Progress In Electromagnetics Research, Vol. 127, 371–387, 2012

SYNTHESIS OF THINNED LINEAR AND PLANAR ANTENNA ARRAYS USING BINARY PSO ALGORITHM

W.-B. Wang[1,2], Q.-Y. Feng[1,*], and D. Liu[3]

[1]School of Information Science & Technology, Southwest Jiaotong University, Chengdu, Sichuan 610031, China

[2]School of Electrical and Information Engineering, Xihua University, Chengdu, Sichuan 610039, China

[3]School of Electrical Engineering, Southwest Jiaotong University, Chengdu, Sichuan 610031, China

Abstract—Traditional optimization methods are not well suitable for thinning large arrays to obtain a low sidelobe level (SLL). The chaotic binary particle swarm optimization (CBPSO) algorithm is presented as a useful alternative for the synthesis of thinned arrays. The proposed algorithm can be improved by nonlinear inertia weight with chaotic mutation to increase the diversity of particles. Two examples have been presented and solved. Simulation results are proposed to compare with published results to verify the effectiveness of the proposed method for both linear and planar arrays.

1. INTRODUCTION

Array thinning is related to the removal of radiating elements from a uniformly spaced or periodic array to create a desired radiation pattern [1–3]. The main purpose of thinning is to reduce the cost, weight and power consumption [4, 5]. In thinned array synthesis, the positions of the elements will be fixed, and each element will present two states: "on" (when the element is fed) and "off" (when the element is passively terminated in an impedance equal to the source impedance of the fed elements). However, synthesis of antenna arrays is a tough challenge, and it is hard to solve these synthesis problems with simple analytical methods [6–10].

Received 3 February 2012, Accepted 17 April 2012, Scheduled 26 April 2012

* Corresponding author: Quanyuan Feng (fengquanyuan@163.com).

In order to reduce the number of elements while keeping the radiation properties of the original structures, several thinning techniques have been successfully introduced [11, 12]. In recent years, the use of global optimization approaches has led to significant advancements [13]. These methods, such as particle swarm optimization (PSO) [14, 15], genetic algorithm (GA) [16, 17], differential evolution (DE) [18–20] and ant colony optimization (ACO) [21], have already been used to design thinned arrays for various applications.

The main contribution of this paper is to employ a modified binary PSO to synthesize thinned arrays with a lower SLL. The chaotic sequences are embedded in the proposed algorithm to determine the inertia weight of the binary PSO.

This paper is organized as follows: Section 2 describes the theoretical formulations for the thinned linear and planar arrays. The principle of the CBPSO is presented in Section 3. Numerical results for thinned arrays are given and analyzed in Section 4 while the conclusions are discussed in Section 5.

2. THINNED ARRAY

2.1. Linear Array

According to the structure shown in Figure 1, where there are $2N$ isotropic radiators placed symmetrically along the x-axis, the array factor AF at an angle θ in xz plane for a linear antenna array can be expressed as [22, 23]:

$$AF(\mathbf{I}, \mathbf{x}, \boldsymbol{\varphi}, \theta) = \sum_{n=-N}^{N} I_n e^{j\left(\frac{2\pi}{\lambda} x_n \sin\theta + \varphi_n\right)} \tag{1}$$

where $\mathbf{I}$ is the vector of the excitation amplitudes of the array elements, $\mathbf{x}$ the vector of the element positions, $\boldsymbol{\varphi}$ is the vector of the excitation phases, and λ the wavelength. In thinned array synthesis, the excitation amplitude I_n is 1 if the state of the nth element is "on" and 0 if the state of the nth element is "off".

In our cases, as shown in Figure 1, there is no element located at the axis origin, the distance between elements is 0.5λ, all elements have the uniform excitation phase ($\varphi_n = 0$). Thus, Equation (1) can be written as [17]:

$$AF(\mathbf{I},\theta) = 2\sum_{n=1}^{N} I_n \cos[\pi(n - 0.5)\sin\theta] \tag{2}$$

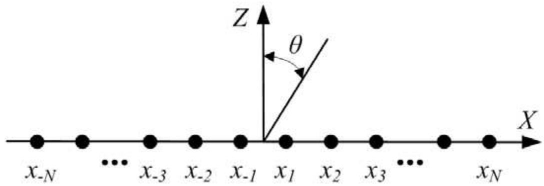

Figure 1. Geometry of a $2N$-element symmetric linear array.

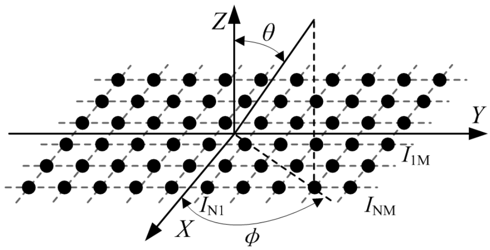

Figure 2. Geometry of a $2N \times 2M$-element symmetric planar array.

where the element number n is sequenced from the array center of the positive axis.

2.2. Planar Array

Figure 2 shows a planar array structure of $2N \times 2M$ elements. Assuming the same considerations as in the linear array, the array factor in this structure is given by [3, 24]:

$$AF(\mathbf{I}, \theta, \phi) = 4 \sum_{n=1}^{N} \sum_{m=1}^{M} I_{nm} \cos[\pi(n-0.5)\sin\theta\cos\phi] \cdot \cos[\pi(m-0.5)\sin\theta\sin\phi] \quad (3)$$

where θ is the elevation angle with respect to the z-axis, and ϕ is the azimuth angle with respect to x-axis. The interelement spacing is equal to 0.5λ, and the amplitude of excitations (I_{nm}) and interelement spacing are both symmetrical about the x and y axes. Thus, the array factor can be simplified by computing a quarter of the rectangular array.

Therefore, it is necessary to find out which array elements should be turned on or off to get the desired radiation pattern characteristics.

2.3. Fitness Function

The first and most important parameter in antenna pattern synthesis is the normalized sidelobe level that is desired to be as low as possible [25].

In this paper, the normalized peak sidelobe level (PSLL) of the antenna array can be formulated as follows

$$F_{\mathrm{PSLL}}(\mathbf{I}) = \max_{\forall\,\theta \in S} \left\{ 20 \log \left| \frac{AF(\mathbf{I},\theta)}{AF_{\max}} \right| \right\} \tag{4}$$

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

To suppress SLL, we use the following objective function:

$$f(\mathbf{I}) = F_{\mathrm{PSLL}}(\mathbf{I}) \tag{5}$$

For Equation (3), the fitness function is the sum of the maximum PSLL in $\phi = 0^\circ$ and $\phi = 90^\circ$ planes, that can be expressed as [26]:

$$f(\mathbf{I}) = F_{\mathrm{PSLL}}(\mathbf{I})\,|_{\phi=0^\circ} + F_{\mathrm{PSLL}}(\mathbf{I})\,|_{\phi=90^\circ} \tag{6}$$

Thus, the synthesis problem of the thinned array can be formulated as 0–1 integer optimization problem for a binary PSO algorithm:

$$\begin{cases} \min\{f(\mathbf{I})\} \\ s.t.\ I_{nm} \in \{0,1\}, \quad n = 1,2,\ldots,N, \qquad m = 1,2,\ldots,M \\ \mathbf{I} = (I_{11}, I_{12}, \ldots, I_{1M}; \ldots; I_{N1}, I_{N2}, \ldots, I_{NM}) \end{cases} \tag{7}$$

3. MODIFIED BINARY PSO

3.1. Particle Swarm Optimization

The PSO algorithm was proposed by Kennedy and Eberhart in 1995, motivated by social behavior of organisms such as bird flocking and fish schooling [27]. It is not only a tool for optimization, but also a tool for representing sociocognition of human and artificial agents. In a PSO system, particles fly around in a multi-dimensional search space. During the flight, each particle adjusts its position according to its own experience and the experience of neighboring particles, making use of the best position encountered by itself and its neighbors The basic PSO method updates the velocity and position of each particle according to the equations given below:

$$v_{id}(t+1) = w(t)v_{id}(t) + c_1 r_{1d}[p_{id}(t) - x_{id}(t)] + c_2 r_{2d}[p_{gd}(t) - x_{id}(t)] \tag{8}$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \tag{9}$$

where $i = 1, \ldots, N$, $d = 1, \ldots, D$, N is the number of particles; D represents the number of optimal parameters to be determined; w is the inertia weight factor to control the exploration and exploitation; c_1 and c_2 are the acceleration constants; $r_{1\mathrm{d}}$ and $r_{2\mathrm{d}}$ are two random numbers within the range $[0, 1]$; v_{id} and x_{id} are the velocity and position of the current particle i at time step t in the dth-dimensional search space respectively; $\mathbf{p}_i$ is the previous best position of particle i, also called "personal best"; "global best" $\mathbf{p}_g$ is the best position found in all particles. The first part of Equation (8) is the momentum, which prevents velocity from changing abruptly. The inertia weight w is employed to control the impact of the previous history of velocities on the current velocity. The second part of Equation (8) is the "cognition", which represents the private thinking of the particle itself. The third part is the "social" part, which represents the collaboration between the particles.

3.2. Binary PSO

For single objective, PSO algorithms have been classified into two types, real-number and binary PSO [28]. The binary PSO (BPSO) was also introduced by Kennedy and Eberhart in 1997 [29]. The BPSO can cover a wide range of applications as the binary sequences can be transformed to meet the requirements of combinatorial optimization problems [30, 31].

In the BPSO, the velocity of a particle in each dimension is represented by v_{id}, where i is the number of particles, and d is the number of dimensions of a given data set. The position vector of a particle is a binary one. The velocity is related to the possibility that the position of the particle takes a value of 1 or 0. Once the adaptive values "personal best" $\mathbf{p}_i$ and "global best" $\mathbf{p}_g$ are obtained, the features of the $\mathbf{p}_i$ and $\mathbf{p}_g$ particles can be tracked with regard to their position and velocity. Substituted for Equations (9), the position of each particle is updated according to the following equations [29].

$$x_{id}(t+1) = \begin{cases} 1, & rand() < Sigmoid(v_{id}(t+1)) \\ 0, & \text{else.} \end{cases} \tag{10}$$

$$Sigmoid(x) = \frac{1}{1 + \exp(-x)} \tag{11}$$

where $rand(\ )$ is a quasirandom number selected from a uniform distribution in $[0, 1]$.

3.3. Chaotic Binary PSO

Based on the BPSO, we propose the chaotic binary PSO (CBPSO) algorithm to deal with the synthesis of the thinned array mentioned in the previous section.

In order to increase the diversity of the population, the CBPSO utilizes chaotic mutation to determine the inertia weight. The inertia weight controls the balance between the local search ability and global exploration. Appropriate adjustment of the inertia weight value is very important [32]. In [33], a nonlinear variation of the inertia weight along with a particle's old velocity is adopted to improve the speed of convergence as well as fine tune the search in the multidimensional space. In [34], chaotic sequences are employed for binary PSO to prevent the early convergence.

Optimization algorithms based on the chaos theory are stochastic search methodologies that differ from any of the existing evolutionary algorithms. Due to the non-repetition of chaos, it can carry out overall exploration at higher velocities than stochastic and ergodic search that depend on probabilities [35].

In this paper, chaotic mutation is embedded in the CBPSO. The chaotic mutation is generated by the logistic map used in [36]. The equation is the following:

$$y(t+1) = \mu y(t)\left[1 - y(t)\right] \tag{12}$$

where $y(t) \in (0,1)$, μ is usually set to 4 to obtain ergodicity of $y(t+1)$ within (0, 1). When the initial value $y(0) \notin \{0.25, 0.5, 0.75\}$, using Equation (12) we can obtain chaotic sequences.

In [34], the chaotic sequences are used as the substitute for original inertia weight. However, it is possible that large inertia weight will appear at the end of iterations. A large inertia weight facilitates the global search, while a small inertia weight facilitates the local search [34, 37]. In this paper, in order to achieve refined exploration at the end of the iterations, we add the chaotic mutation to the nonlinear inertia weight, as follows:

$$w(t+1) = \left[\left(\frac{t_{\text{max}} - t}{t_{\text{max}}}\right)^{\delta} (w_{\text{start}} - w_{\text{end}}) + w_{\text{end}}\right] y(t) \tag{13}$$

where t is the iteration number at the present time step, t_{max} the maximum number of iterations; w_{start} and w_{end} are the initial and final inertia weights at the given run; δ is nonlinear modulation factor. According to [33], $w_{\text{start}} = 0.9$, $w_{\text{end}} = 0.1$, $\delta = 1.2$. Figure 3 shows the comparison between the inertia weight with chaotic mutation and linear decrease.

From Figure 3, we can see the maximum values of the inertia weight with chaotic mutation are graduated from large scales to small ones as iteration time is increasing, and the inertia weight values are not linearly decreased because of the chaotic mutation. That makes the particles have larger inertial weight to explore new regions at the

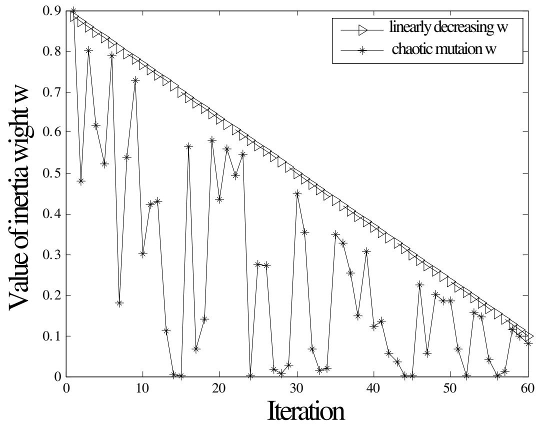

Figure 3. Comparison between the chaotic mutation w and linearly decreasing w.

initial iteration time while the particles have smaller inertial weight to exploit the prominent regions at the end of the iteration time.

In order to limit the exploration to the solution space, a boundary condition should be discussed. This paper uses the reflecting boundary conditions analyzed in [38] to enforce particles to search inside the solution space. The reflecting boundary conditions mean that a particle is relocated at the boundary of the solution space in one of the dimensions when the particle is staying outside the solution space in that dimension. If the velocity of one dimension exceeds $V_{\max}$, the velocity of that dimension is limited to $V_{\max}$. In this paper, $V_{\max} = -V_{\min} = 6$ as mentioned in [34, 39].

A brief description of the CBPSO algorithm is given as follows:

1. Randomly generate an initial population.
2. Evaluate fitness values of all particles.
3. Calculate the inertia weight value with chaotic mutation according to Equations (12) and (13).
4. Update the $\mathbf{p}_i$ and $\mathbf{p}_g$ values. Each particle updates its velocity and position by the CBPSO through Equations (8) and (10).
5. Check the termination criterion. If the $t_{\max}$ is reached, then end the algorithm, otherwise go to step 2.

4. NUMERICAL RESULTS

In this section, the capability and versatility of the proposed algorithm will be assessed by presenting two thinned array cases.

All simulations are conducted in a Windows 7 Professional OS environment using 12-core processors with Intel Xeon (R), 3.33 GHz, 72 GB RAM and the codes are implemented in Matlab 7.10.

4.1. Linear Array

The first case discussed the geometry of a 100-element thinned linear array symmetrically spaced 0.5λ apart along the x-axis with its center at the origin in order to generate a broadside symmetric pattern [3, 17]. In [3], Quevedo-Teruel and Rajo-Iglesias utilize the ACO in the symmetric linear array synthesis, and Mahanti et al. designed the same array using a real-coded GA [17].

According to the structure shown in Figure 1, the excitation amplitude distribution is symmetric with respect to the center of the linear array. In this case, to take advantage of the entire structure, we turn all elements on to initialize the array. There is no restriction imposed on switching on or off the end element of the thinned array. Because of symmetric structure, only half amplitudes are to be optimized.

In this case, fitness function shown as Equation (7) is minimized using the CBPSO. The best radiation pattern is obtained by the CBPSO with following status table (on = 1, off = 0) for the half elements.

1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 0 1 1 1 1 0 1 1 1
1 0 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 0 0 1 0

Figure 4 shows the pattern results obtain by the CBPSO, and the results are compared with the initial value that all elements are turned on.

To further verify the performance of the CBPSO, it is compared with the ACO [3], GA [17], BPSO (Binary PSO) [29] and BDE (Boolean Differential Evolution) [18], and the obtained array patterns using these optimization algorithms are presented in Figure 5.

Table 1 lists the results obtained by the above algorithms. Clearly, the percentage of thinning obtained by the CBPSO is more than that of 20% in [3] and 22% in [17] without sacrificing sidelobe level. The PSLL obtained by the CBPSO is -21.29 dB, which is lower than that of other algorithms.

In order to test the efficiency and reliability of the proposed algorithms, the CBPSO is further compared with these before-mentioned algorithms in terms of convergence speed. All the experiments have been run 100 times with $t_{\max}$ =300 iterations independently. For each iteration step, the average fitness value is calculated from the 100 fitness values derived at the certain step. Figure 6 shows the variation of the average SLL value as a function of the number of iterations. Table 2 shows the comparisons of the simulation results.

From Table 2, the CBPSO can achieve the best average maximum |SLL|, and it converges faster than the ACO, BPSO and BDE except for the GA. More details can be seen from Figure 6. The result of the unmodified BPSO is worse, and the convergence speed of the CBPSO is slower than that of the ACO, BDE and GA during the early 70 iterations. On the other hand, the CBPSO can find better solutions after approximately 120 iterations.

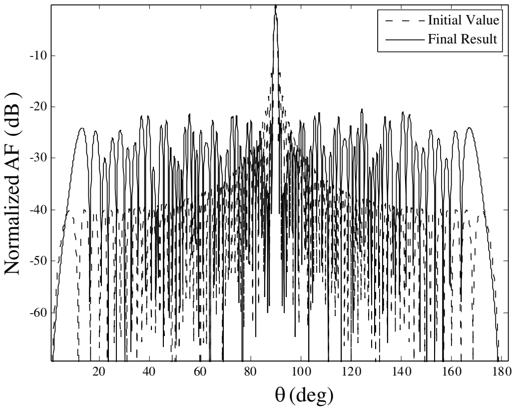

Figure 4. Radiation pattern compared with the initial value.

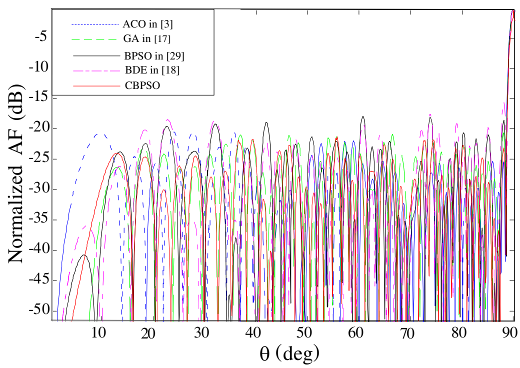

Figure 5. Radiation pattern of 100-element array obtained by the CBPSO and other algorithms.

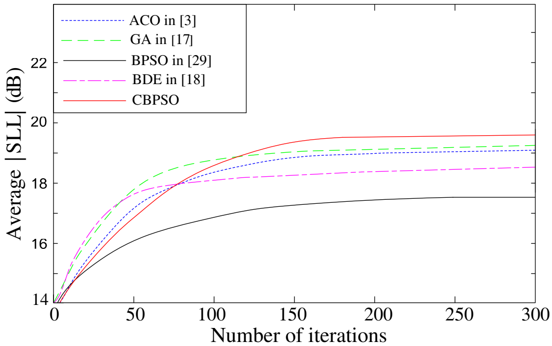

Figure 6. Convergence of the |SLL| value versus the number of iterations.

Table 1. Comparative results in 100-element thinned array.

Design Parameters	ACO in [3]	GA in [17]	BPSO in [29]	BDE in [18]	CBPSO
Percentage of thinning	20	22	20	22	**24**
PSLL (dB)	−20.52	−20.56	−17.92	−18.53	**−21.29**

Table 2. Comparisons of the simulation results.

Simulation results	ACO in [3]	GA in [17]	BPSO in [29]	BDE in [18]	CBPSO
Average maximum |SLL|(dB)	19.1	19.2	17.6	18.5	**20.0**
Average convergence iterations	203	**164**	236	193	181

4.2. Planar Array

The second case discussed the thinned planar array with 20×10 elements in a square lattice [2, 3, 18]. Haupt applied the GA algorithm to the 20×10 planar array [2], Quevedo-Teruel et al., utilized the ACO algorithm in the array synthesis [3], and Zhang et al., designed the same array using the BDE (Boolean Differential Evolution) [18].

In this case, the sidelobe level is suppressed in $\phi = 0°$ and $\phi = 90°$ planes. Equation (6) is selected as the fitness function optimized by the CBPSO. Figure 7 shows the radiation pattern of the optimized array achieved by the proposed algorithm.

The obtained solution is given for a quadrant of the array elements plotted in Figure 8, where the white blocks indicate elements that are turned on, and the black ones are elements that are turned off.

The array patterns using the CBPSO, ACO [3], GA [2], BPSO [29] and BDE [18] in the planes $\phi = 0°$ and $\phi = 90°$ are shown in Figures 9 and 10, respectively. Table 3 lists the performance comparisons of various algorithms.

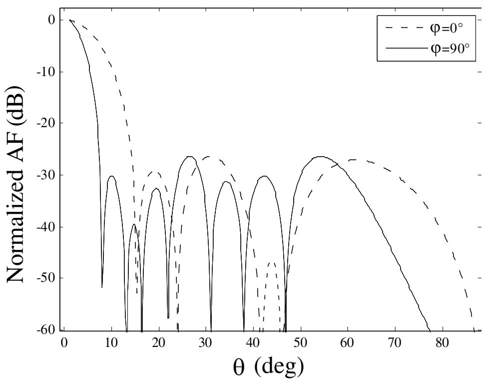

Figure 7. Radiation pattern of 20×10 thinned planar array achieved by the CBPSO in the planes $\phi = 0°$ and $\phi = 90°$.

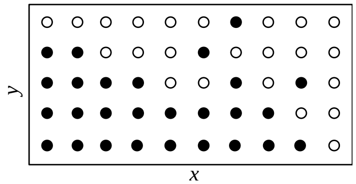

Figure 8. A quadrant configuration of the thinned planar array with 20×10 elements.

Table 3. Comparative results in the thinned planar array.

Design Parameters	ACO in [3]	GA in [2]	BPSO in [29]	BDE in [18]	CBPSO
Percentage of thinning	32	46	44	46	**46**
PSLL (dB) ($\phi = 0°$)	-25.76	-20.07	-21.39	-26.09	$\mathbf{-26.39}$
PSLL (dB) ($\phi = 90°$)	-25.674	-19.76	-21.94	-25.09	$\mathbf{-26.33}$

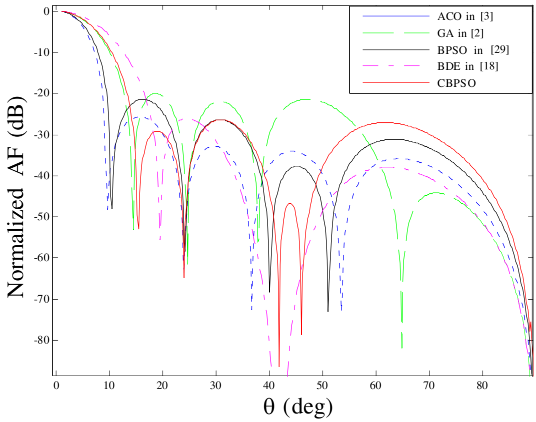

Figure 9. Performance comparisons of 20×10 thinned planar array in the planes $\phi = 0°$.

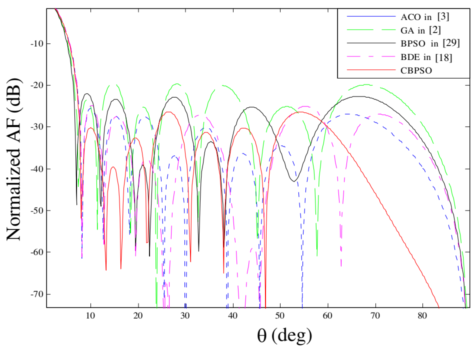

Figure 10. Performance comparisons of 20×10 thinned planar array in the planes $\phi = 90°$.

The comparisons in Table 3 demonstrate that the CBPSO can achieve the best PSLL in $\phi = 0°$ and $\phi = 90°$ planes, and the percentage of thinning obtained by the CBPSO is as same as the ones of [2] and [18], and it outperforms the ones of [3] and [29].

Details of comparative studies in terms of the convergence process are carried out and shown in Figure 11 and Table 4.

Table 4 illustrates that the CBPSO can achieve the best average maximum |SLL|, and it has a faster convergence speed and outperforms the GA and BPSO except for the ACO and BDE. As observed in Figure 11, the result of the unmodified is worse, the convergence speed of the CBPSO is slower than that of the ACO, BDE and GA during the early 40 iterations, while the CBPSO can find better solutions after approximately 170 iterations.

From the above results, in the synthesis of thinned linear and planar arrays, it can be observed clearly that the proposed algorithm improved by nonlinear inertia weight with chaotic mutation can take a good balance between the local search ability and global exploration. Thus, the convergence speed of the CBPSO is slower than that of the ACO, BDE and GA during the early iterations, while the CBPSO can maintain the persistence of convergence in the later iterations, and it can achieve better solutions at the end of iterations compared with above algorithms. Furthermore, the CBPSO is improved based on the simple binary PSO, the contrast between the worst results obtained by the unmodified binary PSO and the desirable performance achieved by the CBPSO can also justify the validity and efficiency of the proposed strategy.

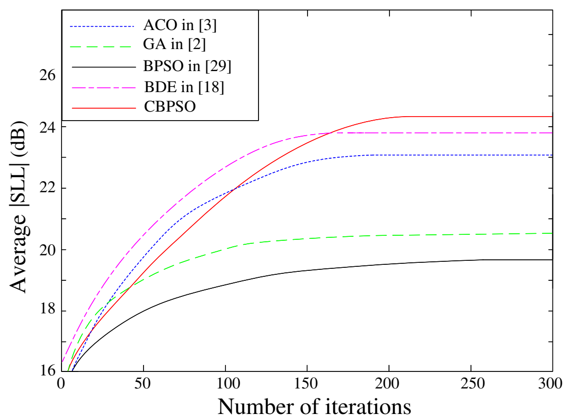

Figure 11. Convergence of the |SLL| value versus the number of iterations.

Table 4. Comparisons of the simulation results.

Simulation results	ACO in [3]	GA in [2]	BPSO in [29]	BDE in [18]	CBPSO
Average maximum |SLL| (dB)	23.1	20.5	19.6	23.8	**24.3**
Average convergence iterations	184	210	257	**162**	206

5. CONCLUSIONS

This paper describes the use of binary PSO algorithm for pattern synthesis of thinned arrays with the constraints of sidelobe reduction. A novel binary PSO algorithm (chaotic binary PSO, CBPSO) has been presented to improve the convergence speed and accuracy in the array synthesis. Because of the ergodicity, regularity and pseudo-randomness of the chaotic sequences embedded in the CBPSO, the proposed algorithm in a hybrid of the chaos system can avoid entrapment in local optima.

An extensive numerical analysis has been performed by addressing thinned linear and planar arrays with SLL suppression. Comparisons of the CBPSO and other techniques, the ACO, GA, binary PSO and Boolean DE show the efficiency of the proposed technique.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (No. 60990320, 60990323), the National 863 Propjet of China under Grant 2012AA012305, the Research Fund of Sichuan Provincial Key Discipline of Power Electronics and Electric Drive (SZD0503-09-0).

REFERENCES

1. Bucci, O. M., T. Isernia, and A. F. Morabito, "A deterministic approach to the synthesis of pencil beams through planar thinned arrays," *Progress In Electromagnetics Research*, Vol. 101, 217–230, 2010.
2. Haupt, R. L., "Thinned arrays using genetic algorithms," *IEEE Transactions on Antennas and Propagation*, Vol. 42, 993–999, 1994.
3. Quevedo-Teruel, O. and E. Rajo-Iglesias, "Ant colony optimization in thinned array synthesis with minimum sidelobe level," *IEEE Antennas and Wireless Propagation Letters*, Vol. 5, 349–352, 2006.
4. Fernandez-Delgado, M., J. A. Rodriguez-Gonzalez, R. Iglesias, S. Barro, and F. J. Ares-Pena, "Fast array thinning using global optimization methods," *Journal of Electromagnetic Waves and Applications*, Vol. 24, No. 16, 2259–2271, 2010.
5. Liu, Y., Z.-P. Nie, and Q. H. Liu, "A new method for the synthesis of non-uniform linear arrays with shaped power patterns," *Progress In Electromagnetics Research*, Vol. 107, 349–363, 2010.

6. Lebret, H. and S. Boyd, "Antenna array pattern synthesis via convex optimization," *IEEE Transactions on Signal Processing*, Vol. 45, 526–532, 1997.
7. Wang, W.-B., Q.-Y. 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.
8. Liu, D., Q.-Y. Feng, W.-B. Wang, and X. Yu, "Synthesis of unequally spaced antenna arrays by using inheritance learning particle swarm optimization," *Progress In Electromagnetics Research*, Vol. 118, 205–221, 2011.
9. Li, W.-T., Y.-Q. Hei, and X.-W. Shi, "Pattern synthesis of conformal arrays by a modified particle swarm optimization," *Progress In Electromagnetics Research*, Vol. 117, 237–252, 2011.
10. Petko, J. S. and D. H. Werner, "Pareto optimization of thinned planar arrays with elliptical mainbeams and low sidelobe levels," *IEEE Transactions on Antennas and Propagation*, Vol. 59, 1748–1751, 2011.
11. Oliveri, G., M. Donelli, and A. Massa, "Linear array thinning exploiting almost difference sets," *IEEE Transactions on Antennas and Propagation*, Vol. 57, 3800–3812, 2009.
12. Oliveri, G., L. Manica, and A. Massa, "ADS-based guidelines for thinned planar arrays," *IEEE Transactions on Antennas and Propagation*, Vol. 58, 1935–1948, 2010.
13. Zaharis, Z. D., D. G. Kampitaki, P. I. Lazaridis, A. I. Papastergiou, A. T. Hatzigaidas, and P. B. Gallion, "Improving the radiation characteristics of a base station antenna array using a particle swarm optimizer," *Microwave and Optical Technology Letters*, Vol. 49, 1690–1698, 2007.
14. Lanza, M., J. R. Perez, and J. Basterrechea, "Synthesis of planar arrays using a modified particle swarm optimization algorithm by introducing a selection operator and elitism," *Progress In Electromagnetics Research*, Vol. 93, 145–160, 2009.
15. Goudos, S. K., Z. D. Zaharis, D. G. Kampitaki, I. T. Rekanos, and C. S. Hilas, "Pareto optimal design of dual-band base station antenna arrays using multi-objective particle swarm optimization with fitness sharing," *IEEE Transactions on Magnetics*, Vol. 45, 1522–1525, 2009.
16. Oliveri, G. and A. Massa, "Genetic algorithm (GA)-enhanced almost difference set (ADS)-based approach for array thinning," *IET Microwaves, Antennas & Propagation*, Vol. 5, 305–315, 2011.

17. 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, 319–328, 2007.
18. Zhang, L., Y.-C. Jiao, Z.-B. Weng, and F.-S. Zhang, "Design of planar thinned arrays using a boolean differential evolution algorithm," *IET Microwaves, Antennas & Propagation*, Vol. 4, 2172–2178, 2010.
19. Rocca, P., G. Oliveri, and A. Massa, "Differential evolution as applied to electromagnetics," *IEEE Antennas and Propagation Magazine*, Vol. 53, 38–49, 2011.
20. Li, R., L. Xu, X.-W. Shi, N. Zhang, and Z.-Q. Lv, "Improved differential evolution strategy for antenna array pattern synthesis problems," *Progress In Electromagnetics Research*, Vol. 113, 429–441, 2011.
21. Oliveri, G. and L. Poli, "Optimal sub-arraying of compromise planar arrays through an innovative ACO-weighted procedure," *Progress In Electromagnetics Research*, Vol. 109, 279–299, 2010.
22. Lin, C., A.-Y. Qing, and Q.-Y. Feng, "Synthesis of unequally spaced antenna arrays by using differential evolution," *IEEE Transactions on Antennas and Propagation*, Vol. 58, 2553–2561, 2010.
23. Mallipeddi, R., J. P. Lie, P. N. Suganthan, S. G. Razul, and C. M. S. See, "A differential evolution approach for robust adaptive beamforming based on joint estimation of look direction and array geometry," *Progress In Electromagnetics Research*, Vol. 119, 381–394, 2011.
24. Rodriguez-Gonzalez, J. A. and F. J. Ares-Pena, "Design of planar arrays composed by an active dipole above a ground plane with parasitic elements," *Progress In Electromagnetics Research*, Vol. 119, 265–277, 2011.
25. Razavi, A. and K. Forooraghi, "Thinned arrays using pattern search algorithms," *Progress In Electromagnetics Research*, Vol. 78, 61–71, 2008.
26. Chen, K.-S., X.-H. Yun, Z.-S. He, and C.-L. Han, "Synthesis of sparse planar arrays using modified real genetic algorithm," *IEEE Transactions on Antennas and Propagation*, Vol. 55, 1067–1073, 2007.
27. Kennedy, J. and R. C. Eberhart, "Particle swarm optimization," *Proceedings of the Conference on Neural Networks*, 1942–1948, Perth, Australia, 1995.
28. Modiri, A. and K. Kiasaleh, "Modification of real-number

and binary PSO algorithms for accelerated convergence," *IEEE Transactions on Antennas and Propagation*, Vol. 59, 214–224, 2011.

29. Kennedy, J. and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, 4104–4108, Orlando, FL, USA, 1997.
30. Menhas, M.-I., L. Wang, M.-R. Fei, and H. Pan, "Comparative performance analysis of various binary coded PSO algorithms in multivariable PID controller design," *Expert Systems with Applications*, Vol. 39, 4390–4401, 2012.
31. Zaharis, Z. D. and T. V. Yioultsis, "A novel adaptive beamforming technique applied on linear antenna arrays using adaptive mutated Boolean PSO," *Progress In Electromagnetics Research*, Vol. 117, 165–179, 2011.
32. Shi, X.-H. and R. C. Eberhart, "Empirical study of particle swarm optimization," *Proceedings of the Congress on Evolutionary Computation*, 1945–1950, Washington, D.C., USA, 1999.
33. Chatterjee, A. and P. Siarry, "Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization," *Computers & Operations Research*, Vol. 33, 859–871, 2006.
34. Chuang, L.-Y., C.-H. Yang, and J.-C. Li, "Chaotic maps based on binary particle swarm optimization for feature selection," *Applied Soft Computing*, Vol. 11, 239–248, 2011.
35. Dos Santos Coelho, L. and A. A. R. Coelho, "Model-free adaptive control optimization using a chaotic particle swarm approach," *Chaos, Solitons and Fractals*, Vol. 41, 2001–2009, 2009.
36. He, Y.-Y., J.-Z. Zhou, X.-Q. Xiang, H. Chen, and H. Qin, "Comparison of different chaotic maps in particle swarm optimization algorithm for long-term cascaded hydroelectric system scheduling," *Chaos, Solitons and Fractals*, Vol. 42, 3169–3176, 2009.
37. Shi, X.-H. and R. Eberhart, "A modified particle swarm optimizer," *Proceedings of the Congress on Evolutionary Computation*, 69–73, Anchorage, AK, USA, 1998.
38. Xu, S. and R. S. Yahya, "Boundary conditions in particle swarm optimization revisited," *IEEE Transactions on Antennas and Propagation*, Vol. 55, 760–765, 2007.
39. Deligkaris, K. V., Z. D. Zaharis, D. G. Kampitaki, S. K. Goudos, I. T. Rekanos, and M. N. Spasos, "Thinned planar array design using Boolean PSO with velocity mutation," *IEEE Transactions on Magnetics*, Vol. 45, 1490–1493, 2009.