*Progress In Electromagnetics Research C, Vol. 114, 217–231, 2021*

# Improved Binary Particle Swarm Optimization and Its Application to Beamforming of Planar Antenna Arrays

**Yan Lv, Fei Cao*, Xiaowei Feng, and Huiqin Li**

**Abstract**—Beamforming can steer the mainlobe of the beam pattern towards the desired signal and set several nulls in the directions of interference signals by adjusting the excitation weights of array elements. These days, a range of meta-heuristic algorithms have been utilized for the beamforming of antenna arrays. However, most of the methods are applied to linear arrays and rarely to planar arrays. In this paper, a novel variant of binary particle swarm optimization (BPSO) is proposed at first, where the global search ability and local optimization ability are both taken into account. Then, the fitness function including the term of peak sidelobe level (PSLL) is constructed, and the improved BPSO is applied to the beamforming of uniform planar array (UPA). Finally, simulation results demonstrate that by setting the parameters reasonably, the proposed algorithm is not only able to suppress PSLL effectively, but also able to form deeper nulls than that of linearly constrained minimum variance (LCMV).

## 1. INTRODUCTION

Over a few decades, particle swarm optimization (PSO) has been increasingly used as a meta-heuristic search algorithm since it is easy to implement compared with other techniques [1–4]. As a binary version, the BPSO is also widely used by scholars in the fields of function optimization, production scheduling, image processing machine learning, etc. [5–8]. The reason for using BPSO is that it has several advantages over other techniques, such as its efficiency, better convergence, simplicity, and robustness [9, 10]. Beamforming is a spatial filtering technique, which can control the optimal excitation weights of array elements to point the mainlobe of the beam pattern towards the direction of signal of interest (SOI) [11], and at the same time set several nulls in the direction of arrival (DOA) [12] of interference signals. The directions mentioned above can be calculated by DOA estimation algorithms [13, 14]. Beamforming techniques are applied in numerous areas such as radar, sonar, modern communications, and medical imaging [15]. In recent years, more and more meta-heuristic algorithms have been utilized and gradually developed into novel techniques suitable for the beamforming of antenna arrays [16–18].

A hybrid algorithm of BPSO and convex optimization was proposed in [19] to suppress the PSLL for multibeam imaging sonar arrays, which can guarantee the complete match of the global optimization ability and local optimization ability. The work in [20] described a new beamforming method based on PSO, which considers the PSLL, beamwidth between the first nulls (FNBW), and depth of the nulls as beam controlling attributes. Reference [21] employed several meta-heuristic algorithms, such as PSO, firefly algorithm (FA), and cuckoo search (CS) to optimize the weights of LCMV. Simulation results showed that FA can obtain a better signal to interference plus noise ratio (SINR) results than the others. However, the convergence speed of FA was found to be much slower than CS and PSO. In [22], a novel invasive weed optimization (IWO) variant called adaptive dispersion invasive weed optimization

*Received 20 June 2021, Accepted 29 July 2021, Scheduled 11 August 2021*
* Corresponding author: Fei Cao (lyakhp@163.com).
The authors are with the Xi'an Research Institute of High Technology, Xi'an 710025, China.

(ADIWO) was utilized as a beamformer, which achieved lower PSLL than that of minimum variance distortionless response (MVDR).

In this paper, a novel variant of BPSO called hybrid binary particle swarm optimization (HBPSO) is proposed, where the global search ability and local optimization ability are both taken into account. Then, HBPSO is applied to the beamforming of UPA. Simulations demonstrate that by setting the parameters reasonably, the proposed algorithm can effectively suppress PSLL and form deeper nulls than that of LCMV. The remaining portions of this paper are organized as follows. In Section 2, the signal model of UPA and LCMV beamformer are briefly described. Section 3 provides a detailed introduction to BPSO, HBPSO, and the definition of the fitness function. Results of the simulations and analysis are provided in Section 4. Finally, our study is summarized in the last section.

## 2. FORMULATION

### 2.1. Signal Model

Consider a UPA [23] with $W$ rows and $N$ columns as shown in Figure 1, and the total array elements are $Q = W \times N$, where the elements uniformly distribute along $x$-axis and $y$-axis with spacing $d_x = d_y = \lambda/2$($\lambda$ is the wavelength of signal) [24, 25]. Each element is considered to be an isotropic source. $L$ narrowband far-field uncorrelated sources $s_l(k),\ l = 0, \ldots, L-1, k = 1, \ldots, K$, where $s_0(k)$ and $s_{1,\ldots,L-1}(k)$ denote the SOI and interference signals, respectively, and parameter $k$ denotes the $k$-th snapshot, arrive at the UPA from DOA $(\theta_l, \phi_l),\ l = 0, \ldots, L-1$. Here, the azimuth angle and elevation angle are defined as $\theta$ and $\phi$, respectively. The measurement $\mathbf{y}_q(k),\ q = 1, \ldots, Q$ of UPA includes additive white Gaussian noise (AWGN) $\mathbf{n}(k)$ with mean zero and variance $\sigma_n^2$. Thus, $\mathbf{y}(k)$ is modelled as:

$$\begin{aligned}\mathbf{y}(k) &= \mathbf{a}(\theta_0, \phi_0)s_0(k) + [\mathbf{a}(\theta_1, \phi_1)\mathbf{a}(\theta_2, \phi_2)\ldots\mathbf{a}(\theta_{L-1}, \phi_{L-1})]\begin{bmatrix} s_1(k) \\ \vdots \\ s_{L-1}(k) \end{bmatrix} + \mathbf{n}(k) \\ &= \mathbf{a}(\theta_0, \phi_0)s_0(k) + \mathbf{A}\mathbf{s}(k) + \mathbf{n}(k) = \mathbf{S}(k) + \mathbf{I}(k) \end{aligned} \quad (1)$$

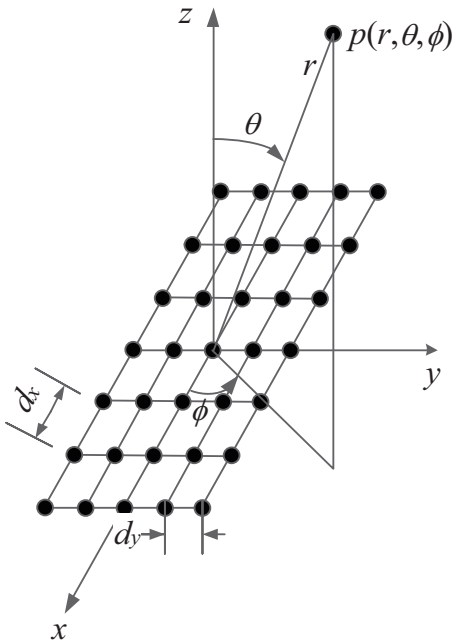


**Figure 1.** Element arrangement of UPA.

where

$$\mathbf{y}(k) = [y_1(k)\ y_2(k)\ \ldots\ y_Q(k)]^{\mathrm{T}} \tag{2}$$

$$\mathbf{a}(\theta_l, \phi_l) = \left[\mathrm{e}^{j2\pi/\lambda(X_1\mu_l+Y_1\upsilon_l)}\mathrm{e}^{j2\pi/\lambda(X_2\mu_l+Y_2\upsilon_l)}\ldots\mathrm{e}^{j2\pi/\lambda(X_Q\mu_l+Y_Q\upsilon_l)}\right]^{\mathrm{T}} \tag{3}$$

$$\mu_l = \sin(\theta_l)\cos(\phi_l),\ \ \nu_l = \sin(\theta_l)\sin(\phi_l) \tag{4}$$

$$\mathbf{A} = [\mathbf{a}(\theta_1, \phi_1)\ \mathbf{a}(\theta_2, \phi_2)\ \ldots\ \mathbf{a}(\theta_{L-1}, \phi_{L-1})] \tag{5}$$

$$\mathbf{s}(k) = [s_1(k)\ s_2(k)\ \ldots\ s_{L-1}(k)] \tag{6}$$

are, respectively, the array output, the steering vector of the $l$-th source, the steering matrix of interferences, and the interference signals. $X_q$ and $Y_q$ are the $q$-th sensor positions in the $x$-axis and $y$-axis, respectively. The vector $\mathbf{S}(k) = \mathbf{a}(\theta_0, \phi_0)s_0(k)$ represents the desired signal, while the vector $\mathbf{I}(k) = \mathbf{A}\mathbf{s}(k)+\mathbf{n}(k)$ denotes the interference plus noise signals. Then, the output of UPA can be written as:

$$Y(k) = \mathbf{w}^{\mathrm{H}}\mathbf{y}(k) = \mathbf{w}^{\mathrm{H}}\mathbf{S}(k) + \mathbf{w}^{\mathrm{H}}\mathbf{I}(k) \tag{7}$$

where $\mathbf{w} = [w_1\ w_2\ \ldots\ w_Q]^{\mathrm{T}}$ is the excitation weight vector. The power for the desired signal can be calculated as follows:

$$\sigma_S^2 = E\left[\left|\mathbf{w}^{\mathrm{H}}\mathbf{S}(k)\right|^2\right] = E\left[\left|\mathbf{w}^{\mathrm{H}}\mathbf{a}(\theta_0, \phi_0)s_0(k)\right|^2\right] = R_{ss}\mathbf{w}^{\mathrm{H}}\mathbf{a}(\theta_0, \phi_0)\mathbf{a}^{\mathrm{H}}(\theta_0, \phi_0)\mathbf{w} \tag{8}$$

where $R_{ss} = E[s_0(k)s_0^*(k)]$ is the mean power of the desired signal. Similarly, the output power for the interference plus noise signals can be written in the following form:

$$\begin{aligned}\sigma_I^2 &= E\left[\left|\mathbf{w}^{\mathrm{H}}\mathbf{I}(k)\right|^2\right] = E\left[\left|\mathbf{w}^{\mathrm{H}}\left[\mathbf{A}\mathbf{s}(k) + \mathbf{n}(k)\right]\right|^2\right] \\ &= \mathbf{w}^{\mathrm{H}}\mathbf{A}\mathbf{R}_{ii}\mathbf{A}^{\mathrm{H}}\mathbf{w} + \mathbf{w}^{\mathrm{H}}\mathbf{R}_{nn}\mathbf{w}\end{aligned} \tag{9}$$

where the matrix $\mathbf{R}_{ii} = E[\mathbf{s}(k)\ \mathbf{s}^{\mathrm{H}}(k)]$ represents the correlation of interference signals, and the matrix $\mathbf{R}_{nn} = E[\mathbf{n}(k)\mathbf{n}^{\mathrm{H}}(k)]$ is the correlation of AWGN, which results $\mathbf{R}_{nn} = \sigma_n^2\mathbf{I}$. Supposing that the interference signals are uncorrelated with each other, to simplify Eqs. (8)–(9), $R_{ss} = 10^{\mathrm{SNR}/10}$, $\mathbf{R}_{ii} = 10^{\mathrm{INR}/10}\mathbf{I}$, and $\sigma_n^2 = 1$ are set, where SNR and INR represent signal to noise ratio and interference to noise ratio, respectively. Then, the parameter $\sigma_{SINR}$ is given below:

$$\sigma_{SINR} = \frac{\sigma_S^2}{\sigma_I^2} = \frac{R_{ss}\mathbf{w}^{\mathrm{H}}\mathbf{a}(\theta_0, \phi_0)\mathbf{a}^{\mathrm{H}}(\theta_0, \phi_0)\mathbf{w}}{\mathbf{w}^{\mathrm{H}}\mathbf{A}\mathbf{R}_{ii}\mathbf{A}^{\mathrm{H}}\mathbf{w} + \sigma_n^2\mathbf{w}^{\mathrm{H}}\mathbf{w}} \tag{10}$$

Finally, $\sigma_{SINR}$ can be expressed in logarithmic units (dB):

$$\sigma_{SINR} = \alpha_{SINR}\,\mathrm{dB} \tag{11}$$

### 2.2. LCMV Beamformer

LCMV beamformer [15] is an adaptive beamforming method that minimizes the output power of $Y(k)$ meanwhile satisfying one or more linear equality constraints. Thus, the optimal excitation weight vector $\mathbf{w}$ can be obtained by solving the equation:

$$\mathbf{w} = \arg\min_{\mathbf{w}} \mathbf{w}^{\mathrm{H}}\mathbf{R}_{yy}\mathbf{w}\ \text{s.t.}\ \mathbf{C}^{\mathrm{H}}\mathbf{w} = \mathbf{f} \tag{12}$$

where $\mathbf{R}_{yy} = E[\mathbf{y}(k)\mathbf{y}^{\mathrm{H}}(k)]$ is the correlation matrix of $\mathbf{y}(k)$; $\mathbf{C}$ is a $Q \times L$ dimensional constrained matrix; and $\mathbf{f}$ is the gain vector corresponding to $\mathbf{C}$. The optimization problem can be solved by the Lagrange method of multipliers, and the solution is:

$$\mathbf{w}_{LCMV} = \mathbf{R}_{yy}^{-1}\mathbf{C}\left(\mathbf{C}^{\mathrm{H}}\mathbf{R}_{yy}^{-1}\mathbf{C}\right)^{-1}\mathbf{f} \tag{13}$$

## 3. PROPOSED ALGORITHM

### 3.1. Binary Particle Swarm Optimization

The search space in BPSO is thought of as a hypercube in which a particle can be moved to a closer or farther corner of the hypercube by flipping a different number of bits [26]. The steps of BPSO are as follows:

Step 1. Randomly generate initial matrix $\mathbf{X}_m$ and $\mathbf{V}_m$ of each particle:

$$\mathbf{X}_m = (x_{m1}, x_{m2}, \ldots, x_{mn}), \quad m = 1, 2, \ldots, M \tag{14}$$

$$\mathbf{V}_m = (v_{m1}, v_{m2}, \ldots, v_{mn}), \quad m = 1, 2, \ldots, M \tag{15}$$

where $\mathbf{X}_m$ is the position of each particle by binary values; $m$ represents the number of particles; and $x_{mn} \in \{0, 1\}$ is set. $\mathbf{V}_m$ is the velocities of each particle defined in terms of the probabilities of corresponding elements in $\mathbf{X}_m$ taking value 1. In addition, the velocities of a particle are confined within $v_{mn} \in [v_{\min}, v_{\max}]$. If $v_{mn}$ is higher than $v_{\max}$, it is set to $v_{\max}$ automatically. Similarly, if $v_{mn}$ is less than $v_{\min}$, $v_{mn}$ is set to $v_{\min}$. Generally, $v_{\max} = 6$ and $v_{\min} = -v_{\max}$ are set.

Step 2. Calculate fitness values according to the fitness function.

Step 3. $\mathbf{P}^b_{best}$ and $\mathbf{G}_{best}$ can be determined according to the following equations:

$$\mathbf{P}^b_{best} = (x_{b1}, x_{b2}, \ldots, x_{bn}) \tag{16}$$

$$\mathbf{G}_{best} = (x_{g1}, x_{g2}, \ldots, x_{gn}) \tag{17}$$

where $\mathbf{P}^b_{best}$ represents the vector indicating the best position experienced thus far by the $b$-th particle, and $\mathbf{G}_{best}$ is the best fitness value particle obtained so far in the whole swarm.

Step 4. The velocities and positions of particles can be updated using the following formulas:

$$v^{t+1}_{mn} = w \times v^t_{mn} + c_1 \times \text{rand}_1 \times \left(x^t_{bn} - x^t_{mn}\right) + c_2 \times \text{rand}_2 \times \left(x^t_{gn} - x^t_{mn}\right) \tag{18}$$

$$x^{t+1}_{mn} = \begin{cases} 1 & \text{rand}_3 < S_{trans}\left(v^{t+1}_{mn}\right) \\ 0 & otherwise \end{cases} \tag{19}$$

where $t$ and $t+1$, respectively, represent the current number of iterations and the next iteration; $w$ denotes the inertia weight parameter; and $c_1$ and $c_2$ are two nonnegative constants called cognitive and social coefficients, respectively. Generally, $w \in (0, 1)$ and $c_1, c_2 \in (0, 2)$ are defined. The parameters $\text{rand}_1$, $\text{rand}_2$, and $\text{rand}_3$ denote three random values that are uniformly distributed between 0 and 1. Besides, $S_{trans}(v^{t+1}_{mn})$ represents the Sigmoid transformation function given by:

$$S_{trans}(v^{t+1}_{mn}) = \frac{1}{1 + \mathrm{e}^{-v^{t+1}_{mn}}} \tag{20}$$

Step 5. Steps 2–4 are repeated until $t$ reaches the maximum number of iterations.

### 3.2. Procedure of HBPSO

The original BPSO algorithm converts different velocities into the probabilities that the codes of current position take 0 or 1 by the Sigmoid (S-shaped) transformation function. When the velocity is positive, it means $x_{bn} > x_{mn}$ or $x_{gn} > x_{mn}$. At this time, the probability that each position in $X_m$ takes value 1 increases. On the contrary, if the velocity is negative, it means $x_{bn} < x_{mn}$ or $x_{gn} < x_{mn}$. In this case, the probability of each position being value 0 increases. It is worth emphasizing that when the velocities all become zero, the position $x_{mn}$ will still be changed, and the probability that takes the value 0 or 1 is 0.5. The above property makes BPSO have strong global search ability, and it is not easy to fall into the local optimum. However, this will affect the local optimization capability and cause the algorithm not to converge well.

A V-shaped transformation function (see Figure 2) based BPSO algorithm is proposed in [27], which can overcome the shortcoming that the position will still be changed with a large probability when the speed becomes zero, so it can improve the local optimization ability and the convergence property of the algorithm.

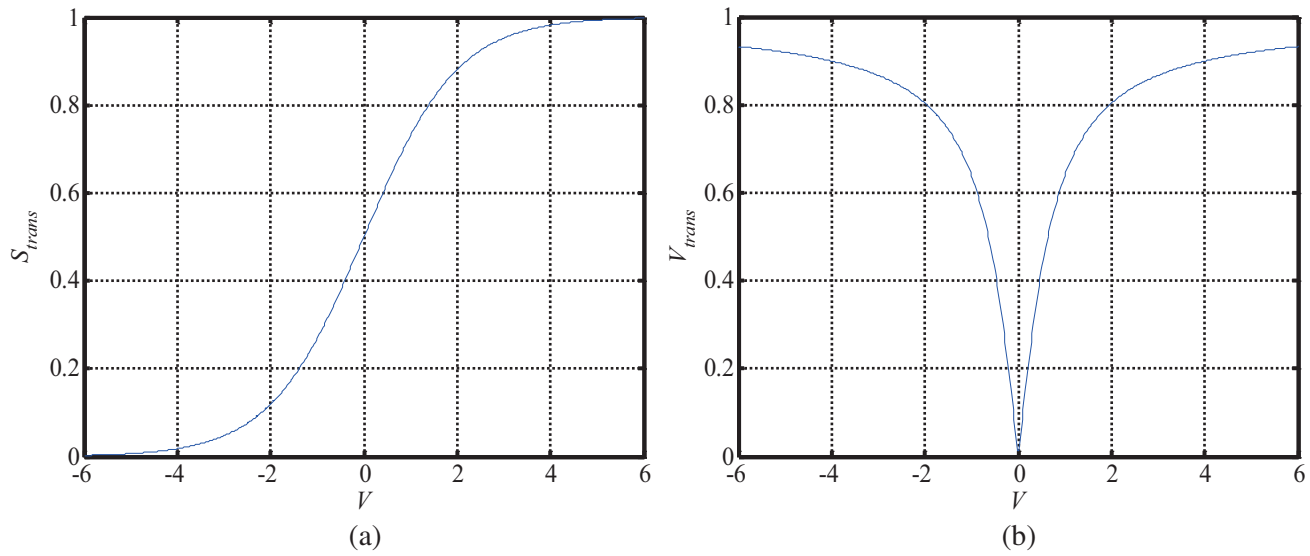


**Figure 2.** Curves of the transformation function. (a) S-shaped. (b) V-shaped.

Based on the two kinds of transformation functions, the paper proposes a hybrid optimization method called HBPSO. In the initial stage of iterations, the S-shaped function is used to improve the global search ability and avoid falling into the local optimum. In the later stage, the V-shaped function is adopted to improve the local optimization ability. Steps 1, 2, 3, and 5 of HBPSO are the same as the original BPSO, and step 4 is modified as follows:

Step 4. If the current number of iterations $t \leq \eta Iter_{\max}$, the positions are updated by using Eq. (19) and Eq. (20). When $\eta Iter_{\max} < t \leq Iter_{\max}$, Eq. (21) and Eq. (22) are used to update the positions:

$$x_{mn}^{t+1} = \begin{cases} {}^{\sim}x_{mn}^{t} & \text{rand}_3 < V_{trans}\left(v_{mn}^{t+1}\right) \\ x_{mn}^{t} & otherwise \end{cases} \tag{21}$$

$$V_{trans}\left(v_{mn}^{t+1}\right) = \left|\frac{2}{\pi}\arctan\left(\frac{\pi}{2}v_{mn}^{t+1}\right)\right| \tag{22}$$

where the parameters $\eta$ and $Iter_{\max}$ are defined as the iteration coefficient and the maximum number of iterations, respectively. ${}^{\sim}x_{mn}^{t}$ is the complement of $x_{mn}^{t}$, and it means that if $x_{mn}^{t} = 1$ or 0, then ${}^{\sim}x_{mn}^{t} = 0$ or 1.

### 3.3. Definition of Fitness Function

The fitness function $F$ in beamforming can be defined as the inverse of $\alpha_{SINR}$ [28]. The smaller $F$ becomes, the larger $\alpha_{SINR}$ is, which also means that the mainlobe is steered towards SOI, and the DOAs of interference signals are set nulls. Besides, we try to reduce the PSLL of the beam pattern in the process of optimization. In this case, $F$ can be redefined as:

$$F = \frac{1}{\alpha_{SINR}} + \xi\left|\frac{1}{P_{SLL}}\right| \tag{23}$$

where $P_{SLL}$ is the value of PSLL, i.e., PSLL $= P_{SLL}$ dB, and $\xi$ denotes the adjustment coefficient used to balance the minimization of the two terms given in Eq. (23).

## 4. SIMULATION RESULTS

### 4.1. Simulation of HBPSO

In order to show the capability of the HBPSO algorithm, 6 benchmark functions [27] are employed (see Table 1), where $n$ that equals Dim is the dimension of the function; Range indicates the boundary of

**Table 1.** Six benchmark functions used in the paper.

| Function's name | Test Function | Dim | Range | $f_{opt}$ |
|---|---|---|---|---|
| Rastrigin | $f_1(x) = \sum_{i=1}^{n} \left[x_i^2 - 10\cos(2\pi x_i) + 10\right]$ | 5 | $[-5.12, 5.12]$ | 0 |
| Rosenbrock | $f_2(x) = \sum_{i=1}^{n-1} \left[100(x_{i+1} - x_i)^2 + (x_i - 1)^2\right]$ | 5 | $[-30, 30]$ | 0 |
| Schaffer | $f_3(x) = \frac{\sin^2\sqrt{\sum_{i=1}^{n} x_i^2} - 0.5}{\left[1 + 0.001\sum_{i=1}^{n} x_i^2\right]^2} - 0.5$ | 5 | $[-100, 100]$ | $-1$ |
| Schwefel | $f_4(x) = \sum_{i=1}^{n} \lvert x_i\rvert + \prod_{i=1}^{n} \lvert x_i\rvert$ | 5 | $[-10, 10]$ | 0 |
| Ackley | $f_5(x) = -20\exp\left(-0.2\sqrt{\sum_{i=1}^{n} x_i^2/n}\right) + 20 + e - \exp\left(\sum_{i=1}^{n} \cos(2\pi x_i)/n\right)$ | 5 | $[-32, 32]$ | 0 |
| Step | $f_6(x) = \sum_{i=1}^{n} (x_i + 0.5)^2$ | 5 | $[-100, 100]$ | 0 |

search space; and $f_{opt}$ is the optimal value of the function. BPSO, VBPSO, and HBPSO are all used to find the global minimum of the functions in Table 1. The 2-D versions of these benchmark functions are illustrated in Figure 3. To represent each continuous variable in binary, a string length of 15 bits for each dimension is used, so the dimension of each particle is 75.

For all BPSO algorithms, the number of particles is set to 30; $w$ decreases linearly from 0.9 to 0.4; the maximum number of iterations is 500; $\eta = 0.4$ is set; and $v_{\max}$ is equal to 6. In Eq. (18), $rand_1 \times (x_{bn}^t - x_{mn}^t)$ and $rand_2 \times (x_{gn}^t - x_{mn}^t)$ are in the range of $[-2, 2]$. When $c_1 = 1.5$ and $c_2 = 1.5$ are set, assuming that $v_{mn}^t$ is not considered, it is possible to make the result of Eq. (18) vary within the range of maximum velocity, i.e., $v_{mn}^{t+1} \in [-6, 6]$. The fitness value of each run is recorded and averaged over 50 independent runs. To observe the convergence properties more intuitively, the convergence characteristics in finding the best fitness values of the benchmark functions are shown in Figure 4. Table 2 shows the best results and average values of 50 independent tests, where *ave* and *best* denote the average values and the best results, respectively.

**Table 2.** Comparison of BPSO, VBPSO and HBPSO on six benchmark functions over 50 independent runs, where *ave* denotes the average values, and *best* stands for the best results.

| Algorithms | | $f_1$ | $f_2$ | $f_3$ | $f_4$ | $f_5$ | $f_6$ |
|---|---|---|---|---|---|---|---|
| BPSO | *ave* | 8.4848 | 484.9910 | −0.8899 | 0.8847 | 4.7783 | 32.4144 |
| | *best* | 3.5554 | 57.9217 | −0.9589 | 0.3566 | 3.2004 | 8.0425 |
| VBPSO | *ave* | 1.7341 | 8.3002 | −0.9779 | 0.0067 | 0.2433 | 0.8622 |
| | *best* | **2.4219e-5** | 3.7581 | **−0.9903** | **0.0018** | **0.0040** | 0.5125 |
| HBPSO | *ave* | **1.4939** | **6.1877** | **−0.9811** | **0.0044** | **0.1407** | **0.6451** |
| | *best* | **2.4219e-5** | **2.2867** | **−0.9903** | **0.0018** | **0.0040** | **4.6109e-4** |

Based on the simulation results shown in Figure 4 and Table 2, the following can be summarized:

(1) If the number of iterations $t \leq \eta Iter_{\max}$, the convergence speed of BPSO and HBPSO is faster than that of VBPSO, so the fitness values are also smaller than VBPSO.

(2) When $t$ reaches the maximum number of iterations, the average fitness values of HBPSO are

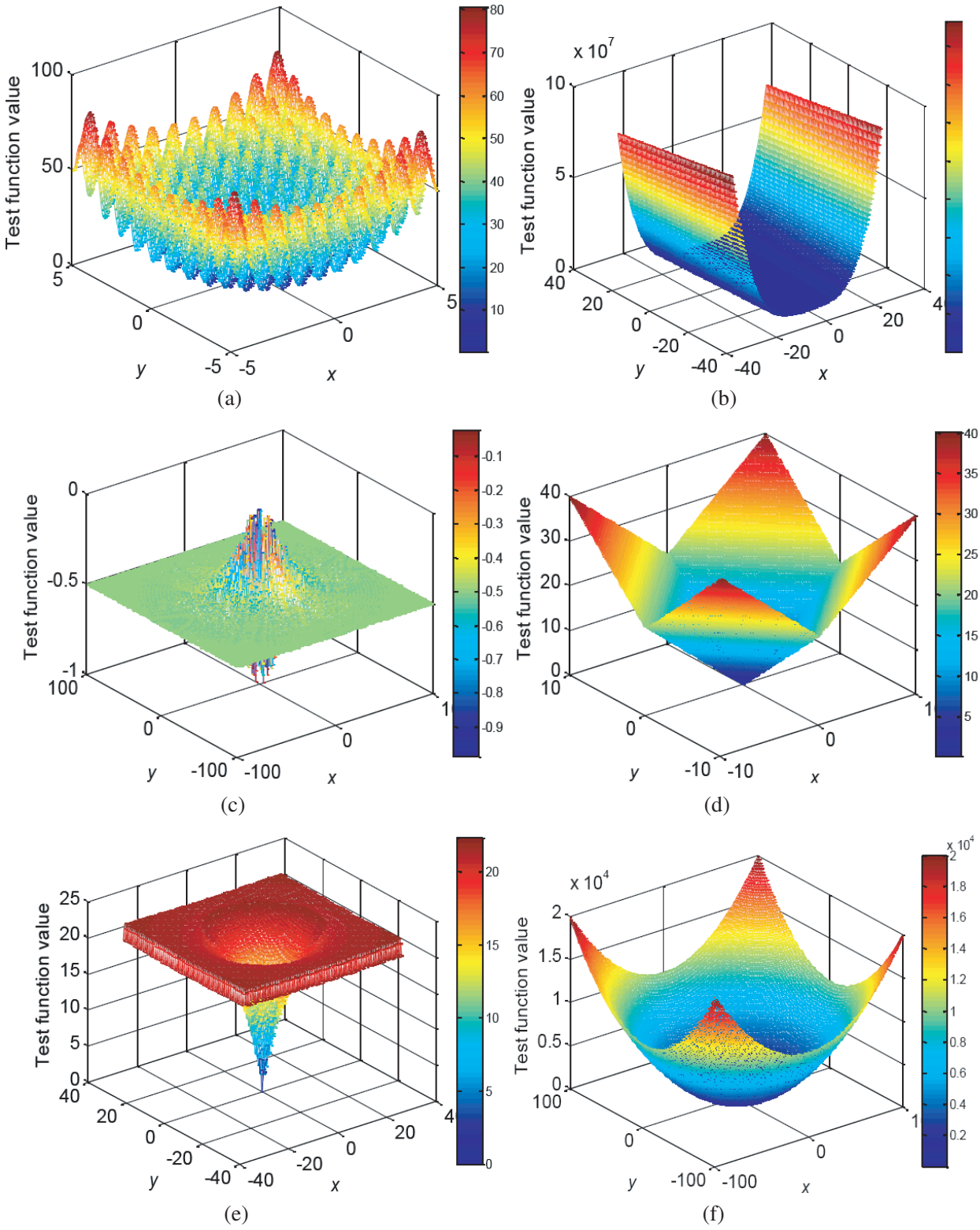


**Figure 3.** 2-D versions of benchmark functions. (a) Rastrigin. (b) Rosenbrock. (c) Schaffer. (d) Schwefel. (e) Ackley. (f) Step.

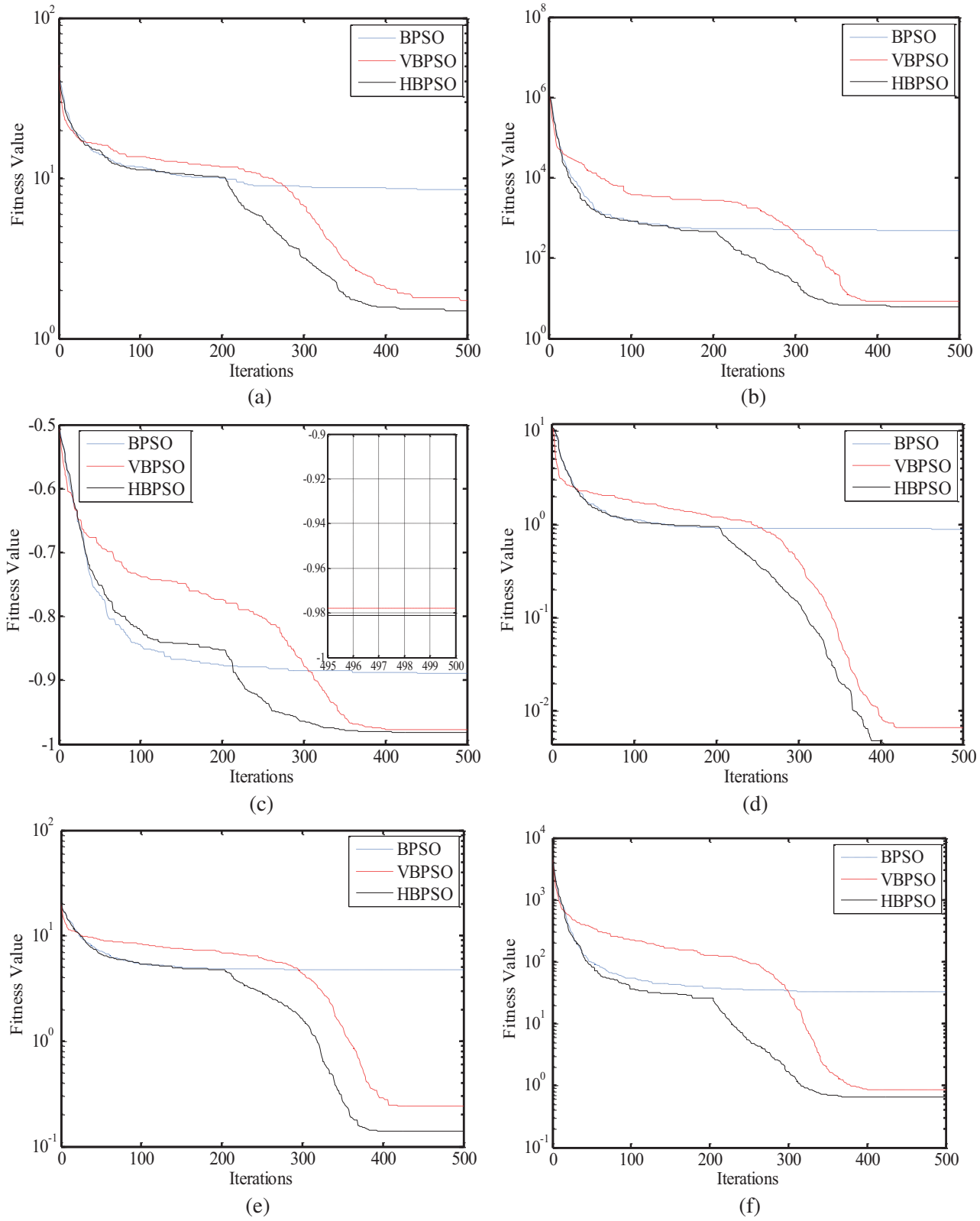


**Figure 4.** Convergence characteristics of BPSO, VBPSO and HBPSO algorithms over 50 independent runs. (a) Rastrigin. (b) Rosenbrock. (c) Schaffer. (d) Schwefel. (e) Ackley. (f) Step.

smaller than that of BPSO and VBPSO, which demonstrates that the results of HBPSO are closer to the optimal values of the functions

(3) If the benchmark functions are $f_1$, $f_3$, $f_4$, and $f_5$, the *best* searched by VBPSO and HBPSO are consistent. However, the *ave* of VBPSO over 50 independent tests is higher than that of HBPSO, which indicates that the optimization performance of HBPSO is more stable. It is worth drawing attention that the *best* of HBPSO of $f_6$ is $4.6109 \times 10^{-4}$, which is much closer to the optimal value than that of VBPSO and BPSO.

### 4.2. Simulation of Beamforming

The HBPSO algorithm is applied to a UPA with 8 rows and 8 columns, and the total number of array elements is 64. The parameters used by the algorithm are: the frequency of the narrowband far-field signals is 6.8 GHz, SNR = 10 dB, INR = 30 dB, $\xi = 0.1$, and the rest of the parameters are set as in Section 4.1. The real part and imaginary part of the excitation weights are encoded respectively, so the length of the encoded sequence is 1920 bits. The UPA receives an SOI arriving from $(\theta_0, \phi_0) = (0°, 10°)$ and 16 interference signals arriving from different directions (see Table 3). In the case studied here, an Intel Core i7 computer with 16 GB RAM is used, and the time taken for each execution is measured around 0.021 seconds. However, this problem can be overcome by using graphics processing units (GPUs), which can make the algorithm execution 10–100 times faster [29].

**Table 3.** DOAs of 16 interference signals.

| interferences | $(\theta_i, \phi_i)$ (°) | | | |
|---|---|---|---|---|
| 1–4 | $(-20, -35)$ | $(-20, -10)$ | $(-20, 16)$ | $(-20, 25)$ |
| 5–8 | $(0, 30)$ | $(-50, -20)$ | $(-5, -10)$ | $(20, 8)$ |
| 9–12 | $(10, -40)$ | $(10, -10)$ | $(10, 40)$ | $(25, -20)$ |
| 13–16 | $(40, -45)$ | $(40, -30)$ | $(40, -15)$ | $(30, 40)$ |

The results of HBPSO are compared with the LCMV technique. Initially, the optimal excitation weights of the two methods are given in Table 4 and Table 5. Meanwhile, the pattern synthesis and contour plots are shown in Figure 5 and Figure 6, respectively. Both techniques succeed to steer the peaks towards SOI and at the same time form nulls in the DOA of interference signals. However, the excitation weights of HBPSO are completely different from that of LCMV, which will essentially affect the depth of nulls and PSLL of them.

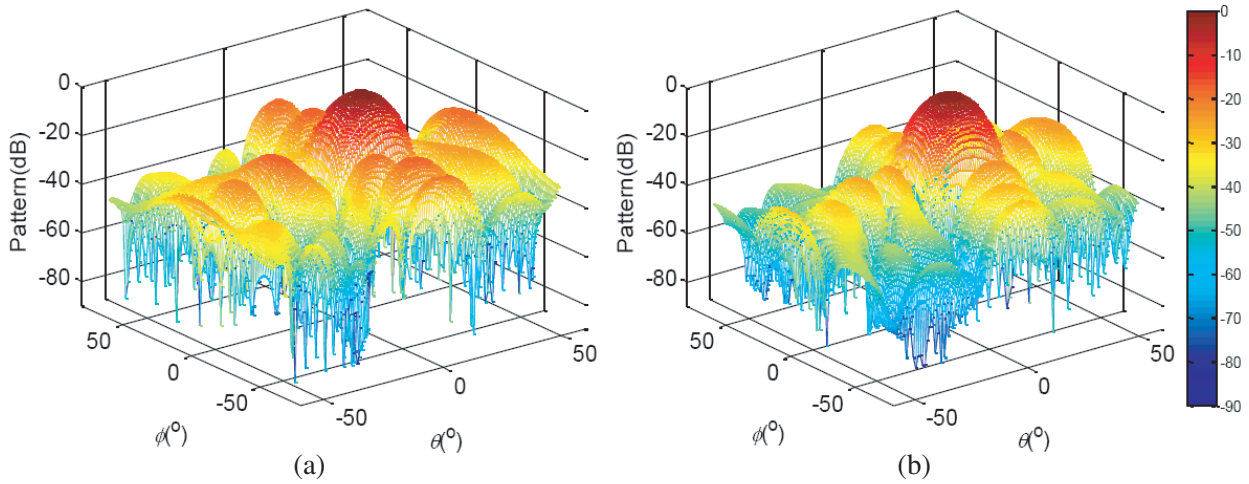


**Figure 5.** Comparison of pattern synthesis between LCMV and HBPSO. (a) LCMV. (b) HBPSO.

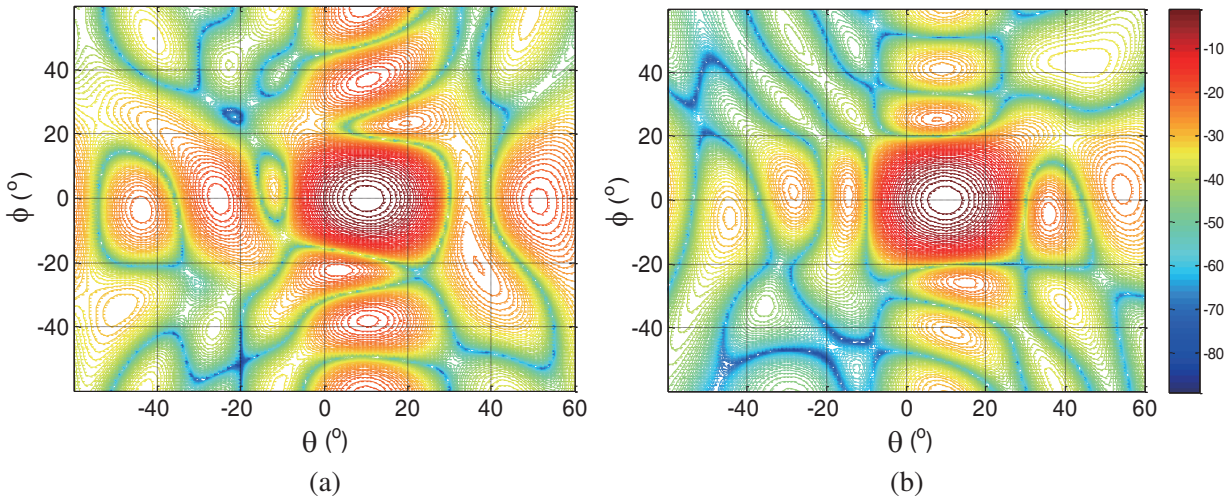


**Figure 6.** Comparison of contour plots between LCMV and HBPSO. (a) LCMV. (b) HBPSO.

**Table 4.** Optimal excitation weights of LCMV beamformer.

| $Q$ | $\mathbf{w}_{LCMV}$ | | | |
|---|---|---|---|---|
| 1–4 | $0.0218 + j0.0049$ | $0.0284 + j0.0025$ | $0.0406 - j0.0008$ | $0.0459 - j0.0061$ |
| 5–8 | $0.0391 - j0.0059$ | $0.0232 - j0.0005$ | $0.0152 + j0.0003$ | $0.0285 - j0.0017$ |
| 9–12 | $0.0178 + j0.0066$ | $0.0226 + j0.0104$ | $0.0386 + j0.0220$ | $0.0457 + j0.0281$ |
| 13–16 | $0.0387 + j0.0233$ | $0.0242 + j0.0153$ | $0.0150 + j0.0113$ | $0.0231 + j0.0157$ |
| 17–20 | $0.0167 + j0.0289$ | $0.0167 + j0.0314$ | $0.0238 + j0.0439$ | $0.0270 + j0.0548$ |
| 21–24 | $0.0234 + j0.0526$ | $0.0181 + j0.0413$ | $0.0169 + j0.0324$ | $0.0260 + j0.0341$ |
| 25–28 | $-0.0028 + j0.0469$ | $-0.0016 + j0.0504$ | $-0.0036 + j0.0687$ | $-0.0108 + j0.0852$ |
| 29–32 | $-0.0165 + j0.0824$ | $-0.0132 + j0.0639$ | $-0.0042 + j0.0480$ | $-0.0042 + j0.0462$ |
| 33–36 | $-0.0322 + j0.0334$ | $-0.0267 + j0.0401$ | $-0.0297 + j0.0580$ | $-0.0388 + j0.0745$ |
| 37–40 | $-0.0449 + j0.0732$ | $-0.0402 + j0.0558$ | $-0.0303 + j0.0403$ | $-0.0272 + j0.0383$ |
| 41–44 | $-0.0417 + j0.0103$ | $-0.0335 + j0.0147$ | $-0.0400 + j0.0208$ | $-0.0512 + j0.0263$ |
| 45–48 | $-0.0553 + j0.0258$ | $-0.0460 + j0.0193$ | $-0.0327 + j0.0140$ | $-0.0312 + j0.0120$ |
| 49–52 | $-0.0278 - j0.0023$ | $-0.0188 - j0.0006$ | $-0.0284 - j0.0032$ | $-0.0447 - j0.0061$ |
| 53–56 | $-0.0532 - j0.0067$ | $-0.0439 - j0.0071$ | $-0.0241 - j0.0060$ | $-0.0180 - j0.0059$ |
| 57–60 | $-0.0211 - j0.0192$ | $-0.0120 - j0.0093$ | $-0.0177 - j0.0149$ | $-0.0268 - j0.0291$ |
| 61–64 | $-0.0319 - j0.0335$ | $-0.0312 - j0.0261$ | $-0.0237 - j0.0158$ | $-0.0200 - j0.0098$ |

The sectional drawings of different azimuth angles are shown in Figure 7. According to these figures, it can be drawn that if $\theta$ is 0°, 10°, or 40°, the nulls are formed in the interference directions, while when $\theta = -20°$, one null is generated at $\phi = 17°$, which is slightly shifted from 16°. The reason may be that the PSLL is added into the fitness function to suppress the sidelobe, which causes the null to be shifted. However, the depth of the null reaches 62.46 dB at $\phi = 16°$, which is lower than that of LCMV and still superior to LCMV in terms of interference suppression.

Then, to compare the effect of different adjustment coefficients on the performance of HBPSO, the beamwidth between mainlobes, PSLL, and the depth of nulls are shown in Table 6, where $main^{\theta}$ and $main^{\phi}$ represent the beamwidths between mainlobes in directions of azimuth and elevation, respectively (the value of beam patterns is reduced to $-3$ dB). Nulls 1–16 indicate the depth of nulls in the DOAs

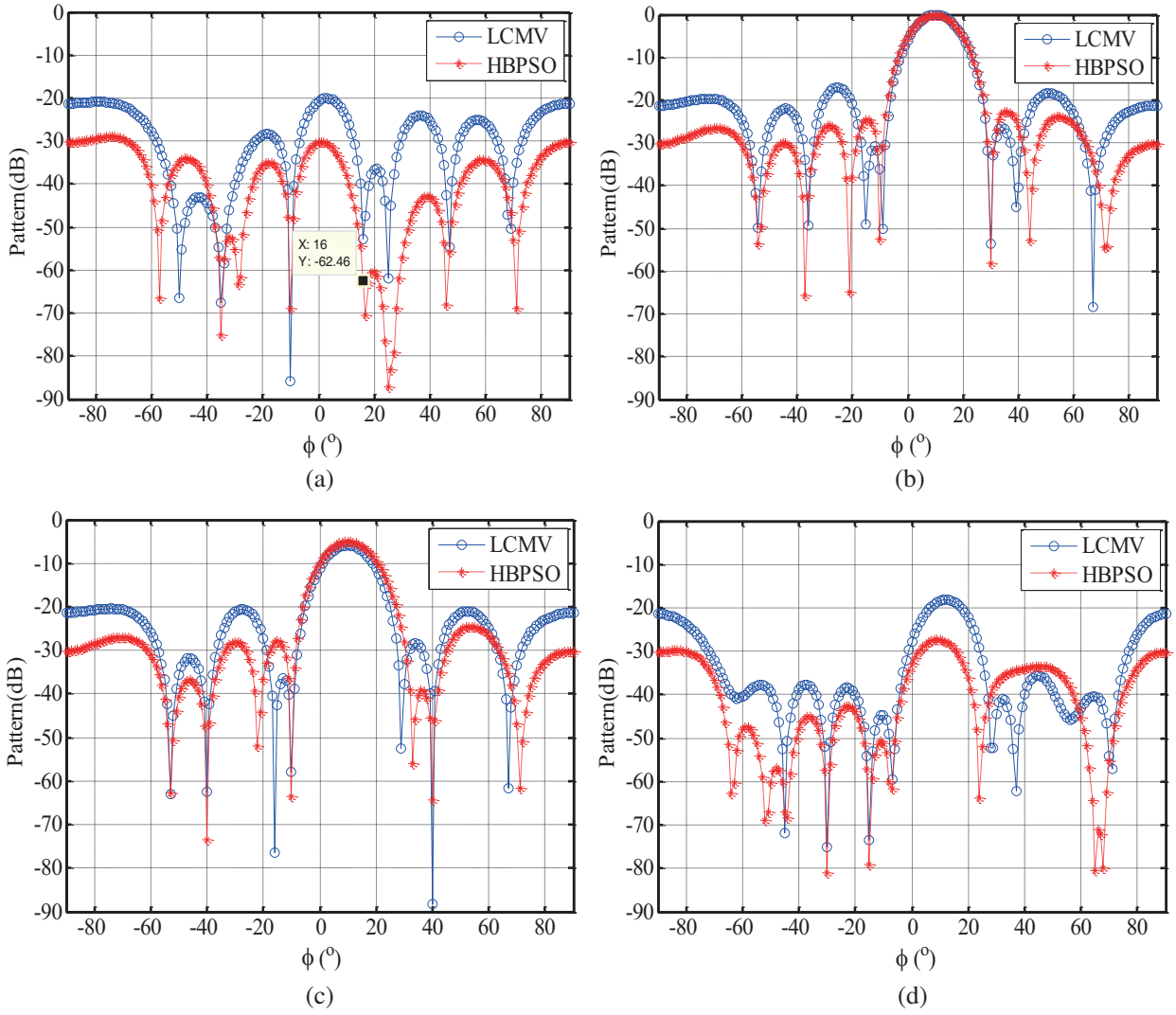


**Figure 7.** Sectional drawings of different azimuth angles. (a) $\theta = -20°$. (b) $\theta = 0°$. (c) $\theta = 10°$. (d) $\theta = 40°$.

of 16 interference signals. The values in bold represent the nulls that are deeper than those of LCMV. Similarly, the data of HBPSO in Table 6 are the average of 50 independent runs. It can be concluded that when $\xi$ changes from 0.05 to 0.2, the PSLL decreases gradually, and the number of nulls that are deeper than the LCMV technique changes from 13 to 4, which is related to the increasing proportion of PSLL in the fitness function.

The comparison of the beamwidth between mainlobes, PSLL, and the depth of nulls at different maximum speeds is shown in Table 7. It is obvious that if the maximum speed is small, the performance of HBPSO degrades severely, and the depth of nulls and PSLL are inferior to LCMV. The reason is that the particles are unable to search the whole hypercube at a low speed. When the maximum speed is set to be large, HBPSO can still maintain strong search ability, and the number of deep nulls is slightly less than that of $v_{\max} = 6$.

Finally, to verify the performance of HBPSO for different received signals, the parameters of the algorithm are fixed, i.e., $\xi = 0.1$ and $v_{\max} = 6$. The UPA receives a new SOI arriving from $(\theta_0, \phi_0) = (-5°, -15°)$ and 16 interference signals that are different from before (see Table 8). The comparison of the beamwidth between mainlobes, PSLL, and the depth of nulls between LCMV and HBPSO in different SNRs and INRs is given in Table 9. It can be concluded that when the received

**Table 5.** Optimal excitation weights of HBPSO algorithm.

| $Q$ | $\mathbf{w}_{HBPSO}$ | | | |
|---|---|---|---|---|
| 1–4 | $0.0089 - j0.0019$ | $0.0103 + j0.0002$ | $0.0124 + j0.0016$ | $0.0146 + j0.0021$ |
| 5–8 | $0.0159 + j0.0013$ | $0.0146 - j0.0005$ | $0.0101 - j0.0012$ | $0.0043 - j0.0005$ |
| 9–12 | $0.0097 + j0.0067$ | $0.0136 + j0.0099$ | $0.0175 + j0.0131$ | $0.0212 + j0.0156$ |
| 13–16 | $0.0229 + j0.0162$ | $0.0206 + j0.0139$ | $0.0145 + j0.0092$ | $0.0076 + j0.0045$ |
| 17–20 | $0.0029 + j0.0110$ | $0.0087 + j0.0186$ | $0.0135 + j0.0275$ | $0.0164 + j0.0335$ |
| 21–24 | $0.0173 + j0.0342$ | $0.0156 + j0.0298$ | $0.0115 + j0.0212$ | $0.0063 + j0.0119$ |
| 25–28 | $-0.0005 + j0.0137$ | $-0.0008 + j0.0264$ | $-0.0018 + j0.0385$ | $-0.0027 + j0.0452$ |
| 29–32 | $-0.0038 + j0.0452$ | $-0.0040 + j0.0389$ | $-0.0035 + j0.0272$ | $-0.0029 + j0.0140$ |
| 33–36 | $-0.0064 + j0.0128$ | $-0.0141 + j0.0234$ | $-0.0211 + j0.0329$ | $-0.0252 + j0.0377$ |
| 37–40 | $-0.0261 + j0.0370$ | $-0.0226 + j0.0311$ | $-0.0158 + j0.0212$ | $-0.0080 + j0.0112$ |
| 41–44 | $-0.0121 + j0.0053$ | $-0.0222 + j0.0093$ | $-0.0307 + j0.0136$ | $-0.0347 + j0.0159$ |
| 45–48 | $-0.0336 + j0.0159$ | $-0.0276 + j0.0130$ | $-0.0184 + j0.0092$ | $-0.0091 + j0.0068$ |
| 49–52 | $-0.0086 - j0.0013$ | $-0.0169 - j0.0018$ | $-0.0245 - j0.0020$ | $-0.0280 - j0.0016$ |
| 53–56 | $-0.0262 - j0.0010$ | $-0.0218 - j0.0007$ | $-0.0167 - j0.0008$ | $-0.0115 - j0.0009$ |
| 57–60 | $-0.0029 - j0.0031$ | $-0.0070 - j0.0071$ | $-0.0109 - j0.0095$ | $-0.0131 - j0.0090$ |
| 61–64 | $-0.0126 - j0.0074$ | $-0.0106 - j0.0064$ | $-0.0080 - j0.0063$ | $-0.0056 - j0.0069$ |

**Table 6.** Comparison of the beamwidth between mainlobes, PSLL and the depth of nulls between LCMV and HBPSO.

| Parameters | LCMV | HBPSO | | |
|---|---|---|---|---|
| | | $\xi = 0.05$ | $\xi = 0.1$ | $\xi = 0.2$ |
| $main^{\theta}/main^{\phi}$ (°) | 14.78/15.16 | 15.42/15.33 | 15.82/15.71 | 15.67/16.30 |
| PSLL (dB) | −17.04 | −19.83 | −21.62 | −22.47 |
| null 1 (dB) | −67.51 | **−80.81** | **−75.21** | −66.09 |
| null 2 (dB) | −85.71 | **−92.74** | −68.89 | −61.37 |
| null 3 (dB) | −52.88 | **−72.04** | **−62.46** | **−54.77** |
| null 4 (dB) | −61.86 | **−73.09** | **−87.16** | **−69.58** |
| null 5 (dB) | −53.71 | **−73.90** | **−58.12** | −53.26 |
| null 6 (dB) | **−90.11** | −81.03 | −87.81 | −66.26 |
| null 7 (dB) | −53.77 | **−68.76** | **−58.54** | −52.97 |
| null 8 (dB) | −52.44 | **−71.41** | **−65.80** | **−55.41** |
| null 9 (dB) | −62.57 | **−71.76** | **−73.52** | −62.52 |
| null 10 (dB) | −57.81 | **−75.43** | **−63.63** | −57.05 |
| null 11 (dB) | **−88.35** | −77.34 | −64.40 | −59.08 |
| null 12 (dB) | **−76.52** | −75.77 | −72.91 | −60.02 |
| null 13 (dB) | −71.87 | **−83.25** | −66.99 | −68.94 |
| null 14 (dB) | −75.20 | **−79.66** | **−80.90** | −64.83 |
| null 15 (dB) | −73.43 | **−83.74** | **−79.05** | **−81.82** |
| null 16 (dB) | −77.62 | **−77.95** | −66.16 | −63.49 |

**Table 7.** Comparison of the beamwidth between mainlobes, PSLL and the depth of nulls under different $v_{\max}$.

| Parameters | $v_{\max}=4$ | $v_{\max}=8$ | Parameters | $v_{\max}=4$ | $v_{\max}=8$ |
|---|---|---|---|---|---|
| $main^{\theta}/main^{\phi}$ (°) | 15.18/15.26 | 15.79/15.73 | PSLL (dB) | −16.88 | −21.17 |
| null 1 (dB) | −60.11 | **−77.68** | null 2 (dB) | −56.39 | −72.35 |
| null 3 (dB) | −47.37 | **−63.17** | null 4 (dB) | −60.42 | **−92.26** |
| null 5 (dB) | **−53.96** | **−58.12** | null 6 (dB) | −56.19 | −76.78 |
| null 7 (dB) | **−57.78** | **−57.69** | null 8 (dB) | **−53.66** | **−66.62** |
| null 9 (dB) | −52.66 | **−70.48** | null 10 (dB) | −56.08 | **−62.10** |
| null 11 (dB) | −55.59 | −64.23 | null 12 (dB) | −55.16 | −75.92 |
| null 13 (dB) | −63.40 | −70.18 | null 14 (dB) | −70.25 | **−80.69** |
| null 15 (dB) | −58.66 | −71.86 | null 16 (dB) | −62.62 | −66.78 |

**Table 8.** DOAs of 16 interference signals.

| interferences | $(\theta_i, \phi_i)$ (°) | | | |
|---|---|---|---|---|
| 1–4 | $(-10,-40)$ | $(-36,-27)$ | $(23,-26)$ | $(-5,5)$ |
| 5–8 | $(0,-33)$ | $(-15,18)$ | $(15,-13)$ | $(-25,21)$ |
| 9–12 | $(35,39)$ | $(15,25)$ | $(5,-34)$ | $(36,-55)$ |
| 13–16 | $(20,-28)$ | $(34,-19)$ | $(-15,-25)$ | $(-16,-20)$ |

**Table 9.** Comparison of the beamwidth between mainlobes, PSLL and the depth of nulls between LCMV and HBPSO in different SNRs and INRs.

| Parameters | SNR = 10 dB, INR = 30 dB | | SNR = 0 dB, INR = 30 dB | | SNR = 10 dB, INR = 20 dB | |
|---|---|---|---|---|---|---|
| | LCMV | HBPSO | LCMV | HBPSO | LCMV | HBPSO |
| $main^{\theta}$ (°) | 13.82 | 14.47 | 13.85 | 14.36 | 14.02 | 14.77 |
| $main^{\phi}$ (°) | 14.65 | 15.21 | 14.50 | 15.13 | 14.58 | 15.19 |
| PSLL (dB) | −17.79 | −20.63 | −16.92 | −20.32 | −17.68 | −20.19 |
| null 1 (dB) | **−60.04** | −52.06 | **−65.9** | −55.51 | **−52.89** | −43.41 |
| null 2 (dB) | **−71.78** | −63.78 | **−70.24** | −67.57 | **−56.86** | −54.35 |
| null 3 (dB) | −44.55 | **−48.25** | −46.29 | **−50.59** | −40.49 | **−43.76** |
| null 4 (dB) | −51.36 | **−51.82** | −55.03 | **−55.49** | **−51.26** | −42.54 |
| null 5 (dB) | −44.29 | **−51.19** | −46.36 | **−53.52** | −39.17 | **−47.48** |
| null 6 (dB) | −62.73 | **−63.40** | −63.95 | **−66.97** | −52.53 | **−54.40** |
| null 7 (dB) | **−66.19** | −51.22 | **−72.82** | −54.78 | **−55.28** | −42.29 |
| null 8 (dB) | **−72.08** | −60.99 | **−68.62** | −64.51 | **−77.31** | −52.31 |
| null 9 (dB) | −64.19 | **−84.10** | −62.54 | **−86.48** | −57.06 | **−79.42** |
| null 10 (dB) | −70.23 | **−70.98** | −68.42 | **−73.86** | −60.51 | **−64.72** |
| null 11 (dB) | −46.27 | **−49.58** | −49.05 | **−52.03** | −42.50 | **−45.01** |
| null 12 (dB) | −71.93 | **−77.98** | −62.55 | **−82.11** | −64.30 | **−67.39** |
| null 13 (dB) | −44.35 | **−49.96** | −47.38 | **−51.85** | −41.36 | **−48.41** |
| null 14 (dB) | **−56.61** | −53.41 | **−59.41** | −56.83 | **−49.27** | −44.82 |
| null 15 (dB) | −59.27 | **−62.34** | −57.50 | **−64.64** | −51.60 | **−61.18** |
| null 16 (dB) | −43.13 | **−47.14** | −47.49 | **−50.63** | −41.11 | **−43.59** |

signal of UCA changes, different SNRs have little impact on the performance of the HBPSO algorithm. When the INR changes from 30 dB to 20 dB, the depths of nulls of both LCMV and HBPSO become shallower. However, the number of nulls that are deeper than LCMV has always remained at a high level, which indicates that the performance of the HBPSO algorithm is still superior to LCMV while the received signal is changed.

## 5. CONCLUSIONS

An improved BPSO algorithm called HBPSO, which combines the S-shaped transformation function and V-shaped transformation function, has been developed in this paper. In HBPSO, the global search ability and local optimization ability are both taken into account, which can improve the optimization performance of the algorithm. Through the tests of six benchmark functions, the results show that HBPSO can be much closer to the optimal values of the functions. Then, HBPSO is applied to the beamforming of UPA. In order to reduce the PSLL, a fitness function including the term of PSLL is constructed. By comparing the pattern synthesis, contour plots, and sectional drawings of different azimuth angles between LCMV and HBPSO, as well as the effect of different adjustment coefficients and different speeds on the performance of HBPSO, it can be concluded that by setting the parameters reasonably, HBPSO succeeds not only to form deeper nulls towards the DOA of interference signals but also to depress the PSLL more than LCMV does. Therefore, the proposed algorithm seems quite promising in the beamforming applications of planar antenna arrays.

## ACKNOWLEDGMENT

The work was supported by the Youth Science Foundation of the National Natural Science Foundation of China (Grant No. 61903375).

## REFERENCES

1. Pham, T. and K. Insoo, "Robust weighted sum harvested energy maximization for SWIPT cognitive radio networks based on particle swarm optimization," *Sensors*, Vol. 17, No. 10, 2275, 2017.
2. Ukasz, K., L. Nowak, and C. Jedryczka, "Optimization of the rotor geometry of the line-start permanent magnet synchronous motor by the use of particle swarm optimization," *Compel International Journal of Computations & Mathematics in Electrical*, Vol. 34, No. 3, 882–892, 2015.
3. Di, J., H. Wen, M. Li, et al., "An optimized 2D-robust adaptive beamforming algorithm based on matrix completion in sparse array," *MATEC Web of Conferences*, Vol. 208, Article ID 01003, 2018.
4. Souto, V., R. D. Souza, B. F. Uchoa-Filho, et al., "Beamforming optimization for intelligent reflecting surfaces without CSI," *IEEE Wireless Communication Letters*, Vol. 99, 1–5, 2020.
5. Wei, B., S. Fan, C. Wan, et al., "Optimization strategy of anti-interference performance based on BPSO," *Journal of Physics: Conference Series*, Vol. 1693, No. 1, Article ID 012061, 2020.
6. Omar, R., N. P. Perez, C. E. Tuna, et al., "A PSO-BPSO technique for hybrid power generation system sizing," *IEEE Latin America Transactions*, Vol. 18, No. 8, 1362–1370, 2020.
7. Kumar, L. and K. K. Bharti, "A novel hybrid BPSO–SCA approach for feature selection," *Natural Computing*, Vol. 20, No. 14, 1–23, 2021.
8. Niwariya, M., "Data mining approach for diabetes prediction using BPSO, SVM, KNN and nave bayes classifiers," *International Journal of Advanced Trends in Computer Science and Engineering*, Vol. 9, No. 1.5, 286–293, 2020.
9. Sancar, N. and D. Inan, "Identification of influential observations based on binary particle swarm optimization in the cox PH model," *Communications in Statistics — Simulation and Computation*, Vol. 49, No. 3, 567–590, 2020.
10. Feng, Q., W. B. Wang, and D. Liu, "Synthesis of thinned linear and planar antenna arrays using binary PSO algorithm," *Progress In Electromagnetics Research*, Vol. 127, No. 1, 371–387, 2012.

11. Chen, W. and J. Elangage, "Nonuniformly spaced array with the direct data domain method for 2D angle-of-arrival measurement in electronic support measures application from 6 to 18 GHz," *International Journal of Antennas and Propagation*, Vol. 2020, No. 2, Article ID 9651650, 2020.
12. Sieh, K. T., S. S. Balasem, P. Siaw, et al., "Minimum variance distortionless response beamformer with enhanced nulling level control via dynamic mutated artificial immune system," *The Scientific World Journal*, Vol. 6, No. 5, 1–9, 2014.
13. Shuang, Q., W. Sheng, X. Ma, et al., "A robust reduced-rank monopulse algorithm based on variable-loaded MWF with spatial blocking broadening and automatic rank selection," *Digital Signal Processing*, Vol. 78, 205–217, 2018.
14. Nan, H., S. Bing, Z. Yi, et al., "Underdetermined DOA estimation method for wideband signals using joint nonnegative sparse bayesian learning," *IEEE Signal Processing Letters*, Vol. 24, No. 5, 535–539, 2017.
15. Khan, M., A. N. Malik, F. Zaman, et al., "Robust LCMV beamformer for direction of arrival mismatch without beam broadening," *Wireless Personal Communications*, Vol. 104, 21–36, 2019.
16. Huang, S., Y. Li, F. J. Han, et al., "Adaptive beamforming algorithm for interference suppression based on partition PSO," *Information Technology, Electronics & Mobile Communication Conference*, IEEE, 2016.
17. Zaharis, Z. D. and C. Skeberis, "Design of a novel antenna array beamformer using neural networks trained by modified adaptive dispersion invasive weed optimization based data," *IEEE Transactions on Broadcasting*, Vol. 59, No. 3, 455–460, 2013.
18. Sharma, M. and K. K. Sarma, "GA based MVDR beamforming in UWB wireless channel," *1st International Conference on Wireless and Mobile Communication Systems (WMCS '13)*, 2013.
19. Xia, W., X. Jin, and F. Dou, "Beam performance optimization of multibeam imaging sonar based on the hybrid algorithm of binary particle swarm optimization and convex optimization," *International Journal of Antennas and Propagation*, Vol. 5, No. 12, 1–6, 2016.
20. Biswas, R. N., A. Saha, S. K. Mitra, et al., "Realization of PSO-based adaptive beamforming algorithm for smart antennas," *Advances in Nature-Inspired Computing and Applications*, Vol. 6, 135–163, 2019.
21. Camellia, D. and K. T. Sieh, "Performance comparison of FA, PSO and CS application in SINR optimisation for LCMV beamforming technique," *Wirel. Pers. Commun.*, Vol. 2, 1–19, 2018.
22. Zaharis, Z. D., C. Skeberis, and T. D. Xenos, "Improved antenna array adaptive beamforming with low side lobe level using a novel adaptive invasive weed optimization method," *Progress In Electromagnetics Research*, Vol. 124, No. 8, 137–150, 2012.
23. Sudantha, P., Y. Pan, Y. Zhang, et al., "A fully reconfigurable polarimetric phased array antenna testbed," *International Journal of Antennas and Propagation*, Vol. 2014, Article ID 439606, 2014.
24. Park, Y., W. Seong, and P. Gerstoft, "Block-sparse two-dimensional off-grid beamforming with arbitrary planar array geometry," *The Journal of the Acoustical Society of America*, Vol. 147, No. 4, 2184–2191, 2020.
25. Chen, J. and Y. Yin, "Novel beam forming approach for rectangular planar array," *Microwave and Optical Technology Letters*, Vol. 62, No. 2, 1–7, 2020.
26. Nezamabadi, H. P., M. S. Rostami, and M. M. Farsangi, "Binary particle swarm optimization: Challenges and new solutions," *The Journal of Computer Society of Iran (CSI) On Computer Science and Engineering*, Vol. 6, No. 1-A, 21–32, 2008.
27. Mirjalili, S. and A. Lewis, "S-shaped versus V-shaped transfer functions for binary particle swarm optimization," *Swarm & Evolutionary Computation*, Vol. 9, 1–14, 2013.
28. Zaharis, Z. D., K. A. Gotsis, and J. N. Sahalos, "Adaptive beamforming with low side lobe level using neural networks trained by mutated boolean PSO," *Progress In Electromagnetics Research*, Vol. 127, 139–154, 2012.
29. Zaharis, Z. D. and T. V. Yioultsis, "A novel adaptive beamforming technique applied on linear antenna arrays using adaptive mutated boolean PSO," *Progress In Electromagnetics Research*, Vol. 117, 165–179, 2011.