Progress In Electromagnetics Research B, Vol. 88, 73–95, 2020

A Beamformer Design Based on Fibonacci Branch Search

Tianbao Dong*, Haichuan Zhang, and Fangling Zeng

Abstract—An approach towards beamforming for a uniform linear array (ULA) based on a novel optimization algorithm, designated as Fibonacci branch search (FBS) is presented in this paper. The proposed FBS search strategy was inspired from Fibonacci sequence principle and uses a fundamental branch structure and interactive searching rules to obtain the global optimal solution in the search space. The structure of FBS is established by two types of multidimensional points on the basis of shortening fraction formed by the Fibonacci sequence, and in this mode, interactive global searching and local optimization rules are implemented alternately to reach global optima, avoiding stagnating in local optimum. At the same time, the rigorous mathematical proof for the accessibility and convergence of FBS towards the global optimum is presented to further verify the validity of our theory and support our claim. Taking advantage of the global search ability and high convergence rate of this technique, a robust adaptive beamformer technique is also constructed here by FBS as a real time implementation to improve the beamforming performance by preventing the loss of optimal trajectory. The performance of the FBS is compared with five typical heuristic optimization algorithms, and the reported simulation results demonstrate the superiority of the proposed FBS algorithm in locating the optimal solution with higher precision and reveal the further improvement in adaptive beamforming performance.

1. INTRODUCTION

Since the conception of the adaptive arrays came into usage in aerospace and military applications via the employment of electronically steered beamformer, the ultilized technique adaptive beamforming (ABF) has drawn significant attention in various fields [1–4]. ABF possesses the potential ability to optimize the radiation pattern in real time, which obtains a larger output signal to interference-plus-noise ratio (SINR) by steering the main lobe of radiation toward a desired signal while placing respective nulls toward several interference signals [5].

Classical adaptive beamforming methods used to extract the excitation weights are based on two main criterions: maximum signal-to-noise ratio (MSINR) and minimum mean square error (MMSE) [6]. Minimum variance distortionless response (MVDR) is one of the typical adaptive beamforming approaches on the basis of MSINR criterion. The design of this beamformer involves minimizing the output power subject to the unit gain constraint to desired signal [7–9]. Although the MVDR beamformer is capable of suppressing the interference and improving system reliability, the weights computed by MVDR are not able to form deep nulls towards the interference source in various interference scenarios on account of the characteristics of this technique. Closed-form (CF) design or the classic algorithm exists to easily find the optimal solution for a actually convex problem, such as the algorithm known as the sample matrix inversion (SMI) technique; however, the weight calculation of this solving process takes into account the interference correlation matrix which makes this beamforming process very difficult, time consuming, and sometimes, in ABF application, is unmanageable. Besides, the closed-form design for the convex ABF model problem always needs to compute the matrix inverse

Received 31 March 2020, Accepted 1 July 2020, Scheduled 3 September 2020

* Corresponding author: Tianbao Dong (dtb_1@163.com).
The authors are with the School of Electronic Countermeasures, National University of Defense Technology, China.

process which introduces high computational complexity especially for large number of array elements and not suitable for certain real-time beamforming situations [10–12]. In addition, the classic closed-form design algorithms have hardware component device limitations. It is not possible to change the hardware of the filter for signal processing or to change the design of the antenna based on the increased number of components, but this is necessary. Another criterion for computing the array weights is to minimize the MMSE. One of the most widely used MMSE-based adaptive algorithm is the least mean square (LMS) approach, and this method needs a training sequence of signal of interest (SOI) to adaptively adjust the complex weights and minimize the difference between the array output and the desired signal for forming the optimum array pattern. Consequently, the inherent shortcomings of above mentioned derivative-based and gradient-based ABF methods have compelled many researchers to explore meta-heuristics (MH) methods for overcoming these difficulties.

The main advantage of the evolutionary heuristics algorithms over the classical derived approaches in antenna systems is that they have no requirements for extra iterative derivations or computationally extensive routines in objective functions of ABF model, and some of the evolutionary algorithms have been dedicated to beamformer implementations for their ability to search the global optimum. Approaches such as genetic algorithms (GA), particle swarm optimization (PSO), and other modified techniques are a set of optimization algorithms that have been suggested in the past decades to solve a variety of ABF problems [13, 14]. Many researches have shown that the excitation weights extracted by these optimization techniques defined by specific criteria can be used to place a maximum beam and null in an array pattern in specified locations, and they also require relatively lower mathematical complexity than derivative-based or iterative-based ABF methods. However, there still exist weaknesses and limitations in the application of ABF with these techniques. Most iterative evolutionary methods are highly dependent on starting points in the case of large number of solution variables [15]. Yet the weights of ABF are regularly associated with a large number of array elements, and the excitation of array elements is complex, i.e., having both amplitude and phase, hence the size of the ABF solution space cannot be very small, and in case, these evolutionary algorithms are not really applicable to beamforming. Besides, the classic optimization methods are prone to get trapped in local minima and not reach the global optimum when solving complex multimodal optimization problems of array weight extraction, resulting in a suboptimum beamforming performance [16, 17]. In addition, most of MH algorithms are population-based optimization techniques which require long execution times to converge, specifically when solving large scale complex ABF engineering problems, and the complexity required for implementing the algorithms would also result in huge cost and consumption of hardware resources.

In consideration of the above-mentioned studies, we propose a novel interactive-based random iterative search strategy, called FBS in this paper to deal with complicated optimization problems of ABF. The motivation for considering the FBS with application to such a robust adaptive beamformer is that we expect to be able to form deep nulls towards the interference source in various interference scenarios without the requirement of apriori information such as the training sequence of signal of interest and avoid complex mathematical processing such as the inverse calculation of the interference correlation matrix compared to the derivative-based approaches, and with respect to a traditional meta-heuristic algorithms applied to beamformer, it is anticipated that the FBS based on beamformer could obtain the near optimal nulling level performance featuring a high global optimization resolution capability and prevents the loss of optimal search trajectory low computational load possesssing to improve the convergence performance with the design flexibility of the algorithm in the framework of ABF.

The concept of the proposed FBS is defined from two aspects: The first one is the generation principle of the Fibonacci branch architecture. The basis for building a branch structure in FBS is to establish search points and the shortening fraction based on the Fibonacci sequence to generate a set of optimization elements composed of two types of search points. The optimization endpoints search for the optimal solution according to the branch's growth path pattern. The second view of the FBS concept is that the construction of interactive iteration applies rules to the calculation of optimization elements. The iterative rules are composed of global searching and local optimization, which are the two phases necessary to update the optimization elements. Global tentative points and local searching points are formulated in the phase of two interaction processes, and the points with the best fitness converge towards the global optimum in searching space. At the same time, computer memory can be fully used to record the optimizing process during the interaction optimization course. Global randomness is one of the important characteristics of FBS, and this mechanism is implemented on those points that are not easy to fall into the local optimum and not able to find a better solution. The novelty of this paper lies in the fact that we design the Fibonacci branch optimization structure and propose the novel global searching and local optimizing interaction iterative technique. In addition, the Fibonacci branch search algorithm proposed here has been applied to antenna array beamforming in several cases and in comparison with other evolutionary optimization-based techniques on several test functions and the robust ABF.

The reminder of this paper is organized as follows. Adaptive beamforming model incorporation with metaheuristic algorithm is described in Section 2. Section 3 presents the proposed FBS optimization algorithm and the accessibility convergence proof of FBS. The validation of the proposed FBS via benchmark functions and the simulation results are reported in Section 4. Section 5 gives the conclusion.

2. ADAPTIVE BEAMFORMING MODEL INCORPORATION WITH METAHEURISTIC ALGORITHM

Consider a uniform linear antenna array of M omnidirectional array elements employed in adaptive beamforming receiver, one desired signal and Q uncorrelated interferences which impinge on the array at the kth snapshot can be expressed by [18]

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

where $s(k)$ and $i_i(k)$ are the desired global navigation satellite systems (GNSS) signal and the ith interference, and $\mathbf{n}(k)$ denotes the complex vector of 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; $d_c = \frac{\lambda}{2}$ is the inter-element spacing; λ is the wavelength of GNSS carrier; 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 stands for 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. Solid incident arrow represents the GNSS signal, and dotted arrow represents the interference.

In the above model, the adaptive beamformer studied in this paper aims to calculate a complex weight vector that satisfies the requirement of achieving a deep nulling level for interference to maximize the output SINR, and is usually a low nulling level for multiple interference sources. The non-global optimal weight vector is one of the main disadvantages of MVDR beamforming technology. Therefore, in this section, the proposed FBS algorithm incorporated into the beamforming model will extract the optimal array excitation weights to achieve the improved performance of the technique.

Altering the radiation pattern of an antenna array by adjusting the weights is an inherently multi-objective problem, since multiple sets of agent weight vectors $\mathbf{w}$ with amplitude and phase to make deep nulls toward the interference place and steer the radiation beam toward the desired user to achieve the maximum SINR must be satisfied. The optimization method to the designed ABF aims at finding the near-global minimum of a fit mathematical function called fitness function; therefore, the best weight vector is determined according to the fitness value obtained from the object function defined based on

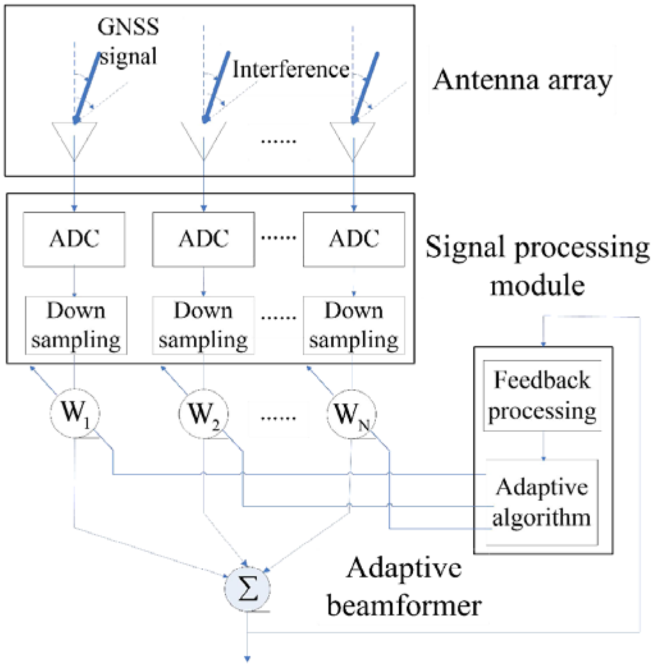


Figure 1. The schematic of the linear antenna array processor.

the SINR. Therefore, a typical fitness function constructed from the perspective of SINR for calculating complex excitation weights using FBS can be appropriately expressed in the following form:

$$Fitness\_Function\left(\mathbf{w}\right) = \frac{P_d}{\sum_{i=1}^{Q} P_i + P_N} \tag{4}$$

where

$$P_d = \frac{1}{2}E\left[\left|\mathbf{w}^T\mathbf{x}_d\right|^2\right] \tag{5}$$

$$P_i = \frac{1}{2}E\left[\left|\mathbf{w}^T\mathbf{x}_i\right|^2\right] \tag{6}$$

are, respectively, the power of the desired signal and the power corresponding to the ith interference, and P_N is the noise power. $\mathbf{x}_d$ and $\mathbf{x}_i$ denote the desired signal and interference component of the received signal in Equation (1), and E is the expectation operator.

Then, the design objective function can be properly stated in the following form

$$Fitness\_Function\left(\mathbf{w}\right) = \frac{\mathbf{w}^T\mathbf{R}_d\mathbf{w}}{\mathbf{w}^T\sum_{i=1}^{Q}\mathbf{R}_i\mathbf{w} + \sigma^2_{\text{noise}}\mathbf{w}^T\mathbf{w}} \tag{7}$$

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

$$\sigma^2_{\text{noise}} = 10^{-\text{SNR}/10} \tag{8}$$

By maximizing Eq. (7), 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, and the optimum performance of the weight corresponding to the maximum SINR can be evaluated based on the best fitness account. Next section of the paper provides a brief description of the implementation steps for extracting the weight using Fibonacci branch search method.

3. PROPOSED FBS ALGORITHM

Since the inception of Fibonacci optimization strategy, the effectiveness of the algorithm for solving a set of nonlinear benchmark functions has been proven in one dimension space; however, this algorithm is seldom applied to the search optimization problem of multidimensional space for the properties and structure of itself, and very few of its variants have been implemented for the beamforming applications in open reported literatures. In this section, the conventional Fibonacci sequence method will be briefly introduced, and the inspired FBS will be explained in principle and provide an in-depth insight into this technique.

3.1. The Standard Principle of Fibonacci Sequence Method

The famous Fibonacci sequence was proposed initially by Etminaniesfahani et al. [19], and the recurrence formula of the general sequence term is given by [20]

$$\begin{cases} F_1 = F_2 = 1 \\ F_j = F_{j-1} + F_{j-2}, \quad j \geq 3 \end{cases} \tag{9}$$

where F_j represents the jth general term of Fibonacci sequence.

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 Fibonacci sequence term, and it has been perceived as one of the most effective strategies to solve one-dimension unimodal problem [21]. 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. Supposing a unimodal $f(x)$ function which is defined on the intervals $[A, B]$. Initially, the technique starts a choice for two feasible points x_1 and $\widetilde{x}_1$, $x_1 < \widetilde{x}_1$ in the given range for the first iteration. Then, it is necessary to reduce the initial box of range to a sufficiently small box region including the minimum solution of $f(x)$ (through an iteration process) for the interval that can be narrowed down provided that the function values are known at two different points in the range. The implementation of the classical Fibonacci serial optimization algorithm can be found in [22], and we will not go into further details here.

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

3.2. Fibonacci Branch Structure and FBS Optimization Algorithm

The standard Fibonacci strategy cannot efficiently solve multi-variate problems and reliably perform the optimum fitness evaluation of multimodal functions [23]. While this is in contradiction to the 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 the searching elements with a dendritic branch structure and interactive searching optimization rules.

The basic structure of FBS expanded to the multi-dimensional space D can be illustrated in Figure 2, 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 which can be determined from the searching rule. A certain proportion of the vectors can be constructed as follows

$$\frac{\|\mathbf{X}_C - \mathbf{X}_A\|}{\|\mathbf{X}_B - \mathbf{X}_A\|} = \frac{\|\mathbf{X}_B - \mathbf{X}_C\|}{\|\mathbf{X}_C - \mathbf{X}_A\|} = \frac{F_p}{F_{p+1}} \tag{10}$$

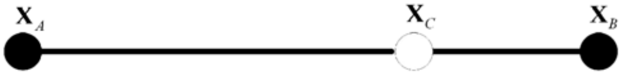


Figure 2. Basic structure of the proposed FBS.

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 search space, the fitness function value calculated by the endpoints in the structure should be evaluated as

$$\mathbf{f}(\mathbf{X}_A) < \mathbf{f}(\mathbf{X}_B) \tag{11}$$

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

$$\mathbf{X}_C = \mathbf{X}_A + \frac{F_p}{F_{p+1}} (\mathbf{X}_B - \mathbf{X}_A) \tag{12}$$

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 basic structure. Although a similar relevant algorithm theory has been studied in [24], it did not elaborate on the steps and principles of the algorithm. In this section, the principle part of FBS is well explained, and the details of the implementation content are fully described including the implantation procedure being regularized, and it is successfully applied to the adaptive beamforming field.

Combining with the basic structure, the process of searching for 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 searching process, which are the corresponding two interactive rules. Let G denote the point sets of the object function searching for in current processing phase, and set $|G|_{num} = F_p$, $p = 1, 2, \cdots, N$. $|\cdot|_{num}$ represents the total number of points in a set, and N is the depth of the Fibonacci branch. The fitness value of the endpoints $\mathbf{X}_A$ and $\mathbf{X}_B$ are initialized using the two corresponding interactive optimization rules, then the segmentation points $\mathbf{X}_C$ can be obtained from Equation (12). By comparison of the fitness values of all the points in the structure, we can get the results that the best fitness value accordingly corresponds to the closest optimal solution. To the next optimization phase, the optimal point with the best fitness value is provided in the forefront position of the set, and the points corresponding to the suboptimal fitness are arranged below the optimal point in accordance with the order from the best to worst. Throughout the operations above, the points set G can be updated in every optimizing phase to attain the aim of growing the Fibonacci branch and global optimization in search space simultaneously.

The two interactive searching 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, which are defined by

$$\{\mathbf{X}_A\} = G_p = \{\mathbf{X}_q | q = [1, F_P]\} \tag{13}$$

$$\{\mathbf{X}_B\} = \left\{ \mathbf{X} | \mathbf{X} \in \prod_{f=1}^{D} \left[X_{lb}^f, X_{ub}^f \right]^U \right\} \tag{14}$$

where G_p is the search points space set in the pth iteration; $\mathbf{X}_q$ are the points in set G_p, and this search point was randomly selected in the whole search space with random search mechanism. q is the sequence number lying on the interval between 1 and the pth Fibonacci number, and it represents the Fibonacci layer used for searching the global optima. $\mathbf{X}_A$ take all the points from G_p of the pth iteration. The other endpoints $\mathbf{X}_B$ take random points in search space, and 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 $\forall \mathbf{X} \in \{\mathbf{X}_B\}$, component x of vector $\mathbf{X}$ is a random variable that satisfies a uniform distribution over the interval $[X_{lb}, X_{ub}]^U$, where the usual character U stands for the uniform distribution of the variable, and the probability distribution of the component can be written as

$$P(x) = U(X_{lb}, X_{ub}) = \frac{1}{X_{ub} - X_{lb}} \tag{15}$$

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 (12).

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

$$\mathbf{X}_{\text{best}} = BEST(\mathbf{G}_P) \tag{16}$$

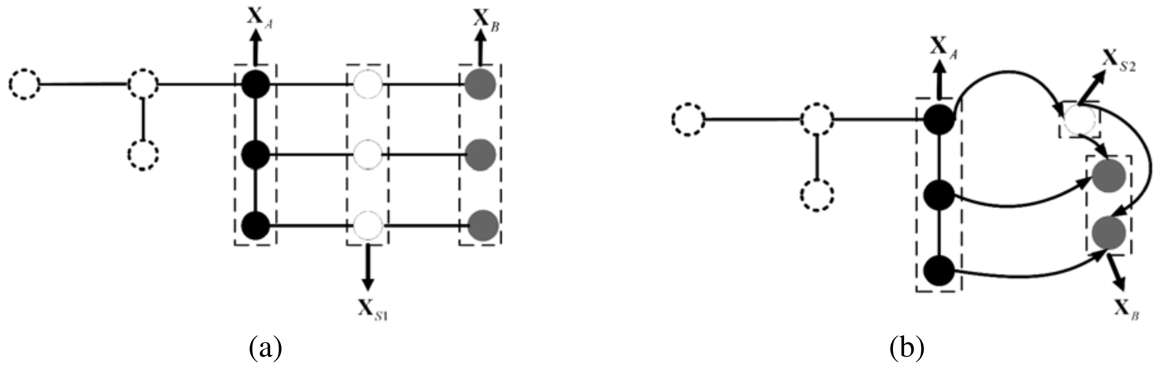


Figure 3. The process of building Fibonacci branch for the global optimization. (a) The first global searching stage. (b) The second local optimization stage.

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

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

$$f(\mathbf{X}_A) = \min\{f(\mathbf{X}_q), q = [1, F_P]\} \tag{17}$$

$$\mathbf{X}_B = \{\mathbf{X}_q | \mathbf{X}_q \in \mathbf{G}_P \wedge \mathbf{X}_q \neq \mathbf{X}_A\} \tag{18}$$

Using the calculation formula of the segmentation point, the segmentation point $\mathbf{X}_{S2}$ of the second local optimization stage can be determined according to the endpoints defined in Eqs. (17) and (18), and the segment search point is computed by Equation (12).

From the two interactive searching rules mentioned above, new points including endpoints $\mathbf{X}_A$, $\mathbf{X}_B$ and segmentation points $\mathbf{X}_{S1}$, $\mathbf{X}_{S2}$ are generated in the two optimization stages, and the total number of points is $3F_P$. Evaluating the cost functions in new points determines their fitness, and all these points are sorted from the best to the worst based on their fitness values. 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 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 search space for the next iteration.

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

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

Figure 4 shows the flowchart of the general procedures for the specific implementation of FBS.

3.3. Implementation Flowchart of Fibonacci Branch Search to Adaptive Beamforming

In this subsection, in light of the results described in detail previously, an optimization scheme of ABF problem combining with the novel FBS is presented to enhance the maximum power for target signal and generate deep nulls for interferences. The basic idea of the design of such an adaptive beamformer is to utilize the global searching and local convergence capability of the novel efficient search algorithm to reduce the local minimum problem of the solution to weight vector for getting the maximum SINR.

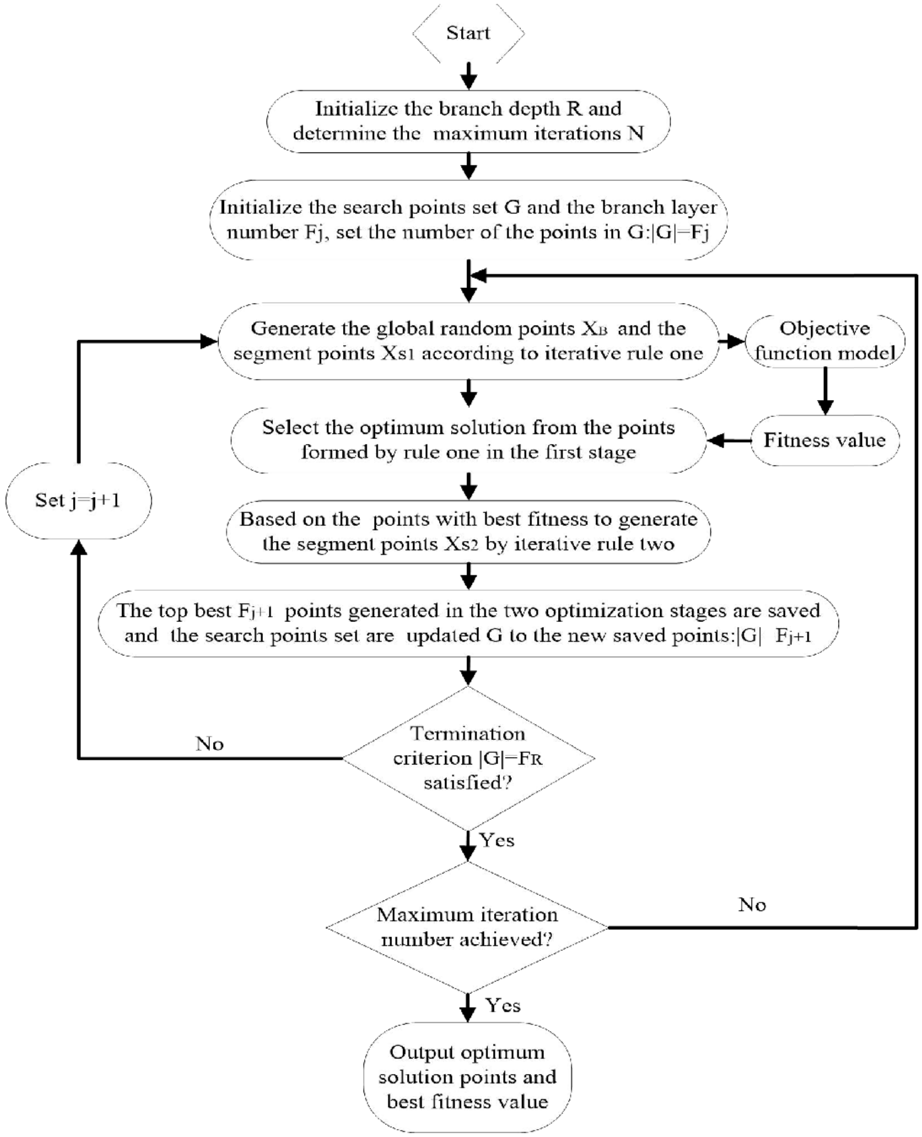

Start
Initialize the branch depth R and determine the maximum iterations N
Initialize the search points set G and the branch layer number Fj, set the number of the points in G:|G|=Fj
Generate the global random points XB and the segment points Xs1 according to iterative rule one
Objective function model
Fitness value
Select the optimum solution from the points formed by rule one in the first stage
Set j=j+1
Based on the points with best fitness to generate the segment points Xs2 by iterative rule two
The top best Fj+1 points generated in the two optimization stages are saved and the search points set are updated G to the new saved points:|G| Fj+1
No
Termination criterion |G|=FR satisfied?
Yes
Maximum iteration number achieved?
No
Yes
Output optimum solution points and best fitness value

Figure 4. Flowchart of Fibonacci branch search optimization method.

The general procedures for the implementation of Fibonacci branch search method with application to adaptive beamforming are presented in Figure 5, in which the key steps are briefly described below.

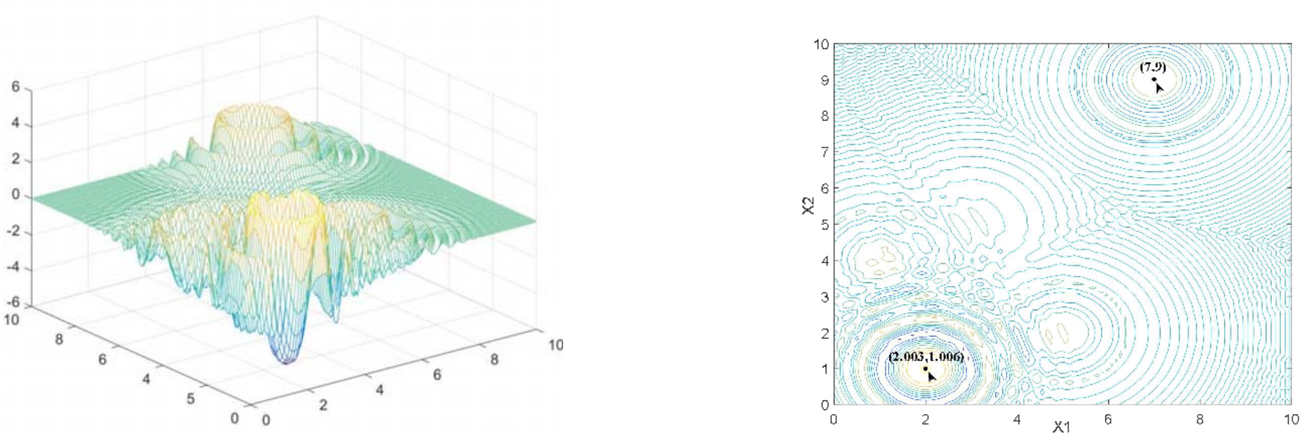
(a) Choose the depth R of Fibonacci branch to determine the population F_R of the top branch layer, and set the maximum number of iterations of the optimization process.

(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 element number of ULA. Also, define the amplitude search space of the weight within $[0, 1]$, and limit the range of 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 initial population of the weight vector set $\mathbf{G_w}$, and the weight element in the vector sets $\mathbf{G_w}$ constructed by the amplitude and phase can be expressed as follows:

$$\mathbf{w}_{jd} = \text{rand}\,[0, 1] \cdot e^{j \cdot \text{rand}[-\pi,\pi]} \tag{19}$$

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



(a)

(b)

Figure 5. Three dimensional and contour plots for the Langermann function. (a) Three dimensional of Langermann function. (b) Contour plots of Langermann function.

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

(e) Set the vector elements of $\mathbf{G}_\mathbf{w}$ and $\mathbf{w}_B$ as the endpoints in Equation (12), and compute the first set of weights $\mathbf{w}_{S1}$ according to the iterative rule one.

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

(g) Generate the second set of weight vectors based on the best weight vector $\mathbf{w}_{\text{best}}$ and the other weights from the weight space set using Equation (12) in iterative rule two.

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

(i) Check the termination criteria $\mathbf{G}_\mathbf{w} = F_R$. 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), or else report the output results and terminate.

3.4. Accessibility and Convergence Proof of FBS towards the Global Optimum for Multimodal Functions

In this section, accessibility and convergence proof of Fibonacci tree optimization algorithm are analyzed and investigated based on global randomness of Fibonacci branch structure. Additional rigorous mathematical proofs were implemented to prove and clarify that the proposed FBS algorithm can achieve global optimality and ensure that FBS converges towards global optimality.

3.4.1. Accessibility Investigation of FBS Algorithm

Mathematical proof one: The solution set in the domain of objective function can be reached via the accessible set in search space by FBS.

According to the principle of interactive searching rules in Fibonacci branch search algorithm, after enough long iterations $n < +\infty$, the endpoints $\mathbf{X}_B$ are generated by rule one in the basic structure of Fibonacci branch obeying the uniform distribution between the space defined by the upper and lower bounds, i.e., $\forall \mathbf{X} \in \mathbf{X}_B$, $\mathbf{X} = (X_d)_{D\times 1}$ where D is the dimension of the vector, and its probability distribution is $P(X_d) = U(X_{\min}, X_{\max})$; therefore, $\mathbf{X}_d$ satisfies the following relation in the domain of the target function

$$\int \overbrace{\cdots}^{D} \int \frac{1}{X_{\max} - X_{\min}} \, dX_d = 1 \tag{20}$$

From the above proof process, $\forall \mathbf{X} \in \mathbf{X}_B$ obey $\mathbf{X} \in B$, namely, the optimization set B is the accessible set of $\mathbf{X}_B$ by FBS algorithm.

Mathematical proof two: The global optima of objective function in search space are accessible for FBS algorithm.

Suppose that the global optima $\mathbf{X}^*$ in search space are in the definition domain B of objective function, then, it can be known that according to the mathematical proof one, the solution $\mathbf{X}^*$ is within the accessible set of B. Then, consider that the probability of the uniformly distributed in search space for random search points is P, and make a hypothesis that the local optimal solution obtained by FBS is $\mathbf{X}^L$. Therefore, after long enough iterations $n < +\infty$, the probability of falling into a local optimal solution for FBS is $P^* \leq \prod_{i=1}^{n}(1-P)$ on the basis of proof one, then we can obtain that

$$\lim_{n\to+\infty} P^* \leq \lim_{n\to+\infty} \prod_{i=1}^{n} (1-P) = 0 \tag{21}$$

From this point of view, the global optima of objective function in search space are accessible for FBS algorithm.

Consequently, it can be demonstrated from the above two mathematical proofs that the global optima in the domain of objective function are reachable for the accessible solution of FBS in search space.

3.4.2. Convergence Analysis of FBS Algorithm

Suppose that long enough iterations $n < +\infty$ are implemented by FBS to search for the global optima of objective function, and the current search element set G of the basic structure obtained by FBS can form the increasingly optimized search point sequences $\{\mathbf{x}_T\} = \{\mathbf{x}_t \,|\mathbf{x}_{\text{best}} \in S_t\}$, $t = 1, 2, \cdots, n$. $\mathbf{x}_{\text{best}}$ is the optimal solution in the current set of iterations. Then, we establish a probability $P(t) = P\left(|\mathbf{X}^* - \mathbf{x}_t| \geq \varsigma\right)$, the chance that FBS algorithm converges to the global optimal solution, and ς represents a fixed variable with a small value. According to the generation rules of Fibonacci branch $P(t) = P_X(t) + P_\varsigma(t)$. $P_X(t)$ is the probability of generating uniformly distributed random points in the definition domain of Rule one, and $P_\varsigma(t)$ is the probability of generating uniformly distributed random search points in a region defined by the radius parameter ς based on Rule two. Therefore, after t iterations by FBS, the probability of the search points not falling into the region of ς spacing distance around the global optima $\mathbf{X}^*$ is $\widetilde{P}(t) = P\left(|\mathbf{X}^* - \mathbf{x}_t| \geq \varsigma\right)$, then we have $\lim_{t\to+\infty} \widetilde{P}(t) = 0$, hence, the following results can be obtained

$$P\left(n < +\infty\right) > P\left(t\right) = 1 - \widetilde{P}\left(t\right), \quad P\left(n < +\infty\right) > 0 \tag{22}$$

Let $t \to +\infty$, and we can get

$$P\left(n < +\infty\right) = 1 \tag{23}$$

So FBS converging to the global optimal solution with probability 100% can be demonstrated and proven.

From the above proof of Section 3.5.1 and Section 3.5.2, we can make a comprehensive conclusion that the proposed FBS algorithm is effective, convergent, and accessible for the global optimal solution of the objective function.

4. SIMULATION RESULTS

4.1. Verification of the Proposed FBS

In order to validate and analyze the efficiency and effectiveness of the proposed FBS, the algorithm is verified from the following aspects in simulation experiments:

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

(2) The convergence of the FBS is proved and discussed by the presented gradient of the iteration curves which manifest the convergence rate speed and the 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 details of the parameter settings for every heuristic algorithm used in the experiments are given in Table 1.

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

Particle Swarm Optimization (PSO)		Genetic Algorithm (GA)		Comprehensive Learning PSO (CLPSO)	
Population size	$Np = 20$	Population size	$Np = 20$	Learning Probability	$0.05 \sim 0.5$
Cognitive ratio	$c1 = 2$	Mutation probability	$Pm = 0.05$	Population size	$Nc = 20$
Social coefficient	$C2 = 2$	Cross probability	$Pc = 0.7$	Cognitive ratio	$c1 = 2$
inertia weight	$0.4 \sim 0.9$	Rate of chromosome elite	$Pe = 0.2$	Social coefficient	$C2 = 2$
				inertia weight	$0.4 \sim 0.9$
Differential Evolution (DE)		**Artificial Bee Colony (ABC)**		**Fibonacci Branch search (FBS)**	
Population size	$Np = 20$	colony size	$Cs = 20$	Nested branch depth	2
scaling factor	$F = 0.6$	onlooker bees percentage	$Lp = 50\%$	Total branch depth	6
Crossover rate	$CR = 0.8$	scout bees	$Sb = 1$	Search Space	[Min, Max]

All the experimental tests are implemented on Intel (R) Core (TM) i7-7700 HQ Core Processor @2.8 GHz and 2.8 GB RAM, and all the meta-heuristics algorithms are coded and carried out in Matlab 2017b version under the Windows10 Professional.

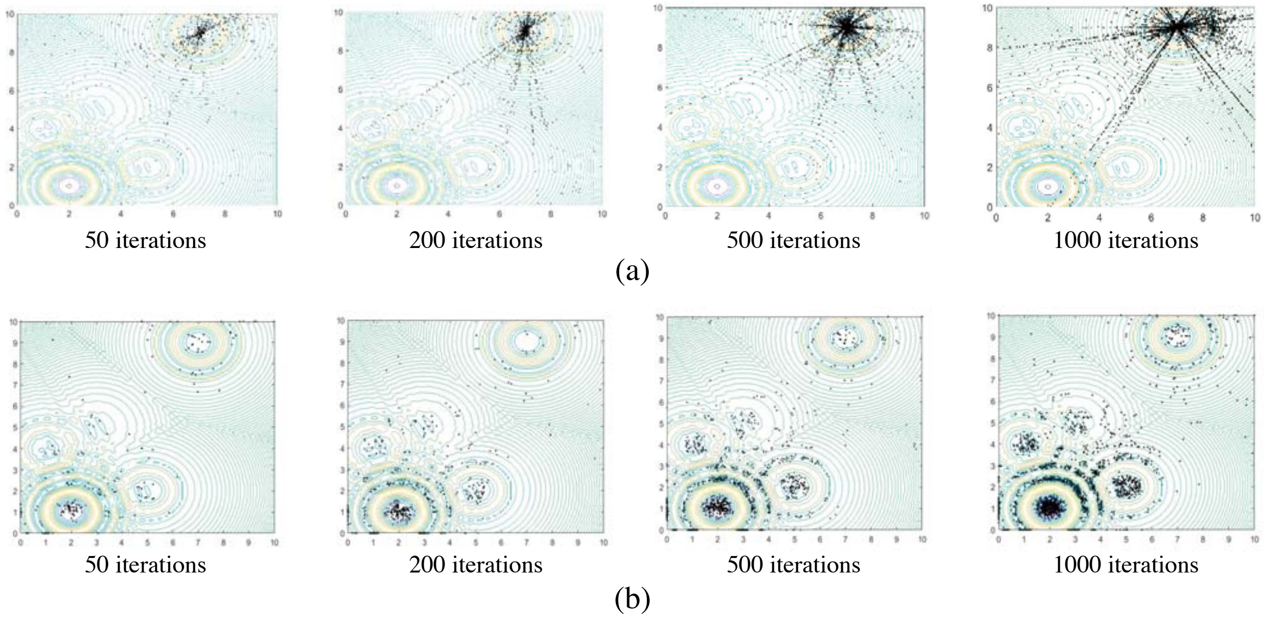
4.1.1. The Location History of the Search Points in FBS for Langermann Function

In this section, the global optimization ability of the proposed FBS is demonstrated by employing the location history of the search points during optimization process for locating the global optimum solution rather than trapping into local optimization of the benchmark example, with results compared against 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 which is taken from [25, 26] and is summarized in Table 2. As can be found in Table 2, two typical extreme points exist in the function. The 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 5.

Table 2. Langermann benchmark function.

Extreme point	Extreme point 1	Extreme point 1
Evaluation in [25]	global optimum solution	local suboptimal solution
Langermann function	$f_L\,(2.003, 1.006) = -5.1612$	$f_L\,(7, 9) = -3$

The performance of the proposed FBS in terms of the movement trajectory of the search points scattering around the best solutions and converge towards the optimal point in the search space for Langermann are illustrated in Figure 6(b). This figure shows that the FBS model is able to simulate the position history of search points in three dimensional and trajectory contour plots over different iterations. For the verification of the results, we compare our algorithm to PSO in the same manner



50 iterations 200 iterations 500 iterations 1000 iterations

(a)

50 iterations 200 iterations 500 iterations 1000 iterations

(b)

Figure 6. Location history of the search points for the Langermann function over the course of different iterations. (a) Visualization results of the points location history in contour plots for PSO. (b) Visualization results of the points location history in contour plots for FBS.

and provide the results in Figure 6(a). The initial positions of the search points in both FBS and PSO are set at the extreme point 2.

As the results exhibited in Figure 6, the search points tend to explore the promising regions of the search space and cluster around the global optima eventually in multimodal Langermann pattern. From the results depicted in Figure 6(a), we can know that as the number of iterations increases, the points of PSO algorithm gradually cluster around the extreme point 2 and proceed toward local optima, and almost no particles enter the region near the global optimum extreme point 1, providing the 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 points as shown in Figure 6(b) that although the Langermann function is non-symmetric and multimodal with different levels of peaks, finding its global optimum is challenging due to many local minima in the search space. Remarkably, FBS is extricated from the initial local optimum at extreme point 2 and jumps out of the trapped solution in a local optimum point assisted by global random searching. It is evident from the location history of the search points during the process of converging toward the global optima that the points grow towards the optimal point from the area of initialization, tending to scatter around extreme points gradually and moving towards the best solutions in the search space in both 2D and 3D spaces over the course of iteration. More than half of the agents have already approached the global optimum valley after the first 50 iterations and begin converging on the optimum. As iteration increases, more agents aggregate at the extreme points and scatter around the extreme point, especially attracted intensively at the global optimum target region. Eventually, the search points find the global optimum and converge toward the global optima. This can be discussed and reasoned according to the global randomness concepts introduced by the endpoints $\mathbf{X}_B$ which is 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 by global searching in space, the best universe is saved and moved to the next iteration. Consequently, these behaviors and abilities will assist the FBS not to become trapped in local optima and converge towards the target point quickly in the iterations of optimization.

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

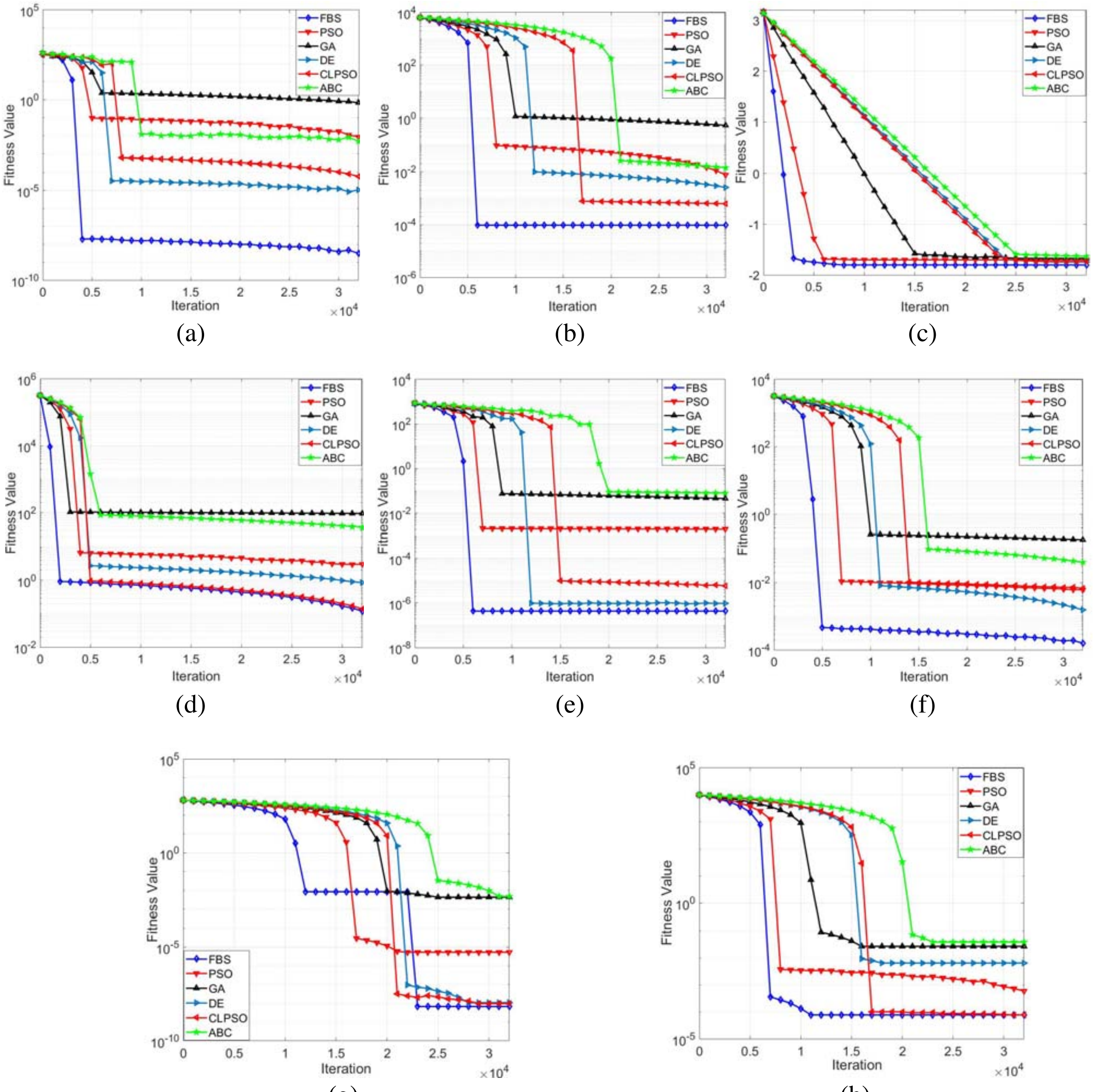
4.1.2. Convergence Performance of the Multimodal Function

To confirm the convergence behaviour of the proposed algorithm, in this subsection, we provide the convergence curves that exhibit the objective fitness value of the typical benchmark functions obtained by the best solutions so far in each iteration. A large set of complex mathematical benchmark functions to be tested are 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 exploration. The illustrated results are compared against those of PSO, GA, CLPSO, DE, and ABC metaheuristic algorithms on the same set of multi-dimensional numerical benmark functions. The properties and formulas of these functions are presented below.

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	$\sum_{i=1}^{D} \left(x_i^2 - 10\cos(2\pi x_i) + 10\right)$	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	$\sum_{i=1}^{D} 100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2$	10	$[-2.048, 2.048]$	0
F5	Ackley	$-20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{D} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{D}\cos(2\pi x_i)\right) + 20 + e$	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	$\sum_{i=1}^{D}\left(\sum_{k=0}^{k\max}\left[a^k\cos\left(2\pi b^k(x_i + 0.5)\right)\right]\right) - D\sum_{k=0}^{k\max}\left[a^k\cos\left(2\pi b^k(x_i \times 0.5)\right)\right]$ $a = 0.5,\ b = 3,\ k\max = 20$	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

Figure 7 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 and converges fast. FBS achieves better results on all multimodal groups than the compared algorithms, and it surpasses all other algorithms apparently on functions 1, 2, 5, and 6, and especially significantly improves the results on functions 1 and 2. The other algorithms show poor performance on the complex problems since they miss the global optimum basin to approach the optimal fitness. The Schwefel's function is a good example, as it traps all other algorithms in local optima, while the FBS successfully avoids falling into the deep local optimum which is far from the global optimum. On the complex multimodal functions with randomly distributed local and global optima, FBS performs the best. It should also be noted that the FBS algorithm in the graphs exhibits superiority regarding the convergence speed over the other five algorithms, and it converge to a global optimum solution with less fitness evaluations and terminates after no more than 5000 iterations on functions 1, 3, and 4, and always converges faster than others on the remaining function problems.



(a) (b) (c)

(d) (e) (f)

(g) (h)

Figure 7. Convergence behavior of the FBS and other optimization algorithms on 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.

These figures also prove that FBS not only improves the accuracy of the approximated optimum initial, but also desirably enhances the convergence speed over the course of iterations that make it converge faster than the other algorithms. Global random property and space region shortening fraction guarantees a satisfactory convergence speed. Other algorithms could not converge as fast as the FBS, since they have a large potential search space. The proposed FBS combines a global searching method and local optimization strategy together to yield a balanced performance that achieves better fitness and faster convergence. Besides, the convergence speed is a crucial parameter of real time applications like adaptive beamforming system, thus, the FBS is highly suitable and affordable for ABF.

4.1.3. Minimization Result of the Tested Benchmark Functions

In this subsection, experiments are conducted on a suite of multimodal functions illustrated in Table 3 to evaluate six optimization algorithms including the proposed FBS. All the test functions are minimized, and the relevant information can be found in [26, 27] for the standard benchmark functions, respectively. The number of iterations for conducting the experiments is 400. For the selected benchmarking problems F1–F8, the dimension of these functions is set to 10. Every algorithm runs 1000 times independently to reduce the statistical error and achieve reliable results [28].

The statistical results considering the average value and the standard deviation function fitness value as well as the success rate (SR) needed to reach the acceptable solution are summarized in Table 4. For the results shown in Table 4, the smaller the mean value is, the better the performance of

Table 4. The comparative and statistical results for benchmarking function problems F1–F8. (SR: success rate).

Method	F1.Griewank				F2.Rastrigin			
	Mean	Stev	SR	Time	Mean	Stev	SR	Time
ABC	4.73E − 03	6.51E − 03	52%	88.20 s	1.43E − 02	4.59E − 03	30%	85.46 s
DE	7.38E − 06	6.51E − 07	100%	86.51 s	2.55E − 03	7.18E − 04	55%	80.03 s
GA	6.81E − 01	7.49E − 02	20%	79.36 s	5.29E − 01	1.21E − 02	11%	73.49 s
PSO	5.27E − 03	3.45E − 03	49%	82.32 s	7.61E − 03	8.29E − 02	38%	77.25 s
FBS	2.54E − 09	1.47E − 08	100%	68.12 s	9.53E − 05	2.17E − 06	86%	63.70 s
CLPSO	6.01E − 05	4.32E − 06	100%	74.94 s	6.18E − 04	7.18E − 03	74%	69.25 s
Method	F3.Michalewicz2				F4.Rosenbrock			
	Mean	Stev	SR	Time	Mean	Stev	SR	Time
ABC	−1.6325E + 00	1.56E − 03	32%	81.20 s	3.67E + 01	1.90E + 00	0%	80.30 s
DE	−1.7428E + 00	8.36E − 07	90%	78.53 s	8.37E − 01	3.27E − 01	8%	76.46 s
GA	−1.6715E + 00	4.83E − 04	44%	69.37 s	9.42E + 01	2.15E − 02	0%	68.04 s
PSO	−1.7002E + 00	3.95E − 03	84%	72.02 s	2.69E + 00	1.90E + 00	6%	71.84 s
FBS	−1.8013E + 00	7.48E − 06	100%	58.92 s	2.98E − 02	5.61E − 04	29%	60.43 s
CLPSO	−1.7332E + 00	6.28E − 05	90%	66.21 s	4.39E − 02	4.03E − 01	12%	65.44 s
Method	F5.Ackley				F6.Schwefel			
	Mean	Stev	SR	Time	Mean	Stev	SR	Time
ABC	8.87E − 03	2.98E − 02	47%	84.74 s	3.85E − 02	3.53E − 03	18%	79.03 s
DE	9.27E − 07	9.69E − 07	100%	79.53 s	1.53E − 03	6.25E − 06	68%	74.36 s
GA	1.46E − 02	5.32E − 03	29%	76.92 s	1.85E − 01	3.74E − 03	11%	70.37 s
PSO	6.09E − 05	5.49E − 04	92%	78.27 s	5.87E − 03	3.68E − 04	66%	73.09 s
FBS	4.34E − 07	5.74E − 08	100%	67.94 s	2.53E − 04	6.25E − 06	74%	65.57 s
CLPSO	5.74E − 06	5.84E − 08	100%	72.82 s	6.53E − 03	7.43E − 03	59%	69.93 s
Method	F7.Weierstrass				F8.Salomon			
	Mean	Stev	SR	Time	Mean	Stev	SR	Time
ABC	6.13E − 05	9.09E − 06	100%	93.85 s	3.85E − 02	1.12E − 03	19%	99.72 s
DE	1.08E − 08	4.17E − 09	100%	87.32 s	6.43E − 03	6.82E − 04	47%	93.73 s
GA	4.28E − 03	9.64E − 06	82%	79.43 s	2.67E − 02	6.63E − 02	29%	87.94 s
PSO	5.27E − 06	3.64E − 06	100%	84.38 s	5.74E − 04	2.63E − 03	68%	89.61 s
FBS	6.77E − 09	7.54E − 11	100%	73.87 s	7.74E − 05	9.48E − 05	85%	79.36 s
CLPSO	9.65E − 09	8.53E − 10	100%	75.38 s	6.24E − 05	9.48E − 04	92%	83.49 s

algorithm is. The lower the standard deviation value is, the stronger the stability of algorithm is. As seen, for most benchmark data sets, the average value and standard deviation calculated by the FBS are both smaller than those other algorithms, and the proposed algorithm surpasses all other algorithms on functions 1, 2, 3, 5, 6, and 7, and especially significantly improves the results on functions 1 and 3. When the other algorithms find their own best fitness of these functions, the proposed FBS could still search better fitness closest to the optimal value. The CLPSO achieves similar results to the FBS on function 7, and they both are much better than the other variants on this problem. The DE also performs well on multimodal problems. The DE performs similarly to the FBS on functions 1, 3, 5, and 7. However, the FBS performs better on much more complex problems when the other algorithms miss the global optimum basin.

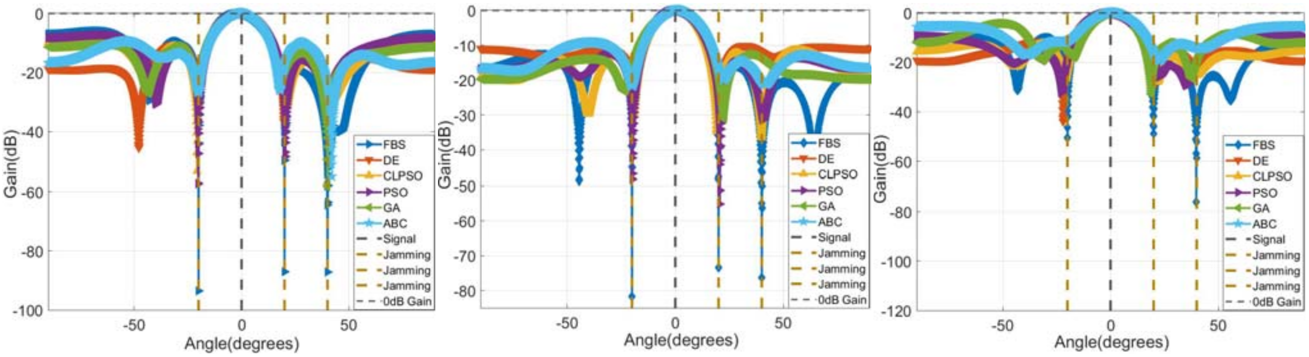
As a result, in terms of performance in the global search ability and the optimization stability for benchmarking function, the proposed FBS outperforms all other heuristics algorithms on the tested functions. In addition, this table illustrates that FBS in comparison with others displays higher percentage and accuracy reaching the acceptable solutions on these test functions. For the mean reliability of test functions F1, F3, F5, F7, FBS exhibits the highest reliability with a 100% success rate and smallest average errors. This performance superior property is due to the FBS's interactive updating rule. With the new updating rule and global randomness, different dimensions may learn from different examples based on the historically optimal search experience, and the FBS explores a larger search space than the others. Because of this, the FBS performs comparably to or better than many meta-heuristic algorithms on most of the multimodal problems experimented in this section.

4.2. The Performance of the FBS Optimization Result in Adaptive Beamforming

To demonstrate the benefits of the FBS optimization with application to adaptive beamforming, in this part of the section, several groups of simulation experiments are conducted using Matlab R2017b. The desired GNSS signal is in the form of QPSK modulation mode with 0° for incident azimuth angle, and the power of GNSS signal is extremely weak on the ground and far less than noise; therefore, the selected SNR should not be too large in our simulation scenes. All the impinged signals are arranged coming from zeniths 45°. Simulation environment is additive white Gaussian noise channel (AWGN). We have taken the structure size of the search element in weight vector to be five and the number of search points per element in the first set 10 layers for Fibonacci branch increases according to the corresponding Fibonacci sequence, and the sum of the element nodes in all the layer of search element is 19. The maximum number of iterations in FBS is 400. 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 signal-to-interference-plus-noise ratio (SINR) of beamformer system. The performance criteria of the given beampattern in Section 5.2 are the nulling level of interference, and the given criterion that we will use to compare the numerical results and make the comparison of SINR is the numerical value.

Four groups of simulation cases corresponding to the defined measure metrics aspects conducted in terms of various experimental critera are considered in this study. The effectiveness of the proposed FBS based beamformer is investigated by the power patterns formed by extracted weights in terms of different input SNRs in the first case. The second case considers the different numbers of array elements for system output SINR using FBS by the suitable weight vectors. The performance measurement on the number of interference sources of output SINR is studied in the third case. The rest case concerns the impact of different INRs of input interference on output SINR in front of a ULA system.

The proposed FBS-based beamforming method is compared with the following five conventional heuristics-based beamforming algorithms: 1) the differential evolution (DE) based beamforming method of [29], 2) the particle swarm optimization (PSO) based beamforming method of [30], 3) the comprehensive learning particle swarm optimization (CLPSO) based beamforming method of [31], 4) the artificial bees colony (ABC) based beamforming method of [32], 5) Genetic Algorithm (GA) based beamforming method of [33]. Besides, in order to fully evaluate FBS-based beamformer comprehensively and accurately from all aspects, the performance of the classic closed-form method [12] for ABF has been presented and compared with the other algorithms specially and separately in corresponding tables. A total of 300 repetitions (independent trials) are implemented and then averaged to obtain each figure of the results.



(a) (b) (c)

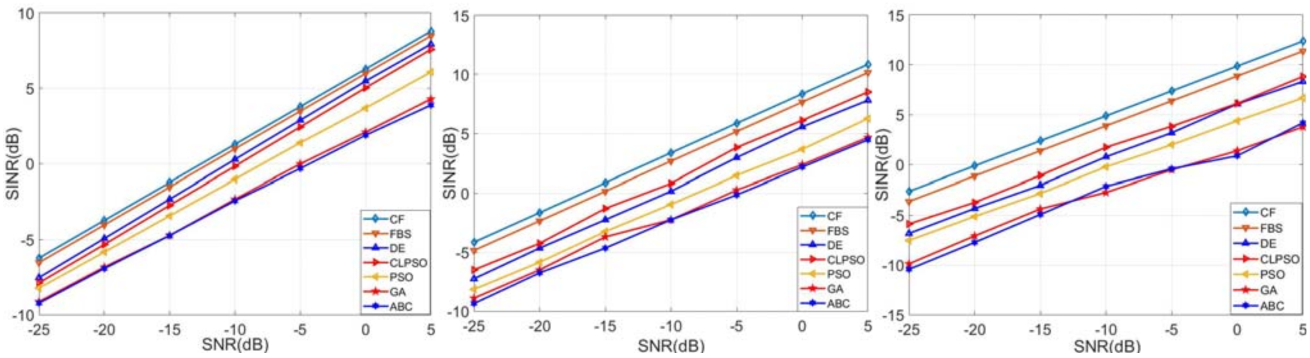
Figure 8. Comparison of the beampattern performance synthesized by the heuristic algorithms with different SNR. (a) SNR = −30 dB; (b) SNR = −10 dB; (c) SNR = 10 dB.

Table 5. Average nulling levels of the FBS and the closed-form method.

Scenario / Method	SNR = −30 dB	SNR = −10 dB	SNR = 10 dB
FBS	−93.02 dB	−112.95 dB	−129.37 dB
closed-form	−98.45 dB	−121.67 dB	−143.29 dB

4.2.1. Input SNR Evaluation of the Beampattern

We first examine the beampattern performance synthesized by the FBS and compared it to the others given in terms of input SNR in this case. A uniform linear array with 6 omnidirectional antenna elements is considered in the simulation. For investigating the effect of the input SNR with different levels, we consider three sets of input SNRs with SNR = −30 dB, SNR = −10, and SNR = 10 to demonstrate the validity of our approach. Figure 8 shows the behavior 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 an inerratic beampattern with deeper nulling (with nulling level exceeding −70 dB) towards the interference compared to the other algorithms. The proposed FBS-based adaptive beamformer has 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 the interference nulling performance for a higher SNR level, i.e., SNR = 10. As the level of SNR decreases, the compared beamformers, especially ABC-based beamformer and GA-based beamformer suffer from performance degradation of the corresponding metaheuristic-based beampatterns, and the nulls do not align precisely with the interference sources. This indicates and demonstrates that the proposed algorithm is more stable and finds better solutions with greater precision in ABF application. To more clearly illustrate and comprehensively examine the nulling degrees of the proposed FBS, we compare the results to the traditional classic closed-form method, and the separately independent average nulling levels of the FBS and the closed-form method are illustrated in Table 5. We can see from the table that compared to the sidelobe level of the closed-form method which performs the best among the compared methods in each case and is 46.79, 52.96, and 71.95 dB for the classic closed-form-based beamformer, either the objective function to be optimized or less computationally extensive, and more computationally stable routines are required or else, specific characteristics of a statistical decision-making problem are desired of; therefore, the superiority of computational complexity for FBS is evident relative to it. The specific experimental comparison results and computational complexity performance advantages of the FBS algorithm are described in Section 4.3.



(a) (b) (c)

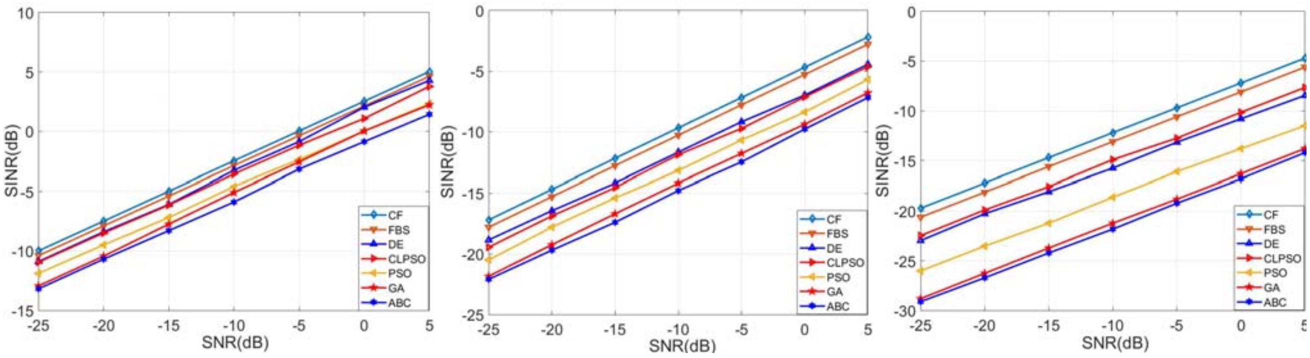
Figure 9. SINR performance versus SNR of the heuristic algorithms with varied array elements. (a) Six elements; (b) ten elements; (c) fourteen elements.

4.2.2. Array Elements Assessment of Output SINR

In this section, the output SINR performance is measured by the number of array elements in three various conditions in terms of the increasing input SNR value, and the SNR is assumed to be changed from −25 dB to 5 dB (centered at −10 dB) in 5 dB steps. The interference with 5 dB INR is considered in the experiment. The linear arrays in different scenarios are considered to be composed respectively of 6, 10, and 14 elements. The results of the output SINR for heuristic-based beamformer and the closed-form method are illustrated in Figure 9. From the graphs, it can be noted that closed-form method achieves the highest SINR value among all the algorithms. The proposed algorithm achieves better performance than the other optimization algorithms for all the array element scenarios and is able to achieve near optimal closed-form performance over the entire range of the input SNR values. CLPSO yields suboptimal higher values of SINR, but FBS yields optimal SINR values consistently in all cases. Moreover, we can also observe that the performance difference in reaching optimum weight vectors between the FBS and other algorithms is increased for the array element increment in each algorithm, which is consistent with our previous analysis owing to the increasing search dimension of the weight vectors solution in array element. With a view to the above fact, the global search capability of the proposed FBS algorithm could achieve the improvement of the output SINR more so than the compared metaheuristic-based beamformer in adaptive beamformer system.

4.2.3. Investigation of INR on the Output SINR

A ULA consisting of 6 monochromatic isotropic elements with different INRs is considered in this scenario. Three groups INRs of the interference are set as 5 dB, 20 dB, 35 dB which are established in different simulation scenarios. Figure 10 displays the SINR performance of these techniques versus the SNR under different power levels of interference sources by using the proposed and other algorithms. From the results depicted in Figure 10 we can know that in general, the optimization algorithms are all able to achieve near-satisfactory closed-form based beamformer SINR performance in the situation of INR = 5 dB, which means the lowest power process of the interference signal. The proposed algorithm achieves improved performance in SINR compared to the other algorithms in all the simulations even under the most severe interference situations when the value of INR is 30 dB. With the increase of INR for the interference, the SINR performances of all the algorithms are degraded, and the proposed algorithms have evident advantages over these algorithms. Therefore, note that the proposed algorithm can perform more robust results and suitable precisions in adaptive beamforming for high interference power levels.



(a) (b) (c)

Figure 10. SINR performance versus SNR of the heuristic algorithms with different INR. (a) INR = 5 dB; (b) INR = 20 dB; (c) INR = 35 dB.

4.2.4. Examination of the Interferences Number Impact on Output SINR

The simulations in this case are conducted to validate the effect of the interference quantity to the output SINR performance. The beamformer is equipped with 6 array elements, and the INR is fixed to 10 dB for the received different numbers of interference signals. Table 6 illustrates the different numbers of interferences and the corresponding incident angle values of the above-mentioned interferences. The SINR performance of the proposed FBS and other optimization algorithms for different numbers of interference signals is shown in Figure 11. As can be seen from Figure 11, closed form method performs the best among all those in comparisons for its individual algorithm solving characteristics. With regard to the proposed FBS and other optimization comparisons, when there are one interference at the receiver, all the optimization algorithms can achieve the close-to-optimal SINR performance by considering the requirement for maximizing SINR, and FBS demonstrates the best improvement, followed by DE. ABC shows the lowest value of SINR. As the number of interferences increases from one to six, the performance advantage of the FBS becomes more evident. The increase in the number of interference sources increases the difficulty of the optimization problem. Failure of ABC, GA, and PSO 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 application.

Table 6. 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

4.3. Time Complexity of the ABF Significant Model to Achieve Optimum Performance

In this part, the experiment results of the performed experiments between the FBS-based beamformer and the classic closed form method are presented in the form of the time complexity in finding the global optima in the ABF significant model. The percentage improvement of the proposed FBS in terms of

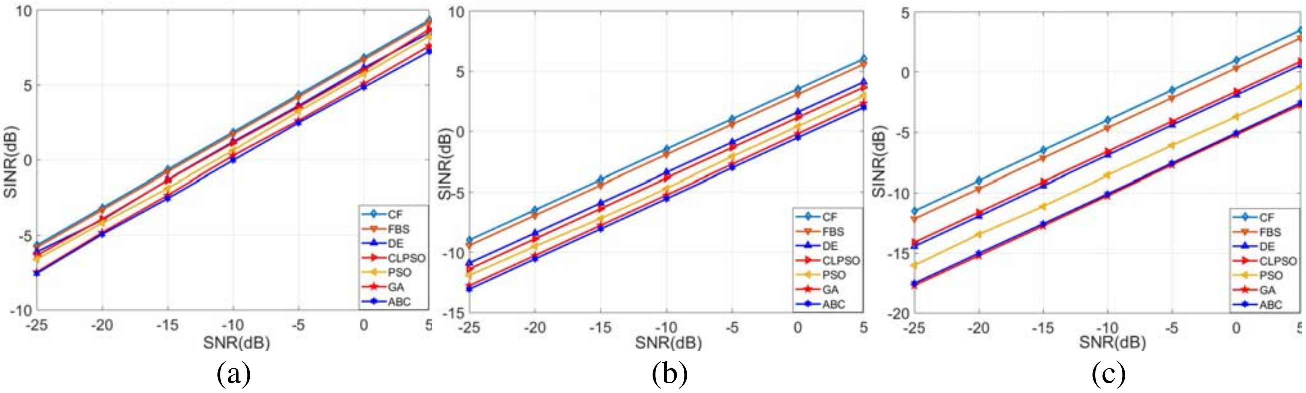


Figure 11. SINR performance versus SNR of the heuristic algorithms with different number of interferences. (a) One interference; (b) three interferences; (c) six interferences.

nulling level is also illustrated in this part. The methods are applied to optimize a ULA consisting of isotropic elements. Different cases with 14, 18, and 22 array elements are simulated to further validate the proposed approach for real-world large-array with more elements applications. The number of iterations for conducting the experiments is 200. Under the same computer hardware configuration in Section 4.1, the average CPU time consumption (in seconds) at key points is measured by the built-in 'Matlab Profiler', which determines the computational complexity proportions.

The optimization results considering the CPU time as well as the time consumption corresponding to computation load for the significant ABF model of the proposed FBS and the comparison algorithms under different numbers of array elements are listed in Table 7, Table 8, and Table 9.

Table 7. Comparison results of the algorithms with 12 array elements.

Algorithm	CPU time (s)	Percentage Improved
FBS	17.26	171.26%
Closed Form	46.82	Initial Comparison

Table 8. Comparison results of the algorithms with 16 array elements.

Algorithm	CPU time (s)	Percentage Improved
FBS	20.68	233.70%
Closed Form	69.01	Initial Comparison

Table 9. Comparison results of the algorithms with 20 array elements.

Algorithm	CPU time (s)	Percentage Improved
FBS	32.74	278.53%
Closed Form	123.93	Initial Comparison

From the above tables, it can be observed that the proposed FBS which achieves the near optimal nulling level possesses a huge advantage in computational complexity compared to closed form method in all cases. FBS requires far fewer evaluations and lower CPU time than closed form method consistent with the computational costs, and FBS presents a more evident advantage over the algorithm in the large array element cases. CPU time cost of FBS in each case is 17.26 s, 20.68 s and 32.74 s, and the improvements of FBS over the closed form are 171.26%, 233.70%, 278.53%. This indicates that FBS requires much fewer function evaluations, and the closed-form approach needs to compute the matrix inverse process which introduces high computational complexity especially for large number of array elements, as shown in Table 8 and Table 9, since more array elements result in a larger dimensionality of the weight vector solution, which requires a greater search time and number of computations to optimize the beamformer. Although FBS achieves slightly less deep nulling level than the closed-form solving algorithm, the percentage improvement of the CPU time can compensate and provide an acceptable performance for the proposed FBS in the experiment for certain real-time beamforming situations. Overall, the FBS approach is quite competitive in complexity computation and the global search optimization compared with the other methods.

5. CONCLUSIONS

In this paper, we present a novel heuristic algorithm, called Fibonacci branch search, for achieving the improved performance of adaptive beamforming. The interactive global and local searching rules are proposed to reduce the probability of falling into the local optima, and the global randomness characteristic and space region shortening fraction guarantee the convergence velocity in global optimization process. In addition, we devise a specific implementation architecture based on FBS for adaptive beamformer, and the amplitudes and phases of the weight vector acting as the solution were acquired in the search space by FBS. The beamforming results synthesized by the vectors are compared with conventional metaheuristic-based beamforming algorithms.

The simulation experiments in Section 4 demonstrate that the proposed FBS achieves increased SINR over the suboptimally performing algorithm refereed to DE, CLPSO for different cases. In addition, the significant improvements over PSO, GA, ABC have been achieved in separate scenarios. With respect to computation complexity, FBS avoids the inverse calculation of the interference correlation matrix compared to the classic closed form solving algorithm. Therefore, it is recognized from academic and practical implications that the global optimization ability and fast convergence searching of FBS-based beamformer would serve significant purpose for real-time spacial filtering electronic countermeasures and large array elements of antenna design in military confrontation field. However, the limitation of FBS caused by the uncertainty of the frequent changing outliers is a major issue that needs careful consideration, and the technique may have unacceptable nulling performance corresponding to dynamic and various interference scenarios. As a future work, the FBS will be explored to apply in a more complicated time-varying situation in ABF field.

ACKNOWLEDGMENT

This research was funded by the National Natural Science Foundation of China (Grant No. 61272333) and the National Key Laboratory of science and Technology Fund (Grant No. 9140C130502140C13068).

REFERENCES

1. Huang, X., L. Bai, I. Vinogradov, and E. Peers, "Adaptive beamforming for array signal processing in aeroacoustic measurements," *Journal of the Acoustical Society of America*, Vol. 131, No. 3, 2152–2161, 2012.
2. Daneshmand, S., N. Sokhandan, M. Zaeriamirani, and G. Lachapelle, "Precise calibration of a GNSS antenna array for adaptive beamforming applications," *Sensors*, Vol. 14, No. 6, 9669, 2014.
3. Leight, J. and B. Toland, "Photonic beamforming technologies for advanced military and commercial SATCOM antennas," *Aerospace Conference*, 1999.
4. Zhao, H., B. Lian, and J. Feng, "Space-time adaptive processing for GPS anti-jamming receiver," *Physics Procedia*, Vol. 33, No. 1, 1060–1067, 2012.

5. Synnevag, J. F., A. Austeng, and S. Holm, "Adaptive beamforming applied to medical ultrasound imaging," *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, Vol. 54, No. 8, 1606–1613, 2007.
6. Khamy, S. E. E. and A. M. Gaballa, "Adaptive arrays for MC-CDMA using the MSINR guided multimodulus algorithm," *2008 National Radio Science Conference*, 2008.
7. Chen, H. W. and J. W. Zhao, "Wideband MVDR beamforming for acoustic vector sensor linear array," *IEE Proceedings — Radar, Sonar and Navigation*, Vol. 151, No. 3, 158–162, 2004.
8. Mu, P., L. Dan, and Q. Yin, "A robust MVDR beamforming based on covariance matrix reconstruction," *International Conference on Graphic & Image Processing*, 2011.
9. Sinha, P., A. D. George, and K. Kim, "Parallel algorithms for robust broadband MVDR beamforming," *Journal of Computational Acoustics*, Vol. 10, No. 1, 69–96, 2002.
10. Shahbazpanahi, S., A. B. Gershman, and Z.-Q. Luo, "Robust adaptive beamforming using worst-case SINR optimization: A new diagonal loading-type solution for general-rank signal models," *Proceedings, IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2003.
11. Mozaffarzadeh, M., A. Mahloojifar, M. Orooji, K. Kratkiewicz, S. Adabi, and M. Nasiriavanaki, "Linear-array photoacoustic imaging using minimum variance-based delay multiply and sum adaptive beamforming algorithm," *Journal of Biomedical Optics*, Vol. 23, No. 2, 026002, 2018.
12. Shahab, S. N., A. R. Zainun, H. A. Ali, M. Hojabri, and H. Nurul, "MVDR algorithm based linear antenna array performance assessment for adaptive beamforming application," *Journal of Engineering Science and Technology*, Vol. 12, No. 5, 1366–1385, 2017.
13. Wei, C. and Y. Lu, "Adaptive beamforming for arbitrary array by particle swarm optimization," *IEEE International Conference on Computational Electromagnetics*, 2015.
14. Vitale, M., G. Vesentini, N. N. Ahmad, and L. Hanzo, "Genetic algorithm assisted adaptive beamforming," *Proceedings IEEE 56th Vehicular Technology Conference*, 2002.
15. He, L. and S. Huang, "Modified firefly algorithm based multilevel thresholding for color image segmentation," *Neurocomputing*, Vol. 240, 152–174, 2017.
16. Sun, K., S. Mou, J. Qiu, T. Wang, and H. Gao, "Adaptive fuzzy control for non-triangular structural stochastic switched nonlinear systems with full state constraints," *IEEE Transactions on Fuzzy Systems*, Vol. 27, No. 8, 1587–1601, 2018.
17. Qiu, J., K. Sun, T. Wang, and H. Gao, "Observer-based fuzzy adaptive event-triggered control for pure-feedback nonlinear systems with prescribed performance," *IEEE Transactions on Fuzzy Systems*, Vol. 27, No. 11, 2152–2162, 2019.
18. Liao,, B. and S. C. Chan, "Adaptive beamforming for uniform linear arrays with unknown mutual coupling," *IEEE Antennas and Wireless Propagation Letters*, Vol. 11, 464–467, 2012.
19. Etminaniesfahani, A., A. Ghanbarzadeh, and Z. Marashi, "Fibonacci indicator algorithm: A novel tool for complex optimization problems," *Engineering Applications of Artificial Intelligence*, Vol. 74, 1–9, 2018.
20. Subasi, M., N. Yildirim, and B. Yildiz, "An improvement on Fibonacci search method in optimization theory," *Applied Mathematics and Computation*, Vol. 147, 893–901, 2004.
21. Yildiz, B. and E. Karaduman, "On Fibonacci search method with k-Lucas numbers," *Applied Mathematics and Computation*, Vol. 143, 523–531, 2003.
22. Omolehin, J. O., M. A. Ibiejugba, A. E. Onachi, and D. J. Evans, "A Fibonacci Search technique for a class of multivariable functions and ODEs," *International Journal of Computer Mathematics*, Vol. 82, 1505–1524, 2005.
23. Ramaprabha, R., M. Balaji, and B. L. Mathur, "Maximum power point tracking of partially shaded solar PV system using modified Fibonacci search method with fuzzy controller," *International Journal of Electrical Power & Energy Systems*, Vol. 43, 754–765, 2012.
24. Wang, X., D. J. Lyu, Y. Dong, et al., "Cutting parameters multi-scheme optimization based on Fibonacci tree optimization algorithm," *Control and Decision*, Vol. 33, 1373–1381, 2018.
25. Kaid Omar, O., F. Debbat, and A. Boudghene Stambouli, "Null steering beamformer using hybrid algorithm based on Honey Bees Mating Optimisation and Tabu Search in adaptive antenna array," *Progress In Electromagnetics Research C*, Vol. 32, 65–80, 2012.

26. Ng, C. K. and D. Li, "Test problem generator for unconstrained global optimization," *Computers & Operations Research*, Vol. 51, No. 51, 338–349, 2014.

27. Yang, Y. and Y. Shang, "A new filled function method for unconstrained global optimization," *Mathematical Problems in Engineering*, Vol. 8, No. 1, 501–512, 2010.

28. Saeed, S., H. C. Ong, and S. Sathasivam, "Self-adaptive single objective hybrid algorithm for unconstrained and constrained test functions: An application of optimization algorithm," *Arabian Journal for Science and Engineering*, Vol. 44, No. 4, 3497–3513, 2019.

29. Mallipeddi, R., J. P. Lie, P. N. Suganthan, S. G. Razul, and C. M. S. See, "A differential evolution approach for robust adaptive beamforming based on joint estimation of look direction and array geometry," *Progress In Electromagnetics Research*, Vol. 119, 381–394, 2011.

30. Banerjee, S. and V. V. Dwivedi, "Performance analysis of adaptive beamforming using particle swarm optimization," *11th International Conference on Industrial and Information Systems (ICIIS)*, 242–246, IEEE, 2016.

31. Ismaiel, A. M., E. Elsaidy, and Y. Albagory, "Performance improvement of high altitude platform using concentric circular antenna array based on particle swarm optimization," *AEU-International Journal of Electronics and Communications*, Vol. 91, 85–90, 2018.

32. Ruchi, R., A. Nandi, and B. Basu, "Design of beam forming network for time-modulated linear array with artificial bees colony algorithm," *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, Vol. 28, No. 5, 508–521, 2015.

33. Yeo, B. K. and Y. Lu, "Adaptive array digital beamforming using complex-coded particle swarm optimization-genetic algorithm," *Microwave Conference*, 2006.

Errata to "A Beamformer Design Based on Fibonacci Branch Search"

by Tianbao Dong, Haichuan Zhang, and Fangling Zeng

in *Progress in Electromagnetic Research B*, Vol. 88, 73–95, 2020

Tianbao Dong[1], Haichuan Zhang[2, *], and Fangling Zeng[1]

1. Correction of Second Author's Affiliation: The affiliation of the second author, Haichuan Zhang, was incorrectly listed as: School of Electronic Countermeasures, National University of Defense Technology. This was an oversight on our part during the final proofreading stage. The correct affiliation should be: Northwest Institute of Nuclear Technology.

2. Designation of Corresponding Author: During the initial submission, Dr. Haichuan Zhang was designated as the corresponding author. However, in the published PDF, this was mistakenly assigned to another author, likely due to an error on our part during the revision process. We request that the corresponding author be corrected to Dr. Haichuan Zhang (wellsmith111111@gmail.com).

Received 19 September 2024, Added 24 September 2024

* Corresponding author: Haichuan Zhang (wellsmith111111@gmail.com).

[1] The School of Electronic Countermeasures, National University of Defense Technology, China. [2] Northwest Institute of Nuclear Technology, China.