IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 26, 2019, accepted October 17, 2019, date of publication October 30, 2019,
date of current version November 14, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2949028

A Fibonacci Branch Search (FBS)-Based Optimization Algorithm for Enhanced Nulling Level Control Adaptive Beamforming Techinique

HAICHUAN ZHANG AND FANGLING ZENG
School of Electronic Countermeasures, National University of Defense Technology, Hefei 230037, China
Corresponding author: Haichuan Zhang (zhanghai_scholar@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61272333.

ABSTRACT The present work introduced an adaptive beamformer based on a Fibonacci branch search (FBS) based heuristic algorithm is proposed for uniform linear arrays. The proposed optimization technique inspired by the Fibonacci sequence principle, designated Fibonacci branch search (FBS), used the new tree's fundamental branch structure and interactive searching rules to obtain the global optimal solution in the search space. The branch structure of FBS is selected by two types of multidimensional points on the basis of the shortening fraction formed by the Fibonacci sequence; in this mode, interactive global and local searching rules are implemented alternately to reach optimal solutions, avoiding stagnating in local optima. The proposed FBS is also used to construct an adaptive beamforming technique as a real-time implementation to achieve near-optimal performance due to its simplicity and high convergence; the performance of FBS is compared with that of five typical heuristic optimization algorithms. Simulation results demonstrate the superiority of the proposed FBS approach in locating the optimal solution with higher precision and further improvement in the adaptive beamforming performance.

INDEX TERMS Adaptive beamformer, uniform linear arrays, shortening fraction, Fibonacci branch search, interactive searching rule.

I. INTRODUCTION

As a versatile approach, adaptive beamforming (ABF) has received considerable attention over the past several decades and has become a fundamental technique for numerous applications, including radio astronomy, applied acoustics, cognitive communications, and medical imaging [1]–[4]. ABF possesses the potential to optimize the radiation pattern in real time, therein obtaining a larger output signal-to-interference-plus-noise ratio (SINR) by steering the main lobe of the radiation towards a desired signal while placing respective nulls towards several interference signals [5].

Classic ABF techniques used to extract the respective array excitation weights are based on two main criteria: the minimum mean square error (MMSE) and the maximum signal-to-noise ratio (MSINR) [6]. A typical ABF algorithm is the minimum variance distortionless response (MVDR) based on the maximum signal-to-interference-plus-noise ratio (MSINR) criterion. The design of this beamformer involves minimizing the output power subject to the unit gain constraint to the desired signal [7], [8]. Although the MVDR beamformer is capable of suppressing the interference and improving system reliability, the weights computed by MVDR are unable to form the deep nulls toward the interference source in various interference scenarios because of the nature of this technique. Conventional methods used to solve this problem are very time consuming and are unmanageable in ABF applications. A variant of the MVDR beamformer, also known as the linearly constrained minimum variance (LCMV) beamformer, requires only knowledge of the desired signal direction of arrival (DOA) to maximize the SINR [9]. However, the low convergence rate of this technique makes it also inappropriate for real-time applications. Another criterion for computing the array weights is to minimize the MMSE. One of

The associate editor coordinating the review of this manuscript and approving it for publication was Huiqing Wen.

the most widely used MMSE-based adaptive algorithms is the least mean square (LMS) approach. This method requires a training sequence of the signal of interest (SOI) to adjust the complex weights adaptively and minimize the difference between the array output for forming the optimum array pattern [10], [11]. However, the method exhibits a high probability of tracking into a local optimal solution and thus may not be applicable for ABF in harsh environments. Consequently, the inherent shortcomings of derivative-based ABF methods have compelled and motivated many researchers to explore meta-heuristics (MH) and evolutionary methods for overcoming these types of difficulties.

The main advantage of the evolutionary heuristic algorithms over the classical derived approaches is the former's ability to search for the global optimum of the objective functions without using the derivatives of the objective functions. In addition, MH optimization algorithms do not require extra iterative derivations or computationally extensive routines for ABF with objective functions. According to the above, an enormous number of evolutionary algorithms have been dedicated to applying several optimization approaches for ABF problems in recent decades. Approaches such as gravitational search algorithm (GSA), particle swarm optimization (PSO) and other modified techniques represent a set of optimization algorithms that have been suggested in recent decades to solve a variety of ABF problems and address various issues facing array systems [12], [13]. Many studies have shown that these algorithms are capable of finding global or strong local optima of nonlinear multimodal functions with multidimensional solutions [14]. Therefore, the weights of the beamformer extracted by the optimization techniques according to the fitness function defined by the specific criterion can be used to place a maximum beam and null in an array pattern in specified locations. Compared to other evolutionary algorithms, PSO is much easier to implement and achieves better performance; thus, many examples have been successfully demonstrated and validated due to the design flexibility of PSO under the framework of ABF arrays [15], [16].

Although the MH-based ABF algorithm has a lower mathematical complexity than derivative-based and iterative-based ABF methods, there still exist weaknesses and limitations to ABF applications. Certain optimization algorithms are highly dependent on the starting points in the case of a large number of solution variables [17]. However, the weights of the ABF are regularly associated with a large number of array elements, and the excitations of the array elements are complex, i.e., having both amplitude and phase; hence, the ABF solution space cannot be very small. In such a case, the conventional evolutionary algorithm would not really be applicable to ABF. In addition, the classic optimization methods are prone to becoming trapped in local minima and not reaching the global optimum when solving complex multimodal optimization problems of array weight extraction, resulting in a suboptimal beamforming performance. In addition, most MH algorithms are population-based optimization techniques that require long execution times to converge, especially when solving large-scale, complex ABF engineering problems [18]. In addition, the complexity of implementing the algorithms also results in huge costs and hardware resource consumptions. Although certain new optimization algorithms, such as the adaptive multiple-elite-guided composite differential evolution algorithm (AMECoDEs), the depth-first search artificial bee colony (DFSABC), and the adaptive population artificial bee colony (APABC), provide a relatively excellent performance, their application to the ABF model has not yet been realized in the research field [19]–[21].

In consideration of the above-mentioned studies, we propose a interactive-based random iterative search strategy called FBS in this paper to address complicated optimization problems for ABF. The concept of the proposed FBS is defined from two aspects. The first aspect is the generation principle of the Fibonacci branch architecture. The establishment of the branch structure in FBS is built upon the optimization process of the search points, and the shortening fraction is designed based on the Fibonacci sequence for the generation of a set of optimization elements consisting of two types of search points. The optimization endpoint searches for the optimal solution according to the growth path mode of the branch. The second aspect of the FBS concept is that the construction of the interaction iteration applies rules for the computation of the optimization elements. The iterative rules are composed of global search and local optimization, which are the two phases necessary to update the optimization elements. The global tentative points and local search points are formulated during two interaction processes, and the points with the best fitness converge toward the global optimum in the search space. Simultaneously, computer memory can be fully utilized to record the optimization process during the interaction optimization. Global randomness is one of the important characteristics of FBS, and this mechanism is implemented for those points that do not readily fall into the local optimum and that are not able to find a better solution. The contribute of this paper lies in the fact that we design the Fibonacci branch optimization structure and propose a global search and iterative local optimization interaction technique. In addition, the FBS algorithm proposed here has been applied to antenna array beamforming in several cases and compared with other evolutionary optimization-based techniques on several test functions and the robust ABF. To the best of our knowledge, the proposed FBS optimization algorithm is being applied for the first time to antenna array beamforming problems.

The remainder of this paper is organized as follows: the ABF system model of uniform linear array (ULA) is described in section 2. Section 3 presents the proposed FBS optimization algorithm. Then, Section 4 introduces the robust ABF based on FBS. The validation of the proposed FBS via benchmark functions and the simulation results are reported in section 5. Section 6 gives the conclusions.

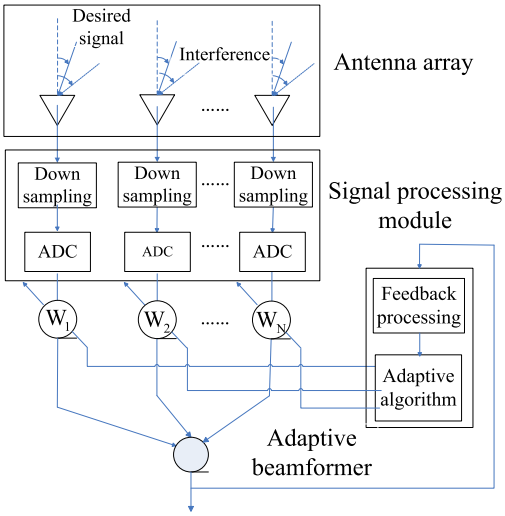


FIGURE 1. Schematic of a linear antenna array processor.

II. BEANMFORMING SYSTEM MODEL OF ULA AND PROBLEM STATEMENT

Consider a uniform linear antenna array of M omni-directional array elements employed in an ABF receiver, one desired signal and Q uncorrelated interference sources that impinge on the array at the kth snapshot. This be expressed by [22]

$$\mathbf{x}(k) = s(k)\,\mathbf{a}(\theta_d) + \sum_{i=1}^{Q} i_i(k)\,\mathbf{a}(\theta_i) + \mathbf{n}(k) \tag{1}$$

where $s(k)$ and $i_i(k)$ are the desired global navigation satellite system (GNSS) signal and the ith interference source, respectively, and $\mathbf{n}(k)$ denotes the complex vector of the sensor noise. $\mathbf{a}(\theta_d)$ and $\mathbf{a}(\theta_i)$ represent $M \times 1$ steering vectors of $s(k)$ and $i_i(k)$, as given by

$$\begin{aligned}\mathbf{a}(\theta_d) &= \left[1, e^{-j2\pi \frac{d_c}{\lambda}\cos\theta_d}, \cdots, e^{-j2\pi \frac{d_c}{\lambda}(M-1)\cos\theta_d}\right]^T \\ \mathbf{a}(\theta_i) &= \left[1, e^{-j2\pi \frac{d_c}{\lambda}\cos\theta_i}, \cdots, e^{-j2\pi \frac{d_c}{\lambda}(M-1)\cos\theta_i}\right]^T\end{aligned} \tag{2}$$

where θ_d and θ_i denote the direction of the desired signal and the ith interference source, respectively;$d_c = \frac{\lambda}{2}$ is the inter-element spacing; λ is the wavelength of the GNSS carrier; and T is the transpose operation.

The array beamformer output can be written as

$$y(k) = \mathbf{w}^H \mathbf{x}(k) \tag{3}$$

where $\mathbf{w}$ is the complex beamforming weight vector of the antenna array and H denotes the Hermitian transpose.

The schematic structure of a linear adaptive antenna array processor on the front end of the receiver is shown in Figure 1.

The extracted weight vector is chosen to maximize the output SINR for improving the performance of the ABF, as given by

$$SINR = \frac{\sigma_S^2 \left|\mathbf{w}^H a(\theta_d)\right|^2}{\mathbf{w}^H R_{i+n} \mathbf{w}} \tag{4}$$

where σ_S^2 is the desired signal power, and R_{i+n} is the interference-plus-noise covariance (INC) matrix.

The adaptive beamformer studied here is an inherently multi-objective problem since multiple sets of agents $\mathbf{w}$ with amplitude and phase must produce deep nulls towards the interference location and steer the radiation beam towards the desired user to achieve the maximum SINR. On the other hand, all the optimization methods to the designed for ABF aim at finding the near-global minimum of a mathematical fitting function called the fitness function; therefore, the best weight vector is determined according to the fitness value obtained from the objective function defined based on the SINR. In the following sections, the proposed FBS optimization is applied to adjust the weights such that the fitness function requirements are achieved to obtain the optimum SINR.

III. FIBONACCI BRANCH SEARCH ALGORITHM

A. THE STANDARD PRINCIPLE OF THE FIBONACCI SEQUENCE METHOD

The famous Fibonacci sequence was first proposed by Italian mathematician Leonardoda in the 12–13th century, and the recursion formula is given by [23], [24]

$$\begin{cases} F_1 = F_2 = 1 \\ F_j = F_{j-1} + F_{j-2}, \quad j \geq 3 \end{cases} \tag{5}$$

where F_j represents the jth general term of the Fibonacci sequence.

The Fibonacci sequence optimization method makes the tentative optimization points in the defined interval converge to the optimal solution by compressing the search interval proportionally based on the Fibonacci sequence term. It has been perceived as one of the most effective strategies to solve one-dimensional unimodal problems [25]. Let us investigate below how the optimization method using the Fibonacci sequence works for a unimodal continuous function in an interval for a minimization problem. Suppose a unimodal $f(x)$ function defined on the interval $[A, B]$. Initially, the technique starts with a choice for two feasible points, x_1 and $\widetilde{x}_1$, where $x_1 < \widetilde{x}_1$ in the given range for the first iteration. Then, it is necessary to reduce the initial box of the range to a sufficiently small box region including the minimum solution of $f(x)$ (through an iteration process) since the interval in which the minimum lies can be narrowed down, provided the function values are known at two different points in the range [26].

Let x_p and $\widetilde{x}_p$ denote the new points over the range $[A_p, B_p]$ to be chosen for shortening the length of the interval at the pth iteration involving the optimal point. Hence, $p = 1, 2, \ldots, N$, where N represents the maximum number of iterations, and the Fibonacci algorithm can be executed as follows.

Step 1 Initialization

Choose $[\mathbf{A}_1, \mathbf{B}_1] = [\mathbf{A}, \mathbf{B}]$ and take $p = 1$. While $p < N$, do

FIGURE 2. Basic structure of the proposed FBS.

Step 2 Calculate the iterative points according to the shortening fraction $\frac{F_p}{F_{p+1}}$ formed by the Fibonacci sequence

$$x_p = A_p + \left(1 - \frac{F_p}{F_{p+1}}\right)(B_p - A_p),$$
$$\widetilde{x}_p = A_p + \frac{F_p}{F_{p+1}}(B_p - A_p) \tag{6}$$

Step 3 Find the values $f(x_p)$ and $f(\widetilde{x}_p)$ using the new points

Step 4 If $f(x_p) < f(\widetilde{x}_p)$, then $A_{p+1} = A_p, B_{p+1} = \widetilde{x}_p$; otherwise, $A_{p+1} = x_p, B_{p+1} = B_p$

Step 5 Set $p = p + 1$. Then, go to step 1 and determine whether to stop the iteration

The minimum point of $f(x)$ can be reached through the above steps due to the convergence trend of the tentative points and the linear convergence quality of the Fibonacci optimization strategy.

B. THE BASIC STRUCTURE OF THE PROPOSED FIBONACCI BRANCH

The standard Fibonacci strategy cannot efficiently solve multi-variate problems and reliably perform the optimum fitness evaluation of multimodal functions [27]. This is in contradiction to classic heuristic optimization algorithms. The FBS algorithm proposed in this paper is used to overcome these defects while avoiding a loss of the optimal search trajectories by using search elements with dendritic branch structures and interactive searching optimization rules.

The basic structure of FBS expanded to a multi-dimensional space D can be illustrated as follows:

where $\mathbf{X}_A$, $\mathbf{X}_B$ and $\mathbf{X}_C$ are the vectors in D-dimensional Euclidean space. $\mathbf{X}_A$ and $\mathbf{X}_B$ represent the endpoints of the search element satisfying the optimization rule, and $\mathbf{X}_C$ denotes the segmentation points that can be determined from the search rule. A proportion of the vectors can be calculated as follows:

$$\frac{\|\mathbf{X}_C - \mathbf{X}_A\|}{\|\mathbf{X}_B - \mathbf{X}_A\|} = \frac{\|\mathbf{X}_B - \mathbf{X}_C\|}{\|\mathbf{X}_C - \mathbf{X}_A\|} = \frac{F_p}{F_{p+1}} \tag{7}$$

where F_p is the pth Fibonacci number.

Considering that the multimodal function with multiple variables $\mathbf{f}(\mathbf{X})$ is to be minimized in the search space, the fitness function value calculated by the endpoints in the structure should be evaluated as

$$\mathbf{f}(\mathbf{X}_A) < \mathbf{f}(\mathbf{X}_B) \tag{8}$$

Then, the coordinate computation formula for segmentation point $\mathbf{X}_C$ can be written as

$$\mathbf{X}_C = \mathbf{X}_A + F_p/F_{p+1}(\mathbf{X}_B - \mathbf{X}_A) \tag{9}$$

C. FIBONACCI BRANCH SEARCH OPTIMIZATION ALGORITHM

The FBS optimization algorithm introduced in this section is based on a framework that is built around the concept of endpoints and segmentation points in a basic structure. Combining with the basic structure, the process of searching for a global optimum solution, which can also be regarded as establishing a search element in FBS, is divided into two stages: the local optimization process and the global search process, which are the two corresponding interactive rules. Let G denote the point sets of the objective function to be searched in the current processing phase, and set $|G|_{num} = F_p, i = 1, 2, \cdots, N$. $|\cdot|_{num}$ represents the total number of points in the set, where N is the depth of the Fibonacci branch. The fitness values of the endpoints $\mathbf{X}_A$ and $\mathbf{X}_B$ are initialized using the interactive optimization rules; then, the segmentation points $\mathbf{X}_C$ can be obtained from equation (9). By comparing the fitness values of each point in the structure, we can obtain the best fitness value corresponding to the closest optimal solution. In the next optimization phase, the optimal point with the best fitness value is provided in the first position of the set, and the points corresponding to a suboptimal fitness are arranged below the optimal point in order from best to worst. Throughout the above operations, the point set G can be updated in every optimization phase towards growing the Fibonacci branch and global optimization in the search space simultaneously.

The two interactive search rules of FBS in the optimization stage are summarized as follows:

Rule One: Let us consider the endpoints $\mathbf{X}_A$ and $\mathbf{X}_B$ in the structure, defined by

$$\{\mathbf{X}_A\} = \mathrm{G}_p = \{\mathbf{X}_q | q = [1, F_P]\} \tag{10}$$
$$\{\mathbf{X}_B\} = \left\{\mathbf{X} | \mathbf{X} \in \prod_{f=1}^{D}\left[X_{lb}^f, X_{ub}^f\right]\right\} \tag{11}$$

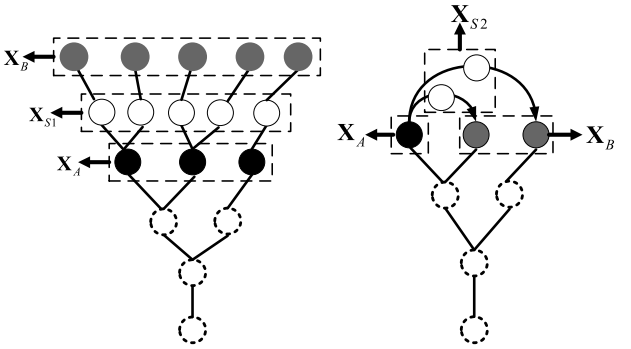
where G_p is the search point space set in the pth iteration, $\mathbf{X}_q$ are the points in set G_p, and q is the sequence number lying on the interval between 1 and the pth Fibonacci number. $\mathbf{X}_A$ take all the points from G_p of the pth iteration. The other endpoints $\mathbf{X}_B$ take random points in the search space, where the number of $\mathbf{X}_B$ is equal to F_P. D is the dimension of the points, and X_{ub}^f are the upper and lower bounds of the search points. Given that $\forall\, \mathbf{X} \in \{\mathbf{X}_B\}$, the component x of the vector $\mathbf{X}$ is a random variable that satisfies a uniform distribution over the interval $[X_{lb}, X_{ub}]$, and the probability distribution of the component can be written as

$$P(x) = U(X_{lb}, X_{ub}) = \frac{1}{X_{ub} - X_{lb}} \tag{12}$$

Using the given endpoints $\mathbf{X}_A$ and $\mathbf{X}_B$, we can determine the segmentation points $\mathbf{X}_{S1}$ in the first global search stage by equation (11).

Rule Two:

Suppose that $\mathbf{X}_{\text{best}}$ is the optimal solution corresponding to the best fitness value of the search space in the current



(a) The first global searching stage (b) The second local optimization stage

FIGURE 3. The process of building the Fibonacci branch for the global optimization.

iteration containing the endpoints and the segmentation points generated from rule one, as given by

$$\mathbf{X}_{\text{best}} = BEST\ (\mathbf{G}_P) \tag{13}$$

where $BEST\ (\cdot)$ represents the best solution of the set at the pth iteration.

Then, we set the endpoints $\mathbf{X}_A = \mathbf{X}_{\text{best}}$ and obtain the following:

$$f\ (\mathbf{X}_A) = \min\left\{f\left(\mathbf{X}_q\right), q = [1, F_P]\right\} \tag{14}$$

$$\mathbf{X}_B = \left\{\mathbf{X}_q | \mathbf{X}_q \in \mathbf{G}_P \wedge \mathbf{X}_q \neq \mathbf{X}_A\right\} \tag{15}$$

Hence, the segmentation points $\mathbf{X}_{S2}$ in the second local optimization stage can be determined based on the endpoints defined by (14) and (15) using the computing formula of the segmentation points.

From the two above-mentioned interactive search rules, new points including the endpoints $\mathbf{X}_A$ and $\mathbf{X}_B$ and segmentation points $\mathbf{X}_{S1}$ and $\mathbf{X}_{S2}$ are generated in the two optimization stages, and the total number of the points is $3F_P$. Evaluating the cost functions at the new points determines their fitness. These points are sorted from best to worst based on their fitness value. The population size of the search points is chosen as the Fibonacci series; thus, the top best F_{P+1} sets of these points are to be saved, and the remaining $3F_P - F_{P+1}$ points need to be dropped. After this procedure, the sets of the search space in the current pth iteration are renewed from the saved points, e.g., the saved points form a new set $\mathbf{G}_{P+1}$ in the search space for the next iteration.

The two stages for building the Fibonacci branch for the global optimization in space are shown in Figure 3.

As shown in Figure 3, the depth of the Fibonacci branch layer illustrated in the figure is initialized as expected, and the number of points in each branch layer remains in the sequence of Fibonacci numbers. The white dashed circle in the figure represents the search point set of the previous iteration, the black solid circle denotes the endpoints $\mathbf{X}_A$ in the current iteration, and the global random endpoints $\mathbf{X}_B$ are represented as grey solid circles. Figure 3(a) shows the first global search stage of the global optimization process. The segmentation points $\mathbf{X}_{S1}$, which are represented by solid-white-lined circles, are constructed based on the global random points and $\mathbf{X}_A$. The second local optimization stage is illustrated in Figure 3 (b). Incorporating the best fitness points $\mathbf{X}_A$ in the search space of the current iteration and the remaining endpoints of $\mathbf{X}_B$, the new segmentation points $\mathbf{X}_{S2}$ can be obtained by following the iteration rules. The fitness values of $\mathbf{X}_A$, $\mathbf{X}_B$, $\mathbf{X}_{S1}$ and $\mathbf{X}_{S2}$ are to be evaluated, and the best F_{P+1} solutions found with the optimum objective function evaluations need to be saved.

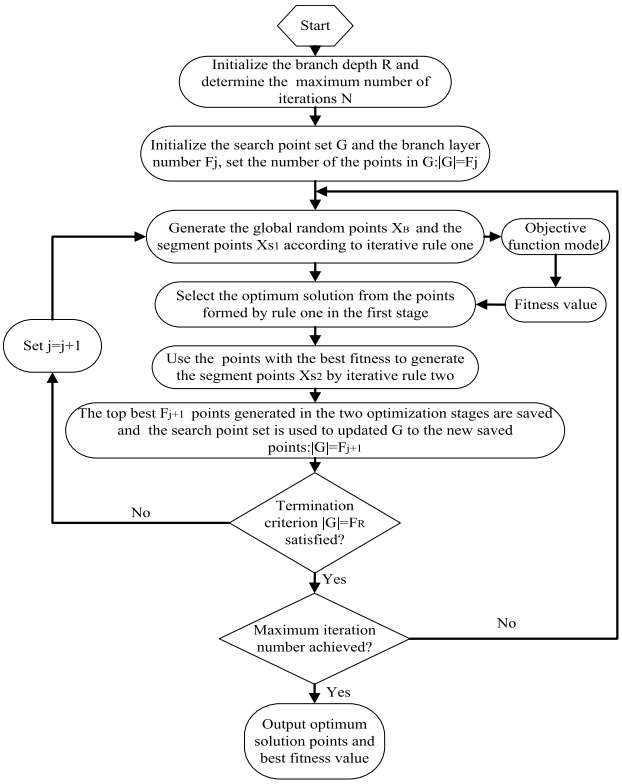


FIGURE 4. Flowchart of the Fibonacci branch search optimization method.

D. FLOWCHART OF THE PROPOSED FBS AND COMPLEXITY ANALYSIS

The basic flowchart of the general procedures for the specific implementation of the FBS algorithm is shown in Figure 4, and the corresponding pseudo-code is presented in Algorithm1.

The proposed above-mentioned FBS algorithm utilizes interactive global search and local optimization to obtain the global optimal solution; it can thus take advantage of global randomness and local convergence to eliminate local minima. The generated points in each layer of the branch depend on the optimum elements with the best fitness value in the previous branch layer, which can ensure that the global optimum solution in the search space is obtained. The search space is reduced as the iterative process proceeds, and the reducing rate is not fixed, therein decreasing from a finite designated random rate to zero due to the related search randomness

Algorithm 1 Pseudo-Code of the Fibonacci Branch Search Algorithm

Initialization

1. Determine the maximum iteration number (N) and the depth of the Fibonacci branch (R)

Iteration

2. **While** predefined maximum number of iterations is not reached

3. **While** the termination criterion $|G| = F_R$ does not satisfy the search and optimization phases

4. Choose the search point set G in the defined space and set the initial number of branch layers F_j

5. Calculate the shortening fraction $\frac{F_j}{F_{j+1}}$ and generate $\mathbf{X}_B$ and $\mathbf{X}_{S1}$ according to rule one

6. Measure the fitness of these points based on the objective function model and perform the analysis to identify the optimal points with the best fitness value

7. Take the best points and $\mathbf{X}_B$ in the current point set to generate $\mathbf{X}_{S2}$ from equation (7) of rule two

8. Save the top F_{j+1} points and update the set G to the new saved points for the next branch layer iteration

9. Set $j = j + 1$ and go to line 3

10. **End while**

11. Save the optimum solution points of the current iteration in the search set space and go to line 2

12. **End iteration and output the optimal points**

property of FBS in the optimization space. Once the optimal solution is found, the space shortening process is stopped.

An effective way to analyse the proposed FBS algorithm is demonstrated here in terms of the computational complexity. The implementation of the FBS is accomplished by the generation of search elements and search branches, therein considering a total number of search elements in the optimization space equal to N_s and the number of components needed for comparison in the search elements set to N_c. Then, the resulting computational complexity of the proposed FBS for the optimization operation is $O\left(C_f N_s \left(N_c\right)^2\right)$, where $C_f = \sum_{j=1}^{N} F_j$ is the ultimate total of the components of the search element in the Fibonacci search branch at the maximum number of iterations.

From the above complexity analysis of FBS, we find that the computational complexity of the proposed optimization algorithm is mainly dependent on the size of the search element and the maximum number of iterations.

IV. FBS-BASED BEAMFORMER DESIGN AND IMPLEMENTATION OF ABF WITH PROPOSED FBS

As explicitly described previously, ABF is an effective technique used to mitigate interference and improve the overall SINR performance by altering the radiation pattern of an antenna array. However, typically, the low nulling levels toward multiple interference sources and non-globally optimal weight vector are two major drawbacks of the MVDR beamforming technique [28]. Therefore, the proposed FBS algorithm is applied to ABF with ULA in this section to demonstrate the high performance of this technique.

A. ABF MODEL INTEGRATED WITH FBS

THE description of the beamforming model was given in section 2. Each adaptive beamformer in the model aims at calculating the complex weight vector that satisfies the requirement of maximizing the output SINR. In this work, the proposed FBS incorporated into the beamforming model will extract the best array excitation weights by maximizing the objective function based on the SINR.

Eq. (16) indicates how the ABF design problem is set up for the FBS-based optimization.

$$X_{mF_i} = w_m, m = 1, 2 \cdots, M, F_i \leq F_R \tag{16}$$

where w_m is the mth excitation weight of the array element, X_{mF_i} is the F_ith search point in the mth dimension of the ith branch layer, M is the total number of ULA, and R is the depth of the branch. The weights are interpreted here as the optimization points of the Fibonacci branch in the search space. The initial population of the weights is determined by the layer I with F_I search points, and the weight vectors of the entire population generated as the search points in the first layer can be written in the following format:

$$\mathbf{W}_{MF_I} = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1F_I} \\ w_{21} & w_{22} & \cdots & w_{2F_I} \\ \vdots & \vdots & & \vdots \\ w_{M1} & w_{M2} & \cdots & w_{MF_I} \end{bmatrix} \tag{17}$$

where $\mathbf{W}_{MF_I}$ is a weight vector containing F_I agents with M sensors representing the dimension of the search points, e.g., every agent will have M weight elements. F_I is the Ith Fibonacci series chosen as the total number of points at the first layer of the Fibonacci branch. Each complex weight of $\mathbf{W}_{MF_I}$ in the array element has an amplitude and a phase. The newly proposed FBS optimization algorithm is deployed to adjust the current amplitude and phase coefficients of the weight factor to provide the maximum beam pattern towards the desired signals as well as generate deep nulls toward the undesired signals to achieve the maximum SINR. Thus, the optimization formulation process of the ABF problem will attempt to maximize the fitness function constructed from the perspective of the SINR accordingly for the calculation of the complex excitation weight using FBS. In addition, the fitness function designed in this text can be written in the following form:

$$Fitness\_Function\left(\mathbf{w}\right) = \frac{P_d}{\sum_{i=1}^{Q} P_i + P_N} \tag{18}$$

where

$$P_d = \frac{1}{2} E\left[\left|\mathbf{w}^T \mathbf{x}_d\right|^2\right] \tag{19}$$

$$P_i = \frac{1}{2} E\left[\left|\mathbf{w}^T \mathbf{x}_i\right|^2\right] \tag{20}$$

Are the power of the desired signal and the power corresponding to the ith interference source, respectively; P_N is the noise power; and $\mathbf{x}_d$ and $\mathbf{x}_i$ denote the desired signal and interference component of the received signal, respectively, in equation (1).

Then, the design objective function (18) can properly be stated in the following form:

$$Fitness\_Function\,(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{R}_d \mathbf{w}}{\mathbf{w}^T \sum_{i=1}^{Q} \mathbf{R}_i \mathbf{w} + \sigma_{noise}^2 \mathbf{w}^T \mathbf{w}} \tag{21}$$

where $\mathbf{R}_i$ and $\mathbf{R}_d$ are the covariance matrix of the ith interference and desired matrix, respectively. The noise variance is calculated from the value of the signal-to-noise ratio (SNR) in dB as follows:

$$\sigma_{noise}^2 = 10^{-SNR/10} \tag{22}$$

By maximizing Eq. (21), the optimal excitation weight corresponding to the minimum level of the interference sources but with the desired user gain of the beamformer can be achieved by the proposed FBS algorithm. In addition, the optimum performance of the weight corresponding to the maximum SINR can be evaluated based on the best fitness account. The next section of this paper provides a brief description of the implementation steps for extracting the weight using the Fibonacci branch search method.

B. IMPLEMENTATION FLOWCHART OF FIBONACCI BRANCH SEARCH FOR ABF

In this subsection, in light of the results previously described in detail, an optimization scheme for the ABF problem combined with the FBS is presented to increase the maximum power of the target signal and generate deep nulls for the interference sources. The basic idea of the design of such an adaptive beamformer is to utilize the global search and local convergence capabilities of the efficient search algorithm to reduce the local minimum problem of the solution to the weight vector and satisfy the requirements on the created multi-objective optimization problem from (21) to obtain the maximum SINR. The general procedures for the implementation of the Fibonacci branch search method with application to ABF are presented in Figure 5, where the key steps are briefly described below.

(a) Choose the depth R of the Fibonacci branch to determine the population F_R of the top branch layer and set the maximum number of iterations of the optimization process.

(b) Initialize the population of the first branch layer F_j and determine the dimension of the weight vectors acting as search points in space according to the number of elements of the ULA. Additionally, define the amplitude search space of the weight within [0, 1] and limit the range of the weight phase to $[-\pi, \pi]$.

(c) Assign the values to the amplitude and phase of the weight elements inside the search space for constructing the

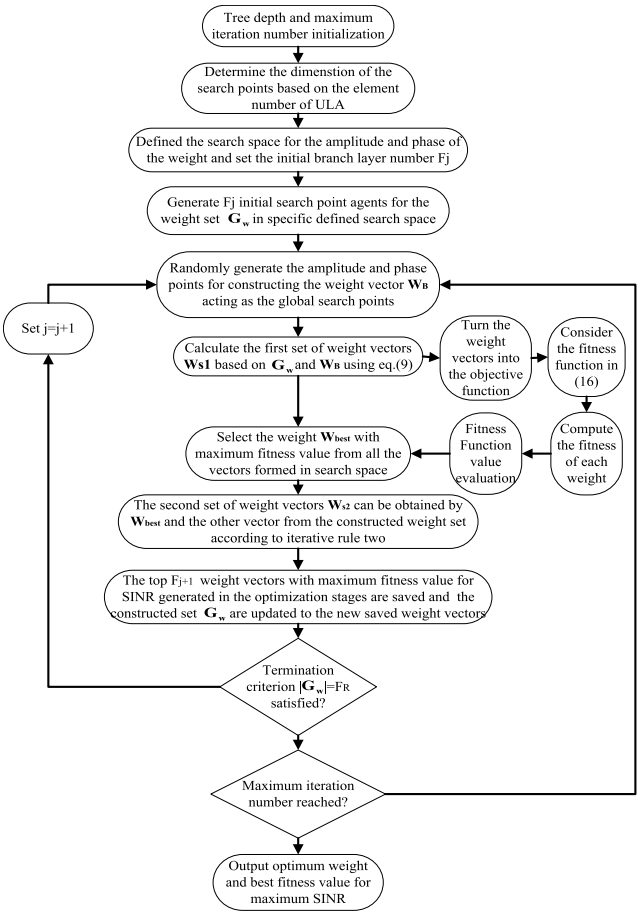


FIGURE 5. Flowchart illustration of the optimization process for ABF using the Fibonacci branch search method.

initial population of the weight vector set $\mathbf{G_w}$. The weight element in the vector set $\mathbf{G_w}$ constructed using the amplitude and phase can be expressed as follows:

$$\mathbf{w}_{jd} = rand\,[0, 1] \cdot e^{j \cdot rand[-\pi, \pi]} \tag{23}$$

where the generated weight $\mathbf{w}_{jd}$ represents the dth dimension of the jth individual in the population, $d \in [1, M\ (dim\ension\ of\ the\ search\ space)]$, $j \in \left[1, F_j\ (population\ of\ the\ vectors)\right]$, and $rand\,[\cdot]$ denotes random value generation in the range.

(d) Take the random amplitude and phase values in the search space for generating the F_j population of the weight vectors $\mathbf{w}_B$, which acts as the global search points.

(e) Set the vector elements of $\mathbf{G_w}$ and $\mathbf{w}_B$ as the endpoints in equation (9) and compute the first set of weights $\mathbf{w}_{S1}$ according to the iterative rule one.

(f) Calculate the fitness with the object function of (18) using $\mathbf{G_w}$, $\mathbf{w}_B$ and $\mathbf{w}_{S1}$; then, give the evaluated results to the values to find the best weight vector $\mathbf{w}_{best}$ with the maximum fitness value among all the vectors in the space.

(g) Generate the second set of weight vectors based on the best weight vector $\mathbf{w}_{best}$ and the other weights from the weight space set using equation (9) in iterative rule two.

(h) Select the top F_{j+1} best weight vectors based on the maximum fitness value in the optimization process, and these best weights are selected to compose the new population of the set $\mathbf{G_w}$.

(i) Check the termination criterion $\mathbf{G_w} = F_R$. If the termination criteria are not satisfied, then increment j and go to Step (d); otherwise, stop.

(j) If the maximum number of iterations is not reached, repeat the algorithm from step (d); else, report the output results and terminate.

V. SIMULATION RESULTS

A. VERIFICATION OF THE PROPOSED FBS

To validate and analyse the efficiency and effectiveness of the proposed FBS, the algorithm is verified in the following regards in the simulation experiments:

(1) The accessibility to the global optimum of the proposed search algorithm for multimodal functions with numerous local optima is revealed by the location history of the search points in the direction of the optimal point.

(2) The convergence of the FBS is proved and discussed based on the presented gradient of the iteration curves, which demonstrated the convergence rate and average fitness of the chosen benchmark function.

(3) The optimization precision of the solution and other relevant optimization assessment aspects of the proposed algorithm are tested on eight representative standard benchmark functions, and the results are compared with typical heuristic algorithms.

The effectiveness can be validated by comparing with classical optimization methods, including the recently published GSA algorithm [13], APABC [20], the artificial flora (AF) algorithm [29], and the bat algorithm (BAT) [30]. The details of the parameter settings for each heuristic algorithm used in the experiments are given in Table 1.

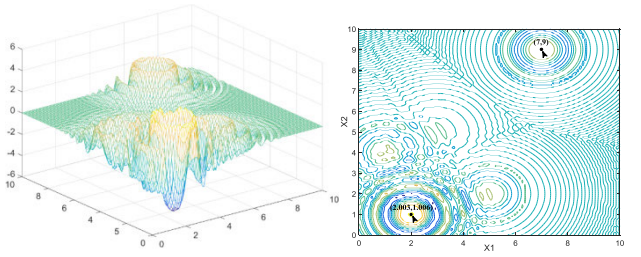
All the experimental tests have been implemented in Matlab R2015a and run on the same PC with Intel(R) Core(TM) CPU 2.8 GHz.

TABLE 1. Reference parameters of the algorithms briefly used in this study.

Particle Swarm Optimization (PSO)		Gravitational Search Algorithm (GSA)		BAT Algorithm (BAT)	
Population size	Np=20	Agent number	Ng=50	Loudness reduction	a=0.8
Cognitive ratio	c1=2	Descending coefficient	a=20	Population size	Np=20
Social coefficient	C2=2	Initial gravitational constant	Go=0.5	Pulse reduction rate	y=0.8
Inertia weight	0.4~0.9	Initial constant	Ko=50		
		Final constant	Kf=1		
Artificial Flora (AF)		**Adaptive population artificial bee colony (APABC)**		**Fibonacci Branch search (FBS)**	
Original plants number	Mp=1	Colony size	Cs=20	Nested branch depth	2
Learning coefficient	c1=0.8	Onlooker bees percentage	Lp=50%	Total branch depth	6
Learning coefficient	c2=1.2	Scout bees	Sb=1	Search Space	[Min,Max]
Maximum branching number	M=50	Population size parameter	SNmax=25 SNmin=15		

TABLE 2. Langermann benchmark function.

Extreme point	Extreme point 1	Extreme point 1
Evaluation in [31]	global optimum solution	local suboptimal solution
Langermann function	f_L 2.003,1.006 $=-5.1612$	f_L 7,9 $=-3$



(a) Three dimensional of Langermann function

(b) Contour plots of Langermann function

FIGURE 6. Three dimensional and contour plots for the Langermann function.

B. VERIFICATION OF THE PROPOSED FBS

1) THE LOCATION HISTORY OF THE SEARCH POINTS IN FBS FOR LANGERMANN FUNCTION

In this section, the global optimization ability of the proposed FBS is demonstrated by employing the location history of the search points during the optimization process for locating the global optimum solution rather than trapping into local optimization of the benchmark example, with results compared against the metaheuristic PSO algorithm. The benchmark function chosen in this section is the Langermann function with several known local optimal points and one global optimum solution point; this function is taken from [31], [32] and is summarized in Table 2. As shown in Table 2, two typical extreme points exist in the function. Extreme point 1 shown in the table is the global optimum solution, and the extreme point 2 is the local suboptimal solution. The three-dimensional Langermann function and contour plots are illustrated in Figure 6.

The performance of the proposed FBS in terms of the movement trajectory of the search points scattering around the best solutions and converging to the optimal point in the search space for the Langermann function is illustrated in Figure 7 (b). This figure shows that the FBS model is able to simulate the position history of the search points in the three-dimensional and trajectory contour plots over different iterations. To verify the results, we compare our algorithm to PSO in the same manner and provide the results in Figure 7(a). The initial position of the search points in both FBS and PSO is set at extreme point 2.

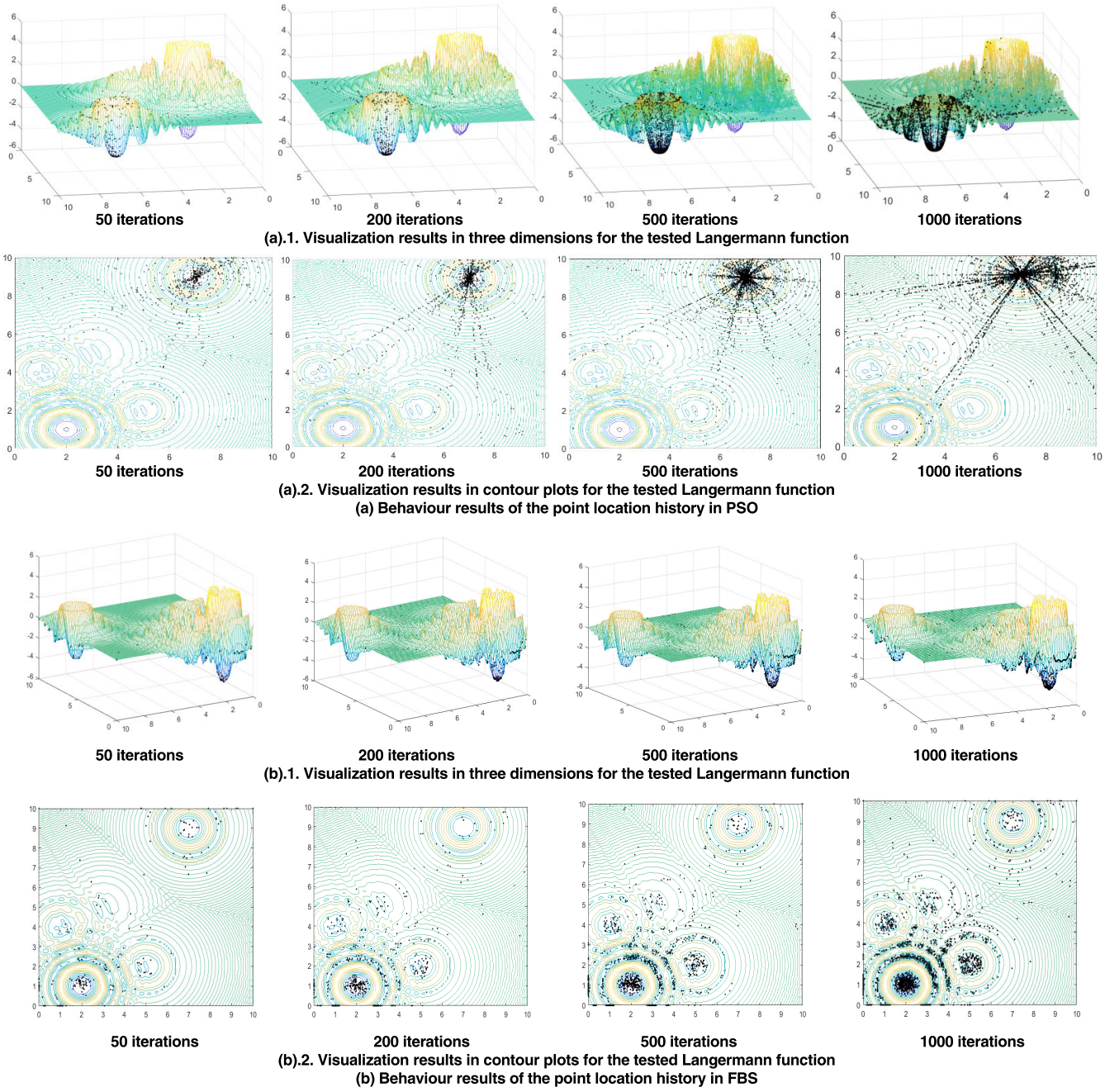


FIGURE 7. Location history of the search points in the three-dimensional and contour plots of the Langermann function over for different iteration numbers.

As shown in Figure 7, the search points tend to explore the promising regions of the search space and cluster around the global optima, eventually producing a multimodal Langermann pattern. From the results depicted in Figure 7(a), we see that, as the number of iterations increases, the points of the PSO algorithm gradually cluster around extreme point 2 and proceed towards the local optima; in addition, very few particles enter the region near the global optimum extreme point 1, providing further evidence that PSO inherently suffers from local optima entrapment and stagnation in the search space. Under the same conditions, it can be seen from the trajectories and 3D version of the search point, as shown in Figure 7(b), that although the Langermann function is non-symmetric and multimodal with different levels of peaks, finding its global optimum is challenging due to the many local minima in the search space. Remarkably, FBS is able to remove itself from the initial local optimum at extreme point 2 and jump out of the trapped solution in a local optimum point assisted by the global random searching. It is evident from the location history of the search points during the process of converging towards the global optima that the points grow toward the optimal point from the area of initialization, tending to

TABLE 3. The details of multimodal benchmark functions (D: dimensions).

No.	Function	Formulation	D	Search Range	Global Optima
F1	Griewank	$\sum_{i=1}^{D}\frac{x_i^2}{4000}-\prod_{i=1}^{D}cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	10	[-600,600]	0
F2	Rastrigin	$\sum_{i=1}^{D}\left(x_i^2-10cos(2\pi x_i)+10\right)$	10	[-5.12,5.12]	0
F3	Michalewicz2	$-\sum_{i=1}^{D}sin(x_i)sin\left(\frac{ix_i^2}{\pi}\right)^{20}$	10	$[0,\pi]$	-1.8013
F4	Rosenbrock	$\sum_{i=1}^{D}100\left(x_{i+1}-x_i^2\right)^2+\left(x_i-1\right)^2$	10	[-2.048,2.048]	0
F5	Ackley	$-20exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{D}x_i^2}\right)-exp\left(\frac{1}{n}\sum_{i=1}^{D}cos(2\pi x_i)\right)+20+e$	10	[-32,32]	0
F6	Schwefel	$418.9829\times D-\sum_{i=1}^{D}x_i\,sin\left(\lvert x_i\rvert^{\frac{1}{2}}\right)$	10	[-500,500]	0
F7	Weierstrass	$\sum_{i=1}^{D}\left(\sum_{k=0}^{kmax}\left[a^k\,cos\left(2\pi b^k\left(x_i+0.5\right)\right)\right]\right)-D\sum_{k=0}^{kmax}\left[a^k\,cos\left(2\pi b^k\left(x_i\times 0.5\right)\right)\right]$ $a=0.5,\ \ b=3,\ \ k\,max=20$	10	[-0.5,0.5]	0
F8	Salomon	$-cos\left(2\pi\sqrt{\sum_{i}^{D}x_i^2}\right)+0.1\sqrt{\sum_{i}^{D}x_i^2}+1$	10	[-100,100]	0

gradually scatter around extreme points and moving toward the best solutions in the search space in both 2D and 3D spaces over the course of the iteration process. More than half of the agents had already approached the global optimum valley after the first 50 iterations and began converging to the optimum. As the iteration process continued, there were more agents aggregating at the extreme points and scattering around the extreme point, strongly attracting towards the global optimum target region. Eventually, the search points found the global optimum and converged towards the global optima. This can be discussed and reasoned considering the global randomness concepts introduced by the endpoints $\mathbf{X}_B$, which are generated in rule one of FBS. Furthermore, the convergence of FBS is guaranteed by the local exploitation optimization ability emphasized in the other endpoints $\mathbf{X}_A$ of the proposed algorithm. Since the global random points tend to move from a less fit universe to a more fit universe through the global search in space, the best universe is saved and transferred to the next iteration. Consequently, these behaviours and abilities help the FBS to not become trapped in local optima and quickly converge to the target point in the optimizationP process.

The above simulations and discussions demonstrate the effectiveness of the FBS algorithm in finding the global optimum in the search space. The convergence performance and speed of obtaining the global optima of the proposed algorithm when employing a set of mathematical functions are to be investigated in the next sections.

2) CONVERGENCE PERFORMANCE OF THE MULTIMODAL FUNCTION

To confirm the convergence behaviour of the proposed algorithm, in this subsection, we provide the convergence curves of the objective fitness value of the typical benchmark functions obtained for the best solutions so far in each iteration. A large set of complex mathematical benchmark functions to be tested is listed in Table 3. These functions have many local optima, which make them highly suitable for benchmarking the performance of the metaheuristic algorithms in terms of optimization and convergence performance. The illustrated results are compared against those of the PSO, GSA, AF, BAT, and APABC metaheuristic algorithms on the same set of multi-dimensional numerical benchmark functions. The properties and formulas for these functions are presented below.

Figure 8 presents the convergence characteristics in terms of the best fitness value of the median run of each algorithm for the test functions. Comparing the results and the convergence graphs, among these six algorithms, we observe that the proposed algorithm has a good global search ability. FBS achieved better and more accurate fitness results on most of the multimodal groups than the comparison algorithms; FBS surpassed all other algorithms on functions 1, 2, 4, 5 and 7 and achieved significantly improved results on functions 5 and 7. The convergence maps also show that FBS always converges faster than PSO, GSA and BAT but was slightly slower than APABC and AF; however, the improved performance is a good trade-off with the slight sacrifice of converge speed.

The PSO and GSA algorithms showed poor performance on complex problems since they miss the global optimum basin when approaching the optimal fitness. Schwefel's function is a good example since it traps the two algorithms in local optima, whereas the FBS successfully avoids falling into the deep local optimum, which is far from the global optimum. The APABC and AF algorithms perform much better than the other comparison algorithms, with success rates exceeding 60% on most

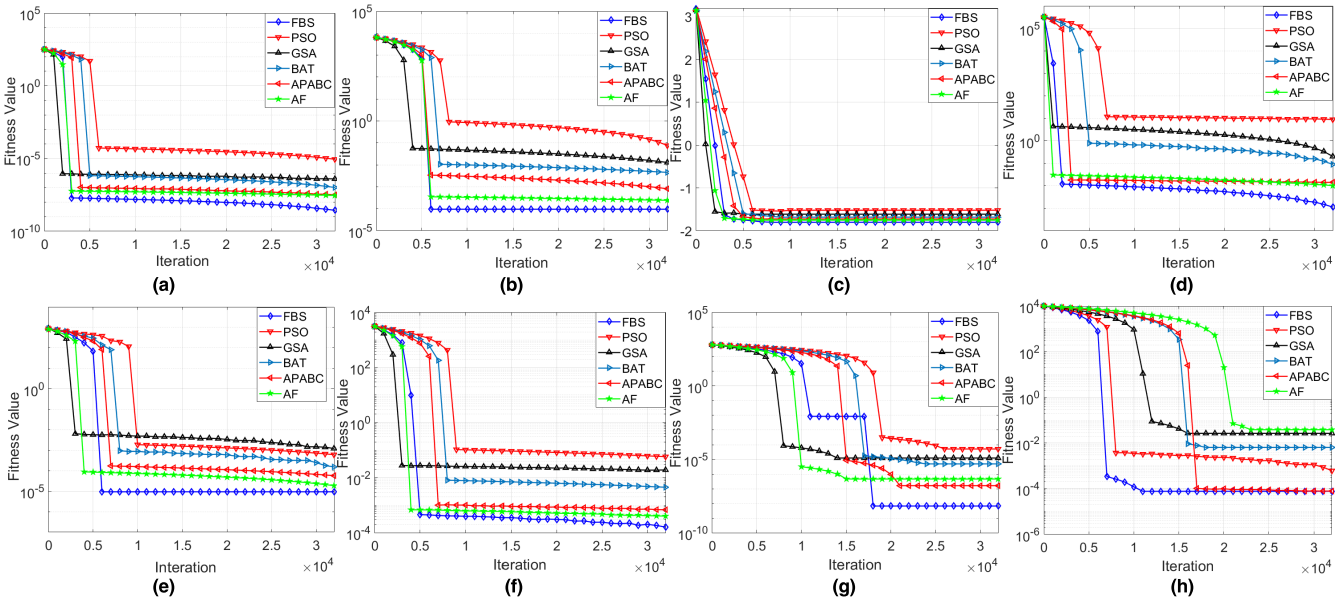


FIGURE 8. Convergence behaviour of the FBS and other optimization algorithms on the 10-D benchmarking functions F1- F8. (a) F1: Griewank; (b) F2: Rastrigin; (c) F3: Michalewicz2; (d) F4: Rosenbrock; (e) F5: Ackley; (f) F6: Schwefel; (g) F7: Weierstrass; (h) F8: Salomon.

problems. Note that the FBS algorithm in the graphs converges to a global optimum solution with fewer fitness evaluations and terminates after no more than 5000 iterations on functions 1, 3 and 4.

These figures also prove that FBS not only improves the accuracy of the initially approximated optimum but also achieves increased convergence speed over the course of the iteration process, making it converge faster than most of the other algorithms. The global random property and space region shortening fraction guarantees a satisfactory convergence speed. The proposed FBS combines a global search method and local optimization strategy to yield a balanced performance that achieves better fitness and faster convergence. The global optimization ability and convergence speed are two crucial parameters of real-time applications, such as ABF systems; thus, the FBS is highly suitable and efficient for ABF.

3) MINIMIZATION RESULT OF THE TESTED BENCHMARK FUNCTIONS

In this subsection, experiments are conducted on the suite of multimodal functions illustrated in Table 3 to evaluate six optimization algorithms, including the proposed FBS. All the test functions are to be minimized, and the relevant information can be found in [33] and [34] for the standard benchmark functions. The dimensions of the selected benchmarking problems F1-F8 are set to 10. Every algorithm runs 1000 times independently to reduce the statistical variance and achieve reliable results.

The statistical results considering the average value and the standard deviation function fitness value as well as the success rate needed to obtain an acceptable solution are summarized in Table 4. In the results shown in Table 4, the mean value is smaller, and the performance of the algorithm is better. The standard deviation is lower, and the stability of the algorithm is stronger. As seen, for most benchmark data sets, both the average value and the standard deviation calculated by the FBS are smaller than those of the other algorithms. In addition, the proposed algorithm outperforms all other algorithms on functions 1, 2, 3, 5, 6, and 7 and especially significantly improves the results on functions 2 and 3. When the other algorithms find their own best fitness for these functions, the proposed FBS could still search for the better fitness closest to the optimal value. The relatively well-performing algorithm AF achieved similarly optimal results to the FBS on functions 1, 5 and 7, and both algorithms significantly outperform the other variants on this problem. The APABC also performs well on multimodal problems. This algorithm performed similarly to the FBS on functions 1 and 5. However, the FBS performs better on substantially more complex problems when the other algorithms miss the global optimum basin.

As a result, in terms of global search ability and optimization stability for the benchmarking function, the proposed FBS outperformed all other heuristics algorithms on the tested functions. Additionally, this table illustrates that FBS, in comparison to the other algorithms, obtains the highest percentage and accuracy in reaching the acceptable solutions on test functions F1, F2, F3, and F5. For the mean reliability of test functions F1, F3, F5, and F6, FBS shows the highest reliability, with a 100% success rate and smallest average errors. This superior performance is due to the FBS's interactive updating rule. With the new updating rule and global randomness, different dimensions may learn from different exemplars based on the historically optimal search experience, and the FBS explores a larger search space

TABLE 4. The comparative and statistical results for benchmarking function problems F1-F8.

Method	F1.Griewank			F2.Rastrigin			F3.Michalewicz2			F4.Rosenbrock		
	Mean	Stev	SR	Mean	Stev	SR	Mean	Stev	SR	Mean	Stev	SR
AF	1.73E-08	5.38E-08	100%	2.43E-04	6.93E-05	70%	-1.7725E+00	9.16E-06	89%	1.03E-2	4.75E-3	21%
APABC	4.87E-08	4.63E-07	100%	8.18E-04	3.23E-04	59%	-1.7232E+00	8.93E-06	72%	1.39E-02	9.47E-03	16%
BAT	9.99E-08	2.51E-07	96%	4.55E-03	2.64E-03	42%	-1.6828E+00	3.46E-05	53%	8.37E-2	8.88E-03	12%
PSO	5.27E-05	3.75E-05	55%	5.19E-02	1.29E-02	14%	-1.5202E+00	6.36E-04	27%	8.23E+00	1.90E+00	2%
FBS	2.54E-09	1.47E-08	100%	9.53E-05	2.17E-06	86%	-1.8013E+00	7.48E-06	100%	9.98E-03	3.27E-03	29%
GSA	8.43E-07	7.64E-06	87%	1.29E-02	4.37E-03	29%	-1.5738E+00	2.38E-04	45%	2.43E-01	9.34E-02	12%
Method	F5.Ackley			F6.Schwefel			F7.Weierstrass			F8.Salomon		
	Mean	Stev	SR	Mean	Stev	SR	Mean	Stev	SR	Mean	Stev	SR
AF	1.87E-05	3.84E-05	100%	3.84E-04	1.56E-04	63%	4.73E-07	8.87E-07	100%	1.92E-04	3.53E-04	72%
APABC	5.74E-05	1.57E-05	92%	6.15E-04	8.36E-05	59%	1.65E-07	9.27E-07	100%	3.28E-04	6.25E-04	64%
BAT	1.27E-04	7.34E-04	73%	4.53E-03	4.83E-04	46%	5.08E-06	1.46E-06	97%	6.43E-04	3.74E-04	51%
PSO	6.09E-03	7.28E-04	49%	5.87E-02	3.95E-03	23%	5.27E-06	6.09E-05	62%	5.74E-03	3.68E-03	29%
FBS	4.34E-06	5.74E-07	100%	2.53E-04	7.48E-04	74%	6.77E-09	4.34E-09	100%	7.74E-05	6.25E-06	85%
GSA	9.09E-04	4.37E-05	64%	1.85E-02	6.28E-03	35%	9.99E-05	5.74E-05	83%	2.67E-03	7.43E-03	42%

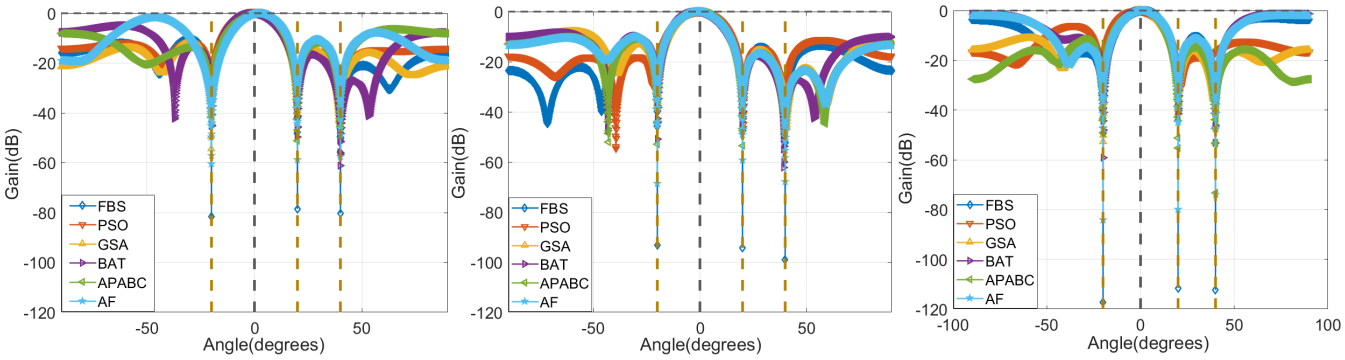


FIGURE 9. Comparison of the beampattern performance synthesized by the heuristic algorithms with different SNRs: (a) SNR = −30 dB; (b) SNR = −10 dB; (c) SNR = 10 dB.

than the other compared algorithms. Because of this, the FBS performs comparably to or better than many meta-heuristic algorithms on most of the multimodal problems studied in this section.

C. THE PERFORMANCE OF THE FBS IN ABF MODEL APPLICATION

To demonstrate the benefits of the FBS optimization with application to ABF, in this sub-section, several groups of simulation experiments are conducted using Matlab R2015b. The performance of the FBS-based ABF is evaluated from the following two simulation metric aspects: the beampattern performance of the adaptive antenna array and the steady-state output SINR of the beamformer system. A uniform linear array with an inter-element spacing of a half wavelength is considered in the simulation. The desired signal is in the form of a QPSK modulation mode with 0° incident azimuth angle, and three single-frequency interference sources with an interference-to-noise ratio (INR) of 10 are assumed to impinge on the antenna array from azimuth directions of 20°, 40°, and −20°. The incident angle of the signals corresponding to the elevation and azimuth in three-dimensional space and the array flow pattern used in this paper are a uniform linear array rather than a planar array; thus, the elevation of the different incident signals is the same for the receiving array. Therefore, the desired signal and jammers are all arranged at an elevation of 45°. The simulation environment includes an additive white Gaussian noise channel (AWGN).

The proposed FBS-based beamforming method is compared with the five previously mentioned conventional heuristics-based beamforming algorithms: 1) the GSA-based beamforming method [13], 2) the PSO-based beamforming method [12], 3) the AF-based beamforming method [29], 4) the APABC-based beamforming method [20], and 5) the BAT-based beamforming method [30]. The parameters of all the algorithms are set as in the previous section and as illustrated in Table 1. A total of 300 repetitions are implemented and then averaged to obtain each figure of the results.

1) BEAMPATTERN PERFORMANCE RESULTS

In this subsection, the effectiveness and applicability of the proposed FBS-based beamformer is investigated based on the power patterns formed by different types of metaheuristic-based beamformers. Three groups of

simulation cases conducted and analysed in terms of various simulation metrics are considered in this study. The first simulation example has a different number of interference sources, and the second case considers a different number of array elements. The performance with respect to the input SNR is studied in the third case.

a: SIMULATION CASE ONE: INPUT SNR EVALUATION

We first examine the beampattern performance synthesized by the FBS and compare it to the other algorithms in terms of input SNR in this case. A uniform linear array with 6 omni-directional antenna elements is considered in the simulation. To investigate the effect of the input SNR at different levels, we consider three sets of input SNR, i.e., $SNR = -30$dB, $SNR = -10$ and $SNR = 10$, to demonstrate the validity of our approach. Figure 9 shows the behaviour of the beampatterns synthesized by the weight vectors determined by the optimization algorithms under different SNRs. It can be seen from the figures that the weight vectors found by FBS could synthesize the inerratic beampattern with deeper nulling (with nulling level exceeding -70 dB) towards the interference sources compared to the other algorithms. The proposed FBS-based adaptive beamformer suppressed the jammers in all cases while maintaining the beampattern gain in the direction of the desired signal. The other algorithms are able to achieve satisfactory interference nulling performance for the higher SNR level, i.e., SNR $= 10$. As the SNR decreases, the comparison beamformers, especially the PSO-based beamformer, suffer from a performance degradation of the corresponding metaheuristic-based beampatterns, and the nulls do not align precisely with the interference sources. To more clearly illustrate the nulling degrees and nulling level of each method, the nulling results corresponding to Figure 9 are shown in Table 5. It can be seen from the table that the improvement of the nulling level by FBS, compared to PSO, which performs the worst among the compared methods in each case, is 46.79, 52.96 and 71.95 dB. FBS presents the best performance, and the proposed algorithm obtains the best performance, with a 50.17% improvement over AF, which demonstrates the worst performance of all the algorithms. This indicates that the proposed algorithm is more stable and finds better solutions with greater precision in ABF applications.

TABLE 5. Comparison of average nulling level toward interference sources for different SNR scenarios.

Method \ Scenario	SNR=-30dB	SNR=-10dB	SNR=10dB
FBS	-81.07dB	-95.29dB	-117.62dB
PSO	-34.28dB	-42.33dB	-45.67dB
GSA	-48.67dB	-52.67dB	-58.20dB
BAT	-50.29dB	-57.98dB	-59.25dB
APABC	-54.87dB	-62.72dB	-70.82dB
AF	-61.26dB	-67.66dB	-78.32dB

b: SIMULATION CASE TWO: ARRAY ELEMENT NUMBER INVESTIGATION

The beampattern performance with respect to the number of array elements was evaluated in the second case study. Adaptive beamformer arrays consisting of 6, 10 and 14 elements were considered in the simulation experiments. The input SNR was fixed at -5 dB. The proposed FBS, in comparison with the other metaheuristic algorithms, is applied to search for the optimum element phases and amplitude of the uniform linear array to achieve the target pattern by considering these three cases with different element numbers. The optimization results of the beampatterns are illustrated in Figure 10. The results depicted in Figure 10 show that the proposed FBS outperforms the other algorithms with the same number of array elements. This means that FBS provides a superior performance, especially compared to PSO, since it creates deeper nulls toward the interference direction while maintaining high power in the desired direction. As the number of elements increases from 6 to 14, the beampattern formed by the other methods, particularly for PSO and GSA, deteriorates and almost fails in steady state. Specifically, the performance disparity between the proposed algorithm and the compared algorithms is expected to increase further with a higher number of array elements. This is due to the increase in the number of elements, which results in an increase in the search dimension of the solution, which inherently increases the difficulty of the optimization problem. In addition, the global search optimization ability of the FBS algorithm is more suitable for the multi-dimensional solution of the large array elements in improving the beampattern performance. Table 6 confirms the superiority of the proposed method based on specific nulling levels and nulling degrees. FBS exhibits superior nulling characteristics compared to PSO, and improvements in the nulling levels of 57.73, 73.92 and 90.60 dB are achieved by using FBS in different array element number scenarios. APABC and AF show almost identical but marginally satisfactory nulling performances, while FBS achieves a better ratio, i.e., 46.43%, compared to AF in the best scenario. The results establish FBS's suitability as an interference mitigation algorithm.

c: SIMULATION CASE THREE: INTERFERENCE SOURCE NUMBER ASSESSMENT

To fully verify the beampattern performance, different types of scenarios with multiple interference sources were simulated to validate the proposed approach for ABF applications

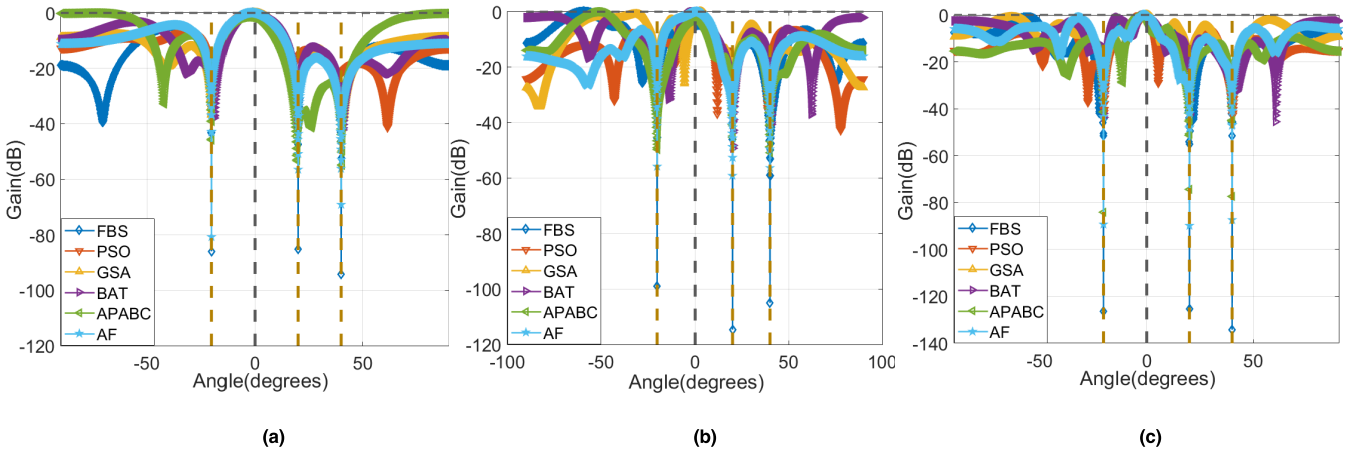


FIGURE 10. Comparison of the beampattern performance synthesized by the heuristic algorithms with different numbers of array elements: (a) six elements; (b) ten elements; (c) fourteen elements.

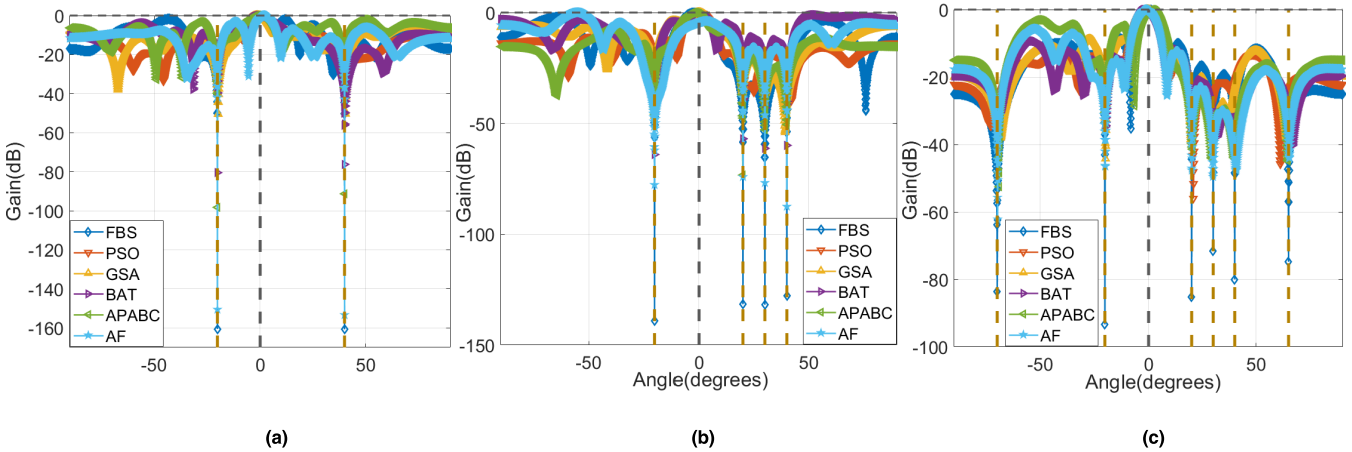


FIGURE 11. Comparison of the beampattern performance synthesized by the heuristic algorithms with different numbers of interference sources: (a) two interference sources; (b) four interference sources; (c) six interference sources.

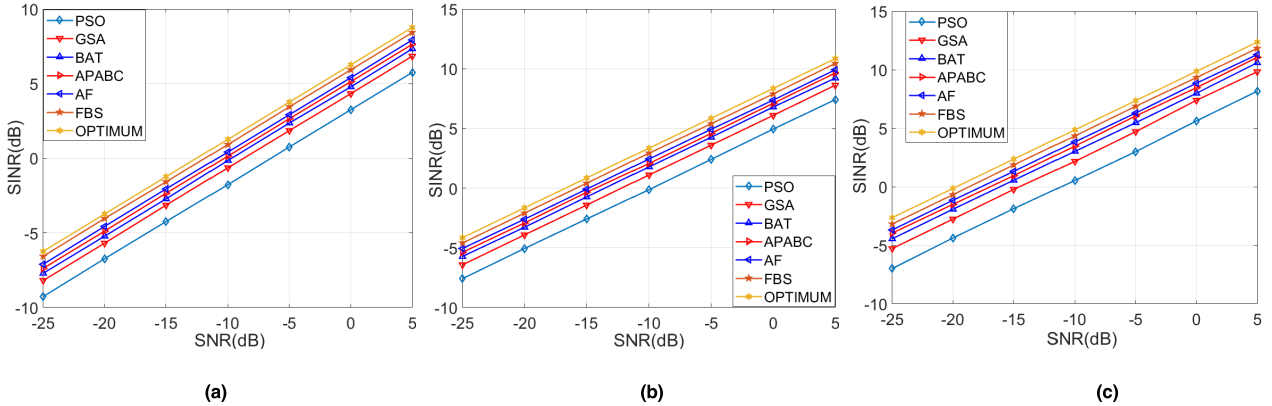


FIGURE 12. SINR versus SNR of the heuristic algorithms with varying array element numbers: (a) six elements; (b) ten elements; (c) fourteen elements.

in the last case. A uniform linear array of 12 omni-directional antenna elements was adopted in this subsection. All scenarios have one desired signal at $\theta_s = 0°$, while the number of interference sources changes in each scenario. The first two target interference sources are assumed to be impinging on the array from 20° and 40°, and the four received interference sources impinging at $\theta_i = \{20°, 40°, 20°, 30°\}$ are considered in the second scenario. Finally, the third scenario concerns six interference sources with incident angles $\theta_i = \{20°, 40°, 20°, 30°, 65°, 70°\}$. In each scenario, the FBS and other optimization algorithms are applied to find the optimal excitation vectors that produce a main lobe toward θ_s

TABLE 6. Comparison of average nulling level toward interference sources for different array elements scenarios.

Method \ Scenario	Six Elements	Ten Elements	Fourteen Elements
FBS	-87.28dB	-110.59dB	-131.44dB
PSO	-29.55dB	-36.67dB	-40.84dB
GSA	-39.01dB	-42.25dB	-49.63dB
BAT	-42.53dB	-49.73dB	-58.73dB
APABC	-57.03dB	-67.62dB	-82.95dB
AF	-61.09dB	-78.83dB	-89.76dB

TABLE 7. Comparison of average nulling level toward interference sources for different numbers of interference scenarios.

Method \ Scenario	Two interference	Four interference	Six interference
FBS	-161.02dB	-142.95B	-89.37dB
PSO	-28.45dB	-21.67dB	-23.29dB
GSA	-53.73dB	-49.65dB	-36.04dB
BAT	-79.91dB	-62.73dB	-39.93dB
APABC	-92.46dB	-69.40dB	-47.37dB
AF	-145.75dB	-77.81dB	-54.03dB

and nulls toward θ_i. The graphs shown in Figure 11 represent the optimized beampattern for all the scenarios studied here. All the radiation pattern results presented in Figure 11 show that the FBS-based beamformer is a robust technique capable of improving the radiation characteristics while outperforming conventional optimized beamformers regardless of the number of interference signals. The FBS-based beamformer exhibits a more prominent behaviour regarding the steering ability and increasing nulling levels in Figure 11 (c), i.e., the superiority of FBS in scenario 3 is more apparent than in the other two cases. When the number of interference sources increases, it is more difficult for the other optimization algorithms to enhance the null level in the interference direction. Therefore, these results demonstrate that the superior exploratory and exploitive properties of FBS applied to ABF have resulted in better beamsteering and interference mitigation performances in all three cases, especially under multiple interference sources. Table 7 presents the results of the average nulling levels determined by the different heuristics-based beamforming algorithms. FBS shows the best performance. The increase in the nulling level of FBS in comparison with PSO is 132.57, 121.28, and 66.08 dB. Even given the relatively high performance of the APABC and AF, Table 7 further evidences the FBS superiority over APABC and AF, i.e., the 10.48%, 83.72%, and 65.41% improvements over AF, which presents a suboptimal performance in all these cases. Thus, FBS demonstrates a better exploitative ability than the other compared algorithms in the ABF model.

2) OUTPUT SINR RESULTS

This section will illustrate different scenarios for system output SINR using FBS and other optimization algorithms for searching for suitable weight vectors to achieve the required SINR. Numerical simulations to investigate the SINR performance of the proposed algorithms in terms of useful metric are carried out. The first two cases are studied with different numbers of array elements and interference sources. The remaining case concerns the different INRs of input interference in front of a ULA system. The SINR performance of all the tested algorithms in the results of the figures is measured by the increasing input SNR value for various simulation conditions, and the SNR is assumed to vary from -25 dB to 5 dB (cantered at -10 dB) in 5 dB steps. All cases have one user at 0°, while the numbers of interference sources and elements as well as the input INR vary in each scenario.

a: SIMULATION CASE ONE: FOUR INTERFERENCE SOURCES WITH 20 DB INR CONSIDERING VARYING ELEMENT NUMBERS

The first case is two interference signals at 20° and 40° with 5 dB INR. The linear arrays are considered to be composed of 6, 10 and 14 elements. The results of the output SINR are illustrated in Figure 12. From the graphs, it can be noted that the proposed algorithm outperforms the other algorithms for all the array element scenarios and is able to achieve near-optimal performance over the entire range of input SNR values. The APABC and AF optimization algorithms yield suboptimal but higher values of SINR; FBS yields optimal SINR values consistently in all cases, and the improvements in the SINR by FBS, compared to the second-best-performing algorithm in these three scenarios, are 21.04%, 27.62%, and 31.73%. We can also observe that the performance difference in reaching the optimum weight vectors between FBS and the other algorithms increases with increasing array element number in each algorithm, which is consistent with the results in Section 5.2.1.2. This is due to the increasing search dimension of the weight vector solution in the array element. In light of the above, the global search capability of the proposed FBS algorithm improved the output SINR more so than the

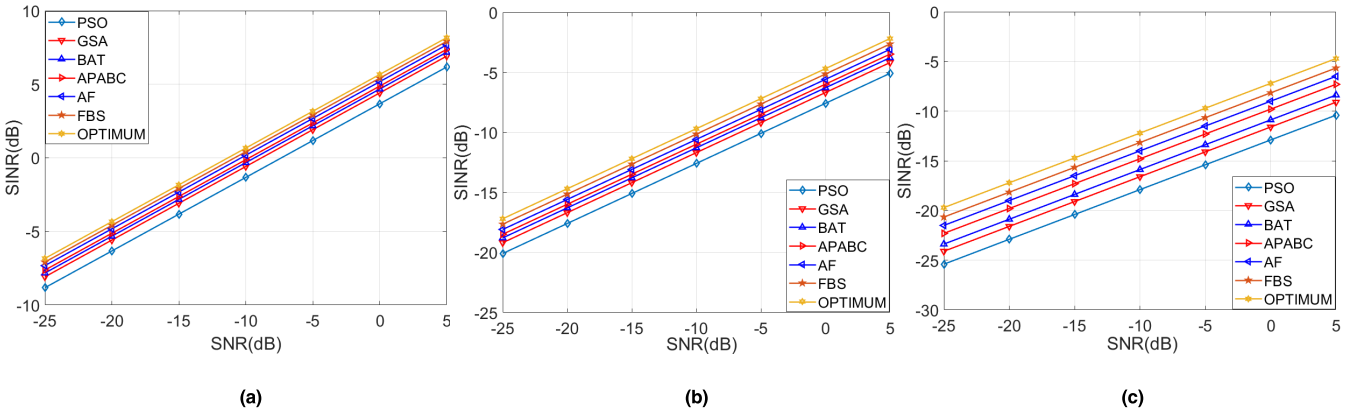


FIGURE 13. SINR versus SNR of the heuristic algorithms with different INRs: (a) INR = 5 dB; (b) INR = 20 dB; (c) INR = 35 dB.

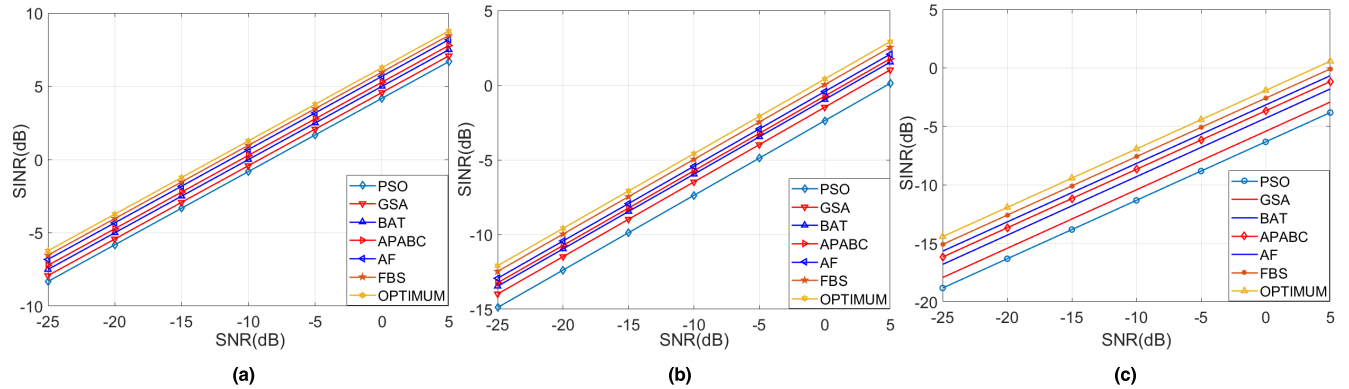


FIGURE 14. SINR performance versus SNR of the heuristic algorithms with different numbers of interferences. (a) One interference; (b) three interferences; (c) six interferences.

compared metaheuristic-based beamformer in an adaptive beamformer system.

b: SIMULATION CASE TWO: FOUR INTERFERENCE SOURCES WITH DIFFERENT INR IMPINGING ON A 6-ELEMENT ARRAY

A ULA consists of 6 monochromatic isotropic elements receiving four interference signals with different INRs from 20°, 40°, 65 and −70. Three groups of interference signals with INRs of 5, 20, and 35 dB are established in the different simulation scenarios. Figure 13 displays the SINR performance of these techniques versus the SNR under different power levels of interference sources by using the proposed and comparison optimization algorithms. From the results depicted in Figure 13, we can know that, in general, the optimization algorithms, including AF, APABC, and BAT, are able to achieve similar near-satisfactory SINR performance to FBS in the situation of INR = 5 dB, i.e., the interference signal with the lowest power. The proposed FBS achieved improved performance in terms of SINR compared with the other algorithms in all simulations even under the most severe interference situations when the INR is 30 dB. With increasing interference INR, the SINR performance of all algorithms is degraded, and the proposed algorithm presents evident advantages over these algorithms. The improvement in the SINR by FBS, compared to the suboptimal comparison algorithms, is 7.75%, 13.84% and 15.71%.

TABLE 8. Different number of interfereces for three scenarios.

Scenario One		Scenario Two		Scenario Three	
Interference	Incident Angle	Interference	Incident Angle	Interference	Incident Angle
1	40	1	40	1	40
				2	-20
		2	-20	3	-45
				4	-30
		3	-45	5	35
				6	60

Note that the proposed algorithm can achieve more robust results and suitable precision in ABF for high interference power levels.

c: SIMULATION CASE THREE: DIFFERENT NUMBERS OF INTERFERENCE SOURCES WITH FIXED INR RECEIVED BY 6-ELEMENT ARRAY

To prove the robustness of the proposed beamformer in this project, the simulations in this case were conducted to validate the effect of the interference source quantity on the SINR performance. The beamformer is equipped with 6 array elements, and the INR is fixed to 10 dB for the received interference signals. Table 8 illustrates the different numbers of interference sources and the corresponding incident angle values of the above-mentioned interference sources. The SINR performance of the proposed FBS and other

TABLE 9. Comparison results of the algorithms with 12 array elements.

Algorithm	Maximum iteration	CPU time(s)	Percentage Improved
FBS	5628	17.26	30.72%
GSA	4271	13.22	10.41%
BAT	6252	19.40	19.53%
AF	4870	15.48	25.82%
PSO	8239	23.63	9.68%
APABC	6134	18.59	21.24%

optimization algorithms for different numbers of interference signals is shown in Figure 14. As seen from Figure 14, when there is one interference signal at the receiver, most of the optimization algorithms can achieve a close-to-optimal SINR performance by considering the requirement for maximizing SINR. FBS demonstrates the best improvement, followed by AF showing a lower SINR, with a 4.56%, 9.84%, and 13.07% improvement in the diverse scenarios.

As the number of interference signals increases from one to six, the superior performance advantage of FBS becomes more evident. The increase in the number of interference sources increases the difficulty of the optimization problem. The failure of PSO and GSA to achieve sufficiently high SINR clearly illustrates their limitations; therefore, the proposed FBS is more versatile and robust than the other optimization methods in ABF applications.

3) MAXIMUM ITERATION AND TIME COMPLEXITY OF THE ABF SIGNIFICANT MODEL TO ACHIEVE OPTIMUM PERFORMANCE

In this section, the comparison results of the performed experiments between the FBS-based beamformer and the five heuristics-based beamformers are presented in the form of the maximum iteration number and time complexity in finding the global optima in the ABF significant model. The percentage improvement of the proposed FBS in terms of nulling level is also illustrated in this part. Each method was applied to optimize a ULA consisting of isotropic elements. Different cases with 14, 18, and 22 array elements were simulated to further validate the proposed approach for real-world large-array applications. The signal of interest is from 0°, and the three target interference signals are assumed to be impinging on the array from 20°, 40° and 70°. The input SNR was fixed at −5 dB. Under the same computer hardware configuration in Section 5.1, the average CPU time consumption (in seconds) at key points were measured by the built-in 'Matlab Profiler', which determines the computational complexity proportions. Each beamforming algorithm runs 100 times independently to reduce the statistical variance.

TABLE 10. Comparison results of the algorithms with 16 array elements.

Algorithm	Maximum iteration	CPU time(s)	Percentage Improved
FBS	6892	20.68	44.72%
GSA	5788	17.72	24.46%
BAT	7694	23.95	31.27%
AF	6217	19.24	35.93%
PSO	9823	27.26	21.20%
APABC	7435	21.01	34.72%

TABLE 11. Comparison results of the algorithms with 20 array elements.

Algorithm	Maximum iteration	CPU time(s)	Percentage Improved
FBS	8920	32.74	60.72%
GSA	7602	28.73	39.46%
BAT	9620	34.89	45.07%
AF	8023	27.30	50.93%
PSO	11831	38.21	32.20%
APABC	9035	33.47	47.70%

The optimization results considering the maximum number of iterations as well as the time complexity for the significant ABF model of the proposed FBS and the comparison algorithms under different numbers of array elements are listed in Table 9, Table 10 and Table 11.

From the above tables, it can be observed that the proposed FBS achieves the deepest nulling level and optimum performance of the ABF model in all cases. In addition, the percentage improvement of FBS for the nulling level is 90% in the first case study, compared to the conventional GA algorithm, which has the worst nulling performance out of all comparison algorithms. As the number of array elements increases, the improvements in the nulling level by FBS, compared to the conventional GA, increase to 38.89% and 65.04%, as demonstrated in Table 9 and Table 10. This is due to the increased number of elements generating a greater search dimension of the solution, which inherently leads to the greater difficulty of the optimization problem. In addition, the global search optimization ability of the FBS algorithm is more suitable for obtaining the multi-dimensional solutions of large array elements and improving the nulling levels.

Concerning the convergence performance and the computational cost, FBS requires far fewer evaluations and lower CPU time than PSO, BAT, and APABC in the three scenarios, and FBS presents a more evident advantage over these algorithms in the large array element cases. This indicates that FBS requires much fewer function evaluations than the three other algorithms, as shown in Table 9 and Table 10, since more array elements results in a larger dimensionality of the weight vector solution. This requires a greater search time and number of computations to optimize the beamformer. Although FBS required slightly more evaluations and higher CPU time compared to GSA and AF, the percentage improvement of the nulling level can compensate and provide an acceptable performance for the proposed FBS in the ABF model. Overall, the FBS approach is quite competitive when compared with other methods.

VI. CONCLUSION

In this paper, we presented a global optimization heuristic algorithm called FBS, which is based on the use of the Fibonacci series, for achieving improved ABF performance. Interactive global and local search rules were proposed to reduce the probability of falling into local optima, and the global randomness characteristic and space region shortening fraction of this technique guarantee the satisfactory convergence speed of the global optimization process. Numerical multimodal benchmarking functions were employed to validate the effectiveness of these global optimization algorithms. We found that the generally FBS achieves a better convergence rate, precision and stability compared to the other methods. In addition, we devised a specific implementation architecture based on FBS for the adaptive beamformer. The amplitudes and phases of the weight vector acting as the solution were acquired in the search space by FBS, and the beamforming results synthesized by the vectors were compared with typical metaheuristic-based beamforming algorithms to validate the improved output SINR.

The simulation experiments in Section 5 demonstrate that the proposed FBS achieves 21.04%, 27.62%, and 31.73% increased SINR over the suboptimally performing comparison algorithms for cases with 6-14 array elements. In addition, it also achieves significant improvements of 7.75%, 13.84% and 15.71% for three different INR cases. With respect to different numbers of interference signals, the SINR percentage improvement by FBS is 4.56%, 9.84%, and 13.07%. Consequently, the proposed FBS is seen as a valuable tool for multi-objective optimization and is well suited for ABF design problems. FBS also seems to be a promising smart antenna technology based on beamforming. In future work, FBS will be explored to apply to a more complicated time-varying situation to the ABF field.

REFERENCES

[1] T. Komljenovic, R. Helkey, L. Coldren, and J. E. Bowers, "Sparse aperiodic arrays for optical beam forming and LIDAR," *Opt. Express*, vol. 25, pp. 2511–2528, Feb. 2017.

[2] A. Madanayake, C. Wijenayake, D. G. Dansereau, T. K. Gunaratne, L. T. Bruton, and S. B. Williams, "Multidimensional (MD) circuits and systems for emerging applications including cognitive radio, radio astronomy, robot vision and imaging," *IEEE Circuits Syst. Mag.*, vol. 13, no. 1, pp. 10–43, Firstquarter 2013.

[3] H. Bing and J. Leger, "Numerical aperture invariant focus shaping using spirally polarized beams," *Opt. Commun.*, vol. 281, no. 8, pp. 1924–1928, 2008.

[4] D. Headland, Y. Monnai, D. Abbott, C. Fumeaux, and W. Withayachumnankul, "Tutorial: Terahertz beamforming, from concepts to realizations," *APL Photon.*, vol. 3, Jan. 2018, Art. no. 051101.

[5] Z. D. Zaharis, C. Skeberis, T. D. Xenos, P. I. Lazaridis, and J. Cosmas, "Design of a novel antenna array beamformer using neural networks trained by modified adaptive dispersion invasive weed optimization based data," *IEEE Trans. Broadcast.*, vol. 59, no. 3, pp. 455–460, Sep. 2013.

[6] X. Yang, P. Yin, Z. Tao, and T. K. Sarkar, "Applying auxiliary array to suppress mainlobe interference for ground-based radar," *IEEE Antennas Wireless Propag. Lett.*, vol. 12, pp. 433–436, Mar. 2013.

[7] Y.-H. Choi, "Doubly constrained robust beamforming method using subspace-associated power components," *Digit. Signal Process.*, vol. 42, pp. 43–49, Jul. 2015.

[8] A. El-Keyi and B. Champagne, "Collaborative uplink transmit beamforming with robustness against channel estimation errors," *IEEE Trans. Veh. Technol.*, vol. 58, no. 1, pp. 126–139, Jan. 2009.

[9] A. Hassani, J. Plata-Chaves, M. H. Bahari, M. Moonen, and A. Bertrand, "Multi-task wireless sensor network for joint distributed node-specific signal enhancement, LCMV beamforming and DOA estimation," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 3, pp. 518–533, Mar. 2017.

[10] M. Roshanaei, C. Lucas, and A. R. Mehrabian, "Adaptive beamforming using a novel numerical optimisation algorithm," *IET Microw., Antennas Propag.*, vol. 3, no. 5, pp. 765–773, Aug. 2009.

[11] E. M. Al-Ardi, R. M. Shubair, and M. E. Al-Mualla, "Performance evaluation of the LMS adaptive beamforming algorithm used in smart antenna systems," in *Proc. IEEE Midwest Symp. Circuits Syst.*, Dec. 2004, pp. 432–435.

[12] C. Wei and Y. Lu, "Adaptive beamforming for arbitrary array by particle swarm optimization," in *Proc. IEEE Int. Conf. Comput. Electromagn.*, Feb. 2015, pp. 79–80.

[13] S. Darzi, K. T. Sieh, I. M. Tariqul, H. R. Soleymanpour, and S. Kibria, "A memory-based gravitational search algorithm for enhancing minimum variance distortionless response beamforming," *Appl. Soft Comput.*, vol. 47, pp. 103–118, Oct. 2016.

[14] T. Sussman and K.-J. Bathe, "The gradient of the finite element variational indicator with respect to nodal and applications in fracture mechanics and mesh optimization," *Int. J. Numer. Methods Eng.*, vol. 21, no. 4, pp. 763–774, 2010.

[15] R. M. Maina, K. Langat, and P. K. Kihato, "Comparative analysis between use of particle swarm optimization and simulated annealing algorithms in beam/null steering in a rectangular array system," *Int. J. Innov. Res. Develop.*, vol. 2, no. 7, pp. 27–33, 2013.

[16] S. Darzi, S. K. Tiong, M. T. Islam, M. Ismail, and S. Kibria, "Optimal null steering of minimum variance distortionless response adaptive beamforming using particle swarm optimization and gravitational search algorithm," in *Proc. IEEE Int. Symp. Telecommun. Technol.*, Nov. 2015, pp. 230–235.

[17] L. He and S. Huang, "Modified firefly algorithm based multilevel thresholding for color image segmentation," *Neurocomputing*, vol. 240, pp. 152–174, May 2017.

[18] P. Korošec, J. Šilc, and B. Filipič, "The differential ant-stigmergy algorithm," *Inf. Sci.*, vol. 192, pp. 82–97, Jun. 2012.

[19] L. Cui, G. Li, Q. Lin, Z. Du, W. Gao, J. Chen, and N. Lu, "A novel artificial bee colony algorithm with depth-first search framework and elite-guided search equation," *Inf. Sci.*, vols. 367–368, pp. 1012–1044, Nov. 2016.

[20] L. Cui, G. Li, Z. Zhu, Q. Lin, Z. Wen, N. Lu, K.-C. Wong, and J. Chen, "A novel artificial bee colony algorithm with an adaptive population size for numerical function optimization," *Inf. Sci.*, vol. 414, pp. 53–67, Nov. 2017.

[21] L. Cui, G. Li, Z. Zhu, Q. Lin, K.-C. Wong, J. Chen, N. Lu, and J. Lu, "Adaptive multiple-elites-guided composite differential evolution algorithm with a shift mechanism," *Inf. Sci.*, vol. 422, pp. 122–143, Jan. 2018.

[22] B. Liao and S.-C. Chan, "Adaptive beamforming for uniform linear arrays with unknown mutual coupling," *IEEE Antennas Wireless Propag. Lett.*, vol. 11, pp. 464–467, 2012.

[23] A. Etminaniesfahani, A. Ghanbarzadeh, and Z. Marashi, "Fibonacci indicator algorithm: A novel tool for complex optimization problems," *Eng. Appl. Artif. Intell.*, vol. 74, pp. 1–9, Sep. 2018.
[24] M. Subasi, N. Yildirim, and B. Yildiz, "An improvement on Fibonacci search method in optimization theory," *Appl. Math. Comput.*, vol. 147, pp. 893–901, Jan. 2004.
[25] B. Yildiz and E. Karaduman, "On Fibonacci search method with k-Lucas numbers," *Appl. Math. Comput.*, vol. 143, pp. 523–531, Nov. 2003.
[26] J. O. Omolehin, M. A. Ibiejugba, A. E. Onachi, and D. J. Evans, "A fibonacci search technique for a class of multivariable functions and ODEs," *Int. J. Comput. Math.*, vol. 82, no. 12, pp. 1505–1524, 2005.
[27] R. Ramaprabha, M. Balaji, and B. Mathur, "Maximum power point tracking of partially shaded solar PV system using modified Fibonacci search method with fuzzy controller," *Int. J. Elect. Power Energy Syst.*, vol. 43, no. 1, pp. 754–765, 2012.
[28] O. O. Kaid, F. Debbat, and S. A. Boudghene, "Null steering beamformer using hybrid algorithm based on honey bees mating optimisation and tabu search in adaptive antenna array," *Prog. Electromagn. Res.*, vol. 32, pp. 65–80, Aug. 2012.
[29] L. Cheng, X.-H. Wu, and Y. Wang, "Artificial flora (AF) optimization algorithm," *Appl. Sci.*, vol. 8, no. 3, p. 329, 2018.
[30] X. S. Yang, "Bat algorithm for multi-objective optimisation," *Int. J. Bio-Inspired Comput.*, vol. 3, no. 5, pp. 267–274, 2012.
[31] E. P. Adorio and U. Diliman, "MVF—Multivariate test functions library in C for unconstrained global optimization," Dept. Math. U.P., Diliman Quezon City, Philippines, 2005, pp. 100–104.
[32] M. Molga and C. Smutnicki, "Test functions for optimization needs," Dept. Ind. Eng., Fac. Eng., Univ. Tehran, Tehran, Iran, 2005, vol. 101.
[33] L. Cui, G. Li, X. Wang, Q. Lin, J. Chen, N. Lu, and J. Lu, "A ranking-based adaptive artificial bee colony algorithm for global numerical optimization," *Inf. Sci.*, vol. 417, pp. 169–185, Nov. 2017.
[34] C. Zhang, J. Ning, S. Lu, D. Ouyang, and T. Ding, "A novel hybrid differential evolution and particle swarm optimization algorithm for unconstrained optimization," *Oper. Res. Lett.*, vol. 37, no. 2, pp. 117–122, 2009.

HAICHUAN ZHANG was born in Qinhuangdao, China, in 1991. He received the B.S. and M.S. degrees from the School of Electronic Countermeasures, National University of Defense Technology (NUDT), Hefei, China, in 2016, where he is currently pursuing the Ph.D. degree. His research interests include adaptive beamforming and its applications, array signal processing, and digital communication systems.

FANGLING ZENG was born in Zigong, China, in 1970. She received the B.S., M.S., and Ph.D. degrees from the University of Science Technology of China (USTC), Heifei, China, in 1996, 1999, and 2002, respectively. She is currently a Professor with the National University of Defense Technology. She is the coauthor of five books, more than 20 articles, and more than 15 inventions. Her main research interests include precise point positioning, phased array signal processing, optimization algorithms development, and GNSS/INS Precise Timing.