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

# APPLICATION OF CHAOTIC PARTICLE SWARM OPTIMIZATION ALGORITHM TO PATTERN SYNTHESIS OF ANTENNA ARRAYS

**W.-B. Wang** †

School of Electric Information
Xihua University, Chengdu, Sichuan 610039, China

**Q.-Y. Feng and D. Liu**

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

**Abstract**—To deal with pattern synthesis of antenna arrays, a chaotic particle swarm optimization (CPSO) is presented to avoid the premature convergence. By fusing with the ergodic and stochastic chaos, the novel algorithm explores the global optimum with the comprehensive learning strategy. The chaotic searching region can be adjusted adaptively. To evaluate the performance of CPSO, several representative benchmark functions are minimized using various optimization algorithms. Numerical results demonstrate that the proposed approach improves the performance of the algorithm significantly, in terms of both the convergence speed and exploration ability. Moreover, CPSO was applied to array synthesis examples, including the equally spaced linear array, unequally spaced linear array and conformal array, compared with other optimization methods. Experimental results show its high performance in the pattern synthesis with low side lobe, multi-nulls and shaped beam.

## 1. INTRODUCTION

The purpose of antenna array synthesis is to find appropriate excitation vector and layout of the array that produces the radiation pattern

*Received 23 January 2011, Accepted 17 March 2011, Scheduled 28 March 2011*
Corresponding author: Wei-Bo Wang (okwwb@163.com).
† Also with School of Information Science & Technology, Southwest Jiaotong University, Chengdu, Sichuan 610031, China.

which is closest to the desired pattern. Antenna array synthesis plays a very important role in communication systems [1–7].

Various techniques have been developed to array synthesis. Classic techniques such as the Dolph-Chebyshev and Taylor methods have many practical difficulties in the array design especially if there are some restricted conditions [3]. Moreover, synthesis of antenna arrays also imposes tough challenges that mainly come from the nonlinear and non-convex dependency of the array factor to positions, excitation phases and amplitude of elements [7]. The constraints placed on steering nulls also increase the difficulty of synthesis. Furthermore, it is a tough task in the synthesis of conformal arrays which is conformal to a curved surface. New far field pattern behaviors emerge, and some traditional linear and planar array synthesis methods are not valid.

In recent years, several new optimization techniques have emerged. The evolutionary algorithms (EAs) for array synthesis have been extensively studied. Several global optimization algorithms such as differential evolution (DE) [1, 7], genetic algorithm (GA) [4, 6], simulated annealing (SA) [8], ant colony optimization (ACO) [9], particle swarm optimization (PSO) [10–14] are used in antenna array pattern. However, these methods present certain drawbacks with the possibility of premature convergence to a local optimum.

In this paper, a novel chaotic PSO algorithm (CPSO) is proposed. Based on the ergodicity, regularity and pseudo-randomness of the chaotic variable, chaotic search is used to explore better solutions. In order to verify its effectiveness and versatility, CPSO has been first applied to classical benchmark functions and then used for the optimization of antenna array synthesis.

This paper is organized as follows: Section 2 describes the principle of CPSO and its simulations on classical benchmark functions. Array synthesis problems are then addressed in Section 3, and the numerical results are compared with those obtained in literatures. Finally, the conclusions are discussed in Section 4.

## 2. PRINCIPLE OF CPSO

### 2.1. Basic Particle Swarm Optimization

Inspired by the social behaviors of animal, bird flocking and fishing, PSO was developed by Kennedy and Eberhat [16]. The particle is endowed with two factors: velocity and position which can be regarded as the potential solution in the $D$ dimension problem space. In basic PSO, they can be updated by following formulas:

$$v_{id}(t+1) = wv_{id}(t) + c_1 r_{1d}(p_{id}(t) - x_{id}(t)) + c_2 r_{2d}(p_{gd}(t) - x_{id}(t)) \quad (1)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \tag{2}$$

where $i = 1, \ldots, N$, $d = 1, \ldots, D$, $N$ is the number of particles. $w$ is the inertia weight factor to control the exploration and exploitation. $r_{1d}$ and $r_{2d}$ are two random numbers within the range [0, 1]. $v_{id}(t)$ and $x_{id}(t)$ are the velocity and position of the current particle $i$ at time step $t$ in the $d$th-dimensional search space respectively. When $v_{id}(t)$ and $x_{id}(t)$ are beyond the boundary, the solution may be illegal. So, the treatment of boundaries in the PSO method is important in order to prevent the swarm from explosion [28]. In many practical problems, the search range $x_{id}$ is in $[X_{\min}, X_{\max}]^D$. $v_{id}$ should be clamped to a maximum magnitude $V_{\max}$. $p_i$ is the previous best position of particle $i$ which is also called "personal best", and its $d$th-dimensional part is $p_{id}$. The "global best" $p_g$ is the best position found in the whole particles, and its $d$th-dimensional part is $p_{gd}$. $c_1$, $c_2$ are the acceleration constants which change the velocity of a particle towards the $p_i$ and $p_g$.

## 2.2. Modification Techniques in CPSO

The basic PSO uses $p_g$ as neighborhood topology. Each particle learns from its $p_i$ and $p_g$. Restricting the social learning part to $p_g$ can make basic PSO converge quickly. However, because all particles in the swarm learn from the $p_g$ even if the current $p_g$ is far from the global optimum, particles may easily be attracted to the area and trapped in a local optimum. Furthermore, the fitness value of a particle is determined by all dimensions. A particle that has discovered the region corresponding to the global optimum in some dimensions may have a low fitness value because of the poor solutions in other dimensions [17].

In order to acquire more beneficial information from the entire swarm, we define $p_c$ as "comprehensive best position".

$$p_c = \left\{ \frac{\sum_{i=1}^{N} p_{i1}}{N}, \frac{\sum_{i=1}^{N} p_{i2}}{N}, \ldots, \frac{\sum_{i=1}^{N} p_{id}}{N}, \ldots, \frac{\sum_{i=1}^{N} p_{iD}}{N} \right\} \tag{3}$$

where $i = 1, \ldots, N$. Thus Equation (1) is modified as

$$v_{id}(t+1) = wv_{id}(t) + c_1 r_{1d}(p_{id}(t) - x_{id}(t)) + c_2 r_{2d}(p_{cd}(t) - x_{id}(t)) \tag{4}$$

where $p_{cd}$ is the $d$th-dimensional part of $p_c$. By using $p_c$ instead of $p_g$, all particles' $p_i$ can potentially be used as the exemplars to guide their flying direction. The comprehensive learning strategy yields a larger potential search space than that of the basic PSO. On the other hand, a particle can learn from $p_g$, as well as its personal best and the other particles' best, so that the particle can learn from particle itself, the elite and other particles. The strategy can increase the initial diversity and enable the swarm to overcome premature convergence problem.

Basic PSO has shown some important advances by providing high speed of convergence in specific problems. However it does exhibit some shortages [18]. During the process of evolution, sometimes particles lose their abilities of exploration and will be stagnated. When some particles' velocity is be close to zero, other particles will quickly fly into the region near the inactive particles' position that guided by $p_i$ and $p_g$. Because of the particles' randomicity in initialization and evolution process, the updating sometimes looks aimless. As a result, when $p_g$ is trapped in a local optimum, the whole swarm becomes premature convergence, and the exploration performance will not be improved.

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 searches that depend on probabilities [19]. Chaotic PSO can be divided into two types. In the first type, chaos is embedded into the velocity updating equation of PSO. In [18], $c_1$ and $c_2$ are generated from the iterations of a chaotic map instead of using the rand function. In [20], a chaotic map is used to determine the value of $w$ during iterations. In the second type, chaotic search is fused with the procedures of PSO. This type is a kind of multi-phase optimization technique that chaotic optimization and PSO can switch to each other according to certain conditions [21].

Therefore, this paper provides a new strategy, which not only introduces chaotic mapping with certainty, ergodicity and stochastic property into PSO algorithm, but also proposes multi-phase optimization integrated by chaotic search and PSO evolution. The multi-phase optimization of chaotic PSO includes: $v_{id}$ and $x_{id}$ are updated by basic PSO with comprehensive learning strategy. If the swarm is stagnated, chaotic disturbance would be introduced.

Here, variance $\sigma^2$ demonstrates the converge degree of all particles.

$$\sigma^2 = \sum_{i=1}^{N} [(f_i - f_{avg})/f]^2 \tag{5}$$

$$f = \max\{1, \max\{|f_i - f_{avg}|\}\} \tag{6}$$

where $f_i$ is the fitness of the $i$th particle; $f_{avg}$ is the average fitness value; $f$ is the factor of fitness value. The bigger $\sigma^2$ is, the broader ithe particles will spread. Otherwise, they will almost converge.

The chaotic sequence can be generated by the logistic map introduced by Robert May in 1976. It is often cited as an example of how complex behavior can arise from a simple dynamic system without any stochastic disturbance [22]. The equation is the following

$$y_{id}(t+1) = \mu y_{id}(t)(1 - y_{id}(t)) \tag{7}$$

where $y_{id}(t) \in (0, 1)$, $i = 1, \ldots, N$, $d = 1, \ldots, D$. $\mu$ is usually set to 4 to obtain ergodicity of $y_{id}(t+1)$ within (0, 1). When the initial value $y_{id}(0) \notin \{0.25, 0.5, 0.75\}$, using Equation (7) we can obtain chaotic sequences.

In order to increase the population diversity and prevent premature convergence, we add adaptively chaotic disturbance upon $p_c$ at the time of stagnation. Thus, $p_c$ is modified as $p'_c$.

$$p'_{cd}(t+1) = p_{cd}(t) + R_{id}(2y_{id}(t) - 1) \tag{8}$$

where $p'_{cd}$ is the $d$th dimension part of $p'_c$, and $p_{cd}$ is the $d$th dimension part of $p_c$. The chaotic searching radius $R_{id}$ is defined as

$$R_{id} = \beta \left| p_{cd}(t) - p_{id}(t) \right| \tag{9}$$

where $\beta$ is the region scale factor. Because $y_{id} \in (0, 1)$, the second part of Equation (8) is in the range of $(-|R_{id}|, |R_{id}|)$ that would restrict the searching area around $p_c$. In addition, the searching range can be adaptively adjusted by the distance between $p_i$ and $p_c$. If $p_c$ is surrounded with the previous best positions $p_i$, it means that a good region may have been found, and it is reasonable to search elaborately in a small area. On the contrary, if $p_i$ is far from $p_c$, this probably suggests that a good area has not yet been found. For better solution, searching region should be enlarged [27].

Thus in CPSO, the new position can be expressed as

$$v_{id}(t+1) = wv_{id}(t) + c_1 r_{1d}(p_{id}(t) - x_{id}(t)) + c_2 r_{2d}(p'_{cd}(t) - x_{id}(t)) \tag{10}$$

Different from Equation (1), $p_g$ is replaced by $p'_c$ at the time of stagnation when $\sigma^2$ is less than the stagnation factor $\xi$. Chaotic search is restricted into a small range to obtain high performance in local exploration. Additionally, the algorithm keeps a dynamic balance between global and local searches due to its adaptive mechanism. With the new updating rule, different exemplars are used in different dimensions to explore a larger search space than the basic PSO. In addition, chaotic disturbance is embedded in different dimensions to maintain the diversity which plays an important role in avoiding early convergence.

### 2.3. Simulation of CPSO to Benchmark Functions

To verify its effectiveness, CPSO has been applied to classical benchmark functions. All simulations are conducted in a Windows 7

Professional OS using 12-core processors with Intel Xeon(R), 3.33 GHz, 72 GB RAM, and the codes were implemented in Matlab 7.10.

In this section, four benchmark functions including unimodal and multimodal functions in [17] are employed. Test functions, search range $[X_{\min}, X_{\max}]$ and optimal goal for functions are listed in Table 1.

**Table 1.** Benchmark functions.

| Function Name | Search Range | Goal | Dimension |
|---|---|---|---|
| Sphere | [−100, 100] | 0.01 | 30 |
| Rastrigin | [−5.12, 5.12] | 100 | 30 |
| Ackley | [−32, 32] | 0.1 | 30 |
| Griewank | [−600, 600] | 0.1 | 30 |

Experiments were carried out with 3000 iterations for the population size of 50. All experiments were run 30 times. $c_1 = c_2 = 2$, $w$ is linearly decreased from 0.9 to 0.4 during the iterations. Region scale factor $\beta$ is set to 0.3, and stagnation factor $\xi$ is equal to 0.15.

The initiation of CPSO is as follows. $x_{id} \in [0, X_{\max}]$, $v_{id} \in [-V_{\max}/2, V_{\max}/2]$, where the maximum velocity $V_{\max} = 0.5(X_{\max} - X_{\min})$.

The reflecting boundary conditions discussed in [27, 28] are used to ensure particles to search inside the solution space. During the iterations, if $v_{id} > V_{\max}$, then $v_{id} = \mathrm{sign}(x_{id})V_{\max}$. If $x_{id} > X_{\max}$, then $x_{id} = 2X_{\max} - x_{id}$, and if $x_{id} < X_{\min}$, then $x_{id} = 2X_{\min} - x_{id}$.

The performance of CPSO is compared with inertia weight PSO (WPSO) [23], fully informed PSO (FIPSO) [24], perturbation PSO (PPSO) [25]. The simulation results such as the average iteration time, success rate, average best fitness value, standard variation are shown in Table 2. The average simulation time for different algorithms to reach the optimal goal is also obtained. The algorithm is thought to fail in evolution if the fitness cannot reach the goal after the maximum iteration time. The best results are listed in bold in Table 2. In the form of logarithm with base 10, the convergence of mean best fitness values of benchmark functions is shown in Figure 1.

From Table 2, we can see that the CPSO obviously perform better than other three algorithms. It converges successfully and quickly in all functions. CPSO can achieve better evolution results after maximum iteration except in the multimodal function Greiwank. The average best fitness value achieved by FIPSO in Greiwank is better than other algorithms, though convergence speed of FIPSO in Greiwank is the slowest. It is probably due to its strategy of fully informed goniometry which can obtain better results with slow evolutionary progress. From

**Table 2.** Simulation results on benchmark functions.

| Benchmark functions | Applied algorithm | Average iterations | Success rate (%) |
|---|---|---|---|
| Sphere | WPSO | 736 | 1 |
| | FIPSO | 553 | 1 |
| | PPSO | 469 | 1 |
| | CPSO | **346** | 1 |
| Rastrigin | WPSO | 420 | 1 |
| | FIPSO | 303 | 1 |
| | PPSO | 346 | 1 |
| | CPSO | **292** | 1 |
| Ackley | WPSO | 617 | 0.95 |
| | FIPSO | 542 | 1 |
| | PPSO | 921 | 1 |
| | CPSO | **507** | 1 |
| Griwank | WPSO | 605 | 1 |
| | FIPSO | 710 | 1 |
| | PPSO | 455 | 1 |
| | CPSO | **420** | 1 |

| Benchmark functions | Average value | Standard variation | Average time (ms) |
|---|---|---|---|
| Sphere | 6.47e-29 | 7.23e-29 | 1542 |
| | 2.45e-34 | 4.56e-33 | 1769 |
| | 2.32e-30 | 1.03e-30 | 1593 |
| | **5.19**e-**36** | 5.47e-36 | **1301** |
| Rastrigin | 66.23 | 32.59 | 1357 |
| | 74.12 | 16.64 | 1415 |
| | 91.86 | 36.56 | 1246 |
| | **61.63** | 15.52 | **1004** |
| Ackley | 4.3e-15 | 1.2e-15 | 1271 |
| | 3.15e-14 | 2.31e-14 | 1647 |
| | 4.21e-8 | 1.34e-8 | 1456 |
| | **7.35**e-**17** | 4.22e-17 | **1129** |
| Griwank | 7.69e-3 | 8.47e-3 | 1216 |
| | **1.03**e-**6** | 1.98e-6 | 1465 |
| | 0.04 | 0.02 | 1097 |
| | 3.51e-4 | 1.73e-4 | **986** |

Figure 1, in Sphere, Rastrigin and Ackley, CPSO converges fastest. In Griewank, CPSO converges quickly. It is obvious that CPSO performs better than WPSO and PPSO due to the adaptive mechanism of the chaotic search and a good balance of exploration and exploitation.

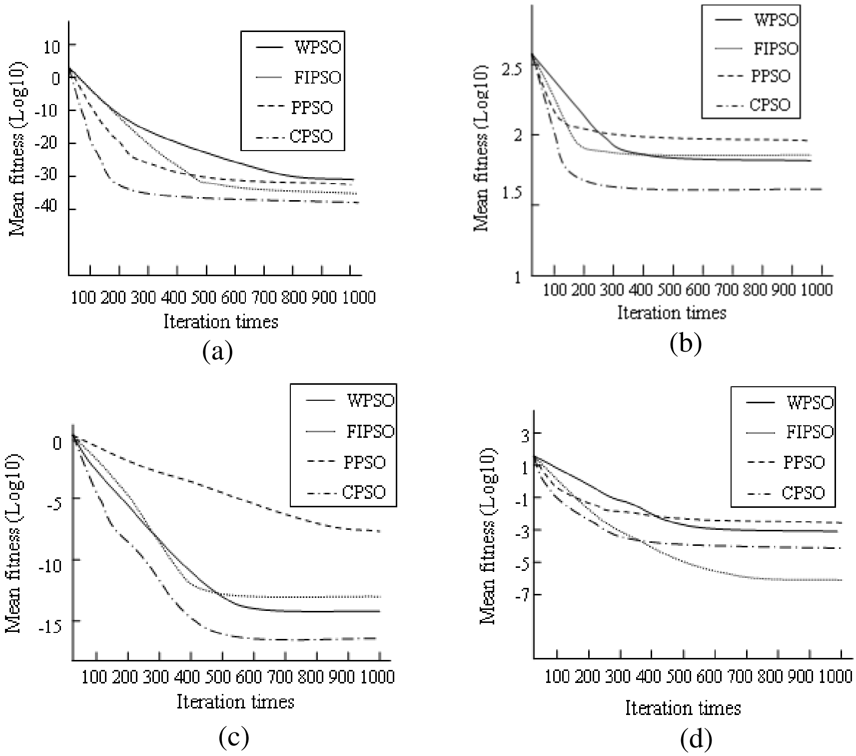



**Figure 1.** Convergence of mean fitness for benchmark functions: (a) Sphere. (b) Rastrigin. (c) Ackley. (d) Griewank.

## 3. ARRAY PATTERN SYNTHESIS USING CPSO

The capabilities and versatility of the proposed CPSO algorithm will be assessed by presenting three different array types: an equally spaced linear array, an unequally spaced linear array and a conformal array.

Parameters setting of CPSO is the same as in the previous simulation in which the parameters are testified effectively. Note that the parameter $D$ is different according to the number of array elements.

To reduce side lobe, steer nulls and shape beams, we define the following objective function in array synthesis.

$$f = \alpha_1 \left| PSLL_c - PSLL_d \right| + \alpha_2 \sum^{\theta_{Null}} \left| MNULL_c - MNULL_d \right| + \alpha_3 \sum^{\theta_{Null}} NULL_s + \alpha_4 \left| BW_c - BW_d \right| + \alpha_5 \left| B_c - B_d \right| \quad (11)$$

where $PSLL_c$ is the calculated peak side lobe level, and $PSLL_d$ is the desired peak side lobe level. $\theta_{Null}$ is the desired null position. $MNULL_c$ denotes the calculated maximum null depth level at prescribed nulls. $MNULL_d$ is the desired maximum null depth level. $NULL_s$ is the variance value of null depth level at prescribed nulls. $BW_c$ and $BW_d$ are the calculated and desired beam widths. $B_c$ and $B_d$ are the calculated beam and desired beams from $\theta_{B1}$ to $\theta_{B2}$. Weighting factors $\alpha_1$ to $\alpha_5$ can be tuned in different array pattern synthesis.

Synthesis times depend on the computing amount or complexity. In the same hardware platform, the computing amount mainly includes two parts: the time cost when updating the position and velocity in PSO, and the time consumption using the position to compute fitness functions. As for array pattern synthesis, the latter part costs much more computing time than the former one. So, the number of fitness function evaluations can be used to indicate synthesis time. In this paper, the number of fitness function evaluations for different algorithms is the same, which means that their synthesis times are almost equivalent. Thus, the results of synthesized pattern can be used to compare the capability of different algorithms.

### 3.1. Equally Spaced Linear Array

Basically, if mutual coupling effects are neglected, and dependence on azimuth angle $\varphi$ with respect to $X$-axis is omitted, the far field radiation pattern of a $Z$-directed linear array at a certain direction given by the elevation angle $\theta$ can be written as [10]

$$FF(\theta) = EP(\theta) \cdot AF(\theta) \tag{12}$$

where $EP(\theta)$ is the element pattern. For a linear array consisting of symmetric $2K$ elements, the array factor $AF$ is given by

$$AF(I, x, \varphi, \theta) = \sum_{n=-K}^{K} I_n e^{j\left(\frac{2\pi}{\lambda} x_n \sin\theta + \varphi_n\right)} \tag{13}$$

In the synthesis of a linear array which is equally spaced with uniform excitation phase ($\varphi_n = 0$ for all elements), Equation (13) becomes

$$AF(I, \theta) = \sum_{n=1}^{2K} I_n e^{j\left(\frac{2\pi(n-1)d}{\lambda} \sin\theta\right)} \tag{14}$$

where $d$ is the equal inter-element spacing distance. To validate the effectiveness of CPSO, we first discuss a 20 elements half-wavelength-spaced linear array [11]. The excitation amplitude is symmetric with respect to the center of the linear array. Only 10 amplitudes are to be

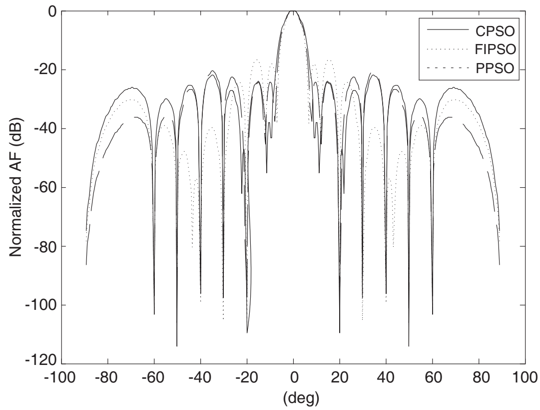



**Figure 2.** Pattern of 20-element array with equally space.

**Table 3.** Performance comparisons of different algorithms.

| Algorithm | $PSLL_c$ (dB) | $MNULL_c@\theta_{Null}$ (dB) | | | | |
|---|---|---|---|---|---|---|
| | | 20° | 30° | 40° | 50° | 60° |
| FIPSO | −20.58 | −97.4 | −97.73 | −97 | −104.14 | −96.89 |
| PPSO | −16.84 | −106.6 | −105.3 | −99.44 | −97.83 | −101.1 |
| CPSO | −21.95 | −109.6 | −96.17 | −114.3 | −105.9 | −103.2 |

optimized. The array is designed with lower side lobe level (SLL) suppression in the region [−100°, 100°], with $PSLL_d$ −15 dB and prescribed nulls at 20°, 30°, 40°, 50° and 60° with $MNULL_d$ −95 dB.

Here, $\alpha_1 = 0.8$, $\alpha_2 = 1$, $\alpha_3 = 0.2$ and $\alpha_4 = \alpha_5 = 0$.

Contrast to FIPSO [24] and PPSO [25], the array pattern obtained by CPSO is shown in Figure 2. Table 3 lists the performance comparisons of different algorithms. The computed amplitudes of elements optimized by CPSO are shown in Table 4. Table 5 illustrates the average synthesis times using CPSO after maximum iterations.

Figure 2 illustrates that the results optimized by CPSO can satisfy the requirements. The performance can be observed in detail from Table 3 that the $MNULL_c$ at $\theta_{Null}$ obtained by CPSO prevails against both FIPSO and PPSO at 20°, 40°, 50° and 60° except at 30°. Note that the returned $PSLL_c$ of CPSO is much better than those of FIPSO and PPSO. As a whole, the CPSO algorithm can easily achieve the optimization goal. From Table 3, it is observed that the suppression of prescribed nulls is obtained by CPSO as well as other optimization methods, and the SLL achieved by CPSO is 5.11 dB less than that

**Table 4.** Computed amplitude optimized by CPSO.

| Number | 1/20 | 2/19 | 3/18 | 4/17 | 5/16 |
|---|---|---|---|---|---|
| Amplitude | 0.1121 | 0.4275 | 0.2342 | 0.4392 | 0.6191 |
| Number | 6/15 | 7/14 | 8/13 | 9/12 | 10/11 |
| Amplitude | 0.6593 | 0.6374 | 0.8430 | 0.9444 | 0.9992 |

**Table 5.** Average synthesis times.

| Synthesis examples | Algorithm (minutes) | | |
|---|---|---|---|
| Equally arrays | CPSO (21.39) | FIPSO (24.41) | PPSO (23.72) |
| Unequally arrays | CPSO (27.29) | | |
| Conformal arrays | CPSO (36.18) | | |

of PPSO and 1.37 dB less than that of FIPSO, which verifies the advantage of the proposed algorithm.

From Table 5, the average synthesis times spending by CPSO and other algorithms are at the same level. The average synthesis times spending by CPSO is smaller than those of other two algorithms. Moreover, it is obvious that the computing times using the particles' position to compute fitness functions in array synthesis are much more than those in benchmark function simulations. These results support what have been discussed above.

### 3.2. Synthesis of Unequally Spaced Linear Array

Compared with the equally spaced array, unequally spaced array enables lower SLL and reduced number of elements. Furthermore, a low SLL design can be obtained by using uniform amplitude excitation of the unequally spaced array to reduce system cost and difficulties in designing feeding network [7]. If we further assume uniform excitation of phase and amplitude ($I_n = 1$ and $\varphi_n = 0$ for all elements), for the synthesis of a position-only and unequally spaced array with symmetric $2K$-element, the array factor is [13]

$$AF(x, \theta) = 2 \sum_{n=1}^{K} e^{j\left(\frac{2\pi}{\lambda} x_n \sin\theta\right)} \tag{15}$$

In the second example, a 32-element array at the direction of 9° with the desired null level at −60 dB is considered. The desired beam width is set to 7.1° ± 14% [13].

Here, $\alpha_1 = 0.3$, $\alpha_2 = 0.5$, $\alpha_3 = 0.1$, $\alpha_4 = 0.5$ and $\alpha_5 = 0$.

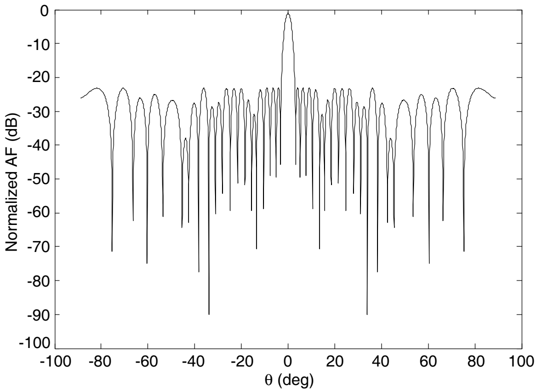



**Figure 3.** Pattern of 32-element array with unequally space.

**Table 6.** Geometry of the 32-element array.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Number | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Position | 0.499 | 0.510 | 0.493 | 0.592 | 0.477 | 0.768 | 0.459 | 0.397 |
| Number | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| Position | 0.235 | 0.384 | 0.035 | 0.296 | 0.122 | 0.206 | 0.766 | 0.036 |

Figure 3 presents the array pattern optimized by CPSO. The normalized array geometry is given in Table 6. The average synthesis time using CPSO after maximum iterations is shown in Table 5.

In Figure 3, $PSLL_c$ found by CPSO is $-23.17$ dB, while that in [13] was $-22.75$ dB and in [26] was $-18.80$ dB. Compared to $-60$ dB in [13] and $-62.12$ dB in [26], the null level at 9° found by CPSO is $-63.16$ dB.

## 3.3. Synthesis of Conformal Array

In [8] and [15], $M \times K$ ($8 \times 8$ in those papers) elements are uniformly spaced by $\lambda/2$ and located on a cylindrical surface with radius $\tau = 5\lambda$ and height $h = (K-1)\lambda/2$. $M$ and $K$ are the number of elements in the $\Phi$- and $Z$-directions. The far-field radiation pattern of the cylindrical array can be computed as

$$FF(I,\chi,\theta,\Phi) = \sum_{k=1}^{K}\sum_{m=1}^{M}\Big[I_{mk}\cdot EP(\theta,\Phi-\Phi_m) \cdot e^{j\{\chi_{mk}+\left(\frac{2\pi}{\lambda}\right)(x_{mk}\sin\theta\cos\Phi+y_{mk}\sin\theta\sin\Phi+z_{mk}\cos\theta)\}}\Big] \quad (16)$$

where $\chi$ is the excitation phase, and its $mk$th part is $\chi_{mk}$. $\Phi$ is the element's azimuth position, and its $m$th-column part in array is $\Phi_m$. Contrary to linear and planar arrays, the element patterns and array factor are not separable for conformal arrays [29]. Each element is towards a different direction. Therefore, $EP(\theta, \Phi)$ is expressed as

$$EP(\theta, \Phi) = \sin\theta \sin\Phi \tag{17}$$

The positions of the $mk$th element in the conformal array are

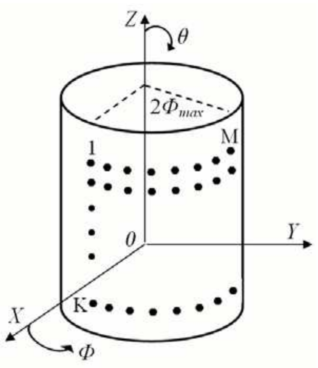
$$\begin{cases} x_{mk} = \tau\cos\Phi_m \\ y_{mk} = \tau\sin\Phi_m \\ z_{mk} = -\frac{h}{2} + (k-1)\frac{\lambda}{2} \end{cases} \tag{18}$$

Range from $-\Phi_{\max}$ to $\Phi_{\max}$, $\Phi_m$ can be given by

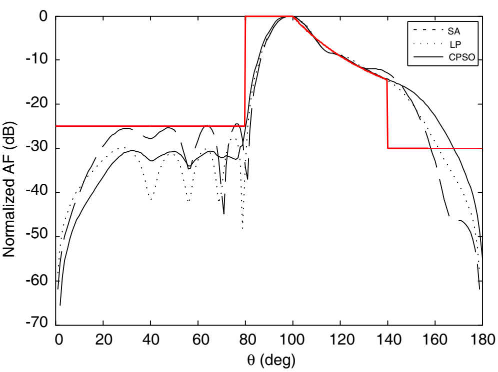
$$\Phi_m = -(M-1)\arcsin\left(\frac{\lambda}{4\tau}\right) + 2(m-1)\arcsin\left(\frac{\lambda}{4\tau}\right) \tag{19}$$

Now the statement of the current problems is simply expressed. Thus our goal is to use the PSO algorithm to adjust the positions or amplitude for the array elements that can result in an array beam with minimum SLL and nulls at specific directions.

In this case, the synthesis of a 3D pattern using a conformal array with $8 \times 8$ elements is addressed. A simulated annealing technique (SA) [8] and linear programming procedure (LP) [15] have been adopted to solve this example. The desired pattern mask has a radar-shaped beam with a cosecant square elevation pattern from 100° to




(a)




(b)

**Figure 4.** (a) Geometry of the $8 \times 8$-elements cylindrical array. (b) Radiation pattern in $\Phi = 0$ plane.

140°. The SLL outside the shaped beam region should remain below −25 dB.

Here, $\alpha_1 = 0.4$, $\alpha_2 = 0$, $\alpha_3 = 0$, $\alpha_4 = 0$ and $\alpha_5 = 0.6$.

The geometry of cylindrical array and the radiation patterns synthesized by SA [8], LP [15] and CPSO are plotted in Figure 4. The amplitudes and phases of the excitation coefficients optimized by CPSO are shown in Table 7. The average synthesis time using CPSO after maximum iterations is shown in Table 5.

**Table 7.** Excitation coefficients ($I_{mk}/\chi_{mk}$ (Deg)).

| $k$ | $m = 1, 8$ | $m = 2, 7$ | $m = 3, 6$ | $m = 4, 5$ |
|---|---|---|---|---|
| 1 | 2.158/151.66 | 2.020/36.12 | 1.687/169.40 | 0.041/111.68 |
| 2 | 0.684/63.70 | 0.435/173.76 | 0.994/36.22 | 3.255/45.17 |
| 3 | 0.566/122.38 | 1.716/87.60 | 3.781/100.69 | 1.805/69.06 |
| 4 | 3.481/230.36 | 4.046/187.62 | 4.759/90.37 | 0.525/54.52 |
| 5 | 0.712/166.73 | 2.830/309.15 | 2.709/195.46 | 4.201/107.27 |
| 6 | 0.314/217.57 | 4.083/129.04 | 2.223/216.23 | 0.862/105.07 |
| 7 | 3.408/223.76 | 1.307/320.41 | 0.672/153.27 | 0.499/68.22 |
| 8 | 3.206/70.01 | 3.401/244.98 | 2.150/26.90 | 3.917/148.90 |

From Figure 4(b), compared with SA and LP, the results obtained by CPSO have the smallest average deviation and ripple in the cosecant square region. The $PSLL_c$ outside the shaped beam region remains below −30.62 dB while that in [15] is −24.5 dB and in [8] −27.51 dB.

## 4. CONCLUSIONS

This paper illustrates the use of PSO in the pattern synthesis of antenna arrays. A hybrid algorithm is proposed by fusing the advantages of both chaotic search and PSO to avoid the prematurity and easy trapping in local optimum. The global best position in basic PSO is replaced by a comprehensive position obtained by the new learning strategy in which a particle makes each dimension learn from the corresponding dimension of other particles' best positions.

The statistical results obtained from the benchmark functions demonstrate the superiority of the proposed CPSO to WPSO, FIPSO and PPSO. This is because CPSO integrates the adaptive and chaotic mechanism into canonical PSO algorithm. In addition, numerical examples of synthesis problems have been presented. In the amplitude synthesis of equally spaced linear array, the peak side lobe level found by CPSO is 5.11 dB less than that using PPSO in [25]. For position-only synthesis of unequally spaced linear array, the peak side lobe level found by CPSO is 4.37 dB less than that in [26], and the null level at certain direction obtained by CPSO is 3.16 dB less than that in [13]. In the application of the conformal array synthesis, the peak side lobe level outside the shaped beam region found by CPSO is 6.12 dB less than that using LP in [15] and 3.11 dB less than that using SA in [8]. Deviation and ripple in the cosecant square region obtained by CPSO are the smallest. The results of the proposed algorithm indicate its potential ability in the antenna designs for a wide class of electromagnetic applications.

## ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (Nos. 10876029, 60990320, 60990323) and the National 863 Project of China under Grant (No. 2009AA01Z230).

## REFERENCES

1. 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.
2. Lanza Diego, M., J. R. Perez Lopez, 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.
3. Qu, Y., G. Liao, S.-Q. Zhu, and X.-Y. Liu, "Pattern synthesis of planar antenna array via convex optimization for airborne forward looking radar," *Progress In Electromagnetics Research*, Vol. 84, 1–10, 2008.
4. Zhang, S., S.-X. Gong, Y. Guan, P.-F. Zhang, and Q. Gong, "A novel IGA-EDSPSO hybrid algorithm for the synthesis of sparse arrays," *Progress In Electromagnetics Research*, Vol. 89, 121–134, 2009.
5. Oliveri, G. and L. Poli, "Synthesis of monopulse sub-arrayed linear and planar array antennas with optimized sidelobes," *Progress In Electromagnetics Research*, Vol. 99, 109–129, 2009.
6. Vaitheeswaran, S. M., "Dual beam synthesis using element

position perturbations and the G3-GA algorithm," *Progress In Electromagnetics Research*, Vol. 88, 43–61, 2008.

7. Lin, C., A.-Y. Qing, and Q.-Y. Feng, "Synthesis of unequally spaced antenna arrays by using differential evolution," *IEEE Trans. Antennas and Propagation*, Vol. 58, 2553–2561, 2010.
8. Ferreira, J. A. and F. Ares, "Pattern synthesis of conformal arrays by the simulated annealing technique," *Electronics Letters*, Vol. 33, 1187–1189, 1997.
9. Hosseini, S. A. and Z. Atlasbaf, "Optimization of side lobe level and fixing quasi-nulls in both of the sum and difference patterns by using Continuous Ant Colony Optimization (ACO) method," *Progress In Electromagnetics Research*, Vol. 79, 321–337, 2008.
10. Perez Lopez, J. R. and J. Basterrechea, "Hybrid particle swarm-based algorithms and their application to linear array synthesis," *Progress In Electromagnetics Research*, Vol. 90, 63–74, 2009.
11. Liu, D. and Q.-Y. Feng, "Pattern synthesis of antennas based on modified PSO algorithm with stagnation detection," *Journal of Radio Science*, Vol. 24, 697–701, 2009.
12. Li, W.-T., et al., "An extended particle swarm optimization algorithm for pattern synthesis of conformal phased arrays," *International Journal of RF and Microwave Computer-aided Engineering*, Vol. 20, 190–199, 2010.
13. Goudos, S. K., et al., "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, 125–129, 2010.
14. Shavit, R. and I. Taig, "Array pattern synthesis using neural networks with mutual coupling effect," *IEE Proc. Microwaves, Antennas and Propagation*, Vol. 152, 354–358, 2005.
15. Fuchs, B., "Shaped beam synthesis of arbitrary arrays via linear programming," *IEEE Antennas and Wireless Propagation Letters*, Vol. 9, 481–484, 2010.
16. Kennedy, J. and R. Eberhart, "Particle swarm optimization," *IEEE Proc. International Conference on Neural Networks*, 1942–1948, Perth, Aust, November 1995.
17. Liang, J.-J., et al., "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Trans. Evolutionary Computation*, Vol. 10, 281–295, 2006.
18. Modares, H., A. Alfi, and M. M. Fateh, "Parameter identification of chaotic dynamic systems through an improved particle swarm optimization," *Expert Systems with Applications*, Vol. 37, 3714–

3720, 2010.

19. Coelho, L. D. and A. A. Coelho, "Model-free adaptive control optimization using a chaotic particle swarm approach," *Chaos, Solitons and Fractals*, Vol. 41, 2001–2009, 2009.
20. Wang, Y., et al., "Improved chaotic particle swarm optimization algorithm for dynamic economic dispatch problem with valve-point effects," *Energy Conversion and Management*, Vol. 51, 2893–2900, 2010.
21. Wu, Q., "A self-adaptive embedded chaotic particle swarm optimization for parameters selection of Wv-SVM," *Expert Systems with Applications*, Vol. 38, 184–192, 2011.
22. He, Y.-Y., et al., "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.
23. Shi, Y. and R. Eberhart, "Modified particle swarm optimizer," *Proceedings of the 1998 IEEE International Conference on Evolutionary Computation*, 69–73, Anchorage, USA, May 1998.
24. Mendes, R., J. Kennedy, and J. Neves, "The fully informed particle swarm: Simpler, maybe better," *IEEE Trans. Evolutionary Computation*, Vol. 8, 204–210, 2004.
25. Jin, R.-H., et al., "The pattern synthesis of antennas based on a modified PSO algorithm," *Journal of Radio Science*, Vol. 21, 873–878, 2006.
26. 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 and Propagation*, Vol. 53, 2674–2679, 2005.
27. Lin, C. and Q.-Y. Feng, "Chaotic particle swarm optimization algorithm based on the essence of particle swarm," *Journal of Southwest Jiaotong University*, Vol. 42, 665–669, 2007.
28. Xu, S.-H. and Y. Rahmat-Samii, "Boundary conditions in particle swarm optimization revisited," *IEEE Trans. Antennas and Propagation*, Vol. 55, 760–765, 2007.
29. Zhou, H.-J., B.-H. Sun, J.-F. Li, and Q.-Z. Liu, "Efficient optimization and realization of a shaped-beam planar array for very large array application," *Progress In Electromagnetics Research*, Vol. 89, 1–10, 2009.