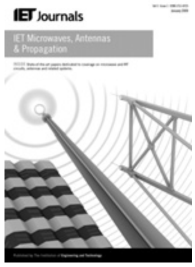
Published in IET Microwaves, Antennas & Propagation
Received on 20th December 2009
Revised on 18th May 2010
doi: 10.1049/iet-map.2009.0630



ISSN 1751-8725

Design of planar thinned arrays using a Boolean differential evolution algorithm

L. Zhang Y.-C. Jiao Z.-B. Weng F.-S. Zhang

National Key Laboratory of Antennas and Microwave Technology, Xidian University, P.O. Box 377, Xi'an, Shaanxi, 710071, People's Republic of China
E-mail: lizhang@mail.xidian.edu.cn

Abstract: As a very powerful optimisation approach, differential evolution algorithm for continuous optimisation problems has been applied to electromagnetic (EM) design problems. However, the optimisation of a thinned array can be formulated as a discrete-variable optimisation problem with solutions encoded as binary strings. Here, a Boolean differential evolution (BDE) algorithm for 0–1 integer programming problems is proposed to design planar thinned arrays with minimum sidelobe levels. The BDE algorithm with only one control parameter is easily implemented. Meanwhile, a fast Fourier transform is employed to speed up the calculation of the pattern. Numerical experiments show that the BDE algorithm is an effective technique.

1 Introduction

Thinned arrays have been of interest for a long time [1–3]. Array thinning involves the removal (turning off) of radiating elements from an array antenna. Active elements are fed with equal amplitude currents, while the remaining ones are turned off. The main motivation to thinning is the reductions in cost, weight and power consumption. This technique allows the realisation of a main beamwidth as almost identical to that for a filled array of equal size. Another advantage is that when the turned-on elements operate with equal amplitude, lower sidelobes can be obtained as for the identical filled array illuminated with uniform weighting. However, the synthesis problem is complex and cannot be solved by analytical methods. Therefore, several global optimisation algorithms, such as genetic algorithm (GA) [4], ant colony optimisation [5] and particle swarm optimisation (PSO) [6], have been introduced to design thinned arrays.

Differential evolution (DE), introduced by Price and Storn [7], is a simple yet powerful optimisation approach for searching continuous-valued spaces. It has been applied to different EM problems [8–12] and has shown excellent performance. However, the nature of reproduction operators in conventional DE limits its application. In fact, the optimisation of a thinned array can be formulated as a discrete-variable optimisation problem with solutions encoded as binary strings. Engelbrecht and Pampará [13] introduced three DE-based methods for generating binary strings, in which the angle-modulated DE [13] shows better performance than others. Nevertheless, when long binary strings are needed, the disadvantage of the amplitude-modulated method is that blocks of bits with the same binary values 0 or 1 are produced. Recently, Greenwood [14] proposed a DE algorithm for solving binary-valued optimisation problems without any modifications to the basic DE. This method obtained significantly high-quality solutions for a subclass of NP-complete graph problems.

The application of the binary version of DE for antenna design has not yet been addressed. The main motivation of this paper is to introduce a new evolutionary computation technique for thinned array design. In this paper, a Boolean differential evolution (BDE) has been introduced for optimising planar thinned arrays with minimum sidelobe levels (SLLs). The BDE is exclusively based on Boolean algebra and it differs entirely from the binary DE [14]. It adopts binary code so that the optimal variables can be directly optimised without being mapped from decimal to binary. Therefore, it is suitable for solving 0–1 integer programming problems. However, the optimal variables of the binary DE [14] are denoted by continuously valued vectors. For binary optimisation problems, the optimal variables are mapped from decimal to binary. Two entirely different philosophy DE-based algorithms (BDE and the binary DE [14]) for solving binary optimisation problems are applied for planar thinned array synthesis. Meanwhile, due to the element-by-element superposition, the calculation of the synthesised pattern is very time-consuming, with fast Fourier transform (FFT) being employed to speed up the calculation of the fitness function. The BDE method with only one control parameter is easily implemented and it is suitable for solving binary optimisation problems. Numerical experiments indicate that the BDE algorithm is an effective technique for thinned arrays synthesis.

2 Boolean differential evolution

Conventional DE has been applied for numerical optimisation [7] and EM problems [8–12]. BDE is a new evolutionary computation technique implemented by means of Boolean algebra to meet the requirements of a real-valued DE algorithm. BDE borrows concepts from binary PSO [15]. It has only one control parameter, the crossover rate (CR). The control parameter is easily chosen and a little time is spent. Assume that $\boldsymbol{X}_{i,G} = (X_{1i,G}, X_{2i,G}, \ldots, X_{Di,G})$, $(i = 1, 2, \ldots, \mathrm{NP})$ is a solution vector in generation G, which is represented by D-bit binary strings, where NP is the population size and D is the problem dimension. For BDE, the mutation, crossover and selection operators are defined as follows.

In the mutation phase, a mutant vector $\boldsymbol{V}_{i,G}$ is generated according to the following equation

$$\boldsymbol{V}_{i,G} = \boldsymbol{X}_{\mathrm{best},G} + F \bullet (\boldsymbol{X}_{r1,G} \oplus \boldsymbol{X}_{r2,G}) \tag{1}$$

where $r1$ and $r2$ are randomly selected from $\{1, 2, \ldots, \mathrm{NP}\}$ such that $r1 \neq r2 \neq i$, F is a random D-bit binary string, which is not a control parameter in contrast to the classic DE algorithm and X_{best}, G is the best individual with the best fitness value in the population at generation G. 'AND', 'OR' and 'XOR' operators denoted by symbols $(\bullet)$, $(+)$ and $(\oplus)$, respectively, are used to carry out mutation. The schematic diagram for procedures of the mutation operator is depicted in Fig. 1.

BDE utilises the crossover operation to generate new solutions by shuffling trial vectors and also to increase the diversity of the population. A trial vector $\boldsymbol{U}_{i,G} = (U_{1i,G}, U_{2i,G}, \ldots, U_{Di,G})$ is calculated according to

$$U_{ji,G} = \begin{cases} V_{ji,G}, & \text{if } \mathrm{rand}_j(0, 1) \leq \mathrm{CR} \text{ or } j = k \\ X_{ji,G}, & \text{otherwise} \end{cases} \tag{2}$$

where $j = 1, 2, \ldots, D$. $\mathrm{CR} \in [0, 1]$ is the crossover rate, which is selected by the user. Hence, CR is the only control parameter for a Boolean DE algorithm. $\mathrm{rand}_j(0, 1)$ is the jth evaluation of a uniformly random number

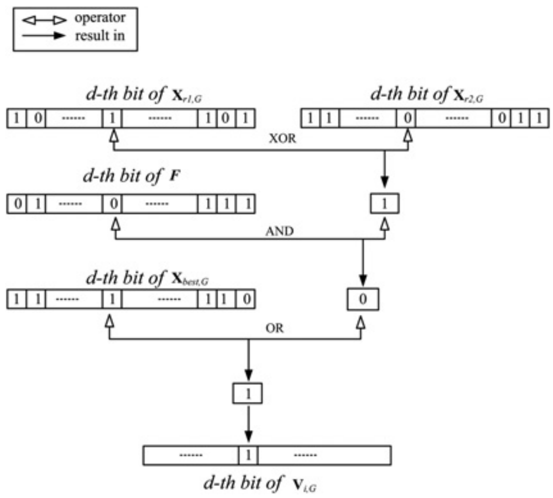


Figure 1 *Procedure of the mutation operator in the BDE*

generator. $k \in \{1, 2, \ldots, D\}$ is a randomly chosen integer, ensuring that $U_{ji,G}$ gets at least one parameter from $V_{ji,G}$.

To decide whether or not it should become a member of the next generation, each child competes with its parent, and survives only if its fitness value is better than that of its parent. The selection procedure can be mathematically written as

$$\boldsymbol{X}_{i,G+1} = \begin{cases} \boldsymbol{U}_{i,G}, & \text{if } f(\boldsymbol{U}_{i,G}) \leq f(\boldsymbol{X}_{i,G}) \\ \boldsymbol{X}_{i,G}, & \text{otherwise} \end{cases} \tag{3}$$

The reproduction (mutation, crossover and selection) cycle iterates until a stopping condition is met.

Parameter settings of the BDE algorithm are as follows. Throughout this paper, population size $\mathrm{NP} = 50$, $\mathrm{CR} = 0.2$ and the maximum number of generations is set to 300 ($G_{\max} = 300$). For the binary DE [14], amplification factor $F = 0.1$ and $\mathrm{CR} = 0.9$ are chosen. Our decision for using those values is based on the proposed values from the literature [14]. Other parameters are the same as for the BDE algorithm. All the experiments are run 20 times independently. Both algorithms were coded in the MATLAB language. The programmes were run on a Pentium IV, 3.06-GHz PC with 2 GB memory in Windows Server 2003 environment.

3 Problem

Consider the planar thinned array consisting of $M \times N$ identical elements with spacings of dx and dy located in the x–y-plane. Assume that the array elements are all isotropic. The radiation pattern of the planar array is calculated by the standard expression of the array factor

$$F(\boldsymbol{a}, u, v) = \sum_{m=1}^{M} \sum_{n=1}^{N} a_{mn} \exp\left(jk(x_m u + y_n v)\right) \tag{4}$$

where $\boldsymbol{a} = (a_{11}, a_{12}, \ldots, a_{1N}; \ldots; a_{m1}, a_{m2}, \ldots, a_{MN})^{\mathrm{T}}$; $a_{mn} = 1$ if (m, n)th element is 'on', otherwise $a_{mn} = 0$; $k = 2\pi/\lambda$, λ is the operating wavelength; $u = \sin\theta\cos\phi$, $v = \sin\theta\sin\phi$, $x_m = (m-1)\,\mathrm{d}x$, $y_n = (n-1)\,\mathrm{d}y$ and $\mathrm{d}x = \mathrm{d}y = 0.5\lambda$.

The peak SLL can be formulated as follows

$$F_{\mathrm{SLL}}(\boldsymbol{a}) = \max_{(u,v)\in S}\left|\frac{F(\boldsymbol{a}, u, v)}{FF_{\max}}\right| \tag{5}$$

where S denotes the sidelobe region and $FF_{\max}$ is the peak of main beam.

To suppress SLL in all the planes, the fitness function can be defined as

$$f(\boldsymbol{a}) = F_{\mathrm{SLL}}(\boldsymbol{a}) \tag{6}$$

In [4], the fitness function is the sum of the maximum SLL in $\phi = 0^\circ$ and $\phi = 90^\circ$ planes, which is expressed as

$$f(\boldsymbol{a}) = F_{\mathrm{SLL}}(\boldsymbol{a})|_{\phi=0^\circ} + F_{\mathrm{SLL}}(\boldsymbol{a})|_{\phi=90^\circ} \tag{7}$$

Hence, the thinned array synthesis problem can be formulated as the following 0–1 integer optimisation problem

$$\begin{aligned} &\min f(\boldsymbol{a}) \\ &s.t.\ a_{mn} \in \{0, 1\},\ m = 1, 2, \ldots M,\ n = 1, 2, \ldots, N \\ &\quad \boldsymbol{a} = (a_{11}, a_{12}, \ldots, a_{1N};\ \cdots;\ a_{m1}, a_{m2}, \ldots, a_{MN})^{\mathrm{T}} \end{aligned} \tag{8}$$

4 Numerical examples

The numerical assessment of the BDE algorithm is conducted. First, two cases of the planar thinned arrays with 6×6 and 10×10 elements, respectively, are evaluated. The sidelobes are suppressed in all planes. Equation (6) acts as a fitness function. In order to achieve minimum SLL, the thinned percentage of the arrays is not constrained. The second example is the planar thinned array with 20×10 elements [4]. The SLL is suppressed in $\varphi = 0^\circ$ and $\varphi = 90^\circ$ planes and the array is 54% filled. Equation (7) is selected as a fitness function.

In the optimisation process, the calculation of the array pattern using the element-by-element superposition principle is very time-consuming. To reduce the computational burden, an FFT is employed to speed up the calculation of the fitness function. The FFT method is very suitable for computing the array with equal spacing. Table 1 shows the average times required for 100 independent evaluations of planar array pattern using two methods. As shown in Table 1, the FFT method can be executed at least 476 times faster than the method based on element-by-element superposition in our study. Therefore, the FFT method is incorporated into the optimisation methods to speed up the calculation of the fitness function. More details on the FFT method are available in [16].

Table 1 Average time for computing the array pattern

Number of elements	Number of sample points	Element-by-element superposition	FFT
6×6	200×200	25.9394 s	0.0545 s
10×10		75.4079 s	0.0605 s

The planar thinned array composed of 6×6 elements is evaluated by the BDE algorithm. Equation (6) acts as the fitness function. Fig. 2 shows the best results obtained with 20 independent runs. The optimised thinned array with 6×6 elements has -16.64 dB peak SLL in all planes. For verification purposes, the best array configuration is shown in Fig. 3. The aperture of the 6×6-element array is filled by a factor of 86%. Next, the planar thinned array with 10×10 elements is optimised. The radiation pattern of the best optimal results is plotted in Fig. 4 with the peak SLL of -19.56 dB. The best array geometry that is obtained is given in Fig. 5. The aperture is filled by 76 elements and other 24 elements are turned-off.

The BDE algorithm is exclusively based on Boolean algebra. For comparison, another binary DE [14] with entirely different philosophy for creating binary strings is also applied for thinned array designs with the minimum SLL. For details about the binary DE, see [14]. We use the DE evolved six-dimensional real vector with six-bit quantisation for the thinned array with 6×6 elements. For the 10×10-element array, a 20-dimensional real vector with five-bit quantisation is employed. The best, mean and worst values of the peak SLL derived after 20 independent executions of the BDE and the binary DE algorithms are

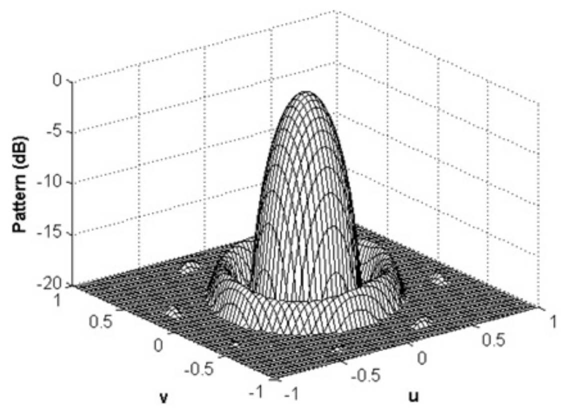


Figure 2 *Radiation pattern of the optimised 6×6-element thinned array*

doi: 10.1049/iet-map.2009.0630

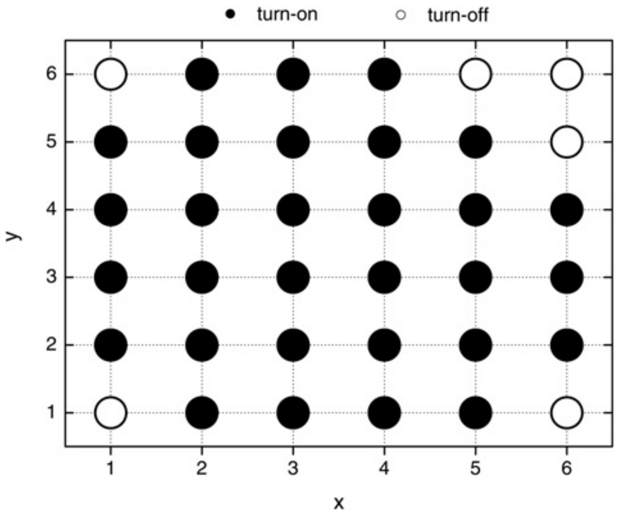


Figure 3 *Best array configuration of the thinned array with 6 × 6 elements*

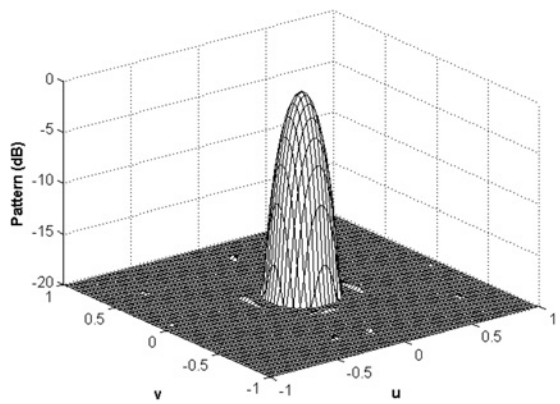


Figure 4 *Radiation pattern of the optimised 10 × 10-element thinned array*

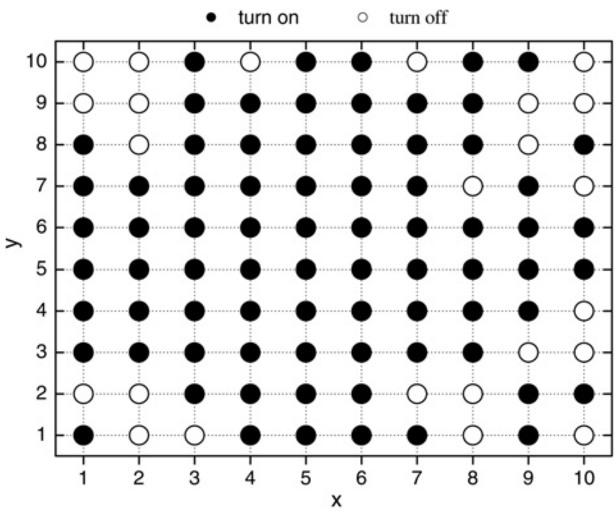


Figure 5 *Best array configuration of the thinned array with 10 × 10 elements*

shown in Table 2. The convergence characteristics of the BDE and the binary DE algorithms are plotted in Figs. 6 and 7, respectively. It can be seen from Table 2, Figs. 6 and 7 that the BDE has a faster convergence and obtains a better fitness value for the thinned array designs and outperforms the binary DE method [14]. Two entirely different binary DE algorithms have been demonstrated in the planar thinned array design. Numerical results indicate that the BDE algorithm is suitable for the thinned array design.

To further show the performance of the BDE, a 20 × 10-element thinned planar array is designed, which is symmetric about the x-axis and y-axis. The thinned planar array was optimised with a GA [4]. The fitness function is the sum of the maximum SLL in $\varphi = 0^\circ$ and $\varphi = 90^\circ$ planes. In this case, (7) acted as the fitness function. The fitness value of the optimal solution by the GA is -39.83 dB [4] (SLL $= -20.07$ dB in $\varphi = 0^\circ$ plane and SLL $= -19.76$ dB in $\varphi = 90^\circ$ plane). In this section, the BDE algorithm is employed to optimise this thinned

Table 2 The best, mean and worst values of the peak SLL derived after 20 times executions of the BDE and the binary DE [14] algorithms

Method	Number of elements	Best value (dB)	Mean value (dB)	Worst value (dB)
BDE	6 × 6	−16.64	−16.64	−16.64
binary DE		−16.64	−16.28	−15.86
BDE	10 × 10	−19.56	−19.24	−18.97
binary DE		−18.47	−17.81	−16.93

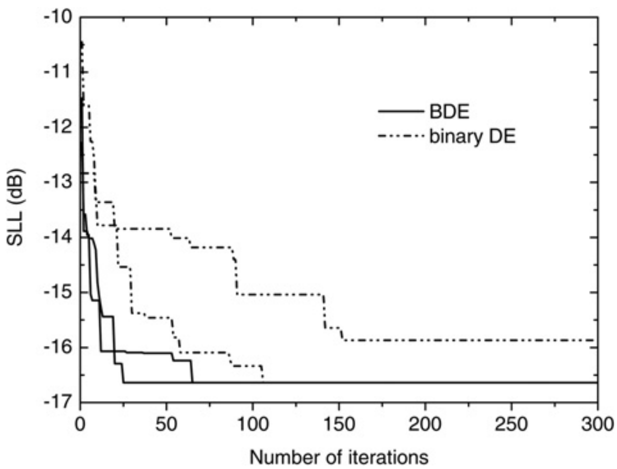


Figure 6 *The best and worst convergence curves of the BDE and the binary DE algorithms for the thinned array with 6 × 6 elements*

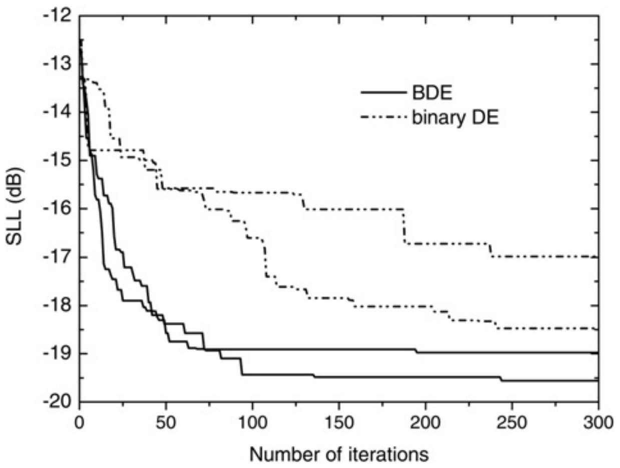


Figure 7 *The best and worst convergence curves of the BDE and the binary DE algorithms for the thinned array with 10 × 10 elements*

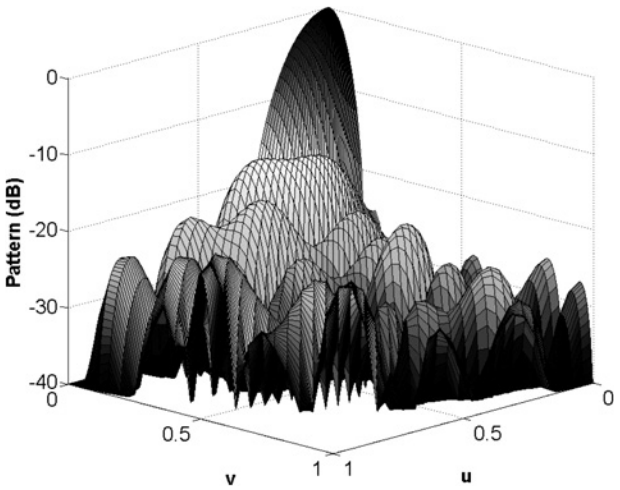


Figure 8 *Radiation pattern of the optimised 20 × 10-element thinned array*

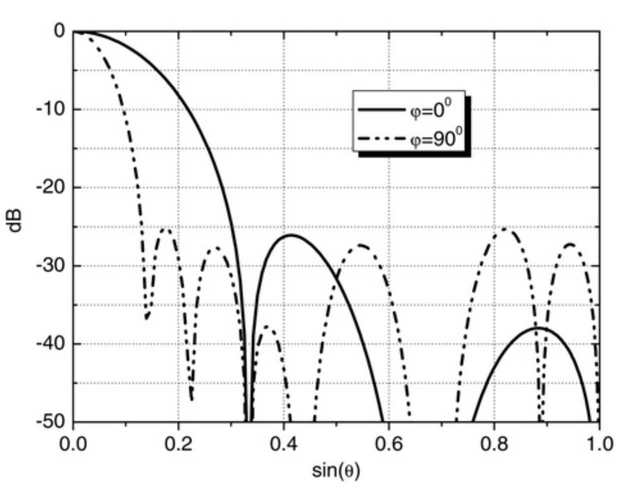


Figure 9 *Principal plane patterns of the optimised 20 × 10-element thinned array*

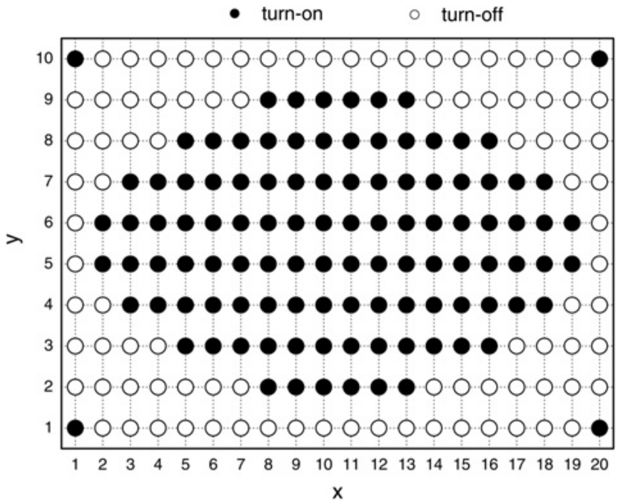


Figure 10 *Best array configuration of the thinned array with 20 × 10 elements*

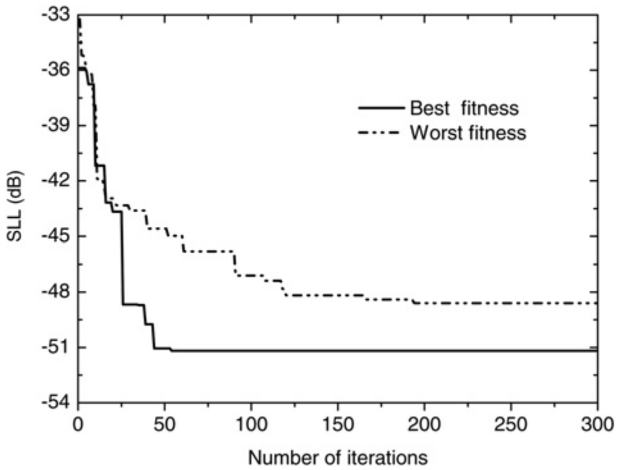


Figure 11 *The best and worst convergence curves of the BDE algorithm for the 20 × 10-element thinned array with 54% filled*

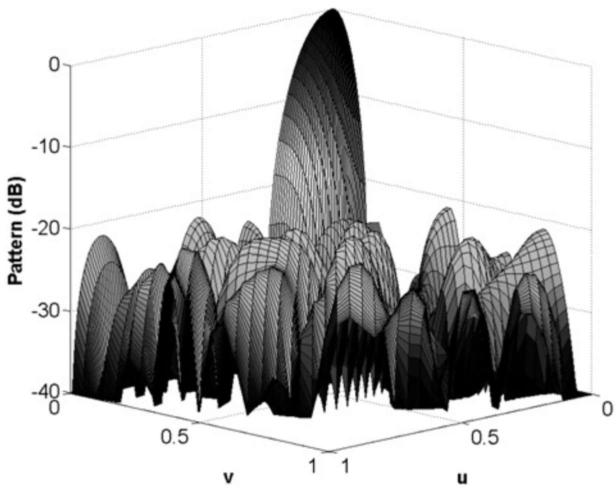


Figure 12 *Radiation pattern of the optimised 20 × 10-element thinned array for suppressing SLL in all the planes*

doi: 10.1049/iet-map.2009.0630

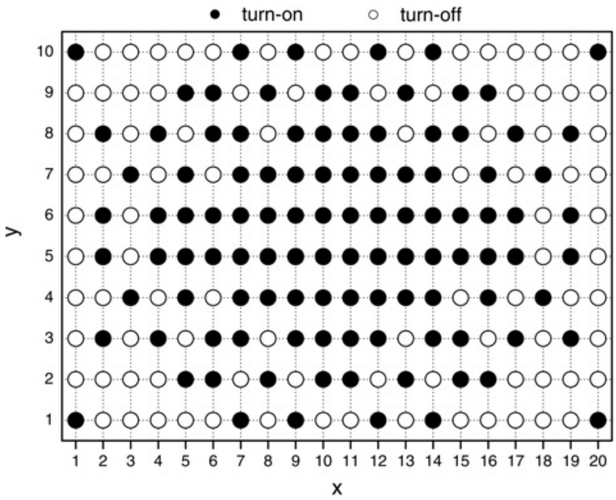


Figure 13 *Best array geometry of the optimised 20 × 10-element thinned array for suppressing SLL in all the planes*

planar array. Equation (7) is also used as the fitness function. The 54% filled array is constrained, which is the same as in [4]. Fig. 8 shows the radiation pattern of the optimised array. The best fitness value is -51.18 dB (SLL $= -26.09$ dB in $\varphi = 0^\circ$ plane and SLL $= -25.09$ dB in $\varphi = 90^\circ$ plane, as shown in Fig. 9). The optimisation result by BDE is 11.35 dB lower than that of the best array in [4]. The configuration of the best array is plotted in Fig. 10. The best and worst convergence curves of the BDE are also shown in Fig. 11. For the 20 × 10-element thinned planar array, another optimisation problem is that SLL is suppressed in all the planes. Equation (6) is selected as the fitness function and the array with 54% filled factor is constrained. The radiation pattern of the optimal array whose maximum SLL is -20.49 dB in all the planes is depicted in Fig. 12. The best array geometry with 108 elements is shown in Fig. 13. Numerical results again demonstrate that the BDE algorithm is an effective technique for thinned array designs.

5 Conclusion

The BDE algorithm with only one control parameter, which is easily implemented, is used in the design of the planar thinned arrays. The experimental results show that the BDE algorithm converges quickly and yields high-quality solutions. As an evolutionary algorithm, the BDE algorithm can be also used to solve continuous optimisation problems. Although the BDE algorithm is used here to design thinned arrays, it is also suitable for other EM optimisation problems.

6 Acknowledgment

This work was supported by the RIM Advanced Antenna Research Program.

7 References

[1] KING D.D., PACKARD R.F., THOMAS R.K.: 'Unequally-spaced broadband antenna arrays', *IRE Trans. Antennas Propag.*, 1960, **8**, (7), pp. 380–384

[2] ISHIMARU A., CHEN Y.S.: 'Thinning and broadbanding antenna arrays by unequal spacings', *IEEE Trans. Antennas Propag.*, 1965, **13**, (1), pp. 34–42

[3] LEEPER D.G.: 'Thinned periodic antenna arrays with improved peak sidelobe level control'. U.S. patent 4071848, January 1978

[4] HAUPT R.L.: 'Thinned arrays using genetic algorithms', *IEEE Trans. Antennas Propag.*, 1994, **42**, (7), pp. 993–999

[5] QUEVEDO-TERUEL Ó., RAJO-IGLESIAS E.: 'Ant colony optimization in thinned array synthesis with minimum sidelobe level', *IEEE Antennas Wireless Propag. Lett.*, 2006, **5**, pp. 349–352

[6] JIN N., RAHMAT-SAMII Y.: 'Advances in particle swarm optimization for antenna design: real-number, binary, single-objective and multiple objective implementations', *IEEE Trans. Antennas Propag.*, 2007, **55**, (3), pp. 556–567

[7] STORN R., PRICE K.: 'Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces', *J Global Optim.*, 1997, **11**, pp. 341–359

[8] KURUP D.G., HIMIDI M., RYDBERG A.: 'Synthesis of uniform amplitude unequally spaced antenna arrays using the differential evolution algorithm', *IEEE Trans. Antennas Propag.*, 2003, **51**, (9), pp. 2210–2217

[9] YANG S., GAN Y.B., QING A.: 'Antenna array pattern nulling using a differential evolution algorithm', *Int. J. RF Microw. Comput. Aided Eng.*, 2004, **14**, (1), pp. 57–63

[10] MASSA A., PASTORINO M., RANDAZZO A.: 'Optimization of the directivity of a monopulse antenna with a subarray weighting by a hybrid differential evolution method', *IEEE Antennas Wireless Propag. Lett.*, 2006, **5**, pp. 155–158

[11] ROCHA-ALICANO C., COVARRUBIAS-ROSALESA D., BRIZUELA-RODRIGUEZ C., PANDURO-MENDOZA M.: 'Differential evolution algorithm applied to sidelobe level reduction on a planar array', *Int. J. Electron. Commun.*, 2007, **61**, pp. 286–290

[12] ZHANG L., CUI Z., JIAO Y.C., ZHANG F.S.: 'Broadband patch antenna design using differential evolution algorithm', *Microwave Opt. Technol. Lett.*, 2009, **51**, (7), pp. 1692–1695

[13] ENGELBRECHT A., PAMPARÁ G.: 'Binary differential evolution strategies'. Proc. IEEE Congress on Evolutionary Computation, Singapore, September 2007, pp. 1942–1947

[14] GREENWOOD G.W.: 'Using differential evolution for a subclass of graph theory problems', *IEEE Trans. Evol. Comput.*, 2009, **13**, (5), pp. 190–1192

[15] AFSHINMANESH F., MARANDI A., SHAHABADI M.: 'Design of a single-feed dual-band dual-polarized printed microstrip antenna using a Boolean particle swarm optimization', *IEEE Trans. Antennas Propag.*, 2008, **56**, (7), pp. 1845–1852

[16] WANG L., FANG D., SHENG W.: 'Combination of genetic algorithm (GA) and fast Fourier transform (FFT) for synthesis of arrays', *Microwave Opt. Technol. Lett.*, 2003, **37**, (1), pp. 56–59

IET Microw. Antennas Propag., 2010, Vol. 4, Iss. 12, pp. 2172–2178

doi: 10.1049/iet-map.2009.0630