Hindawi
International Journal of Antennas and Propagation
Volume 2019, Article ID 1064103, 13 pages
https://doi.org/10.1155/2019/1064103

Research Article

Multiobjective Synthesis of Linear Arrays by Using an Improved Genetic Algorithm

Bo Yang

School of Electronic Science and Engineering, Jilin University, 130012 Changchun, China

Correspondence should be addressed to Bo Yang; yangboall@126.com

Received 18 July 2018; Revised 5 September 2018; Accepted 4 April 2019; Published 22 July 2019

Academic Editor: Shiwen Yang

In this paper, an improved genetic algorithm with dynamic weight vector (IGA-DWV) is proposed for the pattern synthesis of a linear array. To maintain the diversity of the selected solution in each generation, the objective function space is divided by the dynamic weight vector, which is uniformly distributed on the Pareto front (PF). The individuals closer to the dynamic weight vector can be chosen to the new population. Binary- and real-coded genetic algorithms (GAs) with a mapping method are implemented for different optimization problems. To reduce the computation complexity, the repeat calculation of the fitness function in each generation is replaced by a precomputed discrete cosine transform matrix. By transforming the array pattern synthesis into a multiobjective optimization problem, the conflict among the side lobe level (SLL), directivity, and nulls can be efficiently addressed. The proposed method is compared with real number particle swarm optimization (RNPSO) and quantized particle swarm optimization (QPSO) as applied in the pattern synthesis of a linear thinned array and a digital phased array. The numerical examples show that IGA-DWV can achieve a high performance with a lower SLL and more accurate nulls.

1. Introduction

Linear array pattern synthesis with multiobjective optimization has been a hot research topic in recent years [1–9]. The common objectives of such approach are the side lobe level (SLL), directivity, main beam width, and null depth, which may be contradictory to each other. To simplify the feed network of the array, discrete adjustment, instead of continuous adjustment, of the excitation amplitude and phase of the array elements can be carried out. The simple constructions can apply a thinned array by controlling the switch of the elements [10–14] and phase array with the use of an C-bit digital phase shifter [15–17] or some other device.

Many optimization methods construct objective functions into one function, which is defined as aggregating function [1, 2]. The optimization method based on a backtracking search optimization algorithm (BSA) is used to synthesize array pattern with the prescribed nulls and the low SLL [2]. In [10], a suitable mapping method is used, and a modified particle swarm optimization (PSO) is found to be capable of effectively addressing the discrete optimization problems of linear array pattern synthesis. This strategy has been applied to thinned linear array pattern synthesis with a minimum SLL. A modified PSO algorithm called quantized PSO (QPSO) has been described for the synthesis of antenna array patterns by using digital phase shifters. The fitness function includes a weighted sum of the main beam width value, the sum of the null depths at the interference signal directions, and the SLL value [15]. This optimization method by aggregating function may get the result that looks good, but the parameters of coefficients in the fitness function are difficult to decide.

In [4], the authors showed that the linear antenna array design could be modeled as a multiobjective optimization problem (MOP). Evolutionary multiobjective optimization methods, such as NSGA-II, DEMO, SPEA-2, and EM-MOPSO, have been developed for the synthesis of linear antenna arrays and concentric ring antenna arrays, with the main beam width and SLL [1, 4–7] as objectives. NSGA-II [9] is a popular and efficient multiobjective genetic algorithm that has been used in several engineering design problems [18, 19]. For instance, NSGA-II has been extensively applied

in the synthesis of antenna arrays [20–23] and is considered as one of the best evolutionary optimizers for multiobjective problems [7, 24]. What can be focused, the simulation results of optimum solutions distributed in the Pareto front (PF) should keep diversity for every fitness. The classic NSGA-II was applied for 40 elements thinned linear array [22], but the values of the null depth were only -32.7 and -22 dB with low SLL. Some hybrid algorithms, such as memetic generalized differential evolution (MGDE3) [20] and iterative fast Fourier transform (IFFT) with a judge factor introduced into the NSGA-II [23], have been used to try to improve the convergence and solution diversity.

Thinned arrays have been widely used in array pattern synthesis due to the excellent performance [11, 12, 14, 25, 26]. In [10], a suitable mapping method is used, and the modified PSO is found to be capable of effectively addressing the discrete optimization problems of linear array pattern synthesis. This strategy has been applied to thinned linear array pattern synthesis with a minimum SLL. In that study, a rounding strategy and real number PSO (RNPSO) are combined to enable PSO to solve 0-1 discrete optimization problems, integer optimization problems, and mixed optimization problems. In the mapping process, a round-down function is applied instead of a round function to guarantee that each integer value can be equally selected [10].

Phased arrays are considered to be the best solution in beam scanning and widely used in engineering [7, 15–17, 20]. With the advance in technology, digital phase shifters are now widely used in phased arrays to provide beam scanning and interference suppression [24]. A modified PSO algorithm called quantized PSO (QPSO) has been described for the synthesis of antenna array patterns with the use of digital phase shifters [15]. The solution space of the QPSO algorithm is restricted to the finite quantized integer values of the array phase coefficients. The QPSO searches for an optimal solution within the available quantized values of the digital phase shifters to minimize the fitness function, which includes the SLL value and interference suppression, while keeping the main beam unchanged. The fitness function includes a weighted sum of the mean beam width value, the sum of the null depths at the interference signal directions, and the SLL value. The QPSO searches for an optimal solution within the available quantized (discrete) values of the phase shifters that minimize the specified fitness function.

A genetic algorithm (GA) is generally believed to be suitable for discrete optimization problems because it uses a discrete coding method and deals directly with discrete variables. In the present study, an improved genetic algorithm with dynamic weight vector (IGA-DWV) is proposed for array pattern synthesis. The objectives of the linear array are synchronously optimized for different fitness functions. The innovations and the effectiveness obtained in this work are described below:

(1) A multiobjective optimization problem for array pattern synthesis: A multiobjective evolutionary algorithm, instead of a single objective obtained by aggregating functions, is applied to linear array pattern synthesis. The SLL, directivity, and prescribed null positions are the optimized objectives for different arrays.

(2) The application of the dynamic weight vector to address the diversity of solutions: To maintain the diversity of the selected solution in each generation, the solutions closer to the dynamic weight vector uniformly distributed on the PF can be chosen to the new population.

(3) A lower SLL and a more accurate null position: A lower SLL and better null depth and width are obtained by the multiobjective optimization method and the appropriate coded method. The results show a higher performance than those in the existing research literature.

The remainder of this paper is organized as follows. In Section 2, the improved multiobjective genetic algorithm with dynamic weight vector is proposed. In Section 3, the multiobjective optimization problems of array pattern synthesis are presented. Section 4 simulates numerical examples and describes the comparative performance of the proposed technique. Finally, the conclusions are given in Section 5.

2. Improved Multiobjective Genetic Algorithm with Dynamic Weight Vector

The multiobjective optimization problem can be expressed as

$$\begin{aligned} \min \quad & \mathbf{f}(\mathbf{x}) = \left(f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_M(\mathbf{x})\right)^T \\ \text{subject to} \quad & \mathbf{x} = \left(x_1, x_2, \ldots, x_n\right) \in \Omega \end{aligned} \tag{1}$$

where $\mathbf{x}$ is the decision vector, and Ω is the feasible region in the decision space. Here, M is the number of objectives, and n is the size of the decision vector.

Very often, because the objectives in (1) contradict each other, no point in Ω minimizes all the objectives simultaneously; instead, the objectives have to be balanced. The best trade-offs among the objectives can be defined in terms of Pareto optimality.

Considering a minimization problem for each objective and $\mathbf{p}, \mathbf{q} \in R^M$, $\mathbf{p}$ is said to dominate $\mathbf{q}$ (written as $\mathbf{p} \succ \mathbf{q}$) if and only if $p_i \succ q_i$ for every $i \in \{1, ..., M\}$, and $p_i < q_i$ for at least one i. We can obtain a solution $\mathbf{x}^* \in \Omega$, the Pareto optimal to (1), if we cannot find a solution $\mathbf{x} \in \Omega$ such that $F(\mathbf{x}) \succ F(\mathbf{x}^*)$. Then, $F(\mathbf{x}^*)$ is called a Pareto optimal (objective) vector. In other words, any improvement in a Pareto optimal solution in one objective must lead to the deterioration of at least one other objective. The set of all the Pareto optimal solutions is called the Pareto set (PS), and the set of all the Pareto optimal objective vectors is the PF [27]. The multiobjective optimization algorithm aims to find the optimum solutions approximating the actual PF.

In this study, IGA-DWV is proposed for array pattern synthesis. Figure 1 shows the procedure. The key technology focuses on the coding for different problems and the selection operator.

The main step in the process is the nondominated sorting approach which is the same as that in [9]. After the nondominated sorting operator, calculate the dynamic weight vector for each front. Copy the individuals with lower rank to the new population as long as the number of individuals does not overflow the population size Q. In the last front, the individuals closer to the weight vector should be copied to the

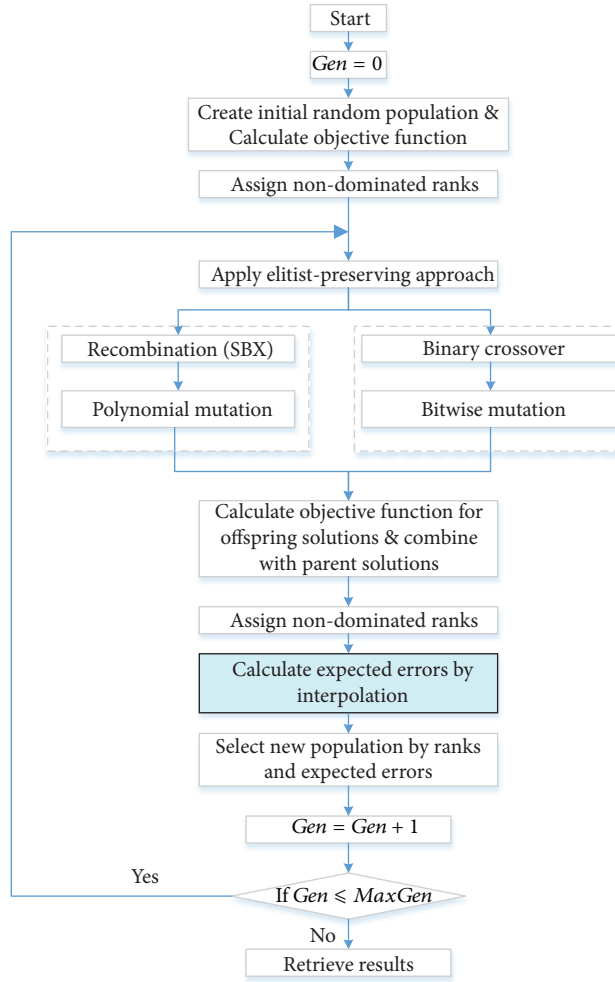

Figure 1: The main flowchart of IGA-DWV.

new population until the total number of individuals reaches Q.

The uniform crossover and bitwise mutation is used for binary-coded GAs. The crossover operator for binary-coded GAs can be expressed as

$$\begin{aligned} \mathbf{x}_{t+1} &= \mathbf{C}_{rc}^{\mathbf{x}_t\mathbf{y}_t}\mathbf{x}_t + \left(1 - \mathbf{C}_{rc}^{\mathbf{x}_t\mathbf{y}_t}\right)\mathbf{y}_t \\ \mathbf{y}_{t+1} &= \mathbf{C}_{rc}^{\mathbf{x}_t\mathbf{y}_t}\mathbf{y}_t + \left(1 - \mathbf{C}_{rc}^{\mathbf{x}_t\mathbf{y}_t}\right)\mathbf{x}_t \end{aligned} \tag{2}$$

where $\mathbf{C}_{rc}^{\mathbf{x}_t\mathbf{y}_t}$ is the binary crossover sequence for the individuals $\mathbf{x}_t$ and $\mathbf{y}_t$ in the t-th generation, which is constituted with the probability p_c.

For real-coded GAs, the simulated binary crossover (SBX) operator and polynomial mutation [28] are used. Every decision in the individual is a real number that can be normalized between 0 and 1. To solve the discrete problems, the real number should be mapped to the binary code. For the C-bit binary code mapping $x_n \longrightarrow I_n$, the procedure can be described as

$$I_n = \text{round}\left\{x_n \times \left(2^C - \varepsilon_1\right) - \left(0.5 - \varepsilon_2\right)\right\} \tag{3}$$

where $\varepsilon_1 = 2\varepsilon_2$, which are two very small numbers, and round means to round the value in the bracket to its nearest integer. This mapping can guarantee that each discrete value has an equal probability of being selected. The small errors ε_1 and ε_2 can constrain the binary data not out of bound, and not destroy the equal probability.

To select the individuals with high performance, non-dominated ranks are effective for the convergence of population. The solutions in the last front should be carefully considered. Here, a group of weight vector uniformly distributed on the PF is expected to be obtained. The weight vector can divide the space to N gridding, and the individuals nearest to this weight vector should be selected to the new population. This method can keep the diversity of final solution, which means that the final solution can better distribute on the PF.

For most MOPs, the PFs are not flat, which is shown in Figure 2 with solid lines. The points uniformly distributed on the straight line must not be uniformly distributed on the PF. So, the weight vector cannot be simply obtained by uniformly dividing every objective function space. The characteristic of the PF is the main key to decide the expected weight vector.

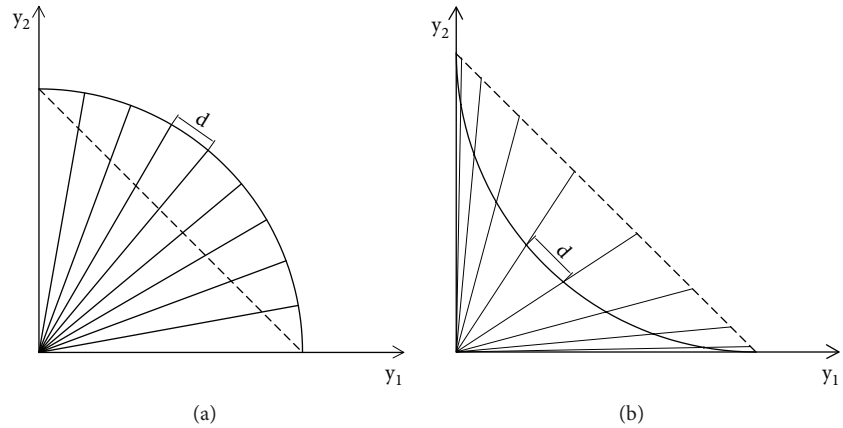

Figure 2: The weight vector uniformly distributed on convex and nonconvex PFs.

To get the uniformly distributed points on the PF, one group of excellent weight vector can achieve this goal, which is calculated according to the slope of the PF. The weight vector will be updated in each generation. In the t-th generation of evolution, the initial weight vector $\lambda^t = [\lambda_1^t, \lambda_2^t, ..., \lambda_Q^t]$ constituted by Q weights is uniformly distributed on every objective function space with unit length simultaneously. Q is the total number of initial weights, and $Q = K * N$. N is the size of population, and K is a constant, which is defined according to the accuracy of distributed uniformness. A larger value of K can get better uniformness.

The main step is to select λ_s^t with size of N from λ^t. In the t-th generation of evolution, one new population is combined by the offspring and the parent. One representative function for the normalized objective function values of the combined population can be calculated by fitting function or interpolation function [24]. The effective weight vector λ_s^t is corresponding to the dynamic reference points uniformly distributed on the representative function. The dynamic reference points can be calculated by dividing the representative function with the same length of the curve.

The length of a certain period of curve is related to the slope here. In the initial weight vector, there are Q points corresponding to λ^t. The length of the curve between q-th and $(q+1)$-th point can be shown as

$$d_{q,q+1} \approx \sqrt{1 + \left(y_q'\right)^2} \tag{4}$$

where y_q' is the first-order derivative of the representative function on the q-th point. The probability of choosing the q-th weight is decided by $d_{q,q+1}$. Associated with Figure 2, λ^t is uniformly distributed on the unit length for every objective function, which is a straight line. So, the length $d_{q,q+1}$ should be normalized too. The normalized length of the q-th points can be calculated as

$$\tilde{d}_{q,q+1} \approx \sqrt{1 + \left(\left|y_q'\right| - 1\right)^2} \tag{5}$$

For the nonconvex and convex PFs, the weight vectors have different origins. In order to unitize different MOPs, $\tilde{d}_{q,q+1}$ should be normalized once again. For the nonconvex PF, the normalized length of the q-th weight can be shown as

$$\tilde{d}_{q,q+1}^{non-c} = \frac{1}{\tilde{d}_{q,q+1}} + \tilde{d}_{q,q+1} \tag{6}$$

For the convex PF, the normalized length of the q-th weight can be shown as

$$\tilde{d}_{q,q+1}^{c} = \frac{1}{\tilde{d}_{q,q+1}^{non-c}} \tag{7}$$

For the nonconvex and convex PFs, the normalized lengths are different from each other. For nonconvex PF, more weights should be selected on both sides, but in the middle for convex PF. The probability of choosing the q-th weight can be calculated as

$$P_q = \frac{\tilde{d}_{q,q+1}}{\sum_{q=1}^{Q} \tilde{d}_{q,q+1}} \tag{8}$$

According to P_q, λ_s^t with size of N can be selected from λ^t. The procedure can be shown as in Algorithm 1.

After getting λ_s^t, every one individual in the combined population nearest to the points corresponding to the weight vector can be chosen to the new population. For every weight in λ_s^t, only one individual can be chosen.

The purpose of this paper is the same as that of [24], which is to improve the diversity of solutions of MOP. The three modifications in [24] are dynamic nondomination strategy, scope-constrained strategy, and front uniformly distributed strategy. These three modifications can improve the diversity of solutions and decrease the computational complexity. The front uniformly distributed strategy finds the expected positions which are uniformly distributed in the current front. The first step is to calculate the entire length of the current front, which is the total distance of every two nearest individuals after sorting the population. Then the expected positions can be obtained by dividing the total distance by the expected number.

Input: The initial weight vector λ^t, and P_q for each individual
Output: The selected weight vector λ_s^t
1: **for** $q = 1$ **to** Q **do**
2: $P_{temp} = P_{temp} + P_q$
3: **if** $P_{temp} \geq 1/N$ **then**
4: Choose the q-th weight to λ_s^t, and $P_{temp} = P_{temp} - 1/N$
5: **end if**
6: **end for**

Algorithm 1: Weight vector selection procedure.

In this paper, IGA-DWV calculates one group of excellent weight vector uniformly distributed in the PF instead of the uniformly distributed points. The first step of IGA-DWV is to calculate the slope in the PF associated with every weight vector. The normalized length of the q-th points and the probability of choosing the q-th weight can be calculated. The process of IGA-DWV is different from that in [24]. The main contribution in [24] is to uniformly divide the total distance, which is hard to deal with the uneven PF in complex MOP, such as the pattern synthesis of large scale array. IGA-DWV use the probability calculated by the slope of the PF to select the weight vector, and IGA-DWV can obtain a set of solutions with good convergence and diversity for the pattern synthesis of a linear array in the following examples.

Considering the complexity of one iteration of IGA-DWV, the basic operations and their worst-case complexities can be divided into three classes. The computational complexity of nondominated sorting is $O(M(2N)^2)$, which is the same as NSGA-II. To determine the Q expected weight vectors, one needs $O(K \times N)$ basic operations. The last operation of selecting individuals has the same computational complexity with the crowded-comparison operator in NSGA-II, which is $O(2N \log(2N))$. The overall complexity of the proposed algorithm is $O(MN^2)$, which is governed by the operator of nondominated sorting.

The fitness function evaluations are the main part of time costing for the methods used in array pattern synthesis. For every iteration, the function value of each individual should be calculated. This operator takes a lot of time, especially for the large array. Hence, the time costing is closely related to the population size and the number of iterations.

3. Pattern Synthesis of Linear Array

3.1. Low-SLL Pattern Synthesis of Linear Thinned Array. Thinned array means turning off some elements from the uniformly spaced array to create a desired radiation pattern [14]. The main purpose of thinning is to decrease the cost, weight, and power consumption [25]. Without loss of generality, the positions for the elements are fixed, and the on or off mode of each element can be controlled by the switch after the feed network. For the simulation, the weight coefficient of 0 or 1 can be used instead of the switch.

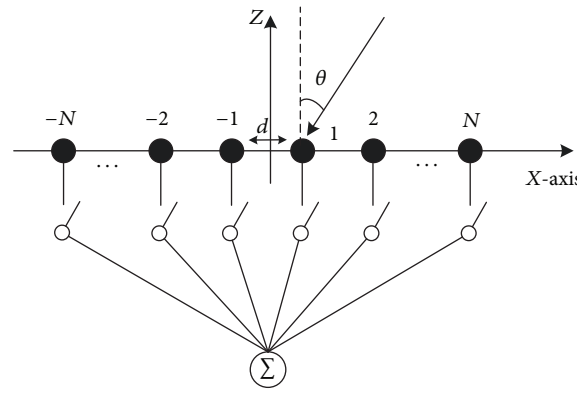

Figure 3: Geometry of the $2N$-element linear thinned array placed along the X-axis.

There are many evolutionary methods for the pattern synthesis of a thinned array. These apply discrete coding to calculate this discrete problem. In [10], the authors proposed a rounding and interval mapping strategy in RNPSO, which has a superior performance compared with GA and iterative FFT [25]. In the present work, the results of the proposed IGA-DWV with a binary-coded method are compared with those of RNPSO for a linear thinned array.

Consider a linear array of $2N$ isotropic elements with a spacing of $d = \lambda/2$, as shown in Figure 3. The position of each sensor along the X-axis is symmetric about the origin. Each element has an independent switch and the same amplitude. The array pattern can be expressed as [10]

$$\begin{aligned} F(\theta) &= \sum_{n=-N}^{-1} w_n e^{jdk(n+0.5)\sin(\theta)} + \sum_{n=1}^{N} w_n e^{jdk(n-0.5)\sin(\theta)} \\ &= 2\sum_{n=1}^{N} w_n \cos\left(dk\left(n-0.5\right)\sin\left(\theta\right)\right) \end{aligned} \tag{9}$$

where θ (measured from Z-axis) is the pitch angle of radiation for the transmit array. Here, $k = 2\pi/\lambda$ is the wavenumber, λ is the wavelength, and w_n is the weight coefficient of the nth sensor. The decision vector is the switch for each sensor. In this paper, $w_n = 0$ or 1 is used in the array pattern synthesis;

Input: $\mathbf{x}_{t+1}^{m}$ (the individual after crossover and mutation)
Output: $\mathbf{x}_{t+1}^{\text{next}}$ (the fill factor is η)
1: Find the position of the elements turned on: **Position** = find($\mathbf{x}_{t+1}^{m} == 1$)
2: Calculate the number of 1 in $\mathbf{x}_{t+1}^{m}$: Num_{all} = length(**Position**)
3: Calculate the number of positions in which the value should be reversed: $Num_e = Num_{all} - N * \eta$
4: Generate a random position distribution according to Num_{all}: $\mathbf{P}_r$ = randperm(Num_{all})
5: **if** $Num_e > 0$ **then**
6: Set the excess number as 0: $\mathbf{x}_{t+1}^{\text{next}} = (\mathbf{x}_{t+1}^{m}(\mathbf{Position}(\mathbf{P}_r(1 : Num_e))) = 0)$
7: **else if** $Num_e < 0$ **then**
8: Set the excess number as 1: $\mathbf{x}_{t+1}^{\text{next}} = (\mathbf{x}_{t+1}^{m}(\mathbf{Position}(\mathbf{P}_r(1 : -1 * Num_e))) = 1)$
9: **end if**

Algorithm 2: Constrain the fill fact procedure.

$w_n = 1$ means that the array element is turned on, and vice versa.

For the thinned array, the objectives of optimization are set as the minimum SLL and the maximum directivity by adjusting the switch of the elements. The fitness function can be defined as

$$\begin{aligned} &\min \quad \text{fitness}_1 \\ &\quad = 20\log_{10}\left(\max_{\theta_{SL}\in[-90^\circ,-\theta_b)\cup(\theta_b,90^\circ]}\left|\frac{F(\theta_{SL})}{\max F(\theta)}\right|\right) \qquad (10) \\ &\min \quad \text{fitness}_2 = -10\log_{10}\left(D(\theta_S)\right) \end{aligned}$$

For the directivity, the minus coefficient can make it a minimization problem. In the fitness function, θ_{SL} is the spanned angle within the side lobe band, except for the range of the main lobe, and $[-\theta_b, \theta_b]$ is the range of the main lobe, which is fixed to a given range according to the design specifications. The first objective is then evaluated, excluding the main beam, and the SLL is measured in decibels. In the second fitness function, $D(\theta_S)$ is the directivity in the signal angle θ_S.

Array directivity is defined as the radiant intensity in a specific direction divided by the isotropic radiant intensity. The isotropic radiant intensity is the total radiated power of the array divided by 4π. In terms of the arrays weights $\mathbf{w}$ and steering vectors $\mathbf{s}$, the directivity can be written as

$$D(\theta,\phi) = 4\pi\frac{\left|F(\theta,\phi)\,\mathbf{w}^H\mathbf{s}\right|^2}{P_{total}} \qquad (11)$$

where P_{total} is the total radiated power from the array. In a discrete implementation, the total radiated power can be computed by the summing intensity values over a uniform grid of angles that covers the full sphere surrounding the array:

$$P_{total} = \frac{2\pi^2}{AB}\sum_{a=0}^{A-1}\sum_{b=0}^{B-1}\left|F(\theta_a,\phi_b)\,\mathbf{w}^H\mathbf{s}(\theta_a,\phi_b)\right|^2\sin\theta_a \qquad (12)$$

where A is the number of elevation grid points, and B is the number of azimuth grid points. For different azimuth angles ϕ and the same θ, the patterns are the same in linear arrays, which can be counteracted in the calculation of P_{total}. For a thinned array, the fill factor is defined as η = (the number of turned-on elements/the number of total elements in the array). After crossover and mutation, the number of 1 in the new individual $\mathbf{x}_{t+1}^{m}$ should be constrained. In the thinned array, this is set by the fixed η. The excess number is eliminated at random. The algorithm of the constrained fill factor can be outlined as Algorithm 2.

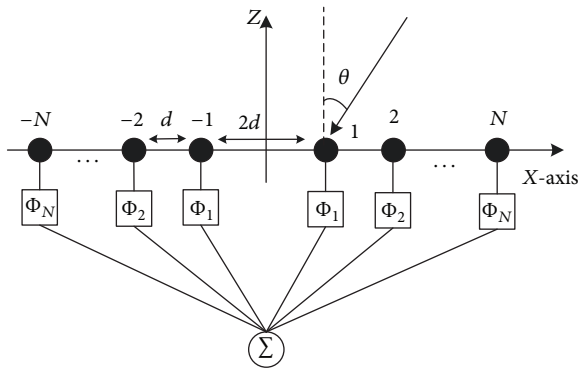

Figure 4: Geometry of the $2N$-element linear array using digital phase shifter.

3.2. Pattern Synthesis of Digital Phased Arrays. Phased array can achieve beam scanning and interference suppression by adjusting the phase without changing the excitation amplitude. With the rapid development of digital phase shifters, phased arrays have been widely used in engineering applications. For the C-bit digital shifter, the number of available phase shifting angles is 2^C. After obtaining the optimum solution, the C-bit binary values can be mapped by (3). Then, the phase values can be calculated by using the C-bit binary values and the angle interval.

In this paper, a linear array of $2N$ isotropic elements is symmetric about the origin along the X-axis. The spacing on each side is $d = \lambda/2$, and the spacing between two elements is $2d$. Each element has the same amplitude and a 4-bit digital phase shifter. Figure 4 shows the array structure.

The array pattern can be expressed as [15]

$$\begin{aligned} F(\theta) &= \sum_{n=-N}^{-1} w_n e^{j(dkn\sin(\theta)+\phi_n)} + \sum_{n=1}^{N} w_n e^{j(dkn\sin(\theta)+\phi_n)} \\ &= \sum_{n=-N}^{-1} w_n e^{j\phi_n} e^{jdkn\sin(\theta)} + \sum_{n=1}^{N} w_n e^{j\phi_n} e^{jdkn\sin(\theta)} \\ &= 2\sum_{n=1}^{N} w_n' \cos(dkn\sin(\theta)) \end{aligned} \tag{13}$$

where $w_n' = w_n e^{j\phi_n}$ is the new weight coefficient with a phase shift. For the digital phase array, the objectives of optimization are set as the minimum SLL and the minimum nulls at the specified positions by adjusting the 4-bit weight of the elements. The fitness function can be defined as

$$\begin{aligned} &\min \quad \text{fitness}_1 \\ &\quad = 20\log_{10}\left(\max_{\theta_{SL}\in[-90^o,-\theta_b)\cup(\theta_b,90^o]}\left|\frac{F(\theta_{SL})}{\max F(\theta)}\right|\right) \\ &\min \quad \text{fitness}_2 = 20\log_{10}\left(\max_{\theta_{null}\in[\theta_{J1},\theta_{J2}]}\left|\frac{F(\theta_{null})}{\max F(\theta)}\right|\right) \end{aligned} \tag{14}$$

The first fitness function is the same as that in the pattern synthesis of a linear thinned array. In the second fitness function, θ_{null} is the direction to the nulls belonging to $[\theta_{J1}, \theta_{J2}]$. The two fitness functions are measured in decibels. In this paper, two cases are implemented for the digital phased array. In the first case, θ_{null} is the spanned angle, where θ_{J1} is set to 30°, and θ_{J2} is set to 31°. Therefore, the second fitness function above is applicable. However, in the second case, θ_{null} is set as several specified angles instead of a region. The second fitness function can be rewritten as

$$\min \quad \text{fitness}_2 = 20\log_{10}\left(\frac{\sum_{\theta_{null}\in\Upsilon}|F(\theta_{null})|}{|\max F(\theta)|}\right) \tag{15}$$

where Υ is the set of all the independent null positions. The positions of null are generated at 30° and 31° in the second simulation of the digital phase array.

For the thinned array and the phased array, the switch and the phase of each element constitute the decision vector of the MOP, respectively. These variable parameters can adjust the relative relation of received or emitted electromagnetic waves. The different gains can be obtained for different steering directions in the space. The above fitness functions are to set the expected gain in the specified steering directions. After the optimal variable parameters are obtained by the proposed IGA-DWV, the expected array pattern can be formed with a high performance. As thus, the useful signals can be received or emitted in the expected directions, and the useless signals can also be effectively suppressed.

The variable parameters in the array pattern synthesis can be optimized by using the proposed multiobjective optimization method of IGA-DWV. The objective functions are obtained by the array factor, which are supposed to be uniformly distributed on the PF. By using IGA-DWV, one group of excellent weight vectors uniformly distributed on the PF can ensure the diversity of the final solutions, and the weight vectors are calculated according to the slope of the PF. With the help of the efficient convergence of IGA-DWV, the smaller values of the objective functions can be optimized, which means that the array pattern has the lower SLL, lower nulls, and higher directivity.

To reduce the computation complexity, one linear transformation similar to a discrete cosine transform (DCT) is used for the pattern synthesis of a linear thinned array and a digital phased array. By discretizing the two array pattern functions with A points, the above equations can be written in a linear transformation as

$$\begin{bmatrix} F(\theta_1) \\ \vdots \\ F(\theta_A) \end{bmatrix} = 2\begin{bmatrix} \psi(\theta_1,1) & \cdots & \psi(\theta_1,N) \\ \vdots & \ddots & \vdots \\ \psi(\theta_A,1) & \cdots & \psi(\theta_A,N) \end{bmatrix} \cdot \begin{bmatrix} w_1 \\ \vdots \\ w_N \end{bmatrix} \tag{16}$$

where $\psi(\theta_i, n) = \cos(dk(n - 0.5)\sin(\theta))$ for a linear thinned array, and $\psi(\theta_i, n) = \cos(dkn\sin(\theta_i))$ for a digital phased array. Then, the array pattern can be expressed by using a precomputed DCT matrix that allows high-speed computation of the fitness function. For the linear thinned array, the weights can be set as $w_n = 0$ for the closed element and as $w_n = 1$ for the open element. For the digital phased array, the phase shifter of each element is written as the new weight w_n'. Hence, the computation of the array pattern for each evolutionary process has the same DCT, adjusting the weights simply.

4. Numerical Examples and Results Analysis

4.1. Low SLL and Maximum Directivity of Linear Thinned Array. For the 200-element ($N = 100$) linear thinned array, the objectives are the SLL and directivity by synthesizing the on/off mode of the elements. The crossover probability is set as 0.9, and the mutation probability is 0.1. The maximum number of generations *MaxGen* is set to 2000, and the population size is set to 400. Fifty independent runs are carried out. The fill factor η is set to 77% and 78%, respectively, for the thinned array. The half power (3 dB) beam width (HPBW) is set to 0.5°, and the angular resolution is 0.1°. All the algorithms are compiled using the same compiler (MATLAB 2012a) in a PC with Intel Core i7-7500 U, at 2.70 GHz with 8 Gb RAM running Windows 10.

Table 1 shows the final results obtained by using IGA-DWV and RNPSO, which contain the best SLL, directivity, and mean time costing for $\eta = 0.77$ and $\eta = 0.78$. The RNPSO results are slightly different from the original ones because of the variance in computational accuracy. In the first case of $\eta = 0.77$, the pattern results with the best SLL and directivity obtained by IGA-DWV and RNPSO are presented in Figure 5. The best SLL obtained by IGA-DWV is -23.26 dB, which is 0.21 dB lower compared with the RNPSO result. The partial enlarged drawing clearly shows the superiority of the SLL, which is the highest level within the side lobe band except the range of the main lobe. The pattern obtained by the proposed IGA-DWV has a better performance than

Table 1: Comparison of results for the linear thinned array.

Design parameter	IGA-DWV	RNPSO	IGA-DWV	RNPSO
η	0.77	0.77	0.78	0.78
SLL (dB)	-23.26	-23.05	-23.05	-22.94
Directivity (dB)	30.44	28.82	30.72	29.33
Mean time costing (m)	15.28	14.96	15.36	15.03

Table 2: Simulation results for the switched-off elements.

Result	Number of switched-off elements
IGA-DWV with $\eta = 0.77$	±54; ±55; ±60; ±64; ±65; ±68; ±69; ±72; ±75; ±76; ±77; ±78; ±81; ±83; ±84; ±85; ±87; ±89; ±90; ±91; ±92; ±94; ±96
IGA-DWV with $\eta = 0.77$	±51; ±52; ±54; ±61; ±64; ±65; ±70; ±71; ±72; ±76; ±77;±79; ±80; ±82; ±84; ±85; ±87; ±88; ±89; ±91; ±93; ±97; ±99
IGA-DWV with $\eta = 0.78$	±52; ±54; ±61; ±64; ±65; ±69; ±71; ±72; ±74; ±78; ±79; ±80; ±82; ±83; ±85; ±86; ±88; ±89; ±90; ±91; ±94; ±95
IGA-DWV with $\eta = 0.78$	±53; ±57; ±60; ±64; ±65; ±69; ±70; ±71; ±73; ±75; ±79; ±81; ±82; ±83; ±84; ±86; ±87; ±89; ±92; ±94; ±95; ±96

RNPSO. Moreover, the directivity of the pattern obtained by IGA-DWV is 30.44 dB, which is better than that by RNPSO. This result is contributed to the lower SLL in the θ from 70° to 90°. Figure 6 shows the distribution of the final solution obtained by IGA-DWV. There is good diversity between the two contradictory fitness functions. One individual in the final solution is used to achieve the array pattern shown in Figure 5. This individual and the result in [10] are presented in the top two rows in Table 2, which show the positions of the switched-off elements. The robustness of the proposed method is verified in 50 independent trials. Figure 7 shows the contrasting distribution of the best SLL obtained in 50 trials by using IGA-DWV and RNPSO with $\eta = 0.77$. The IGA-DWV results are clearly more robust than those achieved with the RNPSO, and the SLL is lower and well concentrated in the interval of [-23.3, -22.7]. On the contrary, the RNPSO results are very diffuse, and not all of them show a good performance.

In the second case of $\eta = 0.78$, the best SLL obtained by IGA-DWV is -23.05 dB, and the directivity is 30.72 dB; these are better than the results reported in [10]. Figure 9 shows the distribution of the final solution obtained by IGA-DWV with $\eta = 0.78$. Figure 8 presents the pattern results with the best SLL and directivity, which is one individual in the final solution. This finding and the result obtained in [10] for the same conditions are shown in the two bottom rows of Table 2. Figure 10 shows the robustness of the proposed method. The results of the 50 trials converge to a small interval of [-23.1, -22.7]. The results of mean time costing for IGA-DWV and RNPSO are listed in Table 1. The results are similar, which are consistent with the above theoretical analysis. The most time is consumed in the function evaluation, and there are the same population size and number of iterations.

4.2. Low SLL with Null Depth for a 4-Bit Digital Phased Array. In this experimental study, a 100-element linear array with a 4-bit digital phase shifter is considered. In this paper, two cases are implemented. The first case is SLL with nulls at

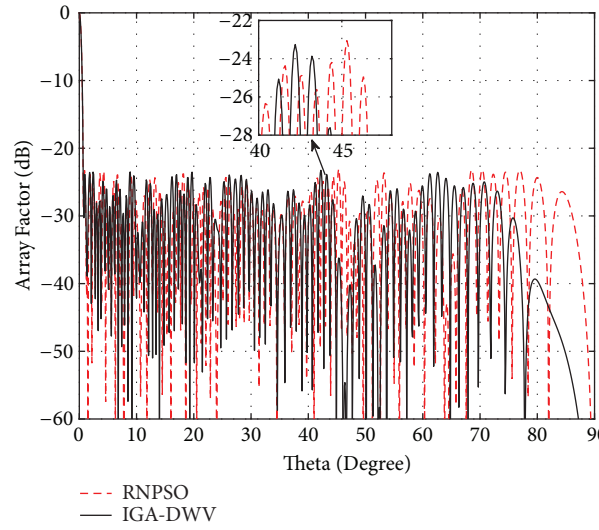

Figure 5: Pattern results for the linear thinned array obtained by using IGA-DWV and RNPSO with $\eta = 0.77$.

30° to 31°. The second case is SLL with nulls at 30° and 31°. Consistent with [10, 15], the phase shift range of the 4-bit phase shifter applied in the experiment is [0°, 114.5°], and the step value is 7.63°. The HPBW is constrained to 0.8° with a dynamic range of 0.1°. A real-coded GA is used to optimize the problem, and the distribution indexes [9] for the crossover and mutation operators are $\eta_c = 20$ and $\eta_m = 20$, respectively. The maximum number of generations and the population size are the same as those described in the above section. The result presented herein is the best one among 50 independent runs.

Figure 11 shows a comparison of the results achieved by using IGA-DWV and QPSO [15]. The time costing is also similar for IGA-DWV and QPSO. Since the number of elements of this experiment is the half of the thinned array, the less time is spent for one function evaluation. The total

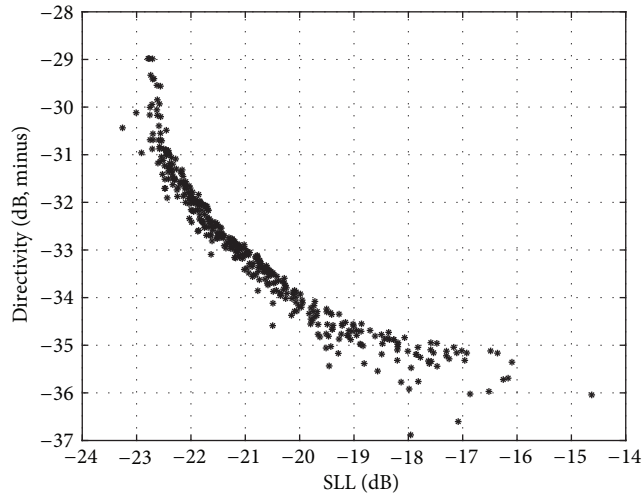

Figure 6: Distribution of the final solution obtained by IGA-DWV with $\eta = 0.77$.

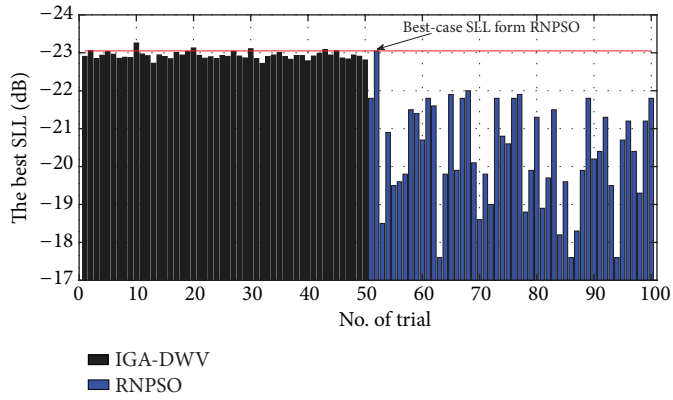

Figure 7: Contrasting distribution of the best SLL obtained in 50 trials by using IGA-DWV and RNPSO with $\eta = 0.77$.

Table 3: The digital phase sequences and results for different solutions.

Result	Digital Phase Sequence	Best SLL (dB)	Null depth (dB)	Mean time costing (m)
Nulls at 30° to 31° obtained by IGA-DWV	787778789877988776896668 6B761F8F0FF0F00E000BEF9858	-18.16	-61.72	8.76
Nulls at 30° to 31° obtained by QPSO	8D6C8EF77049635D76668DDCAC410587A4847784DA79838A6C	-12.03	-37.41	8.62
Nulls at 30° and 31° obtained by IGA-DWV	887788778778888695786974A4C61F0D0F800FFFF00C006999	-18.55	-65.66 and -71.51	8.51
Nulls at 30° and 31° obtained by RNPSO	9588AABA98AA4A68CEB53129A9D99BCEDEB1904ED3771D669A	-14.78	-60.71 and 60.93	8.56

time costing is much shorter than that for the thinned array synthesis. The SLL obtained by IGA-DWV is clearly lower than that by QPSO, and there are more effective null depth and width at 30° to 31°. Table 3 presents the digital phase sequences for the patterns shown in Figure 11. Figure 12 shows that the SLL obtained by IGA-DWV is -18.16 dB, which is more than 6 dB lower than that obtained by QPSO. Moreover, the null obtained by IGA-DWV is -61.72 dB and wide enough for the requested interval. The null obtained by QPSO is clearly too narrow to meet the optimization conditions. The null constrained in the position of 31° appears at 30.5°; thus, the null position obtained by QPSO is not accurate. Although

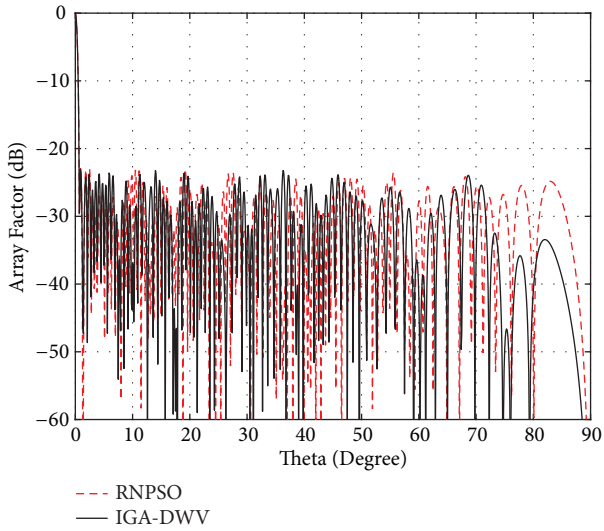

Figure 8: Pattern results for the linear thinned array obtained by using IGA-DWV and RNPSO with $\eta = 0.78$.

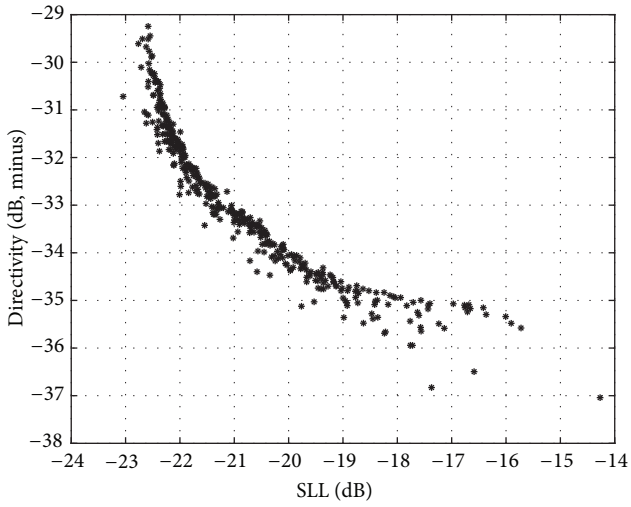

Figure 9: Distribution of the final solution obtained by IGA-DWV with $\eta = 0.78$.

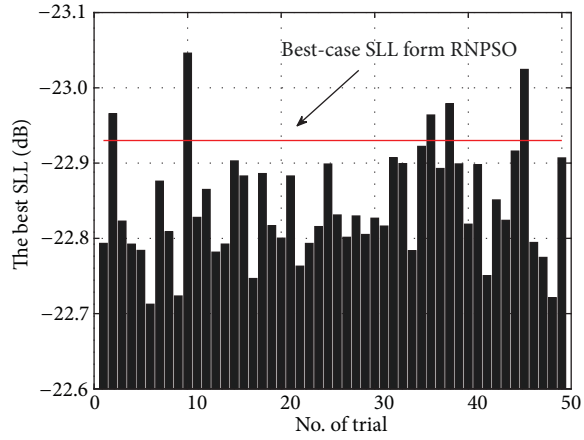

Figure 10: Contrasting distribution of the best SLL obtained in 50 trials by using IGA-DWV with $\eta = 0.78$.

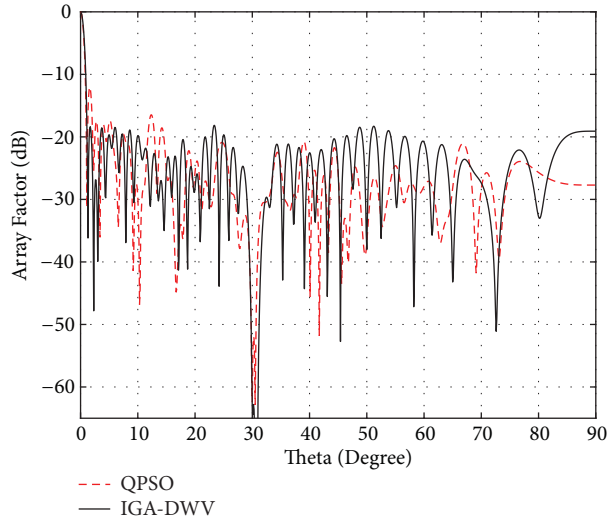

Figure 11: Pattern results for a 4-bit linear array obtained by using IGA-DWV and QPSO in the first case.

the main beam width obtained by IGA-DWV is broader than that obtained by QPSO, the HPBW is within the appointed range. Figure 13 shows the distribution of the final solution obtained by IGA-DWV for the first case. Because the weight coefficients are 4-bit binary values, the fitness function values are discrete rather than continuous.

One more individual is extracted from Figure 13 to show the only minimum SLL. Figure 14 shows the comparison of patterns. The SLL obtained by IGA-DWV is -18.68 dB, which is better than the SLL of -17.47 dB obtained by QPSO. For QPSO, the fitness function should be changed, and the evolutionary process needs to be repeated, whereas, for the multiobjective evolutionary method, the different individuals only need to be extracted to satisfy various conditions.

Table 3 shows the digital phase sequence and the best fitness function values obtained by IGA-DWV and RNPSO in the second case of minimum SLL and nulls at 30° and 31°. Figures 15 and 16 clearly indicate that IGA-DWV has a higher performance, with SLL of -18.55 dB, which is lower than the SLL of -14.78 dB obtained by RNPSO with the same HPBW. The nulls obtained by the two methods are all lower than -60 dB, with IGA-DWV achieving slightly better results. Figure 17 shows the distribution of the final solution obtained by IGA-DWV in the second case.

5. Conclusion

The objective of the proposed IGA-DWV is to obtain a set of solutions with good convergence and diversity for

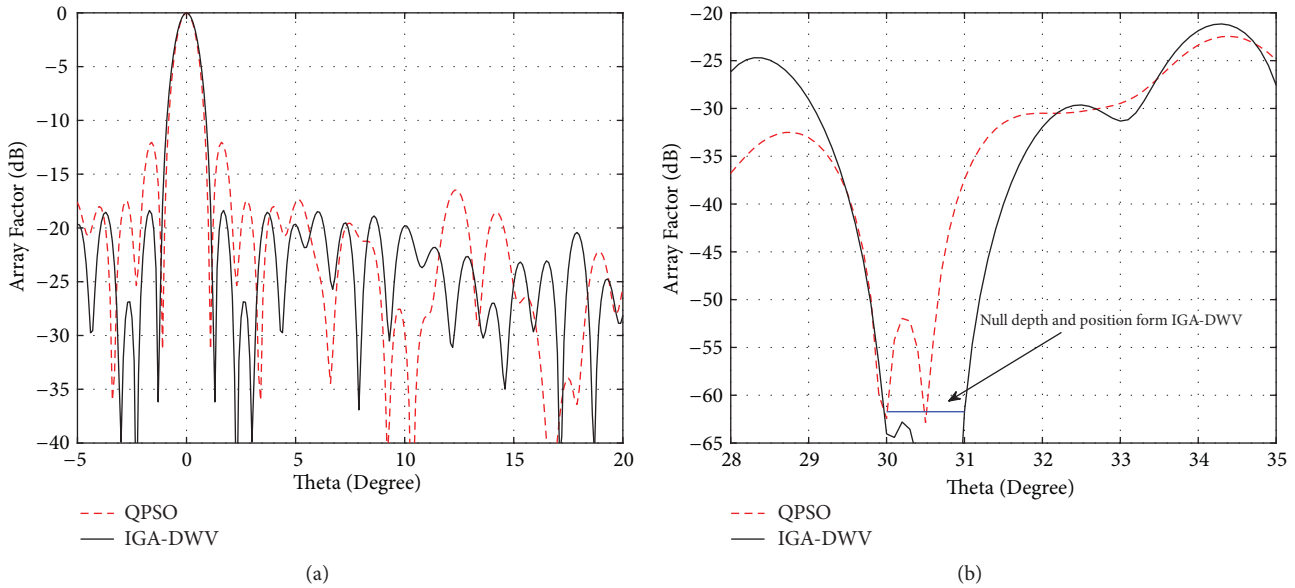

Figure 12: Comparison of patterns for the minimum SLL and nulls at 30° to 31°.

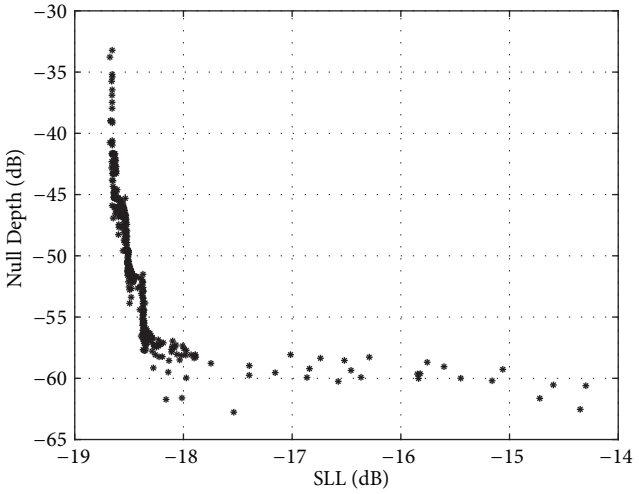

Figure 13: Distribution of the final solution obtained by IGA-DWV in the first case.

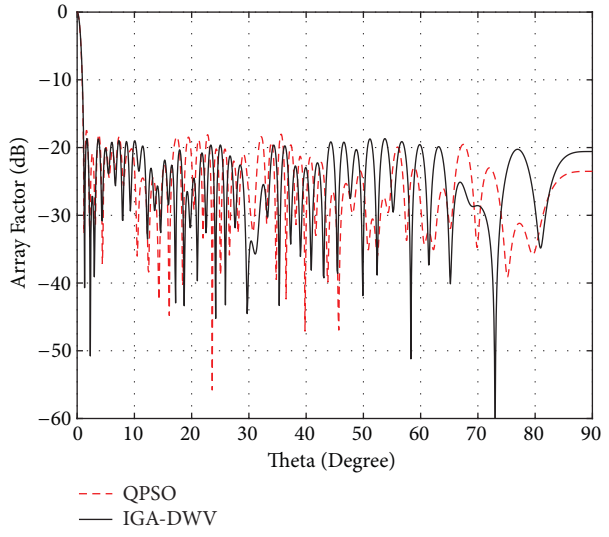

Figure 14: Pattern results for a 4-bit linear array obtained by using IGA-DWV and QPSO just for minimum SLL.

the pattern synthesis of a linear array. A multiobjective evolutionary algorithm, instead of a single objective obtained by aggregating functions, is applied to the linear array pattern synthesis. Contributing to uniformly selecting individuals based on the dynamic weight vector, the final solution can effectively deal with the conflict between the multiobjective functions. For a 200-element linear thinned array with η = 0.77, IGA-DWV obtained an SLL of -23.26 dB, which is 0.21 dB lower than that obtained by RNPSO. The final results for the SLL are well concentrated in the interval of [-23.3, -22.7]. For η = 0.78, a lower SLL and higher directivity are obtained by IGA-DWV. For a 4-bit digital phased array, IGA-DWV obtained a lower SLL of -18.16 dB and nulls of -61.72 dB at 30° to 31°, which indicate a considerable improvement in performance compared with QPSO. In the same calculation, one individual with only low SLL can be extracted, and its SLL is lower than the special calculation for lower SLL. In the case of nulls at 30° and 31°, IGA-DWV clearly achieved a higher performance, with SLL of -18.55 dB, which is lower than that obtained by RNPSO. Thus, IGA-DWV has a high performance in the pattern synthesis of multiobjective optimization.

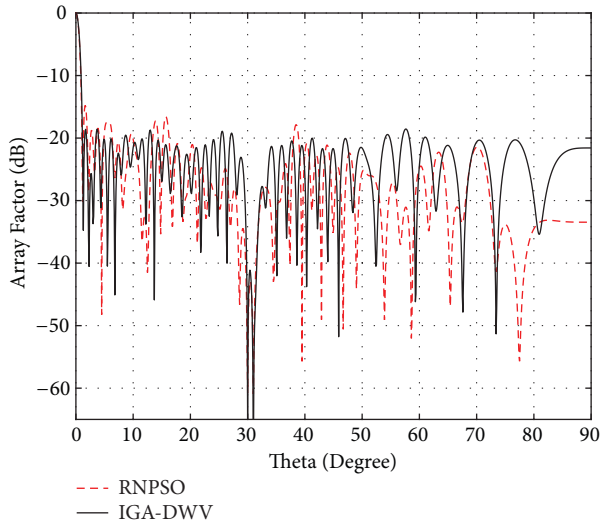

Figure 15: Pattern results for a 4-bit linear array obtained by using IGA-DWV and RNPSO in the second case.

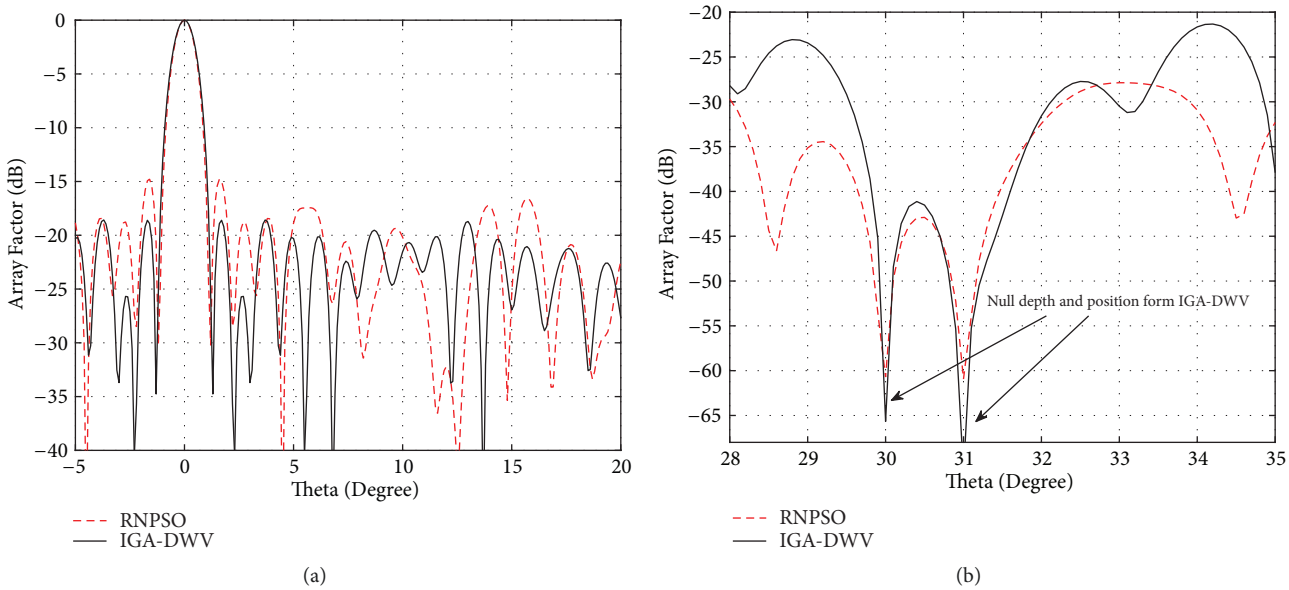

Figure 16: Comparison of patterns for the minimum SLL and nulls at 30° and 31°.

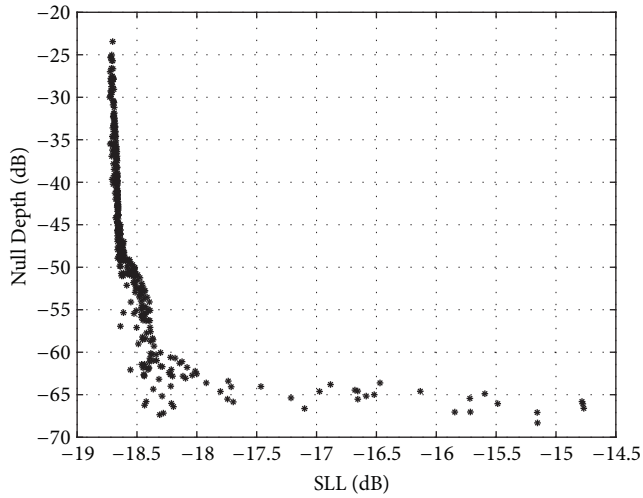

Figure 17: Distribution of the final solution obtained by using IGA-DWV in the second case.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The author declares that there are no competing interests.

References

[1] T. Bauernfeind, P. Baumgartner, O. Bíró, C. A. Magele, K. Preis, and R. Torchio, "PEEC-based multi-objective synthesis of non-uniformly spaced linear antenna arrays," *IEEE Transactions on Magnetics*, vol. 53, no. 6, 2017.

[2] K. Guney and A. Durmus, "Pattern nulling of linear antenna arrays using backtracking search optimization algorithm," *International Journal of Antennas and Propagation*, vol. 2015, Article ID 713080, 10 pages, 2015.

[3] R. Bhattacharya, T. K. Bhattacharyya, and R. Garg, "Position mutated hierarchical particle swarm optimization and its application in synthesis of unequally spaced antenna arrays," *IEEE Transactions on Antennas and Propagation*, vol. 60, no. 7, pp. 3174–3181, 2012.

[4] M. A. Panduro, D. H. Covarrubias, C. A. Brizuela, and F. R. Marante, "A multi-objective approach in the linear antenna array design," *AEU-International Journal of Electronics and Communications*, vol. 59, no. 4, pp. 205–212, 2005.

[5] B. Babayigit, "Synthesis of concentric circular antenna arrays using dragonfly algorithm," *International Journal of Electronics*, vol. 105, no. 5, pp. 784–793, 2018.

[6] M. A. Panduro, C. A. Brizuela, and D. H. Covarrubias, "Design of electronically steerable linear arrays with evolutionary algorithms," *Applied Soft Computing*, vol. 8, no. 1, pp. 46–54, 2008.

[7] M. A. Panduro and C. A. Brizuela, "Evolutionary multi-objective design of non-uniform circular phased arrays," *COMPEL: The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, vol. 27, no. 2, pp. 551–566, 2008.

[8] L. Zhang, Y. Jiao, B. Chen, and Z. Weng, "Multiobjective optimization design of concentric ring arrays with 3D beam scanning using differential evolution algorithm," *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, vol. 26, no. 6, pp. 602–619, 2013.

[9] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[10] D. Liu, Q.-Y. Feng, and W.-B. Wang, "Discrete optimization problems of linear array synthesis by using real number particle swarm optimization," *Progress in Electromagnetics Research*, vol. 133, pp. 407–424, 2013.

[11] A. Razavi and K. Forooraghi, "Thinned arrays using pattern search algorithms," *Progress in Electromagnetics Research*, vol. 78, pp. 61–71, 2008.

[12] K. V. Deligkaris, Z. D. Zaharis, D. G. Kampitaki, S. K. Goudos, I. T. Rekanos, and M. N. Spasos, "Thinned planar array design using boolean PSO with velocity mutation," *IEEE Transactions on Magnetics*, vol. 45, no. 3, pp. 1490–1493, 2009.

[13] P. Rocca, "Large array thinning by means of deterministic binary sequences," *IEEE Antennas and Wireless Propagation Letters*, vol. 10, pp. 334–337, 2011.

[14] W.-B. Wang, Q.-Y. Feng, and D. Liu, "Synthesis of thinned linear and planar antenna arrays using binary PSO algorithm," *Progress in Electromagnetics Research*, vol. 127, pp. 371–387, 2012.

[15] T. H. Ismail and Z. M. Hamici, "Array pattern synthesis using digital phase control by quantized particle swarm optimization," *IEEE Transactions on Antennas and Propagation*, vol. 58, no. 6, pp. 2142–2145, 2010.

[16] W. P. Keizer, "Low sidelobe phased array pattern synthesis with compensation for errors due to quantized tapering," *IEEE Transactions on Antennas and Propagation*, vol. 59, no. 12, pp. 4520–4524, 2011.

[17] J. Song, J. Wang, K. Peng, C. Pan, and Z. Yang, "Quantization error reduction for the phased array with 2-bit phase shifter," *Wireless Personal Communications*, vol. 52, no. 1, pp. 29–41, 2010.

[18] L. Jiang, J. Cui, L. Shi, and X. Li, "Pareto optimal design of multilayer microwave absorbers for wide-angle incidence using genetic algorithms," *IET Microwaves, Antennas & Propagation*, vol. 3, no. 4, pp. 572–579, 2009.

[19] J. Ouyang, X. Luo, J. Yang, K. Zhi Zhang, J. Zhang, and F. Yang, "Analysis synthesis of conformal conical surface linear phased array with volume surface integral equation+AEP (Active Element Pattern) INSGA-II," *IET Microwaves, Antennas & Propagation*, vol. 6, no. 11, pp. 1277–1285, 2012.

[20] S. K. Goudos, K. A. Gotsis, K. Siakavara, E. E. Vafiadis, and J. N. Sahalos, "A multi-objective approach to subarrayed linear antenna arrays design based on memetic differential evolution," *IEEE Transactions on Antennas and Propagation*, vol. 61, no. 6, pp. 3042–3052, 2013.

[21] L. Liu, J. Lu, and S. Yang, "Many-objective optimization of antenna arrays using an improved multiple-single-objective pareto sampling algorithm," *IEEE Antennas and Wireless Propagation Letters*, vol. 11, pp. 399–402, 2012.

[22] F. Tokan and F. Güne, "Pareto optimal synthesis of the linear array geometry for minimum sidelobe level and null control during beam scanning," *International Journal of RF and Microwave Computer-Aided Engineering*, vol. 20, no. 5, pp. 557–566, 2010.

[23] Y. F. Cheng, W. Shao, and S. J. Zhang, "An improved multi-objective genetic algorithm for large planar array thinning," *IEEE Transactions on Magnetics*, vol. 52, no. 3, 2016.

[24] C. Han, L. Wang, Z. Zhang, J. Xie, and Z. Xing, "Linear array pattern synthesis using an improved multiobjective genetic algorithm," *Radioengineering*, vol. 26, no. 4, pp. 1048–1059, 2017.

[25] M. Fernández-Delgado, J. A. Rodríguez-González, R. Iglesias, S. Barro, and F. J. Ares-Pena, "Fast array thinning using global optimization methods," *Journal of Electromagnetic Waves and Applications*, vol. 24, no. 16, pp. 2259–2271, 2012.

[26] W. P. Keizer, "Linear array thinning using iterative fft techniques," *IEEE Transactions on Antennas and Propagation*, vol. 56, no. 8, pp. 2757–2760, 2008.

[27] Q. Zhang and H. Li, "MOEA/D: a multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[28] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Complex Systems*, vol. 9, no. 2, pp. 115–148, 1995.