Progress In Electromagnetics Research, Vol. 137, 371–388, 2013

MODIFIED DIFFERENTIAL EVOLUTION ALGORITHM FOR PATTERN SYNTHESIS OF ANTENNA ARRAYS

Xin Li*, Wentao Li, Xiaowei Shi, Jing Yang, and Jianfeng Yu

National Laboratory of Science and Technology on Antennas and Microwaves, Xidian University, Xi'an, Shanxi 710071, P. R. China

Abstract—A modified differential evolution algorithm (MDE) for pattern synthesis of antenna arrays is proposed in this paper. By employing the novel strategies of best of random mutation and randomized local search, the convergence of standard differential evolution algorithm (SDE) is significantly accelerated. Five standard benchmark functions are optimized to testify the proposed algorithm by comparison with several other optimization algorithms. The numerical results verify the superior performance of the proposed MDE. Furthermore, the MDE is applied to two pattern synthesis examples, including a linear array and a cylindrical conformal array. Experiment results demonstrate that the proposed MDE has better performance than the other optimization methods in both of these two examples, which indicate the proposed algorithm is a competitive optimization algorithm in pattern synthesis.

1. INTRODUCTION

During the past decades, there has been growing interest in the design and application of antenna arrays for remote sensing radar, military and commercial communication systems. The main target of antenna array synthesis is to find appropriate excitation vector and layout of the elements to generate desired radiation pattern. It is therefore a classic optimization problem in electromagnetics. However, the optimization problems in the array synthesis are often highly nonlinear, non-differential, and with multiple extrema. Therefore, local optimization techniques like the conjugate gradient method [1] and alternating projection method [2] which are sensitive to the initial

Received 12 January 2012, Accepted 12 February 2013, Scheduled 23 February 2013
* Corresponding author: Xin Li (xinli@mail.xidian.edu.cn).

variables are difficult or even impossible to meet these challenges. In recent years, population based stochastic methods are proven to be effective for solving these problems. Genetic algorithm (GA) [3, 4], simulated annealing algorithm (SA) [5, 6], particle swarm optimization algorithm (PSO) [7, 8], differential evolution algorithm (DE) [9, 10] and artificial bee colony algorithm (ABC) [11, 12] which belong to these kinds of methods have been widely applied to array synthesis problems. However, these algorithms also present certain drawbacks like complex operation, prematurity convergence or slow convergence speed in some problems.

Proposed by Storn and Price [13], DE is a simple yet powerful population-based stochastic search technique, which is particularly effective at solving multi-variable function optimization problems. It has been successfully applied to array synthesis problems, antenna and filter design problems and some other engineering problems [14–17]. To accelerate the convergence speed of DE, many variants to SDE have been proposed and these improvement measures mainly focus on the parameter settings, evolution strategies (especially mutation strategies), combinations with heuristic or other classic algorithms, etc. Liu and Lampinen [18] devises a fuzzy adaptive DE (FADE) where the control parameters are obtained adaptively according to the population diversity. However, the fuzzy operation of the control parameters is very complex. Qing [19] proposes the dynamic differential evolution (DDE), in which the optimal individual is updated dynamically after each fitness function evaluation and then used in the next mutation equation. However, the risk of premature convergence increases obviously especially in problems containing many local extrema. In [10], an improved differential evolution algorithm (IDE) is proposed by introducing a sub-optimal vector in the mutation operation. This algorithm can accelerate the convergence speed as well as keep the diversity. In [20], the simplified quadratic interpolation is employed to improve the local search ability and reduce the computational overhead of the algorithm. This so called DE-SQI has been proven better than the FADE in several benchmark functions. Nevertheless, the IDE and DE-SQI in complex pattern synthesis applications still suffer from the problem of low convergence speed.

In this paper, a modified differential evolution algorithm (MDE) which utilizes the strategies of best of random mutation and randomized local search is proposed. The advantage of the best of random mutation is that it can make a good balance between convergence speed and population diversity. On the other hand, the randomized local search greatly improves the local search ability, which will be effectively used in the global search process of DE. In order to verify its effectiveness and versatility, the MDE is firstly tested by several standard benchmark functions and then applied in the optimizations of antenna array synthesis.

This paper is organized as follows. The principle of MDE and its optimization results of benchmark functions are described in Section 2. Array synthesis problems are addressed in Section 3 and then the numerical results are compared with some other algorithms. Finally, brief conclusions are discussed in Section 4.

2. MODIFIED DIFFERENTIAL EVOLUTION ALGORITHM

2.1. Standard Differential Evolution Algorithm (SDE)

DE algorithm aims at evolving a population of NP D-dimensional parameter vectors, so-called individuals, towards the global optimum. The individual of population in generation G can be written as:

$$\mathbf{x_{i,G}} = \{x_{1i,G}, x_{2i,G}, \ldots, x_{Di,G}\}, \quad i = 1, 2, \ldots, NP \tag{1}$$

The initial population should cover the entire search space by uniformly randomizing individuals within the search space. The initial value of the jth parameter at the generation $G = 0$ is generated by:

$$\begin{aligned} x_{ji,o} \;=\; & rand(0,1) \cdot \left(x_j^{\text{up}} - x_j^{\text{low}}\right) + x_j^{\text{low}}, \\ & i = 1, 2, \ldots, NP, \quad j = 1, 2, \ldots, D \end{aligned} \tag{2}$$

where $rand(0,1)$ represents an uniformly distributed random variable within the range $[0,1]$; x_j^{low} and x_j^{up} are the lower and upper bounds of the jth parameter. After the initialization of the population, DE algorithm will involve three stages including mutation, crossover and selection in turn.

Mutation operation: DE employs the mutation operation to produce a mutant vector $\mathbf{v_{i,G}}$ with respect to each individual $\mathbf{x_{i,G}}$, so called target vector. The mutant vector can be generated via certain mutation strategy. The two most frequently used mutation strategies in DE are listed as follows:

$$\text{(1) "}DE/rand/1/bin\text{"} \quad \mathbf{v_{i,G}} = \mathbf{x_{r_1,G}} + F \cdot (\mathbf{x_{r_2,G}} - \mathbf{x_{r_3,G}}) \tag{3}$$

$$\text{(2) "}DE/best/1/bin\text{"} \quad \mathbf{v_{i,G}} = \mathbf{x_{best,G}} + F \cdot (\mathbf{x_{r_1,G}} - \mathbf{x_{r_2,G}}) \tag{4}$$

$r_1, r_2, r_3 \in [1, NP]$ and $r_1 \neq r_2 \neq r_3 \neq i$; $\mathbf{x_{best,G}}$ is the vector with the best fitness value in the population at generation G. The scaling factor F is a real and constant factor,and satisfies $F \in [0, 2]$.

Crossover operation: After the mutation operation, a trial vector $\mathbf{u_{i,G}}$ is generated by the target vector $\mathbf{x_{i,G}}$ and its corresponding mutant vector $\mathbf{v_{i,G}}$. The basic crossover strategy is defined as follows:

$$\mathbf{u_{ji,G}} = \begin{cases} \mathbf{v_{ji,G}} & \text{if } rand(0,1) \leq CR \text{ or } j = j_{rand} \\ \mathbf{x_{ji,G}} & \text{otherwise} \end{cases} \tag{5}$$

The crossover factor *CR* is a user-specified constant within the range $[0,1]$, which controls the fraction of parameter values copied from the mutant vector.

Selection: The objective function value of each trial vector $f(\mathbf{u_{i,G}})$ is compared to that of its corresponding target vector $f(\mathbf{x_{i,G}})$, the vector which has smaller fitness function value will remain in the next generation. The selection operation can be expressed as follows:

$$\mathbf{x_{i,G+1}} = \begin{cases} \mathbf{u_{i,G}}, & \text{if } f(\mathbf{u_{i,G}}) < f(\mathbf{x_{i,G}}) \\ \mathbf{x_{i,G}}, & \text{otherwise} \end{cases} \tag{6}$$

The above-mentioned three steps are repeated generation by generation until the specific termination criteria are satisfied.

2.2. Modified Differential Evolution Algorithm (MDE)

To strike a balance between exploration and exploitation and enhance the local search ability of DE, the following modifications are proposed.

2.2.1. Best of Random Mutation

Most evolutionary algorithms including DE try to balance two contradictory aspects: exploration and exploitation. This problem can also be understood by another point of view: the balance between diversity and guidance. The mutation operation in DE could be considered as a kind of local search. The base vectors like $\mathbf{x_{r_1,G}}$ in (3) and $\mathbf{x_{best,G}}$ in (4) act as the center of local search and the scaled vector differences determine the search direction and range around the center. Thus the population diversity and guidance are heavily depended on the selection of base vectors. In SDE, the $DE/rand/1/bin$ is completely random in nature, which helps in preserving the diversity but may lead to a slower convergence. To accelerate the convergence progress, the $DE/best/1/bin$ is introduced. This strategy can provide a faster convergence speed but may lead to loss of diversity, which will result in premature convergence as the search procedure progresses. To strike a relative balance between the convergence and diversity, a novel mutation strategy named best of random mutation [21] is applied here. The mutation equation is as follows:

$$\mathbf{v_{i,G}} = \mathbf{x_{b,G}} + F \cdot (\mathbf{x_{R1,G}} - \mathbf{x_{R2,G}}) \tag{7}$$

where $\mathbf{x_{b,G}}$, $\mathbf{x_{R1,G}}$ and $\mathbf{x_{R2,G}}$ are the individuals which are randomly chosen from the population at generation G, $1 \leq i \neq b \neq R1 \neq R2 \leq NP$. The base vector $\mathbf{x_{b,G}}$ is the best one among the three individuals, which satisfies the condition: $f(\mathbf{x_{b,G}}) \leq \min[f(\mathbf{x_{R1,G}}), f(\mathbf{x_{R2,G}})]$. Being the local elite, the base vector produced by the best of the random mutation provides limited constructive guidance. On the other hand, the randomness of the mutation base vector guarantees its diversity. Therefore, a good balance between exploration and exploitation is achieved, which expedites the convergence and at the same time maintains the population diversity.

2.2.2. Randomized Local Search

Several studies have shown that incorporating some forms of domain knowledge can greatly improve the search capability of evolutionary algorithms [4, 22, 23]. The local search methods like the crossover-based local search and the simplified quartic interpolation are used to explore near the best individual of the population, then the whole fitness function information of the population can be enriched. A novel strategy named randomized local search is proposed in this paper. The randomized local search is a simple and efficient direct search method that does not use numerical or analytic gradients of the objective functions. The proposed randomized local search is defined as follows:

$$x^*_{j,best,G} = x_{j,best,G} + 0.1\left(x^{\max}_{j,G-1} - x^{\min}_{j,G-1}\right) Gauss(0,1) \quad j \in [1, D] \quad (8)$$

where $\mathbf{x_{best,G}}$ is the best individual in generation G; $x_{j,best,G}$ is the jth variables of $\mathbf{x_{best,G}}$; $x^*_{j,best,G}$ is the new trial point; $x^{\min}_{j,G-1}$ and $x^{\max}_{j,G-1}$ are the minimum and maximum value of the jth variables in generation $G-1$, respectively. $Gauss\,(0,1)$ is a random Gaussian variable with zero mean and unitary standard deviation. The random local search is described as follows: $LocalK \leq ceil(\beta \cdot NP)$ is set to avoid the subprogram getting in deadlock, where $ceil(\cdot)$ is the operator that rounds its argument to the nearest integers greater than or equal to it; β is a factor of multiplying factor.

Pseudocode for randomized local search:

Step 0: Find the best and worst individuals in population and their function values: $\mathbf{x_{best,G}}$, , $\mathbf{x_{worst,G}}$, f_b, f_w and set local iteration counter $LocalK = 1$.

Step 1: do

Compute $\mathbf{x^*_{best,G}}$ as stated at formula (8).

Evaluate $f^*_b = f(\mathbf{x^*_{best,G}})$ and set $LocalK = LocalK + 1$.

While $(f^*_b > f_w$ and $LocalK \leq ceil(\beta \cdot NP))$.

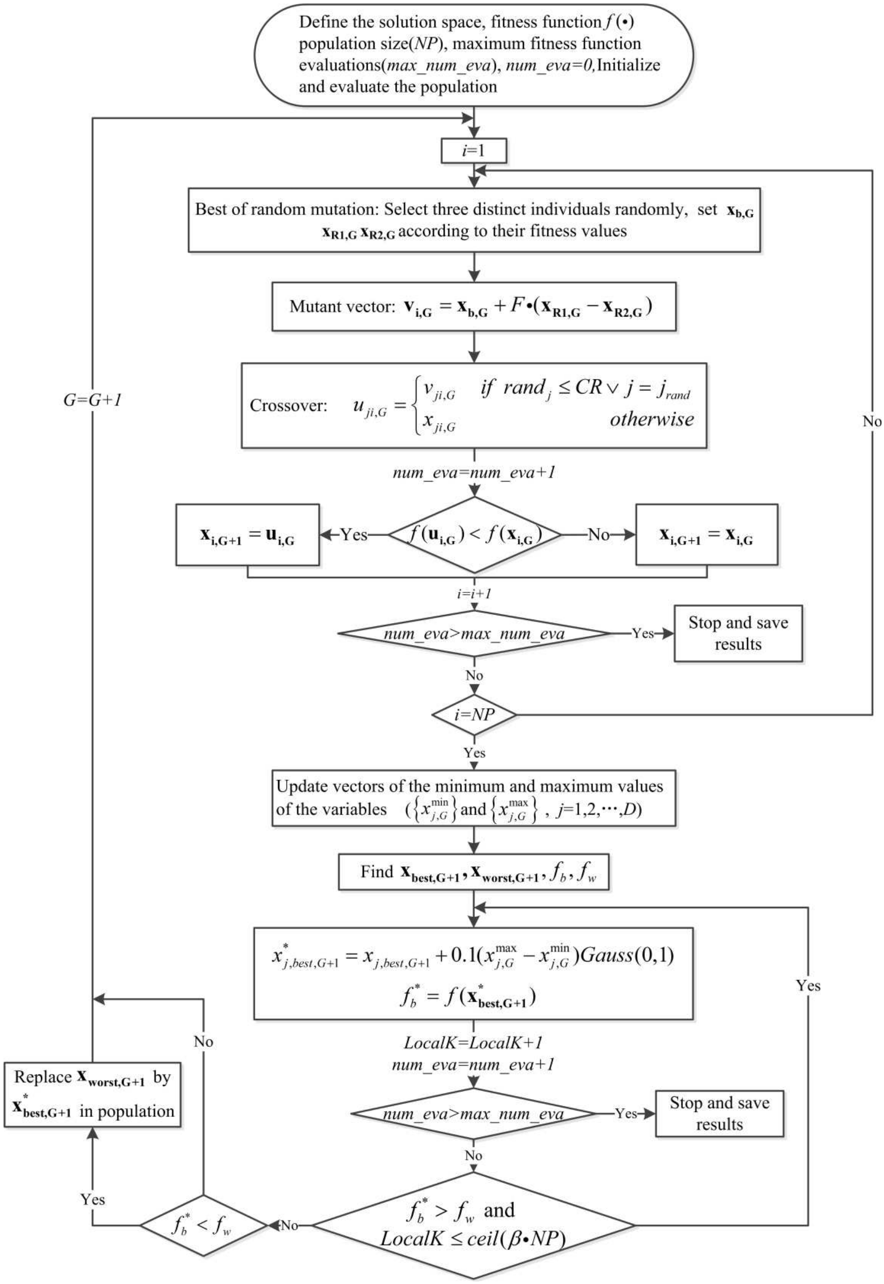

Figure 1. Flowchart of the proposed MDE.

Step 2: If $f_b^* < f_w$, replace $\mathbf{x_{worst,G}}$ by $\mathbf{x^*_{best,G}}$ in the population.

After introducing the two modifications, the flowchart of MDE is presented in Fig. 1.

2.3. Numerical Results of Benchmark Functions

In this section, five benchmark functions [12] including unimodal and multimodal functions are employed to verify the effectiveness of the proposed MDE. These functions, with dimensions of $D = 30$, are executed 50 test runs independently. The initial parameter ranges (IPR) are given in Table 1. In order to demonstrate the superiority of MDE, the performance of MDE is compared with that of standard DE (SDE) [13] and two improved DE algorithms DE-SQI [20] and IDE [10]. Another two stochastic global optimization algorithms PSO [7] and ABC [12] are also employed here for comparison. We set the population size $NP = 150$, maximum number of function evaluations $\max\_num\_eva = 60000$ for all the algorithms. The parameters of MDE, DE-SQI, IDE and SDE are all selected with the same value $F = 0.5$, $CR = 0.9$, which ensures a fair comparison in computation efficiency and solution quality. The parameter β in MDE is set to be 0.05. For PSO, the inertia weight factor ω is set as $\omega_{\max} = 0.9$ and $\omega_{\min} = 0.4$, acceleration constant $c_1 = 2.0$ and $c_2 = 2.0$. We set $limit = NP \cdot D/2$ for ABC.

The average values of the optimal results by all the algorithms are shown in Table 1. The average optimal values found by MDE

Table 1. The average optimal values of all the algorithms.

	PSO	ABC	SDE	DE-SQI
Sphere	$2.03E+02$	$5.72E-05$	$5.80E+00$	$7.98E-02$
Schwefel 2.22	$4.23E+01$	$1.02E-02$	$7.59E+00$	$2.19E-01$
Quartic	$1.14E+00$	$2.14E-01$	$5.56E-02$	$2.35E-02$
Ackley	$1.02E+01$	$3.80E-01$	$1.94E+00$	$1.16E-01$
Griewank	$6.29E+00$	$2.99E-02$	$1.05E+00$	$1.71E-01$
	IDE	MDE	IPR	-
Sphere	$1.06E-03$	**3.64E-07**	$[-100, 100]$	-
Schwefel 2.22	$3.23E-02$	**9.73E-04**	$[-10, 10]$	-
Quartic	$2.75E-02$	**8.35E-03**	$[-1.28, 1.28]$	-
Ackley	$1.09E-02$	**1.93E-04**	$[-32, 32]$	-
Griewank	$1.06E-02$	**1.53E-03**	$[-600, 600]$	-

are smaller in all test cases, which indicates that the proposed MDE exhibits more superior performance than other algorithms.

3. ARRAY PATTERN SYNTHESIS EMPLOYING MDE

The capability and versatility of the proposed MDE algorithm will be assessed by two different array types: an equally spaced linear array and a cylindrical conformal array. Since the evaluations of the cost function tend to dominate the overall computation budget in pattern synthesis problems, a key factor is the number of fitness function evaluations. In this paper, the numbers of fitness function evaluations for different algorithms are the same, which means their synthesis costs are almost equivalent. Thus, the results of synthesized patterns can be used to estimate the capability of different algorithms.

3.1. Synthesis of Linear Array

For a z-direction linear array consisting of K elements, the far field radiation pattern of $FF(\theta)$ is:

$$FF(\theta) = EP(\theta) \cdot \sum_{n=1}^{K} I_n \exp(jkz_n \cos(\theta) + j\psi_n) \tag{9}$$

where $EP(\theta)$ is the element pattern, and $EP(\theta) = \sqrt{\sin(\theta)}$ is used to represent the pattern of a directive array element approximately. I_n, ψ_n and z_n are the excitation amplitude, phase and position of the nth element respectively. To suppress the interferences at certain directions, a broadside pattern with two symmetric nulls located at $40 \sim 45^\circ$ and $135 \sim 140^\circ$ is expected. 32 directive elements are half-wavelength spaced in the linear array. The desired sidelobe level is set as DSLL $= -25\,\text{dB}$ and the desired null depth level DNULL $= -45\,\text{dB}$. Here we only synthesize the excitation amplitudes. Because of the symmetrical distribution, only 16 amplitudes are needed to be optimized.

Since the pattern synthesis processes a linear array with deep notches to suppress the interferences, the objective function to be minimized is defined as the least mean squares of the excess far field magnitude above the specified level, which can be written as:

$$\text{fitness1} = \left(\frac{1}{M_s} \sum_{m=1}^{M_s} \left[U(e_d(\theta)) \times e_d(\theta)^2 \right] \right)^{\frac{1}{2}} \tag{10}$$

$$e_d(\theta) = F_o(\theta) - F_d(\theta) \tag{11}$$

where $F_o(\theta)$ and $F_d(\theta)$ are normalized patterns obtained by optimization and the desired pattern, respectively. M_s is the total sample points number, and $U(t)$ is the unit step function.

For design specifications, the population size and the number of fitness function evaluations of all algorithms are set as $NP = 50$ and $\max\_num\_eva = 5000$, respectively. The rest of parameters in the optimization process are selected the same as them in the previous benchmark functions test. After 50 independent experiments, the average convergence curves are plotted in Fig. 2. Obviously, the convergence speed of MDE is much faster than the other algorithms. The normalized radiation patterns of the median run by all the algorithms at the 50 experiments will be given. Because the PSO and ABC are a little inefficient in this example, only the patterns

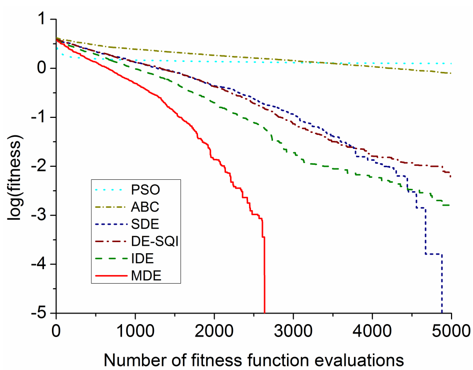

Figure 2. Fitness function value versus the number of fitness function evaluations.

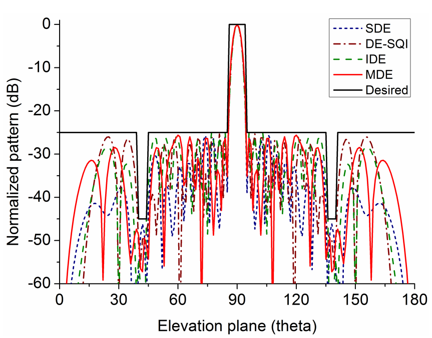

Figure 3. Radiation patterns for the 32-element linear array.

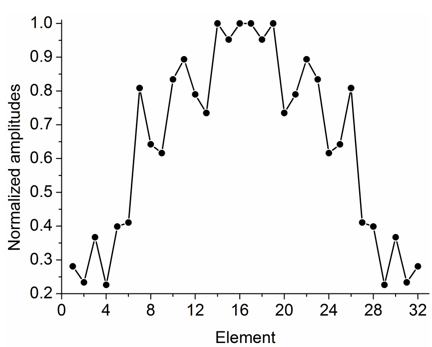

Figure 4. Amplitudes optimized by MDE for this linear array.

by the best performed four algorithms are shown in Fig. 3. From this figure, we can see only the results of SDE and MDE satisfy the requirements. Though the maximum sidelobe levels (MSLL) of them are both about $-25.6\,\mathrm{dB}$, the required number of fitness function evaluations for MDE for the desired pattern is about one half of that for SDE. The corresponding optimized amplitudes by MDE are shown in Fig. 4. Therefore, we can conclude that the proposed MDE is much more efficient in the optimization problems of linear array than other algorithms.

3.2. Synthesis of Cylindrical Conformal Array

3.2.1. Array Element Design

With a center frequency of 3.0 GHz and a bandwidth exceeding 13% (return loss $\leq -10\,\mathrm{dB}$) when in isolation, a compact stacked microstrip antenna is designed as the element of the cylindrical conformal array. Its geometry is given in Fig. 5, which is fabricated on two layers with relative permittivity of 2.65. This antenna, with an overall dimensions of $45 \times 45\,\mathrm{mm}^2$, utilizes a one order quasi-Minkowski fractal patch on the top substrate to reduce the patch size, which also reduces the coupling between adjacent elements.

3.2.2. Conformal Array Configurations

As shown in Fig. 6, an array of $M \times N$ elements are located over a cylindrical surface of radius R, where M and N are the number of elements in z-direction and φ-direction, respectively.

The far-field radiation pattern produced by this $M \times N$ elements cylindrical array can be expressed as:

$$\begin{aligned} FF(\theta,\varphi) \;=\; & \sum_{n=1}^{N}\sum_{m=1}^{M} I_{mn} EP_{mn}(\theta,\varphi)\exp[jk[R\sin\theta\cos(\varphi-\varphi_{mn}) \\ & +z_{mn}\cos\theta]+j\psi_{mn}] \end{aligned} \tag{12}$$

where I_{mn} represents the element excitation current amplitude, ψ_{mn} the excitation current phase, $EP_{mn}(\theta,\varphi)$ the pattern of array element, k the wave number in the free-space, and z_{mn}, φ_{mn} are the individual coordinates in z-direction and φ-direction, respectively. The excitation current phase ψ_{mn} can be calculated by:

$$\psi_{mn} = -k[R\sin\theta_0\cos(\varphi_0-\varphi_{mn}) + z_{mn}\cos\theta_0] \tag{13}$$

where (θ_0,φ_0) is the desired steering angle. It should be noted that the element pattern $EP_{mn}(\theta,\varphi)$ is the active-element-pattern [24] exported

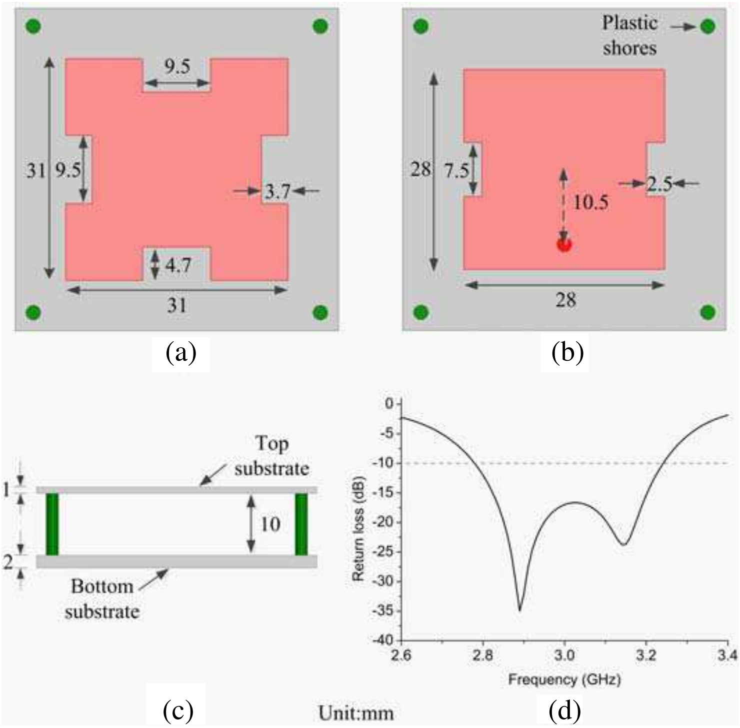

Figure 5. Configurations of the stacked microstrip antenna. (a) Top view of top layer. (b) Top view of bottom layer. (c) Side view. (d) Simulated return loss of this antenna.

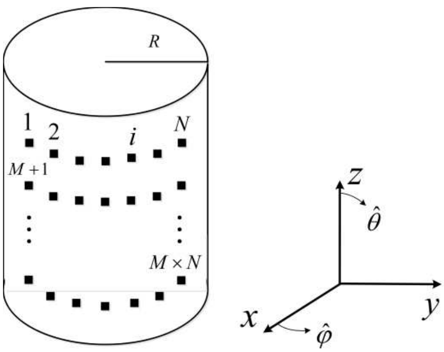

Figure 6. Configurations of the cylindrical conformal antenna array.

from the full-wave simulation software Ansoft HFSSv12 instead of the approximate element pattern used in the previous example. The active-element-pattern, which takes the mutual coupling produced by the conformal carrier and adjacent elements into consideration, can greatly reduce the error between computation and realization. The key parameters of the array dimensions are given as follows: $M = 8$, $N = 12$ and $R = 5.73\lambda_0$; all the elements are uniformly spaced by $\lambda_0/2$ in z-direction and φ-direction. Thus, the angular spacing between consecutive elements in the same row is fixed to be 5° in the φ-direction.

3.2.3. Pattern Synthesis

Our goal is to employ the MDE to synthesize an array pattern with minimum MSLL when the mainbeam scans to different directions. For conformal arrays, it is common to select the excitation phase to focus the beam in the desired direction. With the phase excitation calculated by (13), the specified scan angle can be guaranteed. Thus we only take the amplitude weights as optimization variables. Considering that the optimization for excitation amplitude of each element on this cylindrical array may be a formidable task in engineering practice, a modified Bernstein polynomial for arc arrays is extended here to reduce the number of optimization parameters. The modified Bernstein polynomial [25] is defined as follows:

$$F(u)=\begin{cases} C_0+\frac{1-C_0}{A^{N_0A}(1-A)^{N_0(1-A)}}\cdot u^{N_0A}(1-u)^{N_0(1-A)}, & 0\leq u\leq A \\ C_1+\frac{1-C_1}{A^{N_1A}(1-A)^{N_1(1-A)}}\cdot u^{N_1A}(1-u)^{N_1(1-A)}, & A\leq u\leq 1 \end{cases} \tag{14}$$

where C_0, C_1, N_0, N_1, A are parameters in the polynomial, $u = (i-1)/(n-1)$, $i = 1, 2, \ldots, n$, and n is the number of sampling points. By its definition, the modified Bernstein polynomial shows a smooth distribution which is preferable in the array pattern synthesis. The amplitude weights can be obtained by sampling from the modified Bernstein polynomial illustrated in (14). Therefore, only five variables need to be optimized for each row of this cylindrical conformal array. The differences among the maximum amplitudes of each row are also obtained by a modified Bernstein polynomial. That is to say, for the proposed cylindrical conformal array with of 8 rows and 12 columns, the total number of variables to be optimized is reduced from 8×12 to $5\times(8+1)$, which significantly reduces the overhead of the optimization.

To achieve the lowest MSLL when the desired steering angle is (θ_0, φ_0) in 3D case, the optimization problem is formulated as the following problems:

$$\text{fitness2} = \max_{(\theta,\varphi)\in S} \left| \frac{FF(\theta,\varphi)}{FF_{\max}} \right| \tag{15}$$

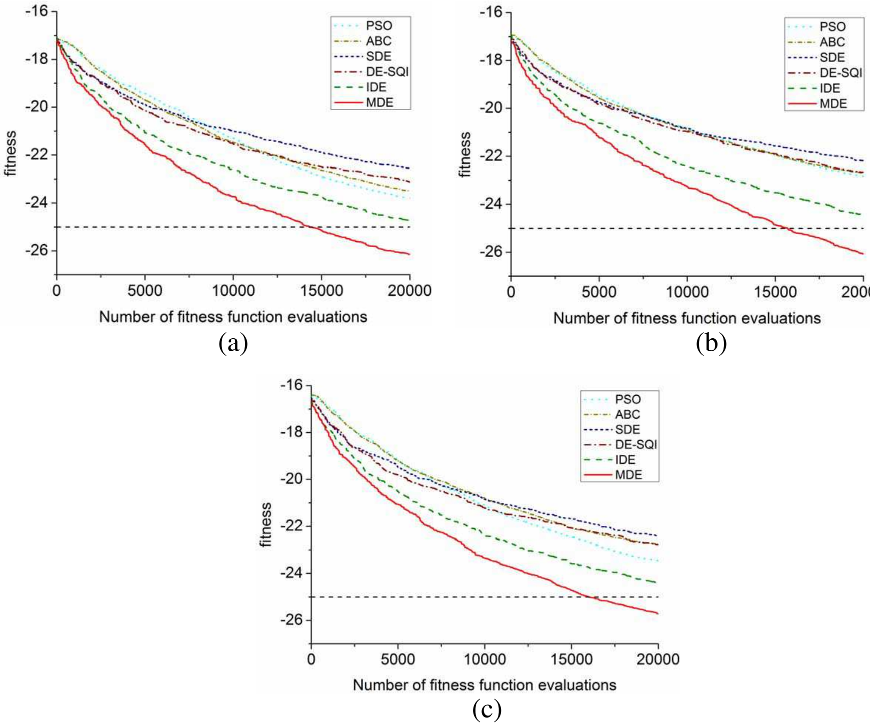

Figure 7. Fitness value versus the number of fitness function evaluations. (a) Mainbeam scan to $(90°, 0°)$. (b) Mainbeam scan to $(90°, 30°)$. (c) Mainbeam scan to $(105°, -20°)$.

where S denotes the sidelobe region; $FF_{\max}$ is the peak value of mainbeam.

By virtue of using the modified Bernstein polynomial, only 45 variables are optimized for the pattern synthesis of the entire cylindrical array. The desired MSLL is lower than -25 dB when the mainbeam scans to $(90°, 0°)$, $(90°, 30°)$ and $(105°, -20°)$. We make the population size $NP = 200$ and the number of fitness function evaluations $\max\_num\_eva = 20000$. The other parameters are selected the same as them in previous examples. All algorithms are conducted 50 times independently. Fig. 7 shows the average fitness curves. Obviously, the proposed MDE possesses the dominant speed and needs the fewest fitness function evaluations among the algorithms.

The normalized radiation pattern of the elevation and azimuth planes of the median run in the 50 independent experiments by all the algorithms are given in Fig. 8. The obtained average MSLL of

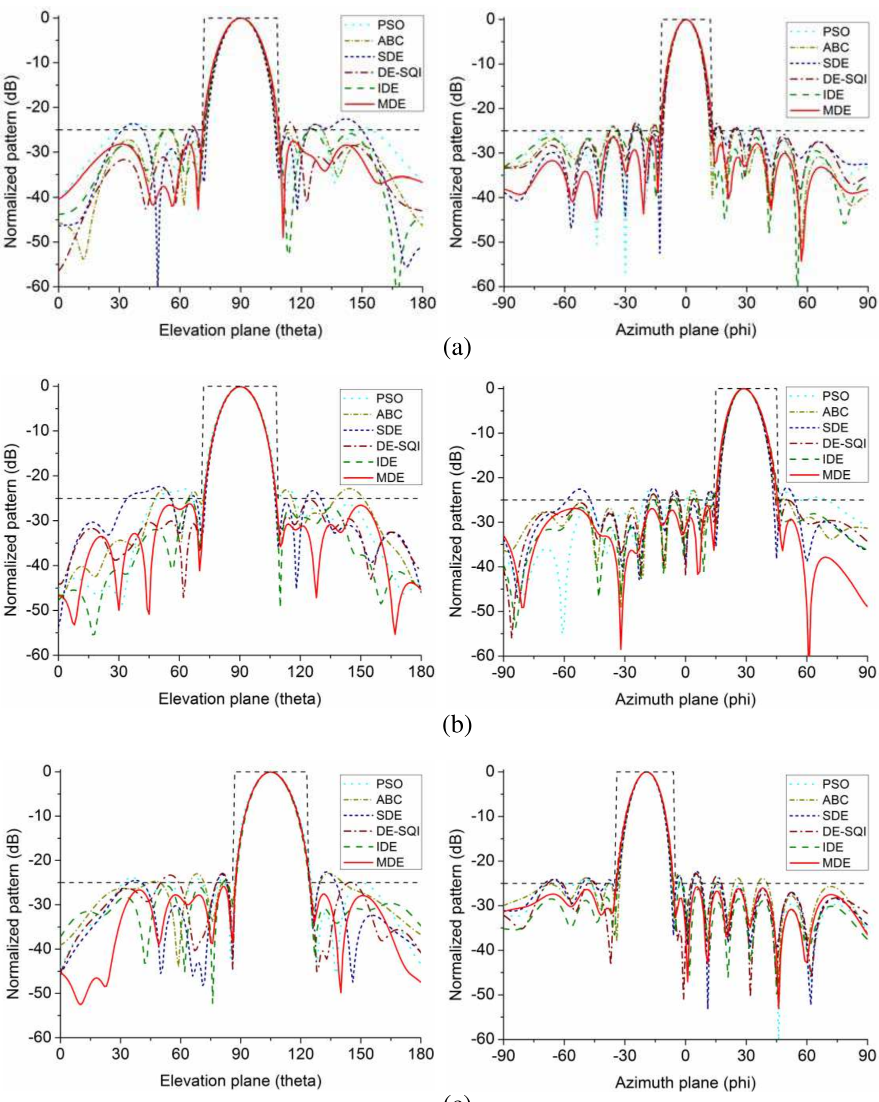

Figure 8. Normalized patterns of the cylindrical array. (a) Mainbeam scan to (90°, 0°). (b) Mainbeam scan to (90°, 30°). (c) Mainbeam scan to (105°, −20°).

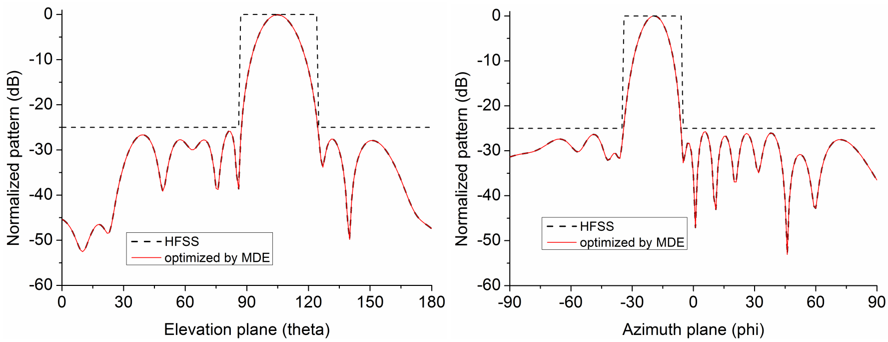

Figure 9. Comparison of the optimized array pattern by MDE and the pattern simulated via HFSSv12.

Table 2. Performance comparisons of different algorithms.

Scan angle	MSLL (dB)					
	PSO	ABC	SDE	DE-SQI	IDE	MDE
(90, 0)	−23.80	−23.50	−22.54	−23.12	−24.73	**−26.15**
(90, 30)	−22.83	−22.71	−22.17	−22.66	−24.44	**−26.06**
(105, −20)	−23.45	−22.76	−22.40	−22.77	−24.40	**−25.71**

the six algorithms are illustrated in Table 2. It is obvious that the MDE can accurately produce the low sidelobe pattern while the other five kinds of algorithms cannot satisfy the design specification, which evidently shows the superior performance of MDE. Fig. 9 depicts the comparison between the radiation pattern obtained by MDE and the full-wave simulation results from Ansoft HFSSv12. Because of limited space, only the result of mainbeam scanning to $(105^\circ, -20^\circ)$ is shown here. Good agreement between those kinds of results validates our method of conformal array synthesis.

4. CONCLUSIONS

In this paper, MDE with novel strategies of best of random mutation and randomized local search is proposed. By taking the advantages of these two strategies, the search capability of the algorithm is greatly improved as well as a relative balance between guidance and diversity is achieved. The proposed algorithm is verified by both classical benchmark functions and pattern synthesis problems of linear and conformal arrays. These experimental results demonstrate that the proposed MDE is able to achieve the optimum design for specified design criteria in an effective manner. The accuracy and the robustness of the proposed MDE indicate that it is a competitive algorithm in function optimization, pattern synthesis and other electromagnetic fields.

ACKNOWLEDGMENT

The authors would like to thank the financial support from the National Natural Science Foundation of China (No. 61101069 and 61072021) and the Fundamental Research Funds for the Central Universities (No. K50510010016).

REFERENCES

1. Sarkar, T. K. and N. Sangruji, "An adaptive nulling system for a narrow-band signal with a look-direction constraint utilizing the conjugate gradient method," *IEEE Trans. Antennas Propagat.*, Vol. 37, 940–944, 1989.
2. Bucci, O. M., G. D'Elia, and G. Romito, "Power synthesis of conformal arrays by a generalized projection method," *IEE Proc. — Microw. Antennas Propag.*, Vol. 142, 467–471, 1995.
3. Mahanti, G. K., N. Pathak, and P. Mahanti, "Synthesis of thinned linear antenna arrays with fixed sidelobe level using real-coded genetic algorithm," *Progress In Electromagnetics Research*, Vol. 75, 319–28, 2007.
4. Xu, Z., H. Li, Q. Z. Liu, and J. Y. Li, "Pattern synthesis of conformal antenna array by the hybrid genetic algorithm," *Progress In Electromagnetics Research*, Vol. 79, 75–90, 2008.
5. Lee, K.-C., "Frequency-domain analysis of nonlinearly loaded antenna arrays using simulated annealing algorithms," *Progress In Electromagnetics Research,* Vol. 53, 271–281, 2005.
6. Xie, P., K. S. Chen, and Z. S. He, "Synthesis of sparse cylindrical arrays using simulated annealing algorithm," *Progress In Electromagnetics Research Letters*, Vol. 9, 147–156, 2009.
7. Khodier, M. M. and C. G. Christodoulou, "Linear array geometry synthesis with minimum sidelobe level and null control using particle swarm optimization," *IEEE Trans. Antennas Propagat.*, Vol. 53, 2674–2679, 2005.
8. Wang, W. B., Q. Feng, and D. Liu, "Application of chaotic particle swarm optimization algorithm to pattern synthesis of antenna

arrays," *Progress In Electromagnetics Research*, Vol. 115, 173–189, 2011.

9. Dib, N. I., S. K. Goudos, and H. Muhsen, "Application of Taguchi's optimization method and self-adaptive differential evolution to the synthesis of linear antenna arrays," *Progress In Electromagnetics Research,* Vol. 102, 159–180, 2010.
10. Li, R., L. Xu, X. W. Shi, and Z. Q. Lv, "Improved differential evolution strategy for antenna array pattern synthesis problems," *Progress In Electromagnetics Research,* Vol. 113, 429–441, 2011.
11. Basu, B. and G. K. Mahanti, "Fire fly and artificial bees colony algorithm for synthesis of scanned and broadside linear array antenna," *Progress In Electromagnetics Research B*, Vol. 32, 169–190, 2011.
12. Karaboga, D. and B. Akay, "A comparative study of artificial bee colony algorithm," *Applied Mathematics and Computation*, Vol. 214, 108–132, 2009.
13. Storn, R. and K. Price, "Differential evolution — A simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, Vol. 11, 341–359, 1997.
14. Mandal, A., H. Zafar, S. Das, and A. V. Vasilakos, "A modified differential evolution algorithm for shaped beam linear array antenna design," *Progress In Electromagnetics Research*, Vol. 125, 439–157, 2012.
15. Chowdhury, A., A. Ghosh, R. Giri, and S. Das, "Optimization of antenna configuration with a fitness-adaptive differential evolution algorithm," *Progress In Electromagnetics Research B*, Vol. 26, 291–319, 2010.
16. Goudos, S. K., Z. D. Zaharis, and T. V. Yioultsis, "Application of a differential evolution algorithm with strategy adaptation to the design of multi-band microwave filters for wireless communications," *Progress In Electromagnetics Research*, Vol. 109, 123–137, 2010.
17. Xie, L. and Y. C. Jiao, "Design of an ultra-wideband antenna using boolean differential evolution algorithm," *Progress In Electromagnetics Research Letters,* Vol. 26, 135–141, 2011.
18. Liu, J. and J. Lampinen, "A fuzzy adaptive differential evolution algorithm. soft computing — A fusion of foundations," *Methodologies and Applications*, Vol. 9, 448–462, 2005.
19. Qing, A., "Dynamic differential evolution strategy and applications in electromagnetic inverse scattering problems," *IEEE Trans. Geoscience and Remote Sensing*, Vol. 44, 116–125, 2006.

20. Zhang, L., Y. C. Jiao, H. Li, and F. S. Zhang, "Hybrid differential evolution and the simplified quadratic interpolation for global optimization," *Proceedings of the First ACM/SIGEVO Summit on Genetic and Evolutionary Computation*, 1049–1052, 2009.
21. Lin, C., A. Qing, and Q. Feng, "Synthesis of unequally spaced antenna arrays by using differential evolution," *IEEE Trans. Antennas Propagat.*, Vol. 58, 2553–2561, 2010.
22. Ishibuchi, H., T. Yoshida, and T. Murata, "Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling," *IEEE Tran. Evolutionary Computation*, Vol. 7, 204–223, 2003.
23. Noman, N. and H. Iba, "Accelerating differential evolution using an adaptive local search. evolutionary computation," *IEEE Tran. Evolutionary Computation*, Vol. 12, 107–125, 2008.
24. He, Q. Q., B. Z. Wang, and W. Shao, "Radiation pattern calculation for arbitrary conformal arrays that include mutual-coupling effects," *IEEE Trans. Antennas Propagt. Mag.*, Vol. 52, 57–63, 2010.
25. Boeringer, D. W. and D. H. Werner, "Efficiency-constrained particle swarm optimization of a modified Bernstein polynomial for conformal array excitation amplitude synthesis," *IEEE Trans. Antennas Propagat.*, Vol. 53, 2662–2673, 2005.