Journal of Electromagnetic Waves and Applications, 2014
Vol. 28, No. 6, 752–764, http://dx.doi.org/10.1080/09205071.2014.888959

Taylor & Francis
Taylor & Francis Group

Unified particle swarm optimization with random ternary variables and its application to antenna array synthesis

Guolong He[a, b*] and Bin Wu[b]

[a]*Space Centre, Tsinghua University, Beijing 100084, China;* [b]*Beijing Institute of Tracking and Telecommunication Technology, Beijing 100094, China*

(*Received 9 December 2013; accepted 24 January 2014*)

Antenna array synthesis sometimes involves both real and integer parameters as a mixed integer optimization problem. In this paper, a modified particle swam optimization (PSO) algorithm is proposed in order to deal with real and integer variables in a unified manner. Two major modifications are made compared to the classical PSO algorithm. First a unified vector having continuous values between 0 and 1 is defined, and at each iteration this vector (or part of it) is mapped to real variables and/or rounded to integer variables, which makes the updating process unified for any type of parameters. Second, random ternary variables are introduced to compensate quantization errors caused by the rounding-off operations, which could accelerate the speed of convergence and lead to improved topology exploration capability. In order to demonstrate the effectiveness of the proposed method, three previous examples about antenna array synthesis are revisited, and better results than those in the existing literatures are obtained in all these examples.

Keywords: unified particle swam optimization; ternary variable; antenna array synthesis

1. Introduction

Antenna arrays play a very important role in modern radar and communication systems. Many optimization techniques have been studied and developed for antenna array synthesis to satisfy various requirements.[1–7] Its purpose is to obtain a desired radiation pattern by designing the geometry of the array and finding the appropriate excitations, i.e. amplitude, phase and position, of the array elements. In practice, antenna array synthesis sometimes involves both real and integer parameters, such as large linear arrays partitioned into contiguous subarrays with unequal sizes, concentric ring arrays with different element densities among rings, and phased arrays using digital phase shifters, etc. For this reason, it is highly desired that optimization algorithms used for array synthesis should be versatile enough to handle a variety of problems involving different types of parameters.

Particle swam optimization (PSO), a recently developed evolutionary optimization algorithm by imitating behaviors of swarm during their food-searching activities, has been found very effective in engineering electromagnetics and is an appealing technique for antenna array synthesis.[6–13] However, it is generally believed that PSO is superior to deal with optimization problems having only real variables, while genetic algorithm (GA) is more effective than PSO to handle optimization problems with integer variables, because PSO was intended to deal with real variables only while GA can translate a binary string to

*Corresponding author. Email: qingchunruyu@qq.com

integer value directly. Therefore, how to modify the classical real PSO (RPSO) algorithm suitable for integer optimization problems while maintaining its simplicity and outstanding performance has become great interest in recent years.[12–15]

Many efforts have been done so far to make PSO suitable for integer optimization problems. In [10], a binary PSO (BPSO) scheme was proposed and applied to thinned antenna array synthesis. For mixed integer optimization problems, BPSO can further be implemented by using binary strings to represent both real and integer variables as that in GA.[12] However, this has several obvious shortages. First, it would make the dimension of the problem extremely large, which leads to the required hardware consuming and computational costs unacceptable. Second, it is always difficult to decide the quantization level, i.e. number of bits needed for representing the real variables, thus the accuracy of optimization results is liable to be affected by this reason. Moreover, BPSO considers each bit having equal updating probability during the optimization process. But it is very critical that they do not have the same effect on the corresponding integer value, changing of the upper bit in a binary string would cause a larger jump than the lower one. In order to overcome this problem, a quantized PSO (QPSO) algorithm was proposed in [16]. It allows the integer variable represented by the binary string to only change one level each time. Unfortunately, QPSO would degrade the speed of convergence in some cases, especially at the beginning stage of the iterations.

Another widely adopted method is to implement evolutions of integer variables by rounding-off associated real values that appear in the optimization process. Many different methods based on this principle have also been proposed in existing literatures.[17–20] For example, an integer PSO (IPSO) scheme was proposed in [17]. It restricts the updating velocity having only integer values, thus position updating of the particles are simply integer calculations. An alternative way is to allow the velocities and positions of the particles still keep real values as that in RPSO, but round the positions to integers during the fitness evaluation process.[20] However, the rounding-off operations would make particles be easily trapped in a local best solution and prevent them reaching better ones, which causes inefficiencies and degrades performance of the algorithm simultaneously. Moreover, in many cases, real and integer variables are considered separately for mixed integer optimization problems, and always different updating mechanisms are applied to them.

In order to overcome these aforementioned drawbacks, a modified PSO algorithm is proposed in this paper. Two major modifications are made compared to the classical RPSO scheme. First, a unified vector having continuous values between 0 and 1 is defined, and at each iteration this vector (or part of it) is mapped to real variables and/or rounded to integer variables. The benefit of this modification is that all the operations occur in the fitness evaluation process, which makes the updating process unified for all types of variables. Second, random ternary variables are introduced to the rounding-off process, in order to compensate quantization errors and obtain improved topology exploration capability.

The rest of this paper is organized in the following structure. First in Section 2, starting from a brief review of the classical RPSO scheme, the modified PSO scheme, named UPSO-m, is proposed to deal with optimization problems having any type of variables. In order to demonstrate the effectiveness of the proposed method, three previous problems about antenna array synthesis are revisited. Section 3 synthesizes a 128-element subarrayed linear array with unequal subarray sizes and different amplitude weights, Section 4 synthesizes a concentric six ring array with different element densities among rings, and Section 5 synthesizes a 100-element phased array with four-bit digital phase shifters. Finally, this paper is summarized in Section 6.

2. The modified PSO algorithm

2.1. The classical RPSO algorithm

PSO, a swarm-based stochastic evolutionary algorithm, was first developed by Kennedy and Eberhart in 1995,[8] and introduced to the antenna community by Robinson and Rahmat-Samii in 2004.[9] Believed to be very robust and effective in multidimensional and nonlinear problems, it has been applied to many different applications.[6–23] Compared to other evolutionary methods, PSO is much easier to understand and implement and requires the least computational costs. In classical RPSO scheme, particles fly throughout the problem hyperspace to search for the optimal solution. Suppose there are N undetermined parameters in the optimization problem, then the position of each particle is denoted by an N-dimensional vector $\mathbf{x} = \{x_1, \ldots, x_n, \ldots, x_N\}$ to represent a possible solution with a velocity vector $\mathbf{v} = \{v_1, \ldots, v_n, \ldots, v_N\}$. First without any prior knowledge of the optimal solution, particles are initialized with random positions and velocities. Then for the kth iteration, these two vectors are updated according to the following equations,

$$\begin{aligned} v_n^{k+1} &= w v_n^k + c_1\phi_1\left(pbest_n^k - x_n^k\right) + c_2\phi_2\left(gbest_n^k - x_n^k\right) \\ x_n^{k+1} &= x_n^k + v_n^k \end{aligned} \tag{1}$$

where w is the inertial weight, c_1 and c_2 are the acceleration constants that specify how much each particle is influenced by the personally encountered best position, *pbest*, and the global best position ever found by the entire swarm, *gbest*, which control the relative proportion of cognition and social interaction in the swarm, respectively, and ϕ_1 and ϕ_2 are two random variables uniformly distributed on [0, 1].

Once an updating process is implemented, the new solution found by each particle is evaluated by a fitness function, then both the *pbest* of each particle and the *gbest* of the entire swarm are updated. If a criterion is met (usually a sufficiently good fitness value or a maximum number of iterations), the iteration process is terminated and the results are obtained. This paper has no intend to review the PSO algorithm extensively, the readers are referred to [7–15] for a much more detailed discussion of the algorithm and how it works.

2.2. The unified PSO algorithm with random ternary variables

Usually, in many mixed integer optimization problems, real and integer variables are treated separately and different updating mechanisms are applied.[12,17–19] Here we propose a method that works with real and integer variables in the same updating scheme. First a unified vector $\mathbf{u} = [\mathbf{u_r}, \mathbf{u_n}]$ having continuous values between 0 and 1 is defined, where $\mathbf{u_r}$ and $\mathbf{u_r}$ are the real and integer part that will be mapped to the corresponding real and integer parameters, $\mathbf{r}$ and $\mathbf{n}$, respectively. During the optimization process, the unified vector is updated by (1), and then the mapping and rounding operations are implemented during fitness evaluation process,

$$\mathbf{r} = r_{min} + \mathbf{u_r} \times (r_{max} - r_{min}) \tag{2a}$$

$$\begin{gathered} \mathbf{n} = n_{min} + \text{round}\left\{\mathbf{u_n} \times (n_{max} - n_{min} + 1) - 0.5\right\} + \mathbf{m} \\ \text{where} \quad \mathbf{m} = \text{random}\{1, 0, -1\} \end{gathered} \tag{2b}$$

where *min* and *max* are the lower and upper bounds for the variables, $r_{min} \leq r \leq r_{max}$, $n_{min} \leq n \leq n_{max}$. The round function is rounding the real value to the nearest integer. In former mapping methods,[20] integers in the range have different possible probabilities to be selected, thus the performance of the algorithms will be influenced. Here, we add the term '−0.5' in the bracket of (2b) to make each integer in the range that can be selected with equal probability.

For integer variable mapping in (2b), a new vector **m** consisting of random ternary variables is introduced. Generally speaking, there are two aspects for one to evaluate the performance of an optimization algorithm. The first aspect is how much time the algorithm costs to find a suitable solution, whereas the second aspect is concerned with the best possible solution ever found by the algorithm. The random ternary vector will influence the algorithm in both aspects. At the beginning stage of the iterations, it may (only may) have some negative effects on the convergence speed of the algorithm. Because updating of the integer part $\mathbf{u_r}$ is the main contribution in (2b) during this period, so the effect of the ternary vector is considered to be very tiny. But at the latter stage of the iterations, the ternary vector can be regarded as additional searching operations nearby the ever found solutions, which leads to better solutions and improved topology exploration capability.

In order to give better understandings of these explanations and demonstrate the effectiveness of the proposed method, we revisit three previous problems in the following sections. We use UPSO-m and UPSO to represent the unified PSO scheme with and without random ternary variables for simplicity, respectively. During the optimization process, the parameters of PSO algorithm are set as suggested by previous literatures,[7–15] w is decreased linearly from 0.9 to 0.4 during iterations, c_1 and c_2 are chosen to be two for better convergence performance, and number of particles are a little larger than the dimension of the optimization problem.

3. Synthesis of the linear array with subarray configuration

Antenna array synthesis is a process to trade-off between different aspects of performances in order to meet specified design goals, we can not improve one aspect significantly without scarifying another.[1–3,24] It is a well-known technique to partition a large linear antenna array into contiguous subarrays in order to reduce cost via common use of components and simplified feed networks. Unfortunately, placing amplitude weights at the subarray ports will create grating lobes (GLs) due to the periodicity and larger spacing between subarrays. As shown in Figure 1, an N-element linear array is placed along x-axis and divided into Q subarrays, with different amplitude weights at the subarray output ports. The elements are equally spaced and the distance d between adjacent elements is 0.5λ. The corresponding array factor $f(\theta)$ is given by,

$$f(\theta) = \sum_{q=1}^{Q} w_q \sum_{i=1}^{n_q} \cos(2\pi x_n \sin\theta) \tag{3}$$

where x_n is the position of the nth element in wavelength, θ is the angle relative to boresight, n_q and w_q are the number of elements and the amplitude weight of the qth subarray, respectively.

First let us consider each subarray having the same number of elements. In RPSO, a Q-dimensional vector is defined to represent the corresponding amplitude weights. A 10-agent swarm is used in optimization for 1000 iterations. Figure 2 illustrates the optimized pattern of a 128-element linear array divided into 16 equal-sized subarrays, along with the pattern of the uniform array. The obtained peak sidelobe level (SLL) is −31.4 dB. Table 1 summarizes the best solutions obtained by different optimization methods. The same array with uniform tapering has a peak SLL of −13.3 dB. For a 31 dB and n = 5 Taylor tapering, the peak SLL is −30.5 dB.[1] The result is −30.9 dB obtained by GA with Nelder–Mead

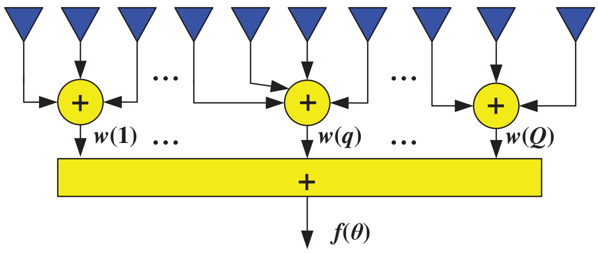

Figure 1. Geometry of the linear array with subarray configuration of unequal subarray sizes and different amplitude weights.

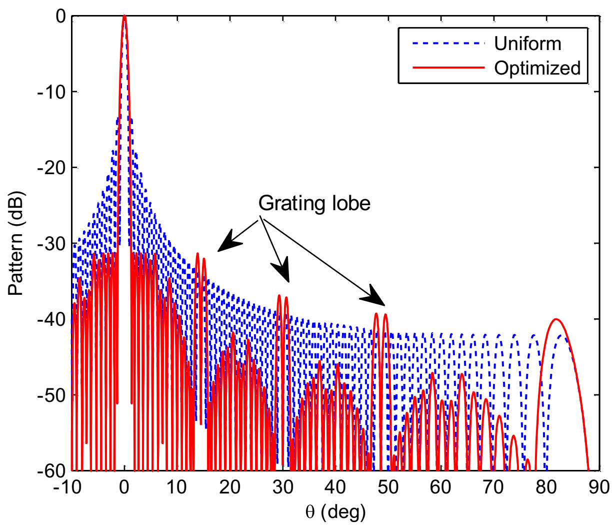

Figure 2. The optimized radiation pattern for the 128-element linear array with equal subarray sizes.

Table 1. The best results obtained by different optimization algorithms.

Method	Subarray sizes	SLL (dB)
Uniform	Equal	−13.3
Taylor tapering	Equal	−30.5
GA	Equal	−30.9
	Unequal	−35.1
UPSO-m	Equal	−31.4
	Unequal	−36.1

downhill simplex method.[25] Although PSO has found the best solution than any other algorithms, but it has been clearly showed that the GLs prevent the optimization process to find better solutions.

Many methods have been proposed to break the periodicity and redistribute the energy of GLs through the array pattern.[25–27] One of these methods is to allow each subarray having a different number of elements. Since there are two different types of parameters

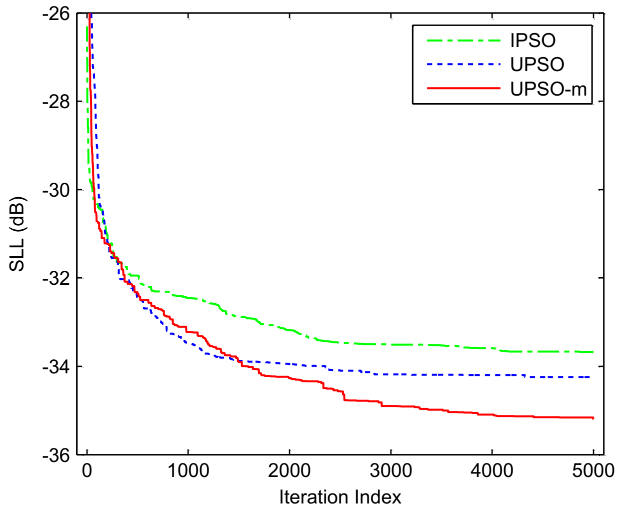

Figure 3. Comparison of convergence performance of different PSO algorithms over 10 independent trials.

Table 2. Comparison of best convergence performance of different PSO algorithms over 10 independent trials.

Method	Average (dB)	Best (dB)
IPSO	−33.67	−35.04
UPSO	−34.24	−35.30
UPSO-m	−35.19	−36.10

in this optimization problem, i.e. integers for the subarray sizes and real numbers for the amplitude weights, UPSO-m is applied for further simulation rather than RPSO. A $2Q$-dimensional unified vector $\mathbf{u} = [\mathbf{u_w}, \mathbf{u_n}] = \{u_1, u_2, \ldots, u_{2Q}\}$ is designated to each candidate design. The first part $\mathbf{u_w} = \{u_1, u_2, \ldots, u_Q\}$ represent the normalized amplitude weights of the subarrays, and the latter part $\mathbf{u_n} = \{u_Q, u_{Q+1}, \ldots, u_{2Q}\}$ are mapped and rounded to the subarray sizes. In order to demonstrate the effectiveness of the method proposed in this paper, UPSO-m is validated and compared with other existing PSO schemes. For each method, a 20-agent swarm is used in optimization for 5000 iterations. Figure 3 illustrates convergence of *gbest* values averaged over 10 independent trials, and the performances are summarized in Table 2. At the end of iteration, an averaged *gbest* value of −35.19 dB is obtained by UPSO-m, compared with −33.67 and −34.24 dB by IPSO and UPSO, respectively. As can be seen in Figure 3, at the beginning stage of the iterations, convergence performance of UPSO-m is a little worse than UPSO. But in the latter stage of the iterations, while the other two PSO schemes have little improvements, UPSO-m continues to find better solutions.

During the 10 independent trials, the best solution ever found by UPSO-m is −36.1 dB, which is also better than IPSO (−35.0 dB), UPSO (−35.3 dB), and GA (−35.1 dB).These results have been clearly showed that UPSO-m has the best convergence performance and topology exploration capability, and most of the improvements should be attributed to the random ternary variables. Figure 4 illustrates the optimized subarray sizes and amplitude

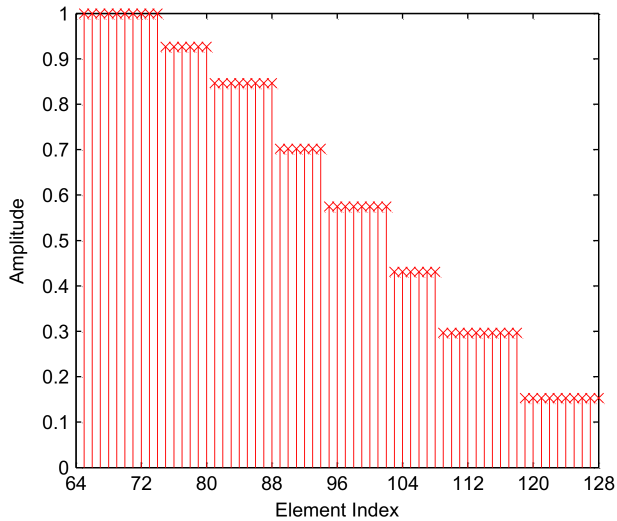

Figure 4. The optimized subarray sizes and amplitude weights for the 128-element linear array.

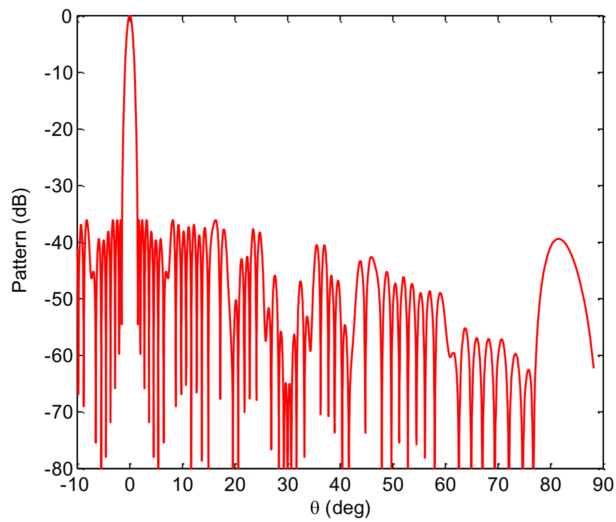

Figure 5. The optimized radiation pattern of the 128-element linear array with unequal subarray sizes.

weights, which $\mathbf{n}$ = [10, 6, 8, 6, 8, 6, 10, 10] and $\mathbf{w}$ = [1.0000, 0.9264, 0.8463, 0.7018, 0.5748, 0.4308, 0.2964, 0.1529]. Since the array is symmetric, only the right hand side is shown. The corresponding optimized array pattern is illustrated in Figure 5, which has nearly equal sidelobe levels without GLs.

4. Synthesis of the concentric ring array with different element densities

A concentric ring array consists of several ring arrays having different radii and sharing the same geometry center. It has several advantages such as azimuthally symmetric pattern,

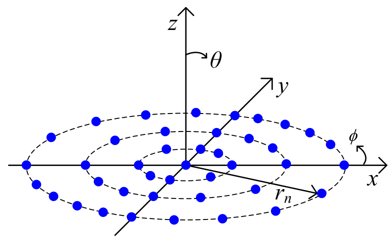

Figure 6. Geometry of a concentric ring array.

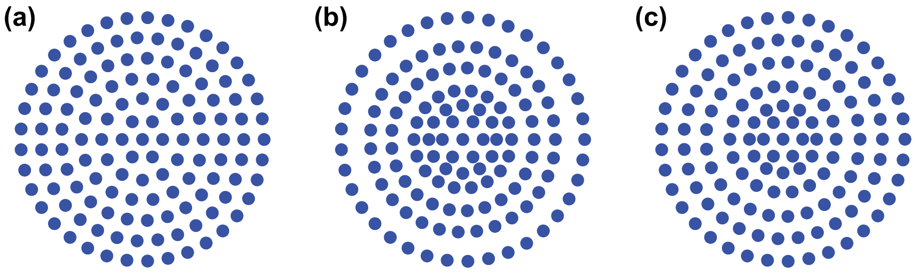

Figure 7. Geometry of the concentric six ring array: (a) uniform array, (b) array with the optimized ring radius only, and (c) array with the optimized ring radius and element density.

maximum utilization of a circular aperture, and reduced grating lobe potential, which has been widely used for remote sensing, radar tracking, direction finding, and many other applications.[28–33] Figure 6 illustrates the geometry of a concentric ring array with Q ring arrays and a single element at the center. Elements are equally located in each ring, r_q and m_q represent the ring radius and number of elements in the qth ring, respectively. The corresponding array factor $f(\theta, \phi)$ is given by,

$$f(\theta,\phi) = 1 + \sum_{q=1}^{Q}\sum_{i=1}^{m_q} \exp\left[jkr_q \sin\theta \cos\left(\phi - (m-1)\phi_q\right)\right] \tag{4}$$

where $k = 2\pi/\lambda$ is the wavenumber, $\phi_q = 2\pi/m_q$ is the azimuth angle between two adjacent elements of the qth ring. Figure 7(a) illustrates the geometry of an uniform concentric six ring array with 0.5λ equally spaced distance between adjacent rings. It has a total element number of 130, and the distance between elements in each ring is chosen to near 0.5λ. The corresponding 3D array pattern is shown in Figure 8(a), which has a peak SLL of −17.34 dB, and a directivity of 26.00 dB.

Usually, the sidelobe levels of the concentric ring arrays are suppressed by amplitude tapering at each ring, which results in inefficiency of the radiation power and difficulty of designing the feed networks.[28,29] A synthesis strategy is proposed in [30] which does not require the exploitation of global optimization procedures. However, this approach cannot be applied to generic planar arrays and generic constraints. The geometry optimization with equal amplitude excitations can also lower the SLLs while making the feeding networks much simpler and cheaper to produce. In previous works,[31–33] this problem has been investigated by GA and modified particle swarm optimization (MPSO) algorithm, which resets the optimal solution in a random manner during the optimization process. Let each

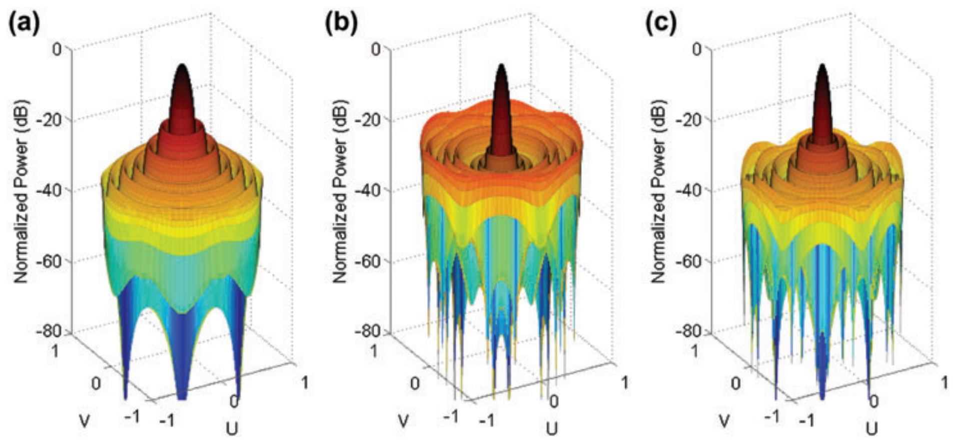

Figure 8. The 3D array pattern of the concentric six ring array: (a) uniform array, (b) array with the optimized ring radius only, and (c) array with the optimized ring radius and element density.

Table 3. The best solutions obtained by different algorithms.

Array	Algorithm	SLL (dB)	Directivity (dB)	Total Number
Uniform	–	−17.34	26.00	130
Opt. m_q only	GA	−22.94	27.38	201
	MPSO	−23.37	27.51	205
	UPSO	−23.66	26.30	183
Opt. m_q and m_q	GA	−27.82	28.89	142
	MPSO	−27.89	29.07	133
	UPSO-m	−29.00	27.83	148

Table 4. Geometries of the optimized arrays.

Array	Geometry	Ring 1	Ring 2	Ring 3	Ring 4	Ring 5	Ring 6
Uniform	r_q	0.5	1.0	1.5	2	2.5	3
	m_q	6	12	18	25	31	37
Opt. r_q only	r_q	0.7608	1.2608	1.8196	2.6587	3.4467	4.5281
	m_q	10	16	23	33	43	57
Opt. r_q and m_q	r_q	0.6967	1.1967	1.9025	2.7560	3.5458	4.3513
	m_q	9	19	30	33	28	28

ring be nonuniformly spaced, and the radius of the qth ring is,

$$r_{q+1} = r_q + d_q$$
$$\text{where} \quad d_q \in [0.5\lambda, 1.5\lambda] \tag{5}$$

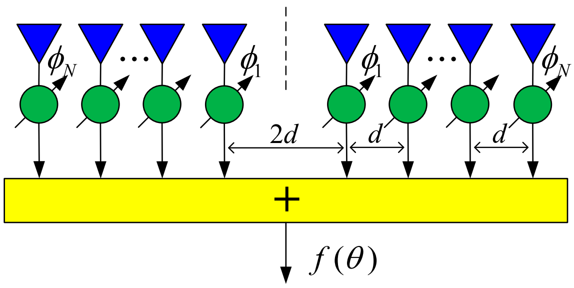

Figure 9. Centro-symmetric phased array with digital phase shifters.

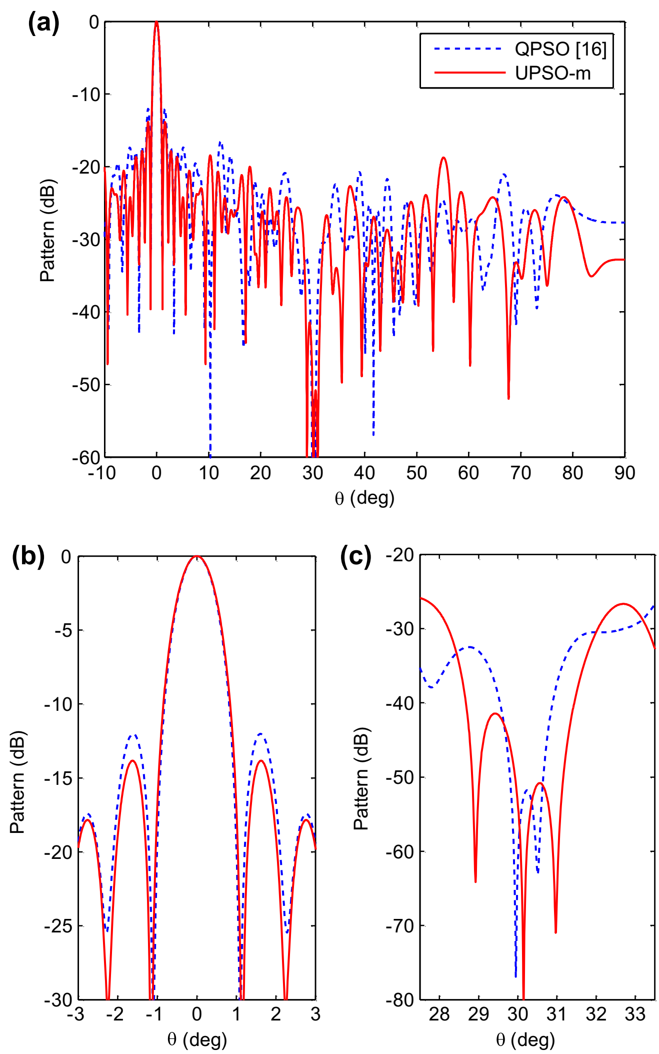

Figure 10. (a) The optimized array pattern for sidelobe reduction and interference suppression; (b) Zoomed array pattern of the main beam and peak SLL; and (c) Zoomed array pattern of the suppressed region of [30°, 31°].

where d_q is the spacing between the qth and $q+1$th ring, and the distance between elements in each ring is to keep near 0.5λ. Since the values of the radii are real, a six-dimensional vector for UPSO is defined to represent each candidate design, and a 10-agent swarm is used in optimization for 100 iterations. The optimized array configuration is illustrated in Figure 7(b), and the corresponding 3D array pattern is shown in Figure 8(b). It has 183 elements in the array, and the optimized peak SLL is −23.66 dB, which is better than −22.94 dB of GA [31] and −23.37 dB of MPSO [32]. An even more effective optimization method is to optimize both the ring radius and the element density in each ring. A $2Q$-dimensional unified vector $\mathbf{u} = [\mathbf{u_r}, \mathbf{u_m}]$ is defined to represent a candidate design, the first part $\mathbf{u_r}$ are mapped to the radii of the rings, and the latter part $\mathbf{u_m}$ are mapped and rounded to number of elements in each ring. A 20-agent swarm is used in optimization for 100 iterations. Figure 7(c) illustrates the optimized geometry of the array, and the corresponding 3D array pattern is shown in Figure 8(c). The total element number of the optimized array is 148, and the peak SLL is −29.00 dB. Compared to the results obtained by GA and MPSO, more than one dB improvement is obtained by UPSO-m. Table 3 summarizes the best solutions obtained by different algorithms, and the geometries of the optimized arrays are given in Table 4.

5. Synthesis of the phased array with digital phase shifters

Synthesis of phased arrays has also been studied extensively over past years.[2,16,34] Since each element of the phased array has a phase shifter used for electronically beam steering, array pattern synthesis can also be realized via phase adjustments of the array elements, which allows simplified uniform amplitude excitation feed networks while avoiding additional hardware and costs. With the advance of modern radio frequency (RF) technology, digital phase shifters are nowadays much more widely used in practice than continuous phase shifters. Thus, the optimized continuous phases obtained by optimization algorithms must be rounded to the closest quantized value for implementation. This often yields approximate results rather than optimal results, and quantization errors will degrade the array sidelobe performance. Therefore, a PSO scheme dealing with integer variables is more suitable than RPSO for phased array synthesis.

Consider a uniform excited $2N$-element centro-symmetric phased array with equally spacing d as shown in Figure 9.[16] Assume phases of the array elements are symmetrical about the center, the array factor $f(\theta)$ is given by,

$$f(\theta) = 2\sum_{n=1}^{N} \cos[2\pi(n-1)d \cdot \sin\theta + \phi_n] \tag{6}$$

where ϕ_n is the excitation phase of the nth element and θ is the angle relative to boresight. In this problem, the performance of the phased array is evaluated by a multi-objective fitness function containing two aspects, sidelobe reduction and interference suppression, which is defined as follows,

$$fitness = \alpha \cdot SLL + \beta \cdot \max_{\theta\in\theta_{SR}} f(\theta) \tag{7}$$

where θ_{SR} means the desired suppressed region, α and β are coefficients reflecting the significance of the corresponding terms. In [16], a 100-element phased array with four-bit digital phase shifters and 0.5λ spaced distance was investigated and optimized by QPSO. The obtained peak SLL is −12.0 dB, and the null depth is −42 dB.

During the optimization process of UPSO-m, α and β are set to 0.8 and 0.2, respectively. Although the phase values can have discrete values ranging from −180° to 180°, we still take the phase shifting range of [0°, 114.5°] with step of 7.63° as the same as in [16]. A 50-agent swarm is used in optimization for 500 iterations. Figure 10 illustrates the optimized array pattern, along with the results reported in [16]. It has a peak SLL of −13.9 dB, and a null depth of −51 dB in the suppressed region of [30°, 31°], both of which are better than the results obtained by QPSO. This validates the effeteness of the method proposed in this paper again.

6. Conclusion

In order to deal with real and integer variables in a unified manner, a modified PSO scheme with random ternary variables, named UPSO-m, is proposed in this paper. Two major modifications are made compared to the classical PSO algorithm. First, a unified vector having continuous values between 0 and 1 is defined, which makes the updating process unified for any type of parameters. Second, random ternary variables are introduced to compensate quantization errors caused by the rounding-off operations. Although this would cause some degradation of the convergence speed at the beginning stage of the iterations, but leads to improved topology exploration capability than other algorithms. In order to demonstrate the effectiveness of the proposed method, three problems about antenna array synthesis are revisited, and better results than those in the existing literatures are obtained in all these problems. The method proposed in this paper not only can be used for antenna array synthesis, but also has potential application to other integer optimization problems.

Funding

This work was supported by National Natural Science Foundation of China [grant number 61271265], [grantnumber 61032003], [grant number 61132002]; National High Technology Research and Development Program [grant number 2012AA121605]; and Tsinghua University Initiative Scientific Research Program [grant number 2011Z05112].

References

[1] Elliott RS. Antenna theory and design. Hoboken (NJ): Wiley-Interscience, IEEE; 2003.
[2] Mailloux RJ. Phased array antenna handbook. Boston (MA): Artech House; 2005.
[3] Haupt RL. Antenna arrays: a computational approach. New York, (NY): Wiley; 2010.
[4] Cheng DK. Optimization techniques for antenna arrays. Proc. IEEE. 1971;59:1664–1674.
[5] Yan KK, Lu Y. Sidelobe reduction in array-pattern synthesis using genetic algorithm. IEEE Trans. Antenn. Propag. 1997;45:1117–1122.
[6] Khodier M, Christodoulou C. Linear array geometry synthesis with minimum sidelobe level and null control using particle swarm optimization. IEEE Trans. Antenn. Propag. 2005;53: 2674–2679.
[7] Boeringer DW, Werner DH. Particle swarm optimization versus genetic algorithms for phased array synthesis. IEEE Trans. Antenn. Propag. 2004;52:771–779.
[8] Kennedy J, Eberhart R. Particle swarm optimization. Proc. IEEE Int. Conf. Neural Netw. 1995;4:1942–1948.
[9] Robinson J, Rahmat-Samii Y. Particle swarm optimization in electromagnetics. IEEE Trans. Antenn. Propag. 2004;52:397–407.
[10] Jin NB, Rahmat-Samii Y. Advances in particle swarm optimization for antenna designs: real-Number, binary, single-objective and multi-objective implementations. IEEE Trans. Antenn. Propag. 2007;55:556–567.
[11] Rahmat-Samii Y, Kovitz JM, Rajagopalan H. Nature-inspired optimization techniques in communication antenna designs. Proc. IEEE. 2012;100:2132–2144.

[12] Jin NB, Rahmat-Samii Y. Hybrid real-binary particle swarm optimization (HPSO) in engineering electromagnetics. IEEE Trans. Antenn. Propag. 2010;58:3786–3794.
[13] Mikki SM, Kishk AA. Quantum particle swarm optimization for electromagnetics. IEEE Trans. Antenn. Propag. 2006;54:2764–2775.
[14] Perez JR, Basterrechea J. Particle swarm optimization with tournament selection for linear array synthesis. Microw. Opt. Techn. Lett. 2008;50:627–632.
[15] Modiri A, Kiasaleh K. Modification of real-number and binary PSO algorithms for accelerated convergence. IEEE Trans. Antenn. Propag. 2011;59:214–224.
[16] Ismail TH, Hamici ZM. Array pattern synthesis using digital phase control by quantized particle swarm optimization. IEEE Trans. Antenn. Propag. 2010;58:2142–2145.
[17] Gaing Z. Constrained optimal power flow by mixed-integer particle swarm optimization. Proc. Power Eng. Soc. Gen. Meet. 2005;1:243–250.
[18] Laskari EC, Parsopoulos KE, Vrahatis MN. Particle swarm optimization for integer programming. IEEE Congress Evolut. Comput. 2002;2:1582–1587.
[19] Hoorfar A, Nelaturi S, Zhu J. Electromagnetic optimization using a mixed-parameter self-adaptive evolutionary algorithm. Microw. Opt. Techn. Lett. 2003;55:267–271.
[20] Haupt RL. Antenna design with a mixed integer genetic algorithm. IEEE Trans. Antenn. Propag. 2007;55:577–582.
[21] Shi Y, Eberhart RC. A modified particle swarm optimizer. In: Proceedings of IEEE International Conference on Evolutionary Computation; 1998; Alaska, USA. p. 69–73.
[22] Eberhart RC, Shi Y. Comparing inertia weights and constriction factors in particle swarm optimization. In: Proceedings of the Congress on Evolutionary Computation; 2000; California, USA. p. 84–88.
[23] Poli R. Analysis of the publications on the applications of particle swarm optimisation. J. Artif. Evol. Appl. 2008:1–10.
[24] Panduro MA, Covarrubias D, Brizuela C, Marante FR. A multi-objective approach in the linear antenna array design. Int. J. Electron. Commun. AEUE. 2005;59:205–212.
[25] Haupt RL. Optimized weighting of uniform subarrays of unequal sizes. IEEE Trans. Antenn. Propag. 2007;53:1207–1210.
[26] Goffer P, Kam M, Herczfeld PR. Design of phased arrays in terms of random subarrays. IEEE Trans. Antenn. Propag. 1994;42:820–826.
[27] Toyama N. Aperiodic array consisting of subarrays for use in small mobile earth stations. IEEE Trans. Antenn. Propag. 2005;53:2004–2010.
[28] Stearns C, Stewart A. An investigation of concentric ring antennas with low sidelobes. IEEE Trans. Antenn. Propag. 1965;13:856–863.
[29] Li YH, Ho KC, Kwan C. 3-D array pattern synthesis with frequency invariant property for concentric ring. IEEE Trans. Signal Proc. 2006;54:780–784.
[30] Morabito AF, Lagana AR, Isernia T. On the optimal synthesis of ring symmetric shaped patterns by means of uniformly spaced planar arrays. Prog. Electrom. Res. B. 2010;20:33–48.
[31] Haupt RL. Optimized element spacing for low sidelobe concentric ring arrays. IEEE Trans. Antenn. Propag. 2008;56:266–268.
[32] Zhang S, Gong SX, Zhang PF. A Modified PSO for Low Sidelobe Concentric Ring Arrays Synthesis with Multiple Constraints. J. Electromagn. Waves Appl. 2009;23:1535–1544.
[33] Reyna A, Panduro MA, Covarrubias D, Mendez A. Design of steerable concentric rings array for low side lobe level. Int. J. Sci. Techn. (Scientia Iranica) Trans. D Elec. Eng. 2012;19:727–732.
[34] Keizer W. Low sidelobe phased array pattern synthesis with compensation for errors due to quantized tapering. IEEE Trans. Antenn. Propag. 2011;59:4520–4524.