Progress In Electromagnetics Research, PIER 80, 461–476, 2008

IMPROVED GA AND PSO CULLED HYBRID ALGORITHM FOR ANTENNA ARRAY PATTERN SYNTHESIS

W. T. Li, X. W. Shi, and L. Xu

National Key Laboratory of Antenna and Microwave Technology
Xidian University
No. 2 SOUTH Taibai ROAD, Xi'an, Shaanxi, P. R. China

Y. Q. Hei

State Key Laboratory of Integrated Services Networks
Xidian University
No. 2 SOUTH Taibai ROAD, Xi'an, Shaanxi, P. R. China

Abstract—In this paper, a new evolutionary learning algorithm based on a hybrid of improved real-code genetic algorithm (IGA) and particle swarm optimization (PSO) called HIGAPSO is proposed. In order to overcome the drawbacks of standard genetic algorithm and particle swarm optimization, some improved mechanisms based on non-linear ranking selection, competition and selection among several crossover offspring and adaptive change of mutation scaling are adopted in the genetic algorithm, and dynamical parameters are adopted in PSO. The new population is produced through three approaches to improve the global optimization performance, which are elitist strategy, PSO strategy and improved genetic algorithm (IGA) strategy. The effectiveness of the proposed algorithm has been compared with GAs and PSO, synthesizing a circular array, a linear array and a base station array. Results show that the proposed algorithm is able to adapt itself to different electromagnetic optimization problems more effectively.

1. INTRODUCTION

With the development of radar and mobile communication systems, special shapes of antenna beams are needed [1–8], which demand more variables to control and form the array pattern. In the array pattern synthesis, the objective function and constraint condition are often highly nonlinear and nondifferentiable. Therefore, analytical methods, such as Taylor method and the Chebishev method, are not applicable any more. To this end, stochastic methods are necessary [9, 10] to efficiently deal with large nonlinear search spaces.

In contrast to traditional computation systems, evolutionary computation [11–14] provides a more robust and efficient approach for solving complex real-world problems. Genetic Algorithm(GA) [15–17] has proven to be a useful method of optimization for difficult and discontinuous multidimensional engineering problems. GA is very efficient at exploring the entire search space, but it is relatively poor in finding the precise local optimal solution in the region where the algorithm converges. A new method of optimization, Particle Swarm Optimization(PSO) [18–20], is able to accomplish the same goal as GA optimization in a new and faster way. Since PSO and GA both work with a population of solutions, combining the searching abilities of both methods seems to be a good approach. Some attempts have been made in this direction, but with a weak integration of the two strategies. Precisely, most of the times one technique has been used just as a pre-optimizer for the initial population of the other technique [21, 22]. In order to improve the speed of convergence of evolutionary algorithms, in this paper, GA and PSO are strong combined for synthesizing three types of antenna arrays.

Based on GA and PSO, a hybrid algorithm called HIGAPSO is presented. Firstly, some improved mechanisms such as non-linear ranking selection, competition and selection among several crossover offspring and adaptive change of mutation scaling are adopted in the genetic algorithm. Then, the improved genetic algorithm is combined with PSO that is improved by dynamical parameters. During each iteration, the population is divided into three parts, which are evolved with the elitist strategy, PSO strategy and the improved genetic algorithm strategy respectively. Therefore, this kind of technique can make balance between acceleration convergence and averting precocity as well as stagnation. The simulation results show the effectiveness of the algorithm in synthesizing conformal array, linear array with prescribed nulls and array with complex pattern.

The remaining sections of this paper are organized as follows: Section 2 briefly explains the formulation of the problem. Section 3 describes the implementations of IGA and PSO in the proposed HIGAPSO algorithm. The numerical examples are then presented in Section 4 and conclusion is made in Section 5.

2. PROBLEMS FORMULATION

In synthesizing of a linear array, the far field array factor of this array can be written as

$$F(\theta) = \sum_{n=1}^{N} a_n \exp\left(j\left(n\frac{2\pi}{\lambda}d\sin\theta + \beta_n\right)\right) \tag{1}$$

where n is the element number, λ is the wavelength, β_n is the excitation current phase of the elements and a_n is the element excitation current amplitude. d denotes the inter-element spacing, and θ is the polar angle of far-field measured from broadside ($-90°$ to $+90°$).

Normalized absolute far-field is

$$F_n(\theta) = \frac{|F(\theta)|}{|F(\theta)|_{\max}} \tag{2}$$

We try to use a modified Bernstein polynomial [23] to reduce the number of variables when calculating the circular array. The modified Bernstein polynomial is

$$F(U) = \begin{cases} B_1 + \dfrac{1-B_1}{A^{MA}(1-A)^{M(1-A)}} U^{MA}(1-U)^{M(1-A)}, & 0 \le U \le A \\ B_2 + \dfrac{1-B_2}{A^{MA}(1-A)^{M(1-A)}} U^{MA}(1-U)^{M(1-A)}, & A \le U \le 1 \end{cases} \tag{3}$$

where B_1, B_2, M, A are parameters in the polynomial, B_1 and B_2 specify the left and right endpoints $F(0)$ and $F(1)$ respectively. For $\theta = 90°$, the far field array factor of the circular array is

$$E(90°, \phi) = \sum_{n=1}^{N} F(U)_n \exp\left(j\frac{2\pi}{\lambda}r\left(\cos(\phi-\beta_n) - \cos(\beta_n)\right)\right) \tag{4}$$

where $F(U)_n$ is the nth excitation amplitude, being the nth equal sampling point value of $F(U)$ and r is the circle radius of the array.

A key point of optimization is the construction of the target function. Although the problem under discussion is a minimization problem, we convert it to a maximization problem. The fitness function to be maximized for array optimization problem can be expressed as follows:

$$Fitness = \frac{U\left(F_o(\theta) - F_d(\theta)\right)}{\alpha \times |MSLL - SLVL| + \beta \times |NULL\_PAT - NLVL| + \gamma \times |F_o(\theta) - F_d(\theta)|} \tag{5}$$

where $MSLL$ is the highest sidelobe level, $SLVL$ is the desired sidelobe level, $NULL\_PAT$ is the average nulls depth, and $NLVL$ is the desired nulls depth. $F_o(\theta)$ and $F_d(\theta)$ are, respectively, the pattern obtained by optimization and the desired pattern. The values of α, β and γ stand for the weight coefficients such that the fitness function is capable of guiding potential solutions to obtain satisfactory array pattern performance with desired properties. $U(t)$ the unit step function, can be depicted as:

$$U(t) = \begin{cases} 1 & t \geq 0 \\ 0 & t < 0 \end{cases} \tag{6}$$

3. HYBRID OF IMPROVED GA AND PSO (HIGAPSO)

The proposed HIGAPSO combines GA with PSO to form a hybrid algorithm. Due to combination of different optimization mechanisms, not only the offspring can keep diversity, but also PSO can keep the balance of global search and local search, so the entire search ability of the algorithm can be improved. In this section, improved GA and PSO are introduced first, followed by a detailed introduction of HIGAPSO.

3.1. Improved Genetic Algorithm

Floating-point GA uses floating-point number representation for the real variables and thus is free from binary encoding and decoding. It takes less memory space and works faster than binary GA. Some practical schemes to improve GA performance are introduced in this paper. According to the optimal results, we can conclude that these measures are effective and helpful in improving convergence property and accuracy.

3.1.1. Nonlinear Ranking Selection

Ranking methods only require the evaluation function to map the solutions to a partially ordered set. All individuals in a population are ranked from best to worst based on their fitness values. It assigns the probability of an individual based on its rank (r) and it is expressed as follows:

$$\begin{cases} p(r) = q'(1-q)^{r-1} \\ q' = \dfrac{q}{1-(1-q)^P} \end{cases} \tag{7}$$

Such that

$$\sum_{r=1}^{P} p(r) = 1 \tag{8}$$

where

$q =$ the probability of selecting the best individual $= [0, 1]$,

$r =$ the rank of the individual $= \begin{cases} 1, & \text{for the best individual} \\ P, & \text{for the worst individual} \end{cases}$

$P =$ the population size

It can be seen that this selection probability doesn't use the absolute value information of fitness value so that it avoid the fitness value scale transformation and control the prematurity to some extent.

3.1.2. Competition and Selection

In natural biological evolution, two parents after crossover can produce several offspring, and the competition also exists among the offspring which are produced by the same parents. Motivate by this phenomenon, we adopt competition and selection among several crossover offspring. Different from the conventional algorithm in which two parents only produce two offspring, the two parents, chromosomes $a_s = [x_1^s \ \ x_2^s \ \ \cdots \ \ x_n^s]$ and $a_t = \left[x_1^t \ \ x_2^t \ \ \cdots \ \ x_n^t\right]$, in this algorithm will produce four chromosomes according to the following mechanisms [24]:

$$b_1 = \left[b_1^1 \ \ b_2^1 \ \ \cdots \ \ b_n^1\right] = \frac{a_s + a_t}{2} \tag{9}$$

$$b_2 = \left[b_1^2 \ \ b_2^2 \ \ \cdots \ \ b_n^2\right] = a_{\max}(1 - w) + \max(a_s, a_t)\, w \tag{10}$$

$$b_3 = \left[b_1^3 \ \ b_2^3 \ \ \cdots \ \ b_n^3\right] = a_{\min}(1 - w) + \min(a_s, a_t)\, w \tag{11}$$

$$b_4 = \left[b_1^4 \ \ b_2^4 \ \ \cdots \ \ b_n^4\right] = \frac{(a_{\max} + a_{\min})(1 - w) + (a_1 + a_2)\, w}{2} \tag{12}$$

$$a_{\max} = [x_1^{\max} \ \ x_2^{\max} \ \ \cdots \ \ x_n^{\max}] \tag{13}$$

$$a_{\min} = \left[x_1^{\min} \ \ x_2^{\min} \ \ \cdots \ \ x_n^{\min}\right] \tag{14}$$

where $w \in [0, 1]$ denotes the weight to be determined by users, $\max(a_s, a_t)$ denotes the vector with each element obtained by taking the maximum among the corresponding element of a_s and a_t. Among b_1 to b_4, the two with the largest fitness value are used as the offspring of the crossover operation. As seen from Eqs. (9) to (12), the potential offspring spreads over the domain. At the same time, (9) and (12) results in searching around the center region of the domain, (10) and (11) can move b_2 and b_3 to be near $a_{\max}$ and $a_{\min}$ respectively. Thus, the offspring generated by this operator, is better than that obtained by arithmetic crossover or heuristic crossover.

3.1.3. Mutation

This is the unary operator responsible for the fine tuning capabilities of the system, so that it can escape from the trap of local optimum. It is defined as follows: For a parent p, if variable p_k was selected at random for this mutation, the result is: $\bar{p} = (p_1 \quad \cdots \quad \bar{p}_k \quad \cdots \quad p_n)$ where

$$\bar{p}_k \in \left\{ \max\left(p_k - \mu \frac{p_k^{\max} - p_k^{\min}}{2}, p_k^{\min} \right), \min\left(p_k + \mu \frac{p_k^{\max} - p_k^{\min}}{2}, p_k^{\max} \right) \right\} \tag{15}$$

and $p_k^{\max}$, $p_k^{\min}$ are upper and lower bounds of p_k respectively, μ decreased with the increase of iterations.

$$\mu(\tau) = 1 - r^{[1-(\tau/T)]^b} \tag{16}$$

where r is uniform random number in $[0, 1]$, T is the maximum number of iterations, τ is the current iteration number, and b is the shape parameter. From (16), at the initial stage of evolution, for small value of r, $\mu(\tau) \approx 1$, the mutation domain is large in this case. However, in the later evolution, when τ approaches T, $\mu(\tau) \approx 0$, the mutation domain become small and search in the local domain.

3.2. Particle Swarm Algorithm

The PSO conducts searches using a population of particles which correspond to individuals in GAs. The population of particles is randomly generated initially. Each particle represents a potential solution and has a position represented by a position vector $\vec{x}_i$. A swarm of particles moves through the problem space, with the moving velocity of each particle represented by a position vector $\vec{v}_i$. At each time step, a function f_i representing a quality measure is calculated by using $\vec{x}_i$ as input. Each particle keeps track of its own best position, which is associated with the best fitness it has achieved so far in a vector $\vec{p}_i$. Furthermore, the best position among all the particles obtained so far in the population is kept track of as $\vec{p}_g$.

At each time step τ, by using the individual best position, $\vec{p}_i(\tau)$, and global best position, $\vec{p}_g(\tau)$, a new velocity for particle i is updated by

$$\vec{v}_i\left(\tau+1\right) = w\vec{v}_i(\tau) + c_1\phi_1\left(\vec{p}_i(\tau) - \vec{x}_i(\tau)\right) + c_2\phi_2\left(\vec{p}_g(\tau) - \vec{x}_i(\tau)\right) \tag{17}$$

where c_1 and c_2 are acceleration constants and ϕ_1 and ϕ_2 are uniformly distributed random numbers in $[0, 1]$. The term $\vec{v}_i$ is limited to its bounds. If the velocity violates this limit, it is set to its proper limit. w is the inertia weight factor and in general, it is set according to the following equation:

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{T} \cdot \tau \tag{18}$$

where $w_{\max}$ and $w_{\min}$ is maximum and minimum value of the weighting factor respectively. T is the maximum number of iterations and τ is the current iteration number.

Based on the updated velocities, each particle changes its position according to the following:

$$\vec{x}_i\left(\tau+1\right) = \vec{x}_i\left(\tau\right) + h\left(\tau\right)\vec{v}_i\left(\tau+1\right) \tag{19}$$

where

$$h\left(\tau\right) = h_{\max} - \frac{\left(h_{\max} - h_0\right)\cdot\tau}{T} \tag{20}$$

where $h_{\max}$ and h_0 are positive constants.

According to (17) and (19), the population of particles tend to cluster together with each particle moving in a random direction. The computation of PSO is easy and adds only a slight computation load when it is incorporated into IGA. Furthermore, the flexibility of PSO to control the balance between local and global exploration of the problem space helps to overcome premature convergence of elite strategy in GAs, and also enhances searching ability. The global best individual is shared by the two algorithms, which means the global best individual can be achieved by the IGA or by PSO, also it can avoid the premature convergence in PSO.

3.3. Hybrid of Improved GA and PSO (HIGAPSO)

The HIGAPSO maintains the integration of IGA and PSO for the entire run, which consists chiefly of genetic algorithm, combined with PSO and the sequential steps of the algorithm are given below. Briefly, the flow of key operations are illustrated in Fig. 1.

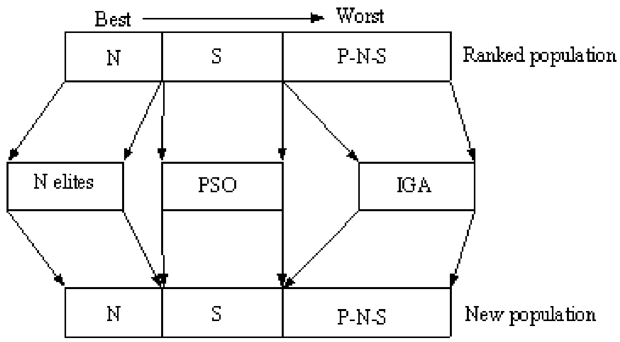

Figure 1. Flow of key operations in HIGAPSO.

Step 1: Randomly initialize the population of P individuals within the variable constraint range.

Step 2: Calculate the fitness of the population from the fitness function, and order ascendingly.

Step 3: The top N individuals are selected as the elites and reproduce them directly to the next generation.

Step 4: The S individuals followed are evolved with PSO and their best positions are updated.

Step 5: The bottom individuals are evolved with IGA and produce P-S-N offspring.

Step 6: Combine the three parts as the new generation and calculate the fitness of the population. Choose the best position among all the individuals obtained so far kept as the global best.

Step 7: Repeat steps 3–6 until a stopping criterion, such as a sufficiently good solution being discovered or a maximum number of generations being completed, is satisfied. The best scoring individual in the population is taken as the final answer.

4. NUMERICAL RESULTS

Three types of antenna arrays have been synthesized to verify the effectiveness and flexibility of the proposed hybrid algorithm. In order to demonstrate the superiority of HIGAPSO, in each simulation, the performance of HIGAPSO is compared with GAs and PSO. The parameters used in GAs and PSO are selected the same as those used in HIGAPSO, which ensure a fair comparison in computation efficiency and solution quality.

4.1. Synthesis a circular array

We consider a circular array of 31 isotropic radiators spaced 0.5λ apart along a circle of radius 6 wavelengths. To reduce the number of variables, we try to use a modified Bernstein polynomial according to Eq. (3). For a candidate set of polynomial parameters A, M, B_1, and B_2, the resulting function $F(U)$ is sampled to obtain amplitude weights for the circular antenna to yield the maximum relative sidelobe level equal to or below -35 dB. The proposed algorithm works to adjust A, M, B_1, and B_2. Each individual has four dimensions, one for each parameter. A, B_1, and B_2 are allowed to vary from zero to one, while M may vary from one to twenty.

For design specifications, the following parameters have been selected for the proposed algorithm: population size = 20; generations = 300; inertia weight factor w is set by Eq. (18), where $w_{\max} = 0.9$ and $w_{\min} = 0.4$; acceleration constant $c_1 = 2.0$ and $c_2 = 2.0$; the probability of selecting the best individual $q = 0.1$; the number of elites $S = 2$; crossover probability $p_c = 0.80$; mutation probability $p_m = 0.02$; the percentage of population in each iteration evolved with IGA $S = 0.2P$. Fig. 2(a) shows the normalized absolute power patterns in dB. Fig. 2(b) shows the optimized distribution by HIGAPSO. It can be seen that all of the four algorithms can obtain the desired beamform, but the pattern obtained by HIGAPSO has a better sidelobe. From Table 1, we can clearly see that the proposed algorithm gives a better solution within a faster convergent speed compared with other algorithms.

Table 1. Performance comparisons for different methods of circular array design.

Method	Best fitness value	Cost function evaluations
SGA	0.9279	5960
IGA	0.9281	3600
PSO	0.9281	3820
HIGAPSO	0.9282	1560

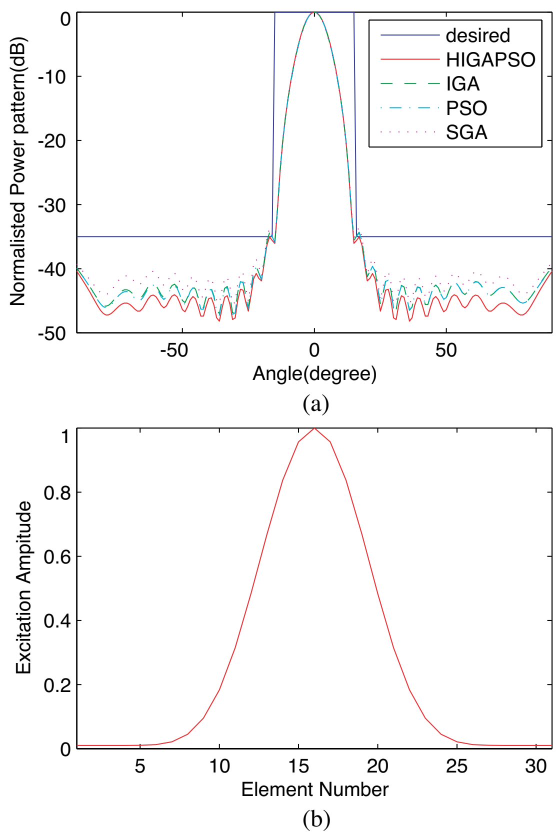

Figure 2. (a) Radiation pattern for the circular array, (b) excitation amplitude distributions for the circular array.

4.2. Synthesis of Amplitude-only Pattern Nulling of Linear Antenna Array

It is well known that the broad nulls are needed when the direction of arrival of the unwanted interference may vary slightly with time or may not be known exactly, and where a comparatively sharp null would require continuous steering for obtaining a reasonable value for the signal-to-noise ratio. To illustrate the broad-band interference suppression capability of the HIGAPSO, the pattern having a broad null located at 47° with $\Delta\theta_i = 5°$ is designed using a uniform linear array of 32 isotropic elements spaced 0.5λ apart. It is desired that the sidelobe level should be lower than $-20\,\text{dB}$ and the null depth level should be deeper than $-40\,\text{dB}$. The excitation amplitude distribution is symmetric with respect to the center of the array.

Because of symmetry, only 16 amplitudes are to be optimized. In applying HIGAPSO, initially, 80 individuals are randomly generated in a population. The rest parameters are the same as those in the first example. Fig. 3(a) shows normalized absolute power pattern in dB for the amplitude-only pattern nulling of linear antenna array. Due to the SGA cannot obtain the desired beamform, the pattern obtained by it is not shown. Fig. 3(b) shows common amplitude distribution obtained by HIGAPSO. It is evident that the pattern obtained by PSO cannot be fully satisfied the design specification, and HIGAPSO can accurately produce the nulling pattern. It can be found from Table 2 that HIGAPSO procedure possesses the dominant speed and precision in the optimization compared with other algorithms.

Table 2. Performance comparisons for different methods of the nulling of linear antenna array design.

Method	Best fitness value	Cost function evaluations
SGA	——	——
IGA	0.2671	29760
PSO	0.2664	32800
HIGAPSO	0.2685	19440

4.3. Synthesis a Base Station Array

A base station array is synthesized in this subsection using a uniform linear array of 16 isotropic elements with desired sidelobe level of −20 dB or below and desired null depth level of −28 dB or below. To realize economical and easy in engineering, we choose distance between elements and relative phase as optimization variables.

In applying HIGAPSO, initially, 80 individuals are randomly generated in a population. The rest parameters are the same as those in the first example. Fig. 4(a) shows normalized absolute power pattern in dB for the amplitude-only pattern nulling of linear antenna array. Also, as the same reason presented above, the result of SGA is not shown here. Fig. 4(b) shows the phase distributions in degree plotted by line and the distance between elements plotted by line with stars for the base station array obtained by HIGAPSO. It is clear that patterns obtained by PSO and IGA cannot satisfy the specifications. It can be concluded that the proposed algorithm can give optimal solution within a faster convergent speed.

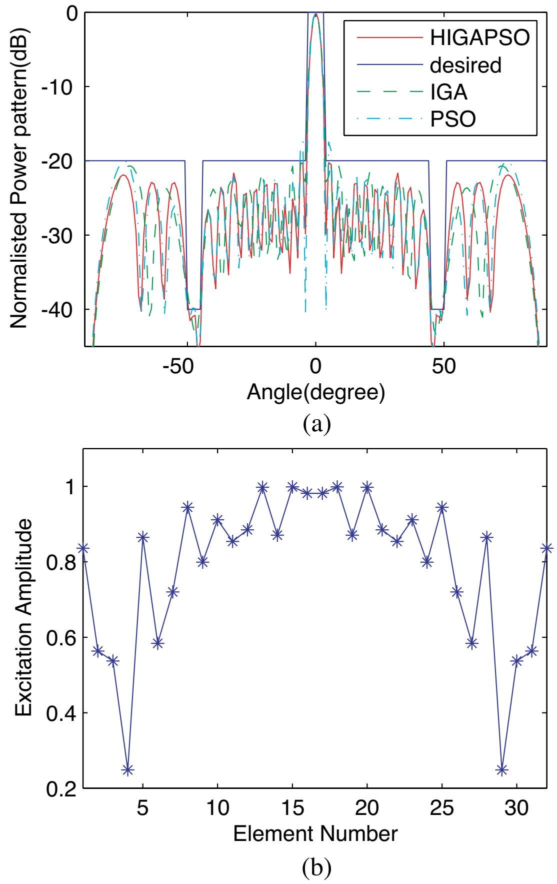

Figure 3. (a) Radiation pattern for the nulling of linear antenna array, (b) excitation amplitude distributions for the nulling of linear antenna array by amplitude-only synthesis.

According to the simulation results, with the increasing complexity of array patterns to obtain and number of variables to optimize, PSO and IGA show their limitations and inefficiency to some extent, respectively. While the HIGAPSO presented in this work has better quality of solutions in all these three cases. From the maximum sidelobe level and the null depth points of view, the performances of the pattern are also excellent. As electromagnetic analysis often has time-consuming cost functions, a key factor is the number of iterations and function evaluations. From the tables above, it is obvious that the proposed algorithm has less cost function evaluations compared

Table 3. Performance comparisons for different methods of the base station array design.

Method	Best fitness value	Cost function evaluations
SGA	——	——
IGA	0.6067	160000
PSO	0.6073	160000
HIGAPSO	0.6146	98480

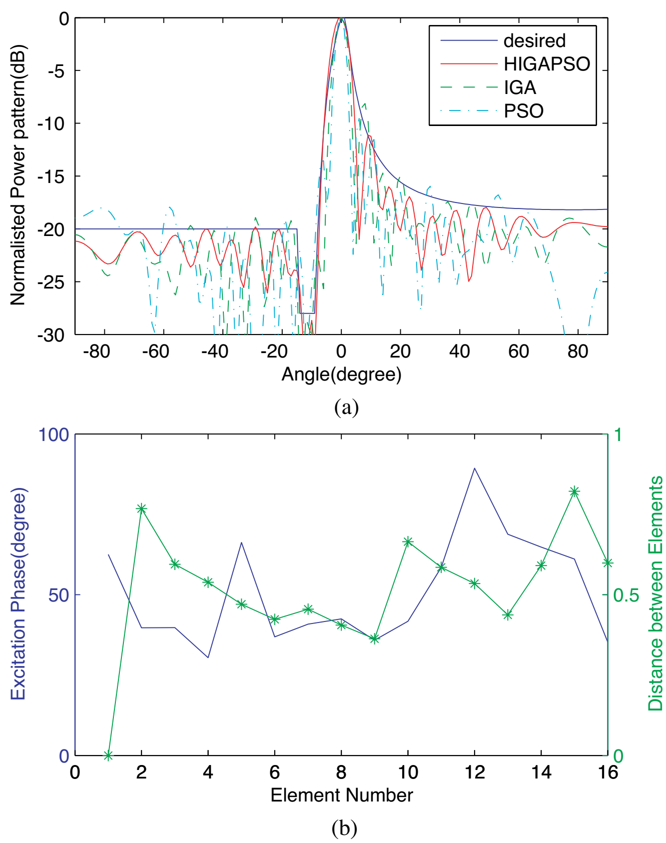

Figure 4. (a) Radiation pattern for the base station array, (b) excitation phases in degree and the distance between elements for the base station array.

with GAs and PSO. Hence, the efficiency of the proposed algorithm is proved.

5. CONCLUSIONS

This paper presents a new algorithm for the beamshaping of antenna arrays. In order to take advantage of the peculiarities of GA and PSO, the proposed algorithm integrates the main features of them into the optimization process. The computer simulation results show a very good agreement between the desired and synthesized pattern. Thus, the HIGAPSO is capable of synthesizing conformal array, linear array with prescribed nulls and base station array with complex pattern. The comparisons with SGA, IGA and PSO, demonstrate the superiority of HIGAPSO in higher convergence accuracy and fewer cost function evaluations. As the HIGAPSO has good accuracy and does not require complicated mathematical functions, it may contribute a practical approach for the pattern synthesis of antenna arrays in electromagnetic analysis.

ACKNOWLEDGMENT

The authors would like to thank the financial support form national natural science fund of P. R. China (No. 60571057).

REFERENCES

1. Guney, K. and M. Onay, "Amplitude-only pattern nulling of linear antenna arrays with the use of bees algorithm," *Progress In Electromagnetics Research*, PIER 70, 21–36, 2007.
2. Babayigit, B., A. Akdagli, and K. Guney, "A clonal selection algorithm for null synthesizing of linear antenna arrays by amplitude control," *Journal of Electromagnetic Waves and Applications*, Vol. 20, 1007–1020, 2006.
3. Hejres, J. A., "Null steering in phased arrays by controlling the positions of selected elements," *IEEE Trans. Antennas Propagat.*, Vol. 52, 2891–2895, 2004.
4. Mahmoud, K. R., M. EI-Adawy, and S. M. M. Ibrahem, "A comparison between circular and hexagonal array geometries for smart antenna systems using particle swarm optimization algorithm," *Progress In Electromagnetics Research*, PIER 72, 75–90, 2007.

5. Keizer, W. P. M. N., "Fast low-sidelobe synthesis for large planar array antennas utilizing successive fast fourier transforms of the array factor," *IEEE Trans. Antennas Propagat.*, Vol. 55, 715–722, 2007.
6. 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.
7. Mouhamadou, M., P. Vaudon, and M. Rammal, "Smart antenna array patterns synthesis: Null steering and multi-user beamforming by phase control," *Progress In Electromagnetics Research*, PIER 60, 95–106, 2006.
8. Vescovo, R., "Beam scanning with null and excitation constraints for linear arrays of antennas," *Journal of Electromagnetic Waves and Applications*, Vol. 21, 267–277, 2007.
9. Shu, L. S., S. Y. Ho, and S. J. Ho, "A novel orthogonal simulated annealing algorithm for optimization of electromagnetic problems," *IEEE Trans. Magnetics*, Vol. 40, 1791–1795, 2004.
10. Donelli, M., F. de Natale, and M. Pastorino, "Linear antenna synthesis with a hybrid genetic algorithm," *Progress In Electromagnetics Research*, PIER 49, 1–22, 2004.
11. Khodier, M. M. and C. G. Christodoulou, "Linear array geometry synthesis with minimum sidelobe level and null control using particle swarm optimization," *IEEE Trans. Antennas Propagat.*, Vol. 53, 2674–2679, 2005.
12. Ares-Pena, F. J., J. A. Rodriguez-Gonzalez, E. Villanueva-Lopez, et al., "Genetic algorithms in the design and optimization of antenna array patterns," *IEEE Trans. Antennas Propagat.*, Vol. 47, 506–510, 1999.
13. Akdagli, A., K. Guney, and B. Babayigit, "Clonal selection algorithm for design of recongigurable antenna array with iscrete phase shifters," *Journal of Electromagnetic Waves and Applications*, Vol. 21, 215–227, 2007.
14. Chen, T. B., Y. L. Dong, Y. C. Jiao, et al., "Synthesis of circular antenna array using crossed particle swarm optimization algorithm," *Journal of Electromagnetic Waves and Applications*, Vol. 20, 1785–1795, 2006.
15. Hoorfar, A., "Evolutionary programming in electromagnetic optimization: A review," *IEEE Trans. Antennas Propagat.*, Vol. 55, 523–537, 2007.

16. Zhai, Y. W., X. W. Shi, and Y. J. Zhao, "Optimized design of ideal and actual transformer based on improved micro-genetic algorithm," *Journal of Electromagnetic Waves and Applications*, Vol. 21, 1761–1771, 2007.
17. Xu, Z., H. Li, and Q. Z. Liu, "Pattern synthesis of conformal antenna array by the hybrid genetic algorithm," *Progress In Electromagnetics Research*, PIER 79, 75–90, 2008.
18. 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 Propagat.*, Vol. 55, 556–567, 2007.
19. 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, 2001–2012, 2006.
20. Liu, X. F. Y. B. Chen, Y. C. Jiao, et al., "Modified particle swarm optimization of patch antenna design based on IE3D," *Journal of Electromagnetic Waves and Applications*, Vol. 21, 1819–1828, 2007.
21. Robinson, J., S. Sinton, and Y. Rahmat-Samii, "Particle swarm, genetic algorithm, and their hybrids: Optimization of a profiled corrugated horn antenna," *IEEE International Symposium on Antennas Propagat.*, Vol. 1, 314–317, 2002.
22. Juang, C. F., "A hybrid of genetic algorithm and particle swarm optimization for recurrent netword design," *IEEE Trans. Syst., Man, Cybern. - Part B: Cybern.*, Vol. 34, 997–1006, 2004.
23. Boeringer, D. W. and D. H. Werner, "Particle swarm optimization versus genetic algorithms for phased array synthesis," *IEEE Trans. Antennas Propagat.*, Vol. 52, 771–779, 2004.
24. Leung, F. H. F., P. K. S. Tam, H. K. Lam, et al., "Tuning of the structure and parameters of a neural network using an improved genetic algorithm," *IEEE Trans. Neural NetWorks*, Vol. 47, 506–510, 2003.