Progress In Electromagnetics Research, PIER 89, 121–134, 2009

A NOVEL IGA-EDSPSO HYBRID ALGORITHM FOR THE SYNTHESIS OF SPARSE ARRAYS

S. Zhang, S.-X. Gong, Y. Guan, P.-F. Zhang, and Q. Gong

National Key Laboratory of Antennas and Microwave Technology
Xidian University
Xi'an, Shaanxi 710071, China

Abstract—Based on the improvements of both Genetic Algorithm and Particle Swarm Optimization, a novel IGA-edsPSO(Improved Genetic Algorithm-extremum disturbed simple Particle Swarm Optimization) Hybrid algorithm is proposed in this paper. An improved performance of GA is achieved by reducing the array space. By discarding the particle velocity vector in the PSO evolutionary equation, the sPSO (simple PSO) can avoid the problem of slow later convergence velocity and low precision caused by determining the maximal velocity vector factitiously. And the edsPSO can overstep local extremum point more effectively with the help of the extremum disturbed factor. The proposed IGA-edsPSO Hybrid algorithm is used in the design of the sparse arrays with minimum element spacing constraint. Given the array aperture and the number of the array elements, the suppression of the peak sidelobe level (PSLL) with a certain half power beamwidth (HPBW) restriction is implemented with a high efficiency by optimizing the HPBW and PSLL synchronously. The simulation results show that faster convergence velocity (which means less computation time) and lower sidelobe level are obtained using IGA-edsPSO compared to IGA, standard PSO, GA-PSO and GA-sPSO.

1. INTRODUCTION

The maximum Peak SideLobe Level (PSLL) of array antennas is an important parameter in judging antenna performance [1, 2]. However, it is quite difficult to determine a Sidelobe Level satisfying the requirements in the practical array synthesis. The synthesis of

Corresponding author: S. Zhang (41539766@qq.com).

unequally spaced arrays [3–5] is more difficult compared to equally spaced ones. The element spacing of the sparse arrays is greater than the half-wave length. The radiation pattern and sidelobe level of the array can be controlled by optimizing the element spacing satisfying a certain constraint condition. The radiation pattern of the sparse array can be further controlled by optimizing the feeding magnitude and phase. As the distance between each two elements of the sparse array can be freely adjusted under the condition of some constraints, it is necessary to search for the best result using numerical optimization method.

On the basis of biological evolution theory, Genetic algorithm (GA) [6–13] accomplishes the search by simulating the natural selection, crossover and mutation in the course of biological evolution, thus possesses an intrinsic flexibility to nonlinear optimization problem. However, GA has a low convergence velocity and may even stagnate when searching near the extremum because of its lack of the local area searching mechanism.

Particle swarm optimization (PSO) algorithm [14–19], in which the optimization is performed in terms of the colony aptitude produced by the cooperation and competition among the particles, carries out the search according to its own velocity. It can memorize the best solution of each particle so far, which results in a higher convergence velocity. However, it also has some demerits. For example, as all the particles fly to the same optimum direction and tend to be identical, the convergence velocity of the PSO in the later evolution process becomes lower, which easily leads to relapsing into local extremum.

The Hybrid algorithm is formed by combining the advantages of GA and PSO [20], which is of great significance in the design and synthesis of Sparse Arrays. IGA-edsPSO Hybrid algorithm is proposed in this paper, which has faster convergence velocity (which means less computation time) and lower sidelobe level than IGA, standard PSO, GA-PSO and GA-sPSO.

2. IGA AND FITNESS FUNCTION

2.1. Optimal Variable and Fitness Function

Let us consider a one-dimensional linear array which is shown in Fig. 1.

The array consists of N-element positioned randomly with the first element as the origin of the coordinate system. For this array of

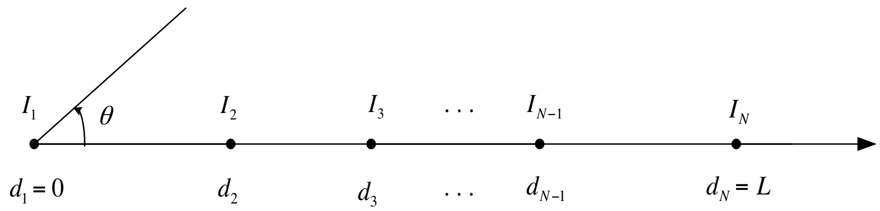

Figure 1. Geometry of sparse linear array.

isotropic ideal elements, the array factor can be written as:

$$E(\theta) = \sum_{n=1}^{N} I_n \exp(jkd_n \cos\theta) \tag{1}$$

where I_n and d_n represent the amplitude of the excitation of the nth element and the distance of the nth element from the origin, respectively; θ represents the steering angle from the broadside of the array satisfying $0 \leq \theta \leq \pi$; and the wavenumber $k = 2\pi/\lambda$, in which λ represents the wavelength. Let $d_1 = 0$, $d_N = L$, thus the aperture dimension is always L. The optimization problem could be expressed as searching for the optimum solution of element excitation amplitude and coordinate vector to minimize the PSLL of the array subjected to the constraints of $\min\{d_i - d_j\} \geq c_0$ and $\max\{d_i - d_j\} \leq c_0'$, $1 \leq j < i \leq N$, where c_0 is the design constraint of the minimum element spacing, and c_0' is that of the maximum one.

Let the optimal variable $\{I_1, I_2 \cdots, I_N, d_1, d_2, \cdots, d_N\}^T$ act as an individual, which means it is a real vector. The coding scheme of GA is real-code. The goal is to minimize the PSLL of the sparse array with a certain HPBW restriction, and the fitness function is defined as follows:

$$\mathit{fitness}(I_1, \cdots, I_N, d_1, \cdots, d_N) = \min\left\{\omega_1 * 20 * \log\left|\frac{E(\theta)}{FF_{\max}}\right| + \omega_2 * HPBW\right\} \tag{2}$$

where $FF_{\max}$ is the peak of the main beam; ω_1 and ω_2 are both weights; HPBW is the half power beamwidth; and the range of θ in which the fitness is valid should exclude the main beam region. Through many experimental attempts, it is found that a better result can be obtained while $\omega_1 = 2.8$ and $\omega_2 = 1.2$.

2.2. Real Coding and Starting Population

2.2.1. Individual

The real coding scheme is used in this paper. First N random real numbers subjected to U(0,1) distribution are produced, then they are expressed as the vector of excitation amplitude $\{I_1, I_2 \cdots, I_N\}$.

Let the coordinate of the first element be 0, and that of the Nth one be L. So $(N-2)$ elements are placed randomly over the aperture. There is a region of the size of $(N-1)c_0$ in which elements are not placed in order to satisfy the design constraint of minimum element spacing c_0. Therefore, the size of the remaining region over the aperture becomes:

$$Y = L - (N-1)c_0 \tag{3}$$

Because the constraint of minimum element spacing is satisfied by using formula (3), much running time is saved in judging whether the condition is satisfied. Then $(N-2)$ random real numbers in the range of $[0, Y]$ are produced using a random-number producer. And their ranking sequence is reset so that they are increasing by degrees, then the random real number sequence can be expressed as the vector notation $S = \{0, e_2, e_3, \cdots, e_{N-1}, L\}^T$, where $0 \leq e_2 \leq e_3 \leq \cdots \leq e_{N-1} \leq L$. And the element coordinate vector is obtained as follows:

$$d = S + P \tag{4}$$

where $P = \{0, c_0, 2c_0, \cdots, (N-1)c_0, 0\}^T$.

2.2.2. Starting Population

The matrix of starting population F (in which each column is an individual) containing M individuals and the matrix X of the constraint of element spacing are expressed as follows:

$$F = \begin{bmatrix} I_{1,1} & I_{1,2} & \cdots & I_{1,M} \\ I_{2,1} & I_{2,2} & \cdots & I_{2,M} \\ \vdots & \vdots & \ddots & \vdots \\ I_{N,1} & I_{N,2} & \cdots & I_{N,M} \\ 0 & 0 & \cdots & 0 \\ d_{2,1} & d_{2,2} & \cdots & d_{2,M} \\ \vdots & \vdots & \ddots & \vdots \\ d_{N-1,1} & d_{N-1,2} & \cdots & d_{N-1,M} \\ L & L & \cdots & L \end{bmatrix}, \quad X = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ c_0 & c_0 & \cdots & c_0 \\ 2c_0 & 2c_0 & \cdots & 2c_0 \\ \vdots & \vdots & \ddots & \vdots \\ (N-2)c_0 & (N-2)c_0 & \cdots & (N-2)c_0 \\ 0 & 0 & \cdots & 0 \end{bmatrix} \tag{5}$$

2.3. Genetic Processing

2.3.1. Selection

The method of competition selection is used in this paper, and elitism is also employed in order to assure convergence. Two individuals are chosen randomly each time, and the one whose fitness is smaller is included in the population of the next generation. This is done until the size of the population is satisfied.

2.3.2. Crossover and Mutation

The preprocessing matrix Z which is formed through the genetic prepocessing and not the individual real-coding themselves is obtained as follows:

$$Z = F_1 - X \tag{6}$$

Take two individuals for example:

$$\begin{aligned} Z_1 &= \{I_{1,1}, I_{2,1} \cdots, I_{N,1}, d_{1,1}, d_{2,1}, \cdots, d_{N,1}\}^T \\ Z_2 &= \{I_{1,2}, I_{2,2} \cdots, I_{N,2}, d_{1,2}, d_{2,2}, \cdots, d_{N,2}\}^T \end{aligned}$$

Two different crossover strategies are used here. The uniform crossover strategy is used for the first N genes, and the single-point crossover strategy is used for the last N genes with the crossover probability of 0.8.

The mutation probability in this paper is 0.05. By reseting the last N genes of each individual from small to big after the mutation operation, we can obtain the new preprocessing matrix Z'. It has been demonstrated that the produced two individuals also satisfy the constraint of element spacing. So the offspring population is expressed as follows:

$$F_2 = Z' + X \tag{7}$$

3. IMPROVED PSO

3.1. PSO

The canonical particle swarm algorithm loops which are shown in literature [16] through a pair of formulas, one for assigning the velocity and another for changing the particle's position:

$$v_{id}(t+1) = \omega * v_{id}(t) + c_1 r_1 (p_{id} - x_{id}(t)) + c_2 r_2 (p_{gd} - x_{id}(t)) \tag{8}$$

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

where $i = 1, 2, \ldots, 2N$, $d = 1, 2, \ldots, M$, r_1 and r_2 are random real numbers subjected to the U(0,1) distribution. In this paper, let $c_1 = c_2 = 2$, $\omega = 0.8$, $v_{id} \in [-v_{\max}, v_{\max}]$; $v_{\max} = 0.2$, p_{gd} is the optimum position found so far by any member of i's topological neighborhood; p_{id} is the optimum position found so far when individual i is at its current position x_{id} in the searching space with velocity v_{id}.

3.2. SPSO

By analyzing the biological model and formulas (8) and (9), it is found that the velocity vector is not necessary. Let i's current position x_i in the searching space be the solution of the current problem, and the goal of optimization is to make x_i infinitely approach the best position. Thus, we only take the direct change of x_i into account. The velocity vector v_i, which only represents the particle's moving speed, does not assure the particle's approaching the best solution effectively. Furthermore, it may even lead to the particle's departing from the correct direction, which results in a slow convergence velocity and a low convergence precision in the later evolution process.

Take one particle as example, we can rewrite (8) and (9) as follows:

$$v(t+1) = \omega * v(t) + (c_1 r_1 + c_2 r_2)\left(\frac{c_1 r_1 p_b + c_2 r_2 p_g}{c_1 r_1 + c_2 r_2} - x(t)\right) \tag{10}$$

$$x(t+1) = x(t) + v(t+1) \tag{11}$$

where p_b is the optimum position found so far of the particle discussed, and p_g is the optimum position in its topological neighborhood.

Iterating the formulas of (10) and (11), we can obtain the following formula:

$$x(t+2) + (c_1 r_1 + c_2 r_2 - \omega - 1) * x(t+1) + \omega * x(t) = c_1 r_1 p_b + c_2 r_2 p_g \tag{12}$$

Although the velocity vector is eliminated in formula (12), it is still "hidden" in it. And formula (13), which is inspired by formula (12), is put forward to discard the velocity vector totally. The simulation results show that it can obtain a better result.

Through the analysis above, the formula which doesn't have the velocity vector is expressed as:

$$x_{id}(t+1) = \omega * x_{id}(t) + c_1 r_1 (p_{id} - x_{id}(t)) + c_2 r_2 (p_{gd} - x_{id}(t)) \tag{13}$$

The significance of formula (13) is to discard the velocity vector, which can avoid the problem of slow convergence velocity in the later evolution process and low precision caused by determining the velocity vector factitiously.

3.3. edsPSO

Experiments show that the particles of the swarm will get together when the evolution is stagnant. They will not disperse until the stagnancy is broken out. Because the particles will get together at the extremum determined by the extremum itself and the extremum of the population, so two methods are given in this paper to sovle this problem by adjusting p_b and p_g respectively so that all the particles will fly to a new direction. Then it is more possible to find the optimum solution.

As shown in formulas (14) and (15), r_3 and r_4 are the disturbed coefficients, T_0 and T_1 are the disturbed threshold valve values. Let $T_0 = 3,\ T_1 = 3,\ r_5, r_6$ and r_7 be the random real numbers in the range of [0,1].

$$\begin{aligned} r_3 &= \begin{cases} 1, & t < T_0 \\ \alpha * [0.8 + 0.2 * r_5], & t \geq T_0 \end{cases}, \\ r_4 &= \begin{cases} 1, & t < T_1 \\ -\alpha * [0.8 + 0.2 * r_5], & t \geq T_1 \end{cases} \end{aligned} \tag{14}$$

$$\alpha = \begin{cases} 1, & r_7 \geq 0.5 \\ -1, & r_7 < 0.5 \end{cases} \tag{15}$$

3.3.1. *IedsPSO*

The formula of IedsPSO (Internal extremum disturbed simple particle swarm optimization) algorithm can be expressed as:

$$x_{id}(t+1) = \omega * x_{id}(t) + c_1 r_1 (r_3 * p_{id} - x_{id}(t)) + c_2 r_2 (r_4 * p_{gd} - x_{id}(t)) \tag{16}$$

3.3.2. *edsPSO*

The formula of edsPSO (External extremum disturbed simple particle swarm optimization) Hybrid algorithm can be expressed as:

$$x_{id}(t+1) = \omega * x_{id}(t) + c_1 r_1 r_3 * (p_{id} - x_{id}(t)) + c_2 r_2 r_4 * (p_{gd} - x_{id}(t)) \tag{17}$$

4. IGA-edsPSO HYBRID ALGORITHM

The flow chart of IGA-edsPSO is shown in Fig. 2:

Step 1: initialize a population M_1;

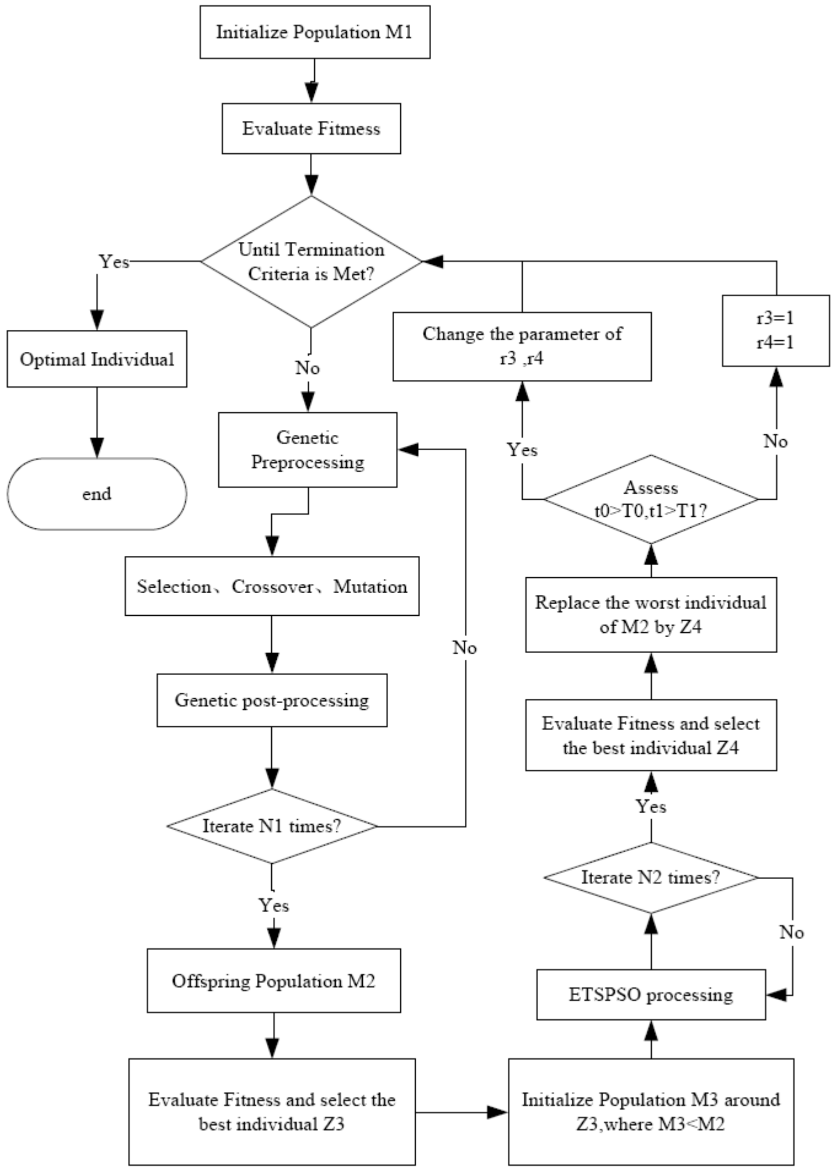

Figure 2. Flow chart of IGA-edsPSO.

Step 2: iterate M_1 for N_1 times by using IGA and obtain the offspring population M_2, then select the best individual Z_3 of the population M_2;

Step 3: initialize a population M_3 whose individuals are all around Z_3;

Step 4: iterate M_3 for N_2 times by using edsPSO and select the best individual Z_4;

Step 5: replace the worst individual of M_2 by Z_4;

Step 6: change r_3 when $t_0 > T_0$ while change r_4 when $t_1 > T_1$;

Step 7: iterate until termination criteria is met.

5. SIMULATION RESULTS AND ANALYSIS

A sparse array synthesis technique called the transformation of Legendre polynomial is proposed in literature [4], and the design constraint of $0.5\lambda \leq c_0 \leq \lambda$ is also described. A 17-element linear sparse array of an aperture of 9.744λ is synthetized, and the optimized PSLL is -19.49 dB. An optimum result of -19.797 dB is achieved for the same sparse array using an improved GA in literature [6], in which the population contains 200 individuals and the number of iterations is 300. It should be noted from the comparative results that the IGA-edsPSO not only achieves a better PSLL performance of -20.56 dB (the best result of 10 runs, and the worst one is -19.73 dB) as shown in Fig. 7, but also requires a smaller population size (30 individuals) and consumes a shorter process time(50 generations). All the algorithms mentioned in this paper run at least for 10 times, and the difference between the best and worst results of each algorithm is less than 0.9 dB, so each algorithm is high in stability. The best result of each algorithm discussed in this paper is used for comparison. The more complex the problem is, the less computation time is cost by IGA-edsPSO than other algorithms mentioned in this paper. A more complex problem is given below. When a lower sidelobe level is required for the 17-element linear array, we can optimize the element excitation amplitude and the coordinate synchronously. As shown in Table 1, less computation time and lower sidelobe level are obtained by using IGA-edsPSO compared to other algorithms mentioned in this paper.

The population size of IGA and standard PSO is 200 and 100, respectively. And the number of generations of them is 300 and 100 respectively for each run.

Then the methods of the improved Hybrid algorithm proposed in this paper are used. Let $M_1 = 50$, $M_3 = 20$, $N_1 = 10$, $N_2 = 20$ (the parameters are shown in the flow chart). A program in FORTRAN

Table 2. The optimization results of the algorithms.

IGA			IGA-PSO			IGA-edPSO		
Element list	Coordinae (λ)	amplitude excitation	Element list	Coordinae (λ)	amplitude excitation	Element list	Coordinae (λ)	amplitude excitation
1	0	0.367	1	0	0.195	1	0	0.413
2	0.635	0.489	2	0.687	0.373	2	0.798	0.345
3	1.382	0.702	3	1.233	0.240	3	1.605	0.772
4	1.952	0.528	4	1.768	0.458	4	2.287	0.820
5	2.464	0.488	5	2.419	0.656	5	2.920	0.858
6	3.056	0.820	6	2.966	0.634	6	3.467	0.538
7	3.642	0.982	7	3.474	0.652	7	3.969	0.829
8	4.284	0.929	8	3.977	0.926	8	4.498	0.806
9	4.863	0.760	9	4.549	0.873	9	5.010	0.804
10	5.410	0.974	10	5.161	0.847	10	5.546	0.942
11	6.044	0.984	11	5.736	0.864	11	6.179	0.962
12	6.672	0.746	12	6.476	0.851	12	6.805	0.655
13	7.256	0.718	13	7.161	0.688	13	7.409	0.683
14	7.879	0.555	14	7.759	0.391	14	8.041	0.521
15	8.458	0.499	15	8.385	0.452	15	8.578	0.371
16	9.117	0.363	16	9.139	0.436	16	9.094	0.407
17	9.744	0.324	17	9.744	0.104	17	9.744	0.260
PSLL	−24.73 (dB)		PSLL	−25.20 (dB)		PSLL	−25 46 (dB)	

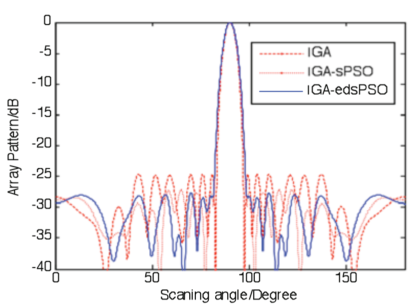

Figure 3. Contrast of the radiation pattern.

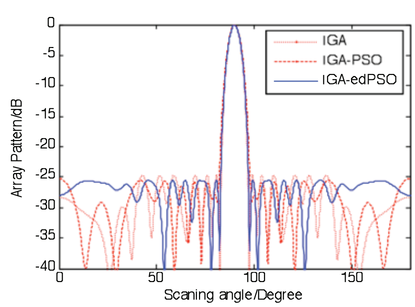

Figure 4. Contrast of the radiation pattern.

Table 3. The optimization results of the algorithms.

IGA-SPSO			IGA-edsPSO		
Element list	Coordinate (λ)	Amplitude excitation	Element list	Coordinate (λ)	Amplitude excitation
1	0	0.301	1	0	0.254
2	0.654	0.521	2	0.654	0.521
3	1.384	0.644	3	1.384	0.644
4	2.015	0.686	4	2.011	0.717
5	2.601	0.741	5	2.545	0.815
6	3.167	0.974	6	3.147	0.937
7	3.721	0.897	7	3.726	0.905
8	4.226	0.872	8	4.286	0.963
9	4.807	0.991	9	4.839	0.784
10	5.392	0.772	10	5.339	0.772
11	5.904	0.828	11	5.891	0.953
12	6.493	0.646	12	6.457	0.622
13	7.076	0.628	13	6.993	0.628
14	7.653	0.591	14	7.652	0.550
15	8.272	0.417	15	8.491	0.417
16	9.046	0.219	16	9.140	0.219
17	9.744	0.184	17	9.744	0.184
PSLL	−27.17 (dB)		PSLL	−27.67 (dB)	

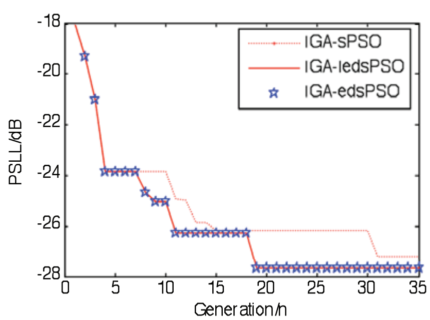

Figure 5. Contrast of convergence velocity.

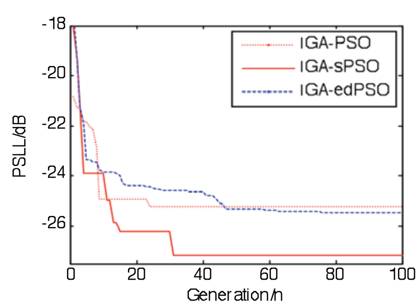

Figure 6. Contrast of convergence velocity.

Table 1. Contrast of the computation time and PSLL among the algorithms.

Algorithm	IGA	Standard PSO	IGA-PSO	IGA-edPSO	IGA-sPSO	IGA-edsPSO
Time(minutes)	50	13	8	13	6	4
PSLL(dB)	−24.73	−24.78	−25.20	−25.46	−27.17	−27.67

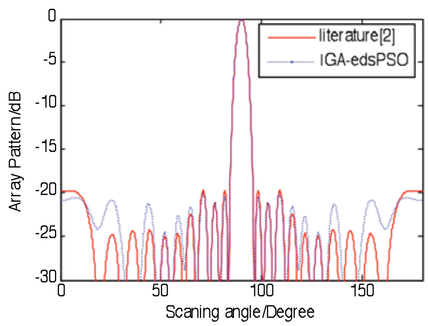

Figure 7. Contrast of the radiation pattern.

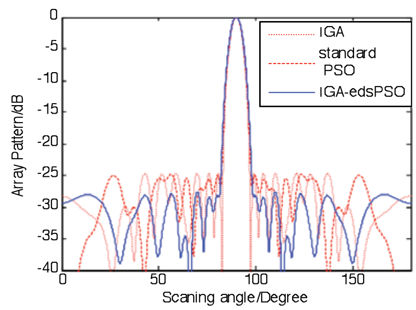

Figure 8. Contrast of the radiation pattern.

language is written for a PC to operate (Inter Celeron processor 3.0 GHz, SDRAM 1 GBytes). The optimized results are shown in Tables 1–3 and Figures 3–8. The results in Table 1 and Fig. 8 show that IGA-edsPSO obtains lower sidelobe level and costs less computation time than IGA and standard PSO. The results in Fig. 4 and Fig. 6 show that compared with IGA-PSO and IGA-edPSO, IGA-SPSO accelerates the convergence velocity in the later evolution process and increases the precision greatly by discarding the velocity vector in PSO. And it is also seen that IGA-edPSO can get a better solution than IGA-PSO with the help of extremum diturbed factor. The results in Fig. 5 shows that the two kinds of disturbed methods proposed in this paper have the similar effect. The results in Fig. 3 and Fig. 5 reveal that IGA-edsPSO, which takes extremum distrubed factor into account and dicards the velocity vetor in PSO, can reduce the time it takes for the particles to overstep the local extremum. Thus, it can speed up the convergence and increase the precison to a large extent. What is more, the HPBW is not broadened with the decreasing of sidelobe level, which achieves the goal of optimization, and the IGA-edsPSO requires a smaller population size (50 individuals) and involves a shorter process time (35 generations). This is a significant saving in the computational effort as compared to that needed by the IGA and standard PSO.

6. CONCLUSION

An IGA-edsPSO Hybrid algorithm is proposed in this paper in the optimization of the sparse array, which is based on the improvements of both GA and PSO, so it inherits the advantages of GA and PSO. The simulation results show that the IGA-edsPSO Hybrid algorithm can accelerate the convergence in the later evolution process and overstep the local extremum rapidly, thus can improve the precision greatly. At present, further work is in progress to extend this synthesis technique to more practical antenna configurations, such as planar arrays of practical antenna elements.

REFERENCES

1. Yan, K.-K. and Y. Lu, "Sidelobe reduction in array-pattern synthesis using genetic algorithm," *IEEE Trans. Antennas and Propag.*, Vol. 45, No. 7, 1117–1122, 1997.
2. 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*, PIER 75, 319–328, 2007.
3. Kumar, B. P. and G. R. Branner, "Generalized analytical technique for the synthesis of unequally spaced arrays with linear, planar, cylindrical or spherical geometry," *IEEE Trans. Antennas and Propag.*, Vol. 53, No. 2, 621–634, 2005.
4. Kumar, B. P. and G. R. Branner, "Design of unequally spaced arrays for performance improvement," *IEEE Trans. Antennas and Propag.*, Vol. 47, No. 3, 511–523, 1999.
5. Lee, K. C. and J. Y. Jhang, "Application of particle swarm algorithm to the optimization of unequally spaced antenna arrays," *Journal of Electromagnetic Waves and Applications*, Vol. 20, No. 14, 2001–2012, 2006.
6. Chen, K., Z. He, and C. Han, "A modifed real GA for the sparse linear array synthesis with multiple constraints," *IEEE Trans. Antennas and Propag.*, Vol. 54, No. 7, 2169–2173, 2006.
7. Haupt, R. L., "Thinned arrays using genetic algorithms," *IEEE Trans. Antennas and Propag.*, Vol. 42, No. 7, 993–999, 1994.
8. Mahanti, G. K., A. Chakrabarty, and S. Das, "Phase-only and amplitude-phase synthesis of dual-pattern linear antenna arrays using floating-point genetic algorithms," *Progress In Electromagnetics Research*, PIER 68, 247–259, 2007.
9. Ares Pena, F. J., J. A. Rodriguez, E. V. Lopez, and

S. R. Rengarajan, "Genetic algorithms in the design and optimization of antenna array patterns," *IEEE Trans. Antennas and Propag.*, Vol. 47, No. 3, 506–510, 1999.

10. Su, D. Y., D. M. Fu, and D. Yu, "Genetic algorithms and method of moments for the design of PIFAS," *Progress In Electromagnetics Research Letters*, Vol. 1, 9–18, 2008.
11. Chen, H.-T., G.-Q. Zhu, and S.-Y. He, "Using genetic algorithm to reduce the radar cross section of three-dimensional anisotropic impedance object," *Progress In Electromagnetics Research B*, Vol. 9, 231–248, 2008.
12. Haupt, R. L., "Antenna design with a mixed integer genetic algorithm," *IEEE Trans. Antennas and Propag.*, Vol. 55, No. 3, 577–582, 2007.
13. Petko, J. S. and D. H. Werner, "An autopolyploidy-based genetic algorithm for enhanced evolution of linear polyfractal arrays," *IEEE Trans. Antennas and Propag.*, Vol. 55, No. 3, 583–593, 2007.
14. Kennedy, J. and R. Eberhart, "Particle swarm optimization," *Proc. IEEE Int. Conf. Neural Networks*, Perth, Australia, 1942–1948, 1995.
15. Trelea, I. C., "The particle swarm optimization algorithm: Convergence analysis and parameter selection," *Information Processing Letters*, Vol. 85, No. 6, 317–325, 2003.
16. Clerc, M. and J. Kennedy, "The particle swarm-explosion,stability and convergence in multid-imensional complex space," *IEEE Trans. Evol. Comput.*, Vol. 6, No. 1, 58–73, Feb. 2002.
17. Robinson, J. and Y. Rahmat-Samii, "Particle swarm optimization in electromagnetic," *IEEE Trans. Antennas and Propag.*, Vol. 52, No. 2, 397–407, 2004.
18. Lu, Z. B., A. Zhang, and X. Y. Hou, "Pattern synthesis of cylindrical conformal array by the modified particle swarm optimization algorithm," *Progress In Electromagnetics Research*, PIER 79, 415–426, 2008.
19. Jin, N. and Y. Rahmat-Samii, "Advances in particle swarm optimization for antenna designs: Real-number, binary, single-objective and multiobjective implementations," *IEEE Trans. Antennas and Propag.*, Vol. 55, No. 3, 556–567, 2007.
20. Donelli, M., S. Caorsi, F. De Natale, M. Pastorina, and A. Massa, "Linear antenna synthesis with a hybrid genetic algorithm," *Progress In Electromagnetics Research*, PIER 49, 1–22, 2004.