays using genetic algorithms," *IEEE Transactions on Antennas and Propagation*, vol. 42, no. 7, pp. 993–999, 1994.
- [16] R. L. Haupt, "Optimized weighting of uniform subarrays of unequal sizes," *IEEE Transactions on Antennas and Propagation*, vol. 55, no. 4, pp. 1207–1210, 2007.
- [17] D. Mandal, A. Chatterjee, and A. K. Bhattacharjee, "Design of fully digital controlled shaped beam synthesis using differential evolution algorithm," *International Journal of Antennas and Propagation*, vol. 2013, Article ID 713680, 9 pages, 2013.
- [18] S. K. Goudos, K. A. Gotsis, K. Siakavara, E. E. Vafiadis, and J. N. Sahalos, "A multi-objective approach to subarrayed linear antenna arrays design based on memetic differential evolution," *IEEE Transactions on Antennas and Propagation*, vol. 61, no. 6, pp. 3042–3052, 2013.
- [19] S. K. Goudos, V. Moysiadou, T. Samaras, K. Siakavara, and J. N. Sahalos, "Application of a comprehensive learning particle swarm optimizer to unequally spaced linear array synthesis with sidelobe level suppression and null control," *IEEE Antennas and Wireless Propagation Letters*, vol. 9, pp. 125–129, 2010.
- [20] S. K. Goudos, K. Siakavara, T. Samaras, E. E. Vafiadis, and J. N. Sahalos, "Self-adaptive differential evolution applied to real-valued antenna and microwave design problems," *IEEE Transactions on Antennas and Propagation*, vol. 59, no. 4, pp. 1286–1298, 2011.
- [21] S. K. Goudos, K. Siakavara, T. Samaras, E. E. Vafiadis, and J. N. Sahalos, "Sparse linear array synthesis with multiple constraints using differential evolution with strategy adaptation," *IEEE Antennas and Wireless Propagation Letters*, vol. 10, pp. 670–673, 2011.
- [22] Y. Wang, Z. Cai, and Q. Zhang, "Differential evolution with composite trial vector generation strategies and control parameters," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 55–66, 2011.
- [23] E. Mezura-Montes, J. Velazquez-Reyes, and C. A. Coello Coello, "A comparative study of differential evolution variants for global optimization," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '06)*, pp. 485–492, Seattle, Wash, USA, 2006.
- [24] S.-M. Guo and C.-C. Yang, "Enhancing differential evolution utilizing eigenvector-based crossover operator," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 1, pp. 31–49, 2015.
- [25] R. Storn, "Differential evolution research—trends and open questions," *Studies in Computational Intelligence*, vol. 143, pp. 1–31, 2008.
- [26] A. W. Iorio and X. Li, "Solving rotated multi-objective optimization problems using differential evolution," in *AI 2004: Advances in Artificial Intelligence*, vol. 3339 of *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science)*, pp. 861–872, Springer, 2004.
- [27] J. Demmel and K. Veselić, "Jacobi's method is more accurate than QR," *SIAM Journal on Matrix Analysis and Applications*, vol. 13, no. 4, pp. 1204–1245, 1992.
- [28] M. Clerc, "The swarm and the queen: towards a deterministic and adaptive particle swarm optimization," in *Proceedings of the 1999 Congress on Evolutionary Computation (CEC '99)*, pp. 1951–1957, IEEE, Washington, DC, USA, July 1999.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

