

Research Paper

Cite this article: Zhang H, Zeng F (2020). Implementation of a novel Fibonacci branch search optimizer for the design of the low sidelobe and deep nulling adaptive beamformer. *International Journal of Microwave and Wireless Technologies* **12**, 660–677. <https://doi.org/10.1017/S1759078719001570>

Received: 28 July 2019

Revised: 19 November 2019

Accepted: 20 November 2019

First published online: 7 January 2020

Key words:

Fibonacci branch search; global optimal solution; interactive searching; robust adaptive beamformer; shortening fraction

Author for correspondence:

Haichuan Zhang,

E-mail: zhanghai4258@163.com

Implementation of a novel Fibonacci branch search optimizer for the design of the low sidelobe and deep nulling adaptive beamformer

Haichuan Zhang  and Fangling Zeng

School of Electronic Countermeasures, National University of Defense Technology, Hefei, China

Abstract

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

Introduction

As a versatile approach, adaptive beamforming (ABF) has received considerable attentions over the past several decades and became a fundamental technique for numerous applications including radio astronomy, applied acoustic, cognitive communications, and medical imaging [1–3]. ABF possesses the potential to optimize the radiation pattern in real-time, which obtained 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 [4].

Classic ABF techniques used to extract the respective array excitation weights are based on two main criteria: minimum mean square error (MMSE) and maximum signal-to-noise ratio (MSINR) [5, 6]. Representative ABF algorithm is the minimum variance distortionless response (MVDR) on the basis of MSINR criterion. Although this method is capable of suppressing the interference and improving system reliability, weights computed by MVDR are not able to form the deep nulls toward the interference source in various interference scenarios on account of the characteristics of this technique. Conventional methods to solve this problem are very time consuming and are unmanageable in ABF application. Another criterion for computing the array weights is to minimize the MMSE, one of the most widely used MMSE-based adaptive algorithms is least mean square approach, this method needs a training sequence of signal of interest to adjust the complex weights adaptively and minimize the difference between the array output for forming the optimum array pattern [7, 8]. However, it exhibits a high probability of tracking into local optimal solution, which may not be applicable for ABF in harsh environments. Consequently, the inherent shortcomings of the derivative-based ABF methods have compelled and motivated many researchers to explore meta-heuristics (MH) and evolutionary methods for overcoming these types of difficulties [9].

The main advantage of the evolutionary heuristics algorithms over the classical derived approaches is their ability to search the global optimum of the objective functions without using the derivatives of the objective functions. In addition, MH optimization algorithms have no requirements for extra iterative derivations or computationally extensive routines of ABF with objective functions. According to the above, an enormous amount of evolutionary algorithms has been dedicated to applying several optimization approaches for ABF problems in the past decades. Many literatures have shown that these algorithms are capable of finding global or strong local optima of non-linear multimodal functions with multidimensional solutions [10, 11]; therefore, weights of the beamformer extracted by the optimization techniques according to the fitness function defined by specific criterion can be used to place a maximum beam and null in an array pattern in specified locations. Compared to other evolutionary

algorithms, the PSO is much easier to implement and outperform better; thus, many examples have been successfully demonstrated and validated the design flexibility of PSO in the framework of ABF arrays [12–14].

Although the MH-based ABF algorithm requires relatively lower mathematical complexity than derivative-based or iterative-based ABF methods, there still exist the weaknesses and limitations in the application of ABF. Some optimization algorithms are highly dependent on starting points in the case of a large number of solution variables, 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 ABF solution space cannot be very small, in that case, the conventional evolutionary algorithm is not really applicable to ABF. Besides, the classic optimization methods are prone to get trapped in local minima and not reach the global optimum [15] when solving complex multimodal optimization problems of array weight extraction, resulting in a suboptimum beamforming performance. 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 [16, 17], and the complexity implementing the algorithms would also result in huge cost and hardware resources.

In consideration of the above-mentioned studies, we propose a novel interactive-based random iterative search strategy, called Fibonacci branch search (FBS) in this paper to deal with complicated optimization problems of ABF. The concept of the proposed FBS is defined from two aspects: The first one is the generation principle of Fibonacci branch architecture. The establishment of branch structure in FBS is built up on the optimization process of the searching points, and the shortening fraction is designed based on Fibonacci sequence for the generation of a set of optimization elements that consist of two types of searching points. The optimization endpoints search for the optimal solution according to the growth path mode of the branch. The second standpoint of the FBS concept is the construction of the interaction iteration that applies rules for the computation of the 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 toward 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 do not readily fall into the local optimum and are 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 FBS 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. To the best of our knowledge, the proposed novel FBS optimization algorithm has been applied for the first time in antenna array beamforming problems.

The reminder of this paper is organized as follows: ABF system model of ULA is described in section “ABF system model of

ULA and problem statement”. Section “Proposed FBS algorithm” presents the proposed FBS optimization algorithm. Later, section “FBS-based beamformer design and implementation of ABF with the proposed FBS” introduces the robust ABF based on FBS. The validation of the proposed FBS via benchmark functions and the simulation results are reported in section “Simulation results”. Section “Conclusions” gives the conclusion.

ABF system model of ULA and problem statement

Consider a uniform linear antenna array of M omni-directional array elements employed in ABF receiver, one desired signal and Q uncorrelated interferences which impinge on the array at k th snapshot can be expressed by [18]

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

where $s(k)$ and $i_i(k)$ are the desired GNSS signal and the i th interference, $\mathbf{n}(k)$ 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

$$\begin{aligned} \mathbf{a}(\theta_d) &= [1, e^{-j2\pi(d_c/\lambda)\cos\theta_d}, \dots, e^{-j2\pi(d_c/\lambda)(M-1)\cos\theta_d}]^T \\ \mathbf{a}(\theta_i) &= [1, e^{-j2\pi(d_c/\lambda)\cos\theta_i}, \dots, e^{-j2\pi(d_c/\lambda)(M-1)\cos\theta_i}]^T, \end{aligned} \quad (2)$$

where θ_d and θ_i denote the direction of the desired signal and the i th interference, $d_c = \lambda/2$ 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

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

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 Fig. 1.

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

$$\text{SINR} = \frac{\sigma_s^2 |\mathbf{w}^H \mathbf{a}(\theta_d)|^2}{\mathbf{w}^H \mathbf{R}_{i+n} \mathbf{w}}, \quad (4)$$

where σ_s^2 is the desired signal power, \mathbf{R}_{i+n} is the interference plus-noise covariance matrix.

The adaptive beamformer studied here is an inherently multi-objective problem, since multiple sets of agents \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. On the other hand, all the optimization methods to the designed ABF aim to find the near-global minimum of a mathematical function fit called fitness function; therefore, the best weight vector is determined according to the fitness value obtained from the object function defined based on SINR. In the following sections, the proposed FBS optimization is applied to adjust the weights, such that the fitness function requirements are achieved for obtaining the optimum SINR.

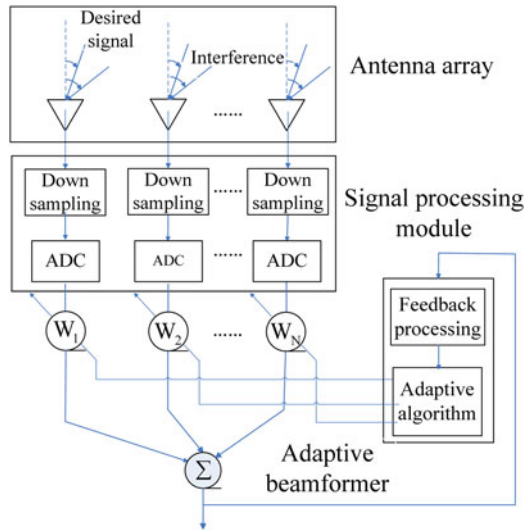


Fig. 1. The schematic of a linear antenna array processor.

Proposed FBS algorithm

The standard principle of Fibonacci sequence method

The famous Fibonacci sequence was proposed by Leonardoda [19], the recurrence formula sequence term is given by [20]

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

where F_j represents the j th general term of the 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, it has been perceived as one of the most effective strategies to solve the one-dimensional unimodal problem [21]. Let us investigate below that how the optimization method using Fibonacci sequence works for a unimodal continuous function in an interval for a minimization problem. Suppose a unimodal $f(x)$ function is defined on the intervals $[A, B]$. Initially, the technique starts a choice of two feasible points x_1 and \tilde{x}_1 , $x_1 < \tilde{x}_1$ 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 in which the minimum lies can be narrowed down provided the function values are known at two different points in the range [22].

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

Step 1 Initialization

Choose $[A_1, B_1] = [A, B]$ and take $p = 1$, while $p < N$ do

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

$$\begin{aligned} x_p &= A_p + \left(1 - \frac{F_p}{F_{p+1}}\right)(B_p - A_p), \\ \tilde{x}_p &= A_p + \frac{F_p}{F_{p+1}}(B_p - A_p) \end{aligned} \quad (6)$$

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

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

Step 5 Set $p = p + 1$ then go to step 1 determine whether to stop the iteration

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

The basic structure of the proposed Fibonacci branch

The standard Fibonacci strategy cannot solve the multiple variables problem efficiently and retain the optimum fitness evaluation of multimodal function reliably. This is in contradiction to the classic heuristic optimization algorithm. The FBS algorithm, proposed in this paper, is used to overcome these defects while avoiding losing optimal search trajectories by using the searching elements with dendritic branch structure and interactive searching optimization rules.

The basic structure of FBS expanded to the multi-dimensional space D can be illustrated where X_A, X_B , and X_C are the vectors in D dimensional Euclidean space. X_A and X_B represent the endpoints of the search element satisfying the optimization rule, 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

$$\frac{\|X_C - X_A\|}{\|X_B - X_A\|} = \frac{\|X_B - X_C\|}{\|X_C - X_A\|} = \frac{F_p}{F_{p+1}}, \quad (7)$$

where F_p is the p th Fibonacci number.

Consider the multimodal function with multiple variables $f(X)$ to be minimized in search space, the fitness function value calculated by the endpoints in the structure should be evaluated as

$$f(X_A) < f(X_B). \quad (8)$$

Then, the coordinate computing formula of segmentation point X_C can be written as

$$X_C = X_A + \frac{F_p}{F_{p+1}}(X_B - X_A). \quad (9)$$

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 the basic structure in Figure 2. 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 to be divided into two stages: local optimization process and global searching process, which are the corresponding two interactive rules. Let G denote the point sets of object function to be searching for in the current processing phase, and set $|G|_{num} = F_p, \quad i = 1, 2, \dots, N, \quad |\cdot|_{num}$ represents the total number of the set, N is the depth of the Fibonacci branch. The fitness values of the endpoint \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 get the results such that the best fitness value accordingly corresponds to the closest optimal solution. To the next optimization phase, the optimal point with 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 order from best to worst. Throughout the operations above, the points set G can be updated in every optimizing phase 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, and defined by

$$\{\mathbf{X}_A\} = G_p = \{\mathbf{X}_q | q = [1, F_p]\} \tag{10}$$

$$\{\mathbf{X}_B\} = \left\{ \mathbf{X} | \mathbf{X} \in \prod_{f=1}^D [X_{lb}^f, X_{ub}^f] \right\}, \tag{11}$$

where G_p is the search points space set in the p th iteration, \mathbf{X}_q are the points in the set G_p , q is the sequence number that lies in the interval between 1 and the p th Fibonacci number. \mathbf{X}_A take all the points from G_p of the p th 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, X_{ub}^f are the upper bound and lower bound 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 \mathbf{X}_{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

$$\mathbf{X}_{best} = \text{BEST}(G_p), \tag{13}$$

where $\text{BEST}(\cdot)$ means the best solution of the set at p th iteration.

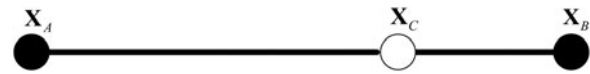


Fig. 2. Basic structure of the proposed FBS.

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

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

$$\mathbf{X}_B = \{\mathbf{X}_q | \mathbf{X}_q \in G_p \wedge \mathbf{X}_q \neq \mathbf{X}_A\}. \tag{15}$$

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 segmentation points.

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 the points is $3F_p$. Evaluating the cost functions in new points determines their fitness, all these points are sorted from the best to the worst based on their fitness value. The population size of the search points is chosen as the Fibonacci serials; thus, the top best F_{p+1} sets of these points are to be saved and the rest corresponding $3F_p - F_{p+1}$ points need to be dropped. After this procedure, the sets of the search space in the current p th iteration are renewed from the saved points, e.g. the saved points form a new set G_{p+1} in search space for the next iteration.

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

As can be seen from Fig. 3, the depth of the Fibonacci branch layer illustrated in the figure is initialized as expected, and the number of the points in every branch layer is kept to the sequence of Fibonacci number. The white dashed circle in the figure represents the search points 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 represent in the gray solid circle. Figure 3(a) shows the first global searching stage of the global optimization process, the segmentation points \mathbf{X}_{S1} which are represented by the solid white line circle are constructed on the basis of global random points and \mathbf{X}_A . The second local optimization stage is illustrated in Fig. 3(b), with the best fitness points \mathbf{X}_A in the search space of the current iteration and the rest endpoints of \mathbf{X}_B , the new segmentation points \mathbf{X}_{S2} can be obtained by iterative 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 found F_{p+1} solutions with optimum object function evaluations need to be saved.

Figure 4 exposes the flowchart of the general procedures for the specific implementation of FBS, and the Pseudo-code of the FBS is shown in Algorithm 1.

Algorithm 1 Pseudo-code of the FBS 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 achieved
3. **While** the termination criterion $|G| = F_R$ not satisfied **do** searching and optimization phase

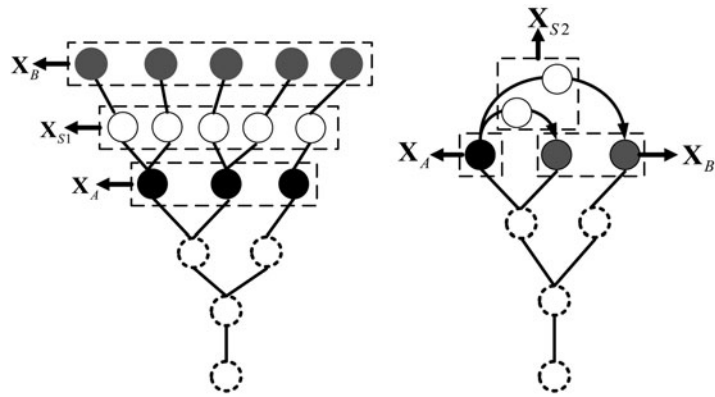


Fig. 3. The process of building the Fibonacci branch for global optimization. (a) The first global searching stage. (b) The second local optimization stage.

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

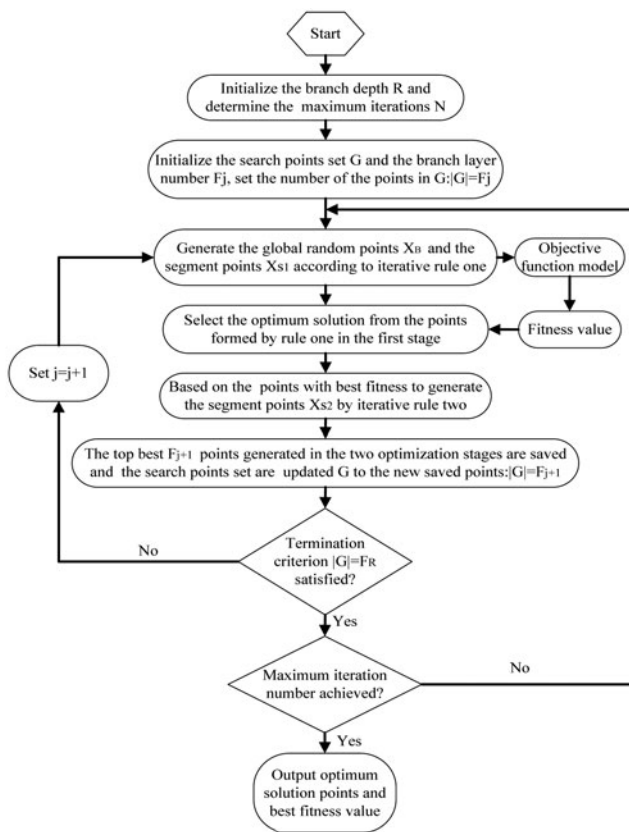


Fig. 4. Flowchart of Fibonacci branch search optimization method.

4. Choose the search points set G in the defined space and set the initial number of branch layer F_j
5. Calculate the shortening fraction F_j/F_{j+1} and generate X_B, 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 X_B in current points set to generate 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 the line 3

10. End while
11. Save the optimum solution points of the current iteration in search set space and go to line 2.
12. End iteration and output the optimal points

The proposed FBS algorithm mentioned above utilizes the interactive global searching and local optimization to obtain the global optimal solution so that it can take advantage of the global randomness and local convergence to get rid of the local minima. The generated points in each layer of the whole branch depend on the optimum elements with the best fitness value in the previous branch layer which can assure the global optimum solution in search space.

FBS-based beamformer design and implementation of ABF with the proposed FBS

As explicitly described previously, ABF is an effective technique used to mitigate interference and improve 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 [23]. Therefore, the proposed FBS algorithm is applied to the ABF with ULA in this section for achieving the improved performance of this technique.

ABF model integrated with FBS

The description of the beamforming model has already been given in section “ABF system model of ULA and problem statement”, 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 through maximizing the objective function based on SINR.

Equation (16) indicates how the ABF design problem is setup for the FBS-based optimization.

$$X_{mF_i} = w_m, \quad m = 1, 2, \dots, M, F_i \leq F_R, \quad (16)$$

where w_m is the m th excitation weight of the array element. X_{mF_i} is the F_i th search point at the m th dimension of the i th branch layer, M is the total number of ULA, 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 entire population generated as the search points at the first layer can be written in the format as below:

$$\mathbf{W}_{MF_I} = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1F_I} \\ w_{21} & w_{22} & \dots & w_{2F_I} \\ \vdots & \vdots & & \vdots \\ w_{M1} & w_{M2} & \dots & w_{MF_I} \end{bmatrix}, \quad (17)$$

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

$$\text{Fitness_Function}(\mathbf{w}) = \frac{P_d}{\sum_{i=1}^Q P_i + P_N}, \quad (18)$$

where

$$P_d = \frac{1}{2} E[|\mathbf{w}^T \mathbf{X}_d|^2] \quad (19)$$

$$P_i = \frac{1}{2} E[|\mathbf{w}^T \mathbf{X}_i|^2] \quad (20)$$

are, respectively, the power of the desired signal and the power corresponding to the i th interference, 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)

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

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

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

$$\sigma_{noise}^2 = 10^{-SNR/10}. \quad (22)$$

In addition, an important feature of the beamformer is that it can form the desired radiation beams, such as the formation of the low sidelobe beams and the flat-top beams. Thus, the next

step in the design process is to formulate the other objective function that is to be minimized for the sidelobe reduction problem. An array consisting of identical and identically oriented elements has a far-field radiation pattern that can be expressed as the product of the element pattern and a factor that is widely referred to as the array factor. The objective function we defined used the array factor in such a way that the objective of the optimization is satisfied. The fitness function to be minimized with the optimization algorithms to introduce the deeper null and reduce the relative sidelobe level (SLL) is given in

$$\text{Fitness}_{SLL} = \sum_k |AF(\theta_k)| + 10 \log_{10} \left(\frac{AF(\theta_{MSL})}{AF(\theta_{ML})} \right), \quad (23)$$

where θ_k is the direction of the interference, θ_{MSL} is the angle of the maximum SLL, θ_{ML} is the angle of mainlobe. $AF(\theta)$ is the array factor of the uniform linear array, and can be written as [23]

$$AF(\theta) = \frac{1}{Q} \sum_{q=1}^Q w_q e^{-j \frac{2\pi}{\lambda} (q-1) d \cos(\theta - \varphi_q)}, \quad (24)$$

where Q is the number of array elements, θ is the angular region of interests. w_q is the weight at the element q . φ_q determines the excitation phase of the q th element.

Consequently, for array design with suppressed SLL and deep null control simultaneously, both terms of equation (23) are used in the fitness evaluation. By optimizing the fitness function defined in this section, the optimal excitation weight corresponding to the minimum level of the interference sources and lowest SLL but with the desired user gain of the beamformer can be achieved by the proposed FBS optimization algorithm. Next section of the paper provides a brief description of the implementation steps for extracting the weight using FBS method.

Implementation flowchart of Fibonacci branch search to ABF

In this subsection, in light of the results described in detail previously, an optimization scheme of ABF problem that combines with the novel FBS is presented to enhance the maximum power for target signal and generate deep nulls for interferences with the lowest sidelobe. 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 and satisfy the requirements to the created multi-objective optimization problem from (21) and (23) for getting the maximum SINR and the minimum SLL. The general procedures for the implementation of the FBS method with application to ABF are presented in Fig. 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 acted 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

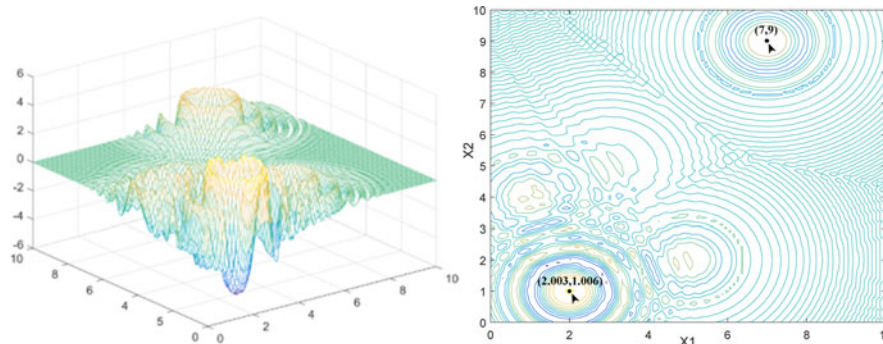


Fig. 5. Three-dimensional and contour plots for the Langermann function. (a) Three-dimensional plots of Langermann function. (b) Contour plots of Langermann function.

(a) Three dimensional of Langermann function

(b) Contour plots of Langermann function

population of the weight vector set \mathbf{G}_w , the weight element in the vector sets \mathbf{G}_w constructed by the amplitude and phase can be expressed as follows:

$$w_{jd} = r \text{ and } [0, 1] \cdot e^{j \cdot \text{rand}[-\pi, \pi]}, \quad (25)$$

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

- (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 is acted 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 in the object function of (21) and (23) using \mathbf{G}_w , \mathbf{w}_B , and \mathbf{w}_{S1} , then give the evaluation to the values to find the best weight vector \mathbf{w}_{best} with the 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}_{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 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}_w .
- (i) Check the termination criteria $\mathbf{G}_w = F_R$. If the termination criteria 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.

Simulation results

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 toward the optimal point.

- (2) The convergence of the FBS is proved and discussed by the presented gradient of the iteration curves which manifested 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 the typical heuristic algorithms.

The details of the parameter settings for every heuristic algorithm used in the experiments are given in Table 1.

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

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 [24, 25] and is summarized in Table 2. As can be found in Table 2, two typical extreme points existed 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 Fig. 5.

The performance of the proposed FBS in terms of the movement trajectory of the search points scattering around the best solutions and converge toward the optimal point in the search space for Langermann is illustrated in Fig. 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 with these same works and provide the results in Fig. 6(a). The initial position of the search points in both FBS and PSO is set at the extreme point 2.

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	$N_p = 20$	Population size	$N_p = 20$	Learning probability	0.05–0.5
Cognitive ratio	$c_1 = 2$	Mutation probability	$P_m = 0.05$	Population size	$N_c = 20$
Social coefficient	$C_2 = 2$	Cross probability	$P_c = 0.7$	Cognitive ratio	$c_1 = 2$
Inertia weight	0.4–0.9	Rate of chromosome elite	$P_e = 0.2$	Social coefficient	$C_2 = 2$
				Inertia weight	0.4–0.9
Differential Evolution (DE)		Artificial Bee Colony (ABC)		Fibonacci Branch search (FBS)	
Population size	$N_p = 20$	Colony size	$C_s = 20$	Nested branch depth	2
Scaling factor	$F = 0.6$	Onlooker bees percentage	$L_p = 50\%$	Total branch depth	6
Crossover rate	$CR = 0.8$	Scout bees	$S_b = 1$	Search space	[Min, Max]

Table 2. Langermann benchmark function

Extreme point	Extreme point 1	Extreme point 1
Evaluation in [29]	Global optimum solution	Local suboptimal solution
Langermann function	$f_L(2.003, 1.006) = -5.1612$	$f_L(7, 9) = -3$

As the results exhibited in Fig. 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 Fig. 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, reflecting the further evidence that PSO inherently suffer 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 Fig. 6(b), although the Langermann function is non-symmetry 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 to be extricated from initial local optimum at extreme point 2 and jump 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 toward the optimal point from the area of initialization, tending to scatter around extreme points gradually and moving toward the best solutions in the search space in both 2D and 3D spaces over the course of iteration. More than half of the agents had already approached the global optimum valley after the first 50 iterations and begun converging on the optimum. As iteration increases, there are more agents that aggregate at the extreme points and scatter around the extreme point, especially attracted intensively at the global optimum target region. Eventually, the search points found the global optimum and converged toward the global optima. This can be discussed and reasoned according to the global randomness concepts introduced by the endpoints 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 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 toward 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; the convergence performance and the rate of obtaining the global optima of the proposed algorithm by employing a set of mathematical functions are to be investigated in the next sections.

Convergence performance of the multimodal function

To confirm the convergence behavior of the proposed algorithm, in this subsection, we provide the convergence curve 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 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 exploration. The illustrated results are compared against that of PSO, GA, CLPSO, DE, and ABC metaheuristic algorithms on the same set of multi-dimensional numerical benchmark functions. The properties and the formulas of these functions are presented below.

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 good global search ability and converges fast. FBS achieved better results on all multimodal groups than the comparison algorithms, 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 exhibited the superiority regarding the convergence speed over the other five

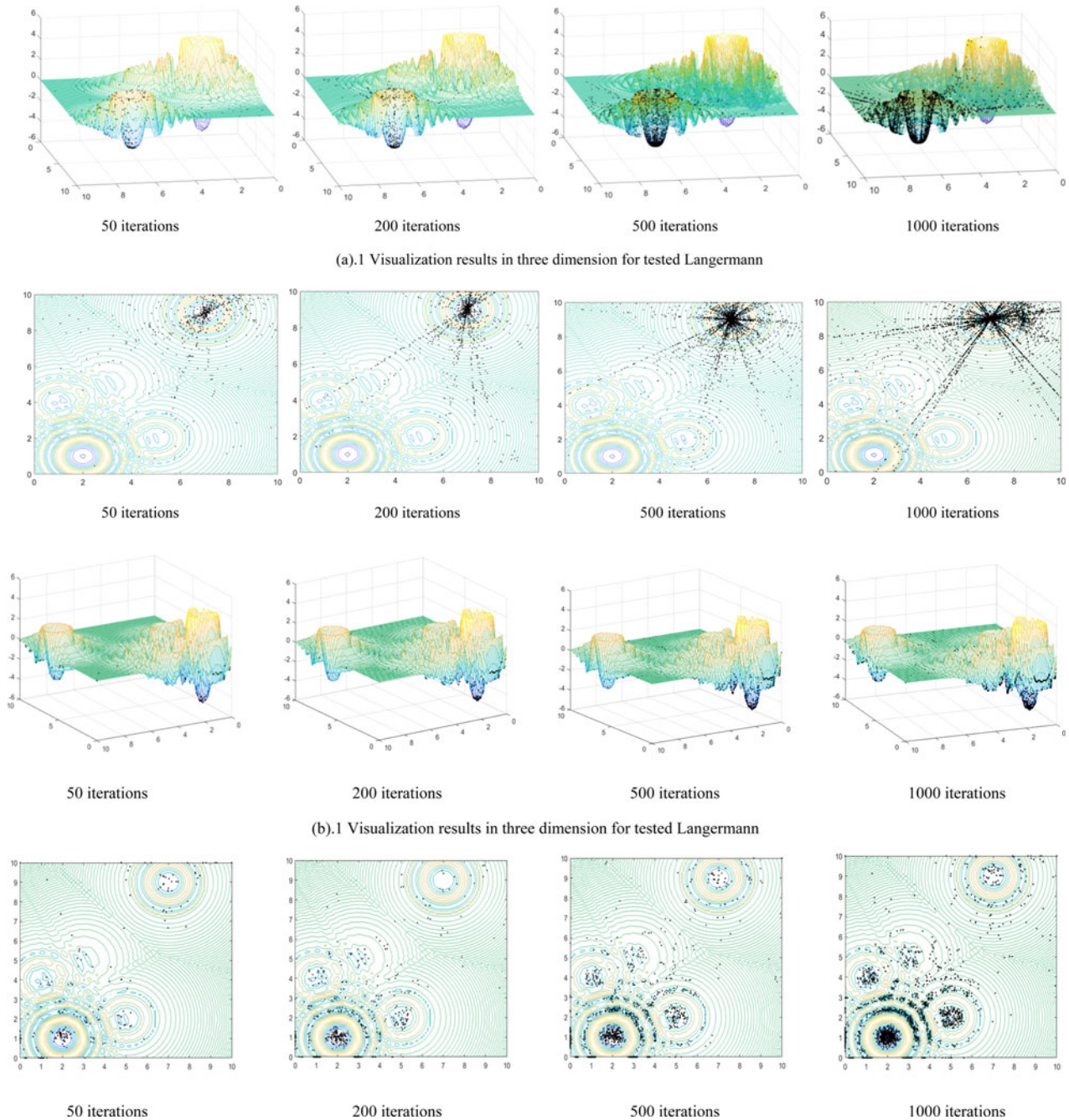


Fig. 6. Location history of the search points in three-dimensional and contour plots for the Langermann function over the course of different iterations. (a) Three-dimensional plots of Langermann function. (b) Contour plots of Langermann function. (a).1 Visualization results in three dimension for tested Langermann. (a).2 Visualization results in contour plots for tested Langermann. (a) Behavior results of the points location history in PSO. (b).1 Visualization results in three dimension for tested Langermann. (b).2 Visualization results in contour plots for tested Langermann. (b) Behavior results of the points location history in FBS.

algorithms, it converges 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 functions.

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 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 ABF system; thus, the FBS is highly suitable and affordable for ABF.

Minimization result of the tested benchmark functions

In this subsection, experiments were conducted on a suite of multimodal functions illustrated in Table 3 to evaluate six optimization algorithms including the proposed FBS. All the test

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	$\sum_{i=1}^D (x_i^2 - 10\cos(2\pi x_i) + 10)$	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	$\sum_{i=1}^D 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2$	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	$418.9829 \times D - \sum_{i=1}^D x_i \sin(x_i ^{1.2})$	10	$[-500, 500]$	0
F7	Weierstrass	$\sum_{i=1}^D \left(\sum_{k=0}^{kmax} [a^k \cos(2\pi b^k(x_i + 0.5))] \right) - D \sum_{k=0}^{kmax} [a^k \cos(2\pi b^k(x_i + 0.5))]$ $a = 0.5, \quad b = 3, \quad kmax = 20$	10	$[-0.5, 0.5]$	0
F8	Salomon	$-\cos\left(2\pi\sqrt{\frac{D}{i}}x_i^2\right) + 0.1\sqrt{\sum_{i=1}^D x_i^2} + 1$	10	$[-100, 100]$	0

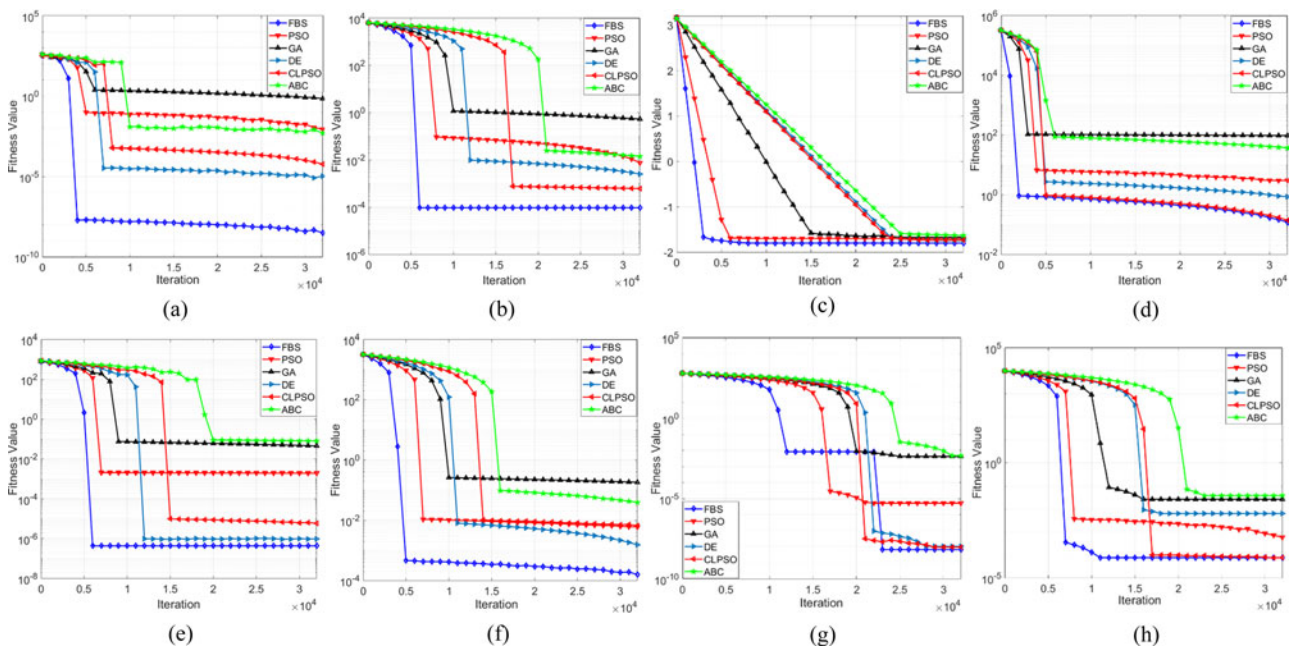


Fig. 7. Convergence behavior of the FBS and other optimization algorithms on 10-D benchmarking functions F1–F8. (a) F1: Griewank; (b) F2: Rastrigin; (c) F3: Michalewicz2; (d) F4: Rosenbrock; (e) F5: Ackley; (f) F6: Schwefel; (g) F7: Weierstrass; (h) F8: Salomon.

functions are to be minimized and the relevant information can be found in [26, 27] for the standard benchmark functions, respectively. 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.

The statistical results considering the average value and the standard deviation function fitness value as well as the success rate needed to reach the acceptable solution are summarized in

Table 4. For the results shown in Table 4, the mean value is smaller, the performance of the algorithm is better. The standard deviation value is lower, the stability of the algorithm is stronger. As seen, for most benchmark data sets, the average value and the standard deviation calculated by the FBS are both smaller than those for 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

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 (%)
ABC	4.73×10^{-3}	6.51×10^{-3}	52	1.43×10^{-2}	4.59×10^{-3}	30	-1.6325×10^0	1.56×10^{-3}	32	3.67×10^1	1.90×10^0	0
DE	7.38×10^{-6}	6.51×10^{-7}	100	2.55×10^{-3}	7.18×10^{-4}	55	-1.7428×10^0	8.36×10^{-7}	90	8.37×10^1	3.27×10^{-1}	8
GA	6.81×10^{-1}	7.49×10^{-2}	20	5.29×10^{-1}	1.21×10^{-2}	11	-1.6715×10^0	4.83×10^{-4}	44	9.42×10^1	2.15×10^{-2}	0
PSO	5.27×10^{-3}	3.45×10^{-3}	49	7.61×10^{-3}	8.29×10^{-2}	38	-1.7002×10^0	3.95×10^{-3}	84	2.69×10^0	1.90×10^0	6
FBS	2.54×10^{-9}	1.47×10^{-8}	100	9.53×10^{-5}	2.17×10^{-6}	86	-1.8013×10^0	7.48×10^{-6}	100	2.98×10^{-2}	5.61×10^{-4}	29
CLPSO	6.01×10^{-5}	4.32×10^{-6}	100	6.18×10^{-4}	7.18×10^{-3}	74	-1.7332×10^0	6.28×10^{-5}	90	4.39×10^{-2}	4.03×10^{-1}	12
Method	F5.Ackley			F6.Schwefel			F7.Weierstrass			F8.Salomon		
	Mean	Stev	SR (%)	Mean	Stev	SR (%)	Mean	Stev	SR (%)	Mean	Stev	SR (%)
ABC	8.87×10^{-3}	2.98×10^{-2}	47	3.85×10^{-2}	3.53×10^{-3}	18	6.13×10^{-5}	9.09×10^{-6}	100	3.85×10^{-2}	1.12×10^{-3}	19
DE	9.27×10^{-7}	9.69×10^{-7}	100	1.53×10^{-3}	6.25×10^{-6}	68	1.08×10^{-8}	4.17×10^{-9}	100	6.43×10^{-3}	6.82×10^{-4}	47
GA	1.46×10^{-2}	5.32×10^{-3}	29	1.85×10^{-1}	3.74×10^{-3}	11	4.28×10^{-3}	9.64×10^{-6}	82	2.67×10^{-2}	6.63×10^{-2}	29
PSO	6.09×10^{-5}	5.49×10^{-4}	92	5.87×10^{-3}	3.68×10^{-4}	66	5.27×10^{-6}	3.64×10^{-6}	100	5.74×10^{-4}	2.63×10^{-3}	68
FBS	4.34×10^{-7}	5.74×10^{-8}	100	2.53×10^{-4}	6.25×10^{-6}	74	6.77×10^{-9}	7.54×10^{-11}	100	7.74×10^{-5}	9.48×10^{-5}	85
CLPSO	5.74×10^{-6}	5.84×10^{-8}	100	6.53×10^{-3}	7.43×10^{-3}	59	9.65×10^{-9}	8.53×10^{-10}	100	6.24×10^{-5}	9.48×10^{-4}	92

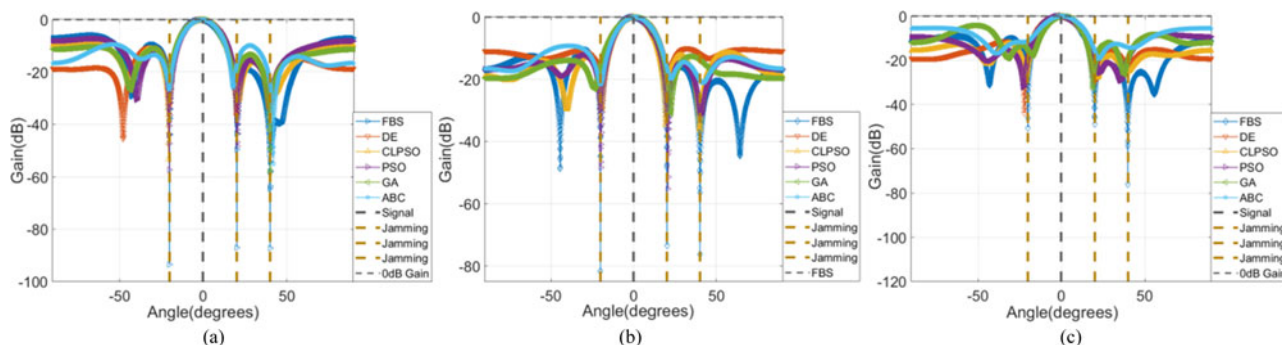


Fig. 8. Comparison of the beam pattern performance synthesized by the heuristic algorithms with different SNR. (a) SNR = 10 dB; (b) SNR = -10 dB; (c) SNR = -30 dB.

proposed FBS could still search the better fitness closest to the optimal value. The CLPSO achieved the similar results as 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 performed similarly with 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 outperformed all other heuristics algorithms on the tested functions. Also, this table illustrates that FBS in comparison with others displays the highest percentage and accuracy of reaching to the acceptable solutions on these test functions. For the mean reliability of test functions F1, F3, F5, F7, FBS exhibits the highest reliability of 100% success rate and smallest average errors. This performance superiority property 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 than the other comparisons. Because of this, the FBS performs comparably to or better than many MH algorithms on most of the multimodal problems experimented in this section.

The performance of the FBS in ABF model application

To demonstrate the benefits of the FBS optimization with application to ABF, in this part of the section, several groups of simulation experiments are conducted using Matlab R2017b. The performance of the FBS-based ABF was evaluated from the following two simulation metric aspects: the beam pattern performance of the adaptive antenna array and the steady state output SINR of the beamformer system. A uniform linear array with the inter-element spacing of half wavelength is considered in the simulation. The desired signal is in the form of QPSK modulation mode with 0° for incident azimuth angle and the three single frequency interfering sources with 10 interference-to-noise (INR) ratio are assumed to impinge on the antenna array from the azimuth directions 20, 40°, and -20°, respectively. The desired signal and jammers are all arranged to be coming from zeniths 45°. Simulation environment is the additive white Gaussian noise channel.

The proposed FBS-based beamforming method is compared with the following four conventional heuristics-based beamforming algorithms: (1) the differential evolution (DE)-based

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

Method	Scenario (dB)		
	SNR = 10 dB	SNR = -10 dB	SNR = -30 dB
FBS	-89.07	-79.29	-61.62
DE	-74.28	-62.33	-55.67
CLPSO	-78.67	-72.67	-58.20
PSO	-70.29	-60.98	-49.25
GA	-54.87	-42.72	-31.82
ABC	-51.26	-37.66	-28.32

beamforming method of [27], (2) the particle swarm optimization (PSO)-based beamforming method of [28], (3) the comprehensive learning particle swarm optimization (CLPSO)-based beamforming method of [29], (4) the artificial bees colony (ABC)-based beamforming method of [30], and (5) Genetic Algorithm (GA)-based beamforming method of [31]. A total of 300 repetitions are implemented and then averaged to obtain each figure of the results.

Beam pattern performance results

In this subsection, the effectiveness and applicability of the proposed FBS-based beamformer are investigated by the power patterns formed by different types of the metaheuristic-based beamformers. The determined optimized antenna element current amplitudes are applied to minimize the peak SLL and to place nulls in the desired directions based on the MATLAB platform, the optimized results are compared with the above-mentioned algorithms. Four groups of simulation cases conducted in terms of various simulation metrics are considered in this study. The first simulation example is with different number of interference sources and the second case considers the different number of array elements. The performance measurement on the input SNR is studied in the third case. The last case illustrates the sidelobe reduction and deep nulls executed by the optimization algorithms based on fitness function (23).

Simulation case one: input SNR evaluation: We first examine the beam pattern performance synthesized by the FBS and compared to the others given in [28-32] in terms of input SNR in this case. A uniform linear array with six omni-directional antenna elements is considered in the simulation. For investigating the effect of the input SNR with different levels, we consider

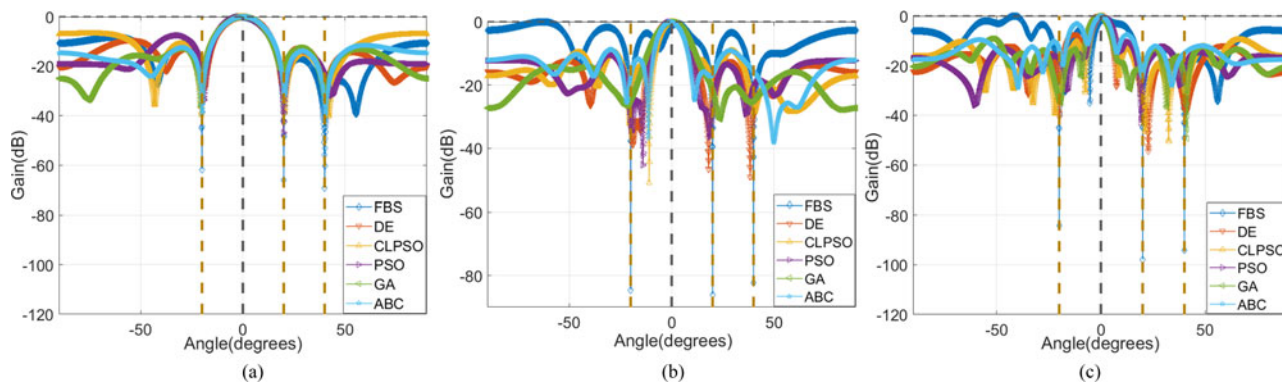


Fig. 9. Comparison of the beam pattern performance synthesized by the heuristic algorithms with a different number of array elements. (a) Six elements; (b) 10 elements; (c) 14 elements.

three sets of input SNR with $SNR = -30$ dB, $SNR = -10$, and $SNR = 10$ to check the validity of our approach. Figure 8 shows the behavior of the beam patterns synthesized by the weight vectors determined by the optimization algorithms under different SNR, and to more clearly illustrate the nulling degrees and nulling level of each method, the nulling results corresponding to Fig. 8 are shown in Table 5. It can be seen from the figures that the weight vectors found by FBS could synthesize the inerratic beam pattern with deeper nulling (with nulling level exceeding -70 dB) toward the interference compared to the other algorithms. The proposed FBS-based adaptive beamformer has suppressed the jammers in all cases while maintaining the beam pattern 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 beam patterns, and the nulls do not align precisely with the interference sources. This indicates that the proposed algorithm is more stable and finds better solutions with greater precision in ABF application.

Simulation case two: array element number investigation: Beam pattern performance regarding the number of array elements has been evaluated in the second case study; adaptive beamformer arrays that consist, respectively, of 6, 10, and 14 were concerned in the simulation experiments. Input SNR was fixed -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 target pattern by considering these three cases with different element numbers. Optimization results of the beam patterns are illustrated in Fig. 9. To avoid showing the confusing results and better distinguish the nulling level, Table 6 illustrates the specific nulling level in different array elements scenarios. The results depicted in Fig. 9 show that the proposed FBS achieves better performance than the other algorithms with the same number of array elements. It means FBS possesses superior performance as compared to DE, CLPSO, GA, PSO, and ABC because it creates deeper null toward the interference direction while maintaining the power at the desired direction. As the number of the elements increases from 6 to 14, the beam pattern formed by the other methods particularly for the ABC and GA are deteriorating and almost fail to work in steady state, that is, the performance disparity between proposed algorithm and the compared algorithms is

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

Method	Scenario		
	Six elements (dB)	Ten elements (dB)	Fourteen elements (dB)
FBS	-69.07	-83.29	-93.62
DE	-64.28	-72.33	-85.67
CLPSO	-61.67	-66.24	-78.20
PSO	-52.29	-60.98	-69.25
GA	-44.87	-52.72	-61.82
ABC	-41.26	-47.66	-58.32

expected to increase further with higher number of array elements, this is due to the increase of the number of elements results in the increase of searching dimension of the solution, which inherently increases the difficulty of the optimization problem, and the global search optimization ability of the FBS algorithm is more suitable for the multi-dimensional solution of the large array elements to improve the beam pattern performance.

Simulation case three: interference sources number assessment: In order to fully verify the beam pattern performance, different types of multiple interference scenarios were simulated to validate the proposed approach for ABF applications in this case. A uniform linear array of 12 omni-directional antenna elements was adopted in this subsection. All scenarios have one desired signal at $\theta_s = 0^\circ$ while the number of interference changes in each one. The first two target interferences source is assumed to be impinging on the array from -20 and 40° , and the four received interferences that impinge at $\theta_i = \{20^\circ, 40^\circ, -20^\circ, 30^\circ\}$ are considered in the second scenario, finally, the third scenario concerns six interferences with incident angle $\theta_i = \{20^\circ, 40^\circ, -20^\circ, 30^\circ, 65^\circ, -70^\circ\}$. In each scenario, the FBS and other optimization algorithms are applied to find the optimal excitation vectors, respectively, that produce a main lobe toward θ_s and nulls toward θ_i . The graphs shown in Fig. 10 represent the optimized beam pattern for all the scenarios studied here. All the radiation pattern results presented in Fig. 10 show that FBS-based beamformer is a robust technique capable of improving the radiation characteristics with better performance

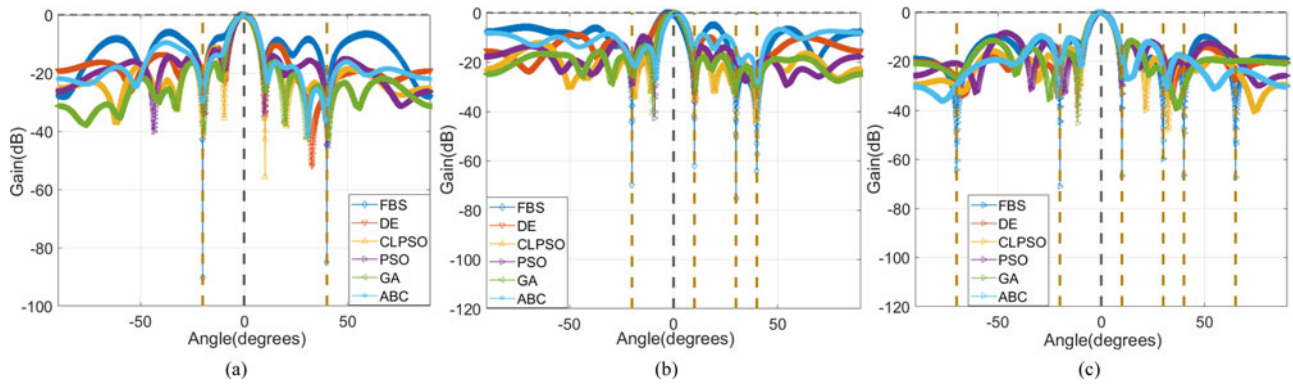


Fig. 10. Comparison of the beam pattern performance synthesized by the heuristic algorithms with a different number of interferences. (a) Two interferences; (b) four interferences; (c) six interferences.

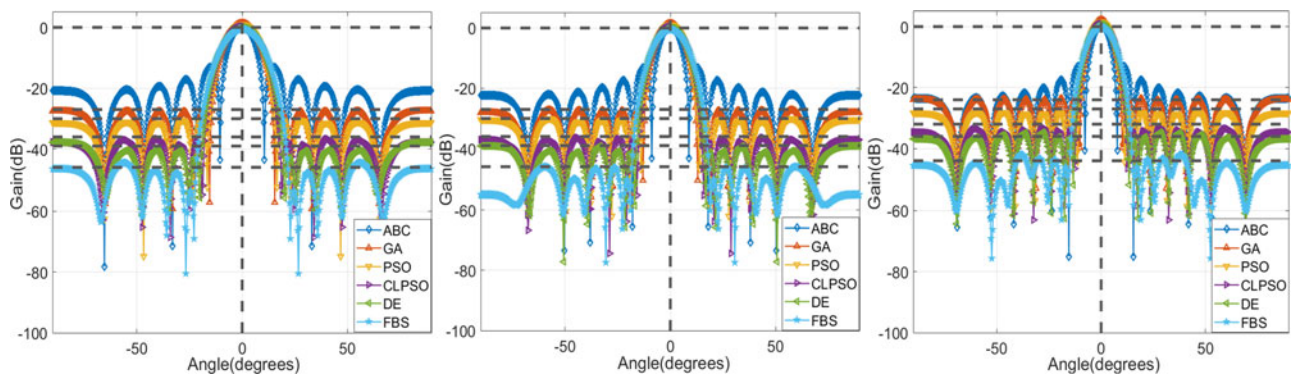


Fig. 11. Beam pattern with reduced sidelobe produced by the proposed FBS and other comparison optimization algorithms for a different number of array elements at diverse interference angles. (a) Twelve elements with prescribed null at 47°, -66°; (b) 14 elements with prescribed null at 50°, -35°; (c) 16 elements with prescribed null at 68°, 56°.

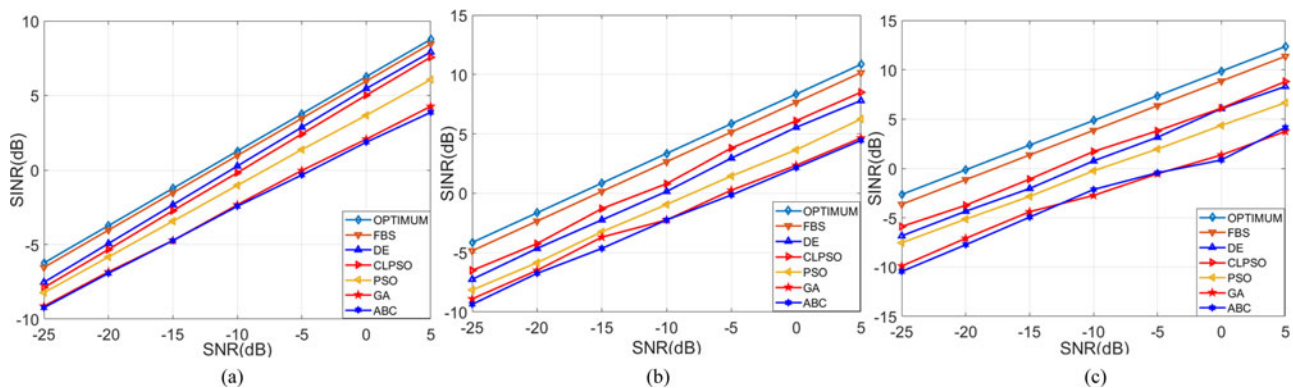


Fig. 12. SINR performance versus SNR of the heuristic algorithms with varied array elements. (a) Six elements; (b) 10 elements; (c) 14 elements.

than the conventional optimized beamformer whatever the number of interference signals is. The FBS-based beamformer exhibits more prominent behavior regarding the steering ability increasing the nulling levels in Fig. 10(c), i.e. the superiority of FBS in scenario 3 is more apparent than 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 beam-steering and interference mitigation performances in all three cases, especially for the multiple interference sources.

Simulation case four: evaluation of the SLL of the beam pattern: The proposed FBS and other selected comparison algorithms in the last case determine the weight of each array element for achieving the lower sidelobe. The receiving linear antenna arrays composed of 12, 14, and 16 monochromatic isotropic radiating elements considered for reference in different simulation

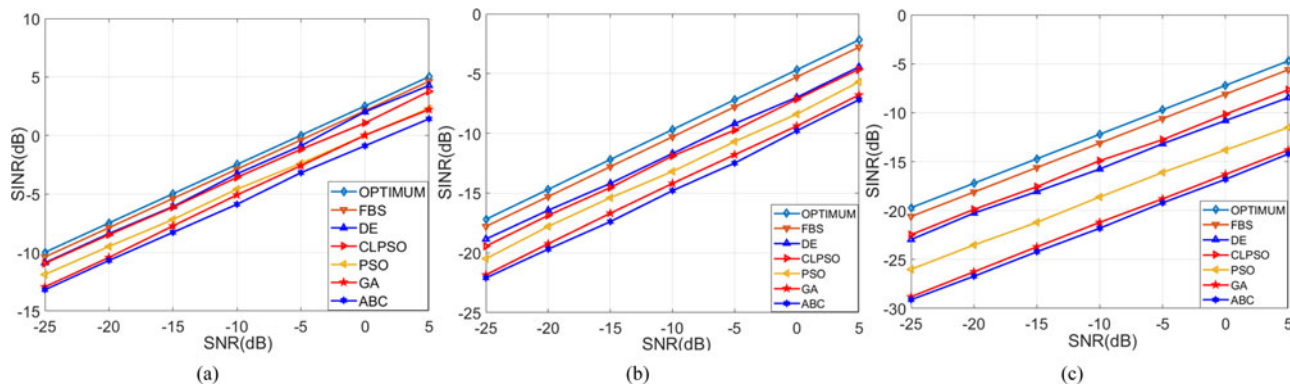


Fig. 13. SINR performance versus SNR of the heuristic algorithms with different INR. (a) INR = 5 dB; (b) INR = 20 dB; (c) INR = 35 dB.

scenarios. Suppose four interferences from respective different angles impinge on the ULA, each scenario has one desired signal at $\theta_s = 0^\circ$. The INR of all the simulation scenarios is fixed at 30 dB. Figure 11 displays the graphical beampattern comparison of the optimized array factor with reduced SLL. It can be seen from Fig. 11 that the symmetry of array factors makes the array radiation pattern more regular than the conventional beamformers, and the proposed FBS provides a better reduction in SLL as compared to arrays optimized using other algorithms; also, it is observed from the figure that null depth in the interference direction synthesized by FBS is superior to the others. The following Table 7 and Table 8 show the specific values of the maximum SLL of each algorithm for the ULA. The simulation results in the table confirm the superiority of the proposed FBS via specific maximum SLL and reduction degrees. The maximum SLL obtained using the FBS with 12, 14, 16 elements is -43.68 , -45.62 , -48.29 dB, respectively, and the improvement of sidelobe reduction level by FBS, compared to ABC which performs worst among comparisons in each case, is 26.96, 26.36, and 25.18 dB. It should be also noted even for relatively good performance of the DE and the CLPSO, the maximum SLL obtained using FBS is better than their results, and the evidence for FBS superiority, which are 12.19, 9.35, and 8.61 dB SLL improvement compared to DE which possesses the suboptimal performance in different cases. Thus, FBS demonstrates a better exploitative ability than other compared algorithms in ABF model.

Output SINR performance results

This section will illustrate different scenarios for system output SINR using FBS and other optimization algorithms for searching for suitable weight vectors feed to fulfill required SINR performance. Numerical simulations to investigate the SINR performance of the proposed algorithms in terms of the useful metric are carried out. The first two cases are studied with a different number of the array elements and interferences. The rest case concerns the different INR 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 be changed from -25 to 5 dB (centered at -10 dB) in 5 dB steps. All cases have one user at 0° , while the number of interferences and elements and input INR changes in each scenario.

Simulation case one: four interference with 20 dB INR considered varied elements: The first case is two interference signals at

Table 7. Maximum sidelobe level achieved by using FBS, DE, CLPSO, PSO, ABC, GA for different number of array elements

Algorithms	Elements number (dB)		
	12	14	16
FBS	-43.68	-45.62	-48.29
DE	-35.02	-36.27	-39.68
CLPSO	-31.49	-32.04	-34.52
PSO	-27.64	-29.04	-30.19
GA	-24.78	-26.81	-28.62
ABC	-16.72	-19.26	-23.11

Table 8. Comparison of average nulling level toward interference sources for different numbers of interference scenarios

Method	Scenario		
	Two interference (dB)	Four interference (dB)	Six interference (dB)
FBS	-86.02	-69.95	-61.37
DE	-68.45	-51.67	-43.29
CLPSO	-63.73	-49.65	-39.04
PSO	-49.91	-32.73	-29.93
GA	-42.46	-29.40	-23.37
ABC	-41.75	-27.81	-24.03

20 and 40° with 5 dB INR. The linear arrays are considered to be composed, respectively, of 6, 10, and 14 elements. The results of the output SINR are illustrated in Fig. 12. From the graphs, it can be noticed that the proposed algorithm outperforms the others for all the array element scenarios and is able to achieve near-optimal 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 increasing for the array element increment in each

Table 9. Different number of interferences 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	3	-45
				4	-30
				5	35
				6	60

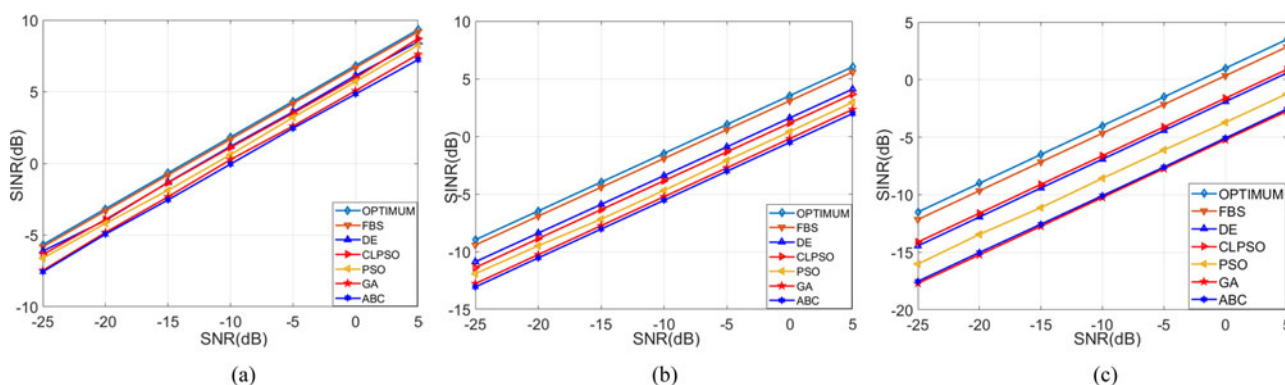


Fig. 14. SINR performance versus SNR of the heuristic algorithms with a different number of interferences. (a) One interference; (b) three interferences; (c) six interferences.

algorithm, this is consistent with previous results in section “Simulation case two: array element number investigation” 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 the adaptive beamformer system.

Simulation case two: four interference with different INR impinge on six-element array: A ULA consisting of six monochromatic isotropic elements receiving four interferences with different INR from respective 20° , 40° , 65° , and -70° is considered in this scenario. Three groups of INRs of the interference set as 5, 20, 35 dB are established in 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 the compared optimization algorithms. From the results depicted in Fig. 13 we can know that, in general, the optimization algorithms are all able to achieve near-satisfactory SINR performance in the situation of INR = 5 dB, which means the lowest power process of the interference signal. The proposed algorithm achieved improved the performance in SINR than the other algorithms in all the simulations even under the most severe interference process when the value of INR is 30 dB. With the increasing of INR for the interference, the SINR performance 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 ABF for high interference power levels.

Simulation case three: different number of interferences with fixed INR received by six-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 quantity to the SINR performance, the beamformer is equipped with six array elements and the INR was fixed to 10 dB for the received interference signals, Table 9 illustrates the different number 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 Fig. 14. As can be seen from Fig. 14, when there is one interference at the receiver, all the optimization algorithms can achieve the close-to-optimal SINR performance by considering the requirement for maximizing SINR, FBS demonstrates the best improvement, followed by DE. ABC shows the lowest value of SINR. As the number of interference increases from one to six, the superior 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.

Conclusions

In this paper, we presented a novel global optimization heuristic algorithm, called FBS, which is based on the use of the Fibonacci number series, to achieve the improved performance of the ABF. The interactive global and local searching rule was

proposed to reduce the probability of falling into the local optima problem, and the global randomness characteristic and space region shortening fraction of this technique guarantee the convergence velocity in the global optimization process. Numerical multimodal benchmarking functions were employed to validate the effectiveness of these global optimization algorithms, and we found that generally FBS achieves a better convergence rate, precision, and stability compared to the others. In addition, we devise the specific implementation architecture based on FBS for adaptive beamformer; the amplitudes and phases of the weight vector acting as the solution were acquired in the search space by the FBS, and the beamforming results synthesized by the vectors were compared with conventional metaheuristic-based beamforming algorithms. The simulation experiments of section “Conclusions” demonstrate that the proposed beamforming method outperformed the compared methods significantly regarding the power beampattern and the output SINR for different scenarios. Consequently, the proposed FBS is a valuable tool for multi-objective optimization and well suited for ABF design problems, and it seems to be a quite promising smart antenna using beamforming technology. In the future work, the FBS will be explored to apply on a more complicated time-varying situation in ABF field.

References

- Komljenovic T, Helkey R, Coldren L and bowers JE (2017). Sparse aperiodic arrays for optical beam forming and LIDAR. *Optics Express* **25**, 2511.
- Madanayake A, Wijenayake C, Dansereau DG, Gunaratne TK, Bruton LT and Williams SB (2013). Multidimensional (MD) circuits and systems for emerging applications including cognitive radio, radio astronomy, robot vision and imaging. *Circuits & Systems Magazine IEEE* **13**, 10–43.
- Bing H and Leger J (2008). Numerical aperture invariant focus shaping using spirally polarized beams. *Optics Communications* **281**, 1924–1928.
- Zaharis ZD, Skeberis C, Xenos TD, Lazaridis PI and Cosmas J (2013). Design of a novel antenna array beamformer using neural networks trained by modified adaptive dispersion invasive weed optimization based data. *IEEE Transactions on Broadcasting* **59**, 455–460.
- El-Keyi A and Champagne B (2009). Collaborative uplink transmit beamforming with robustness against channel estimation errors. *IEEE Transactions on Vehicular Technology* **58**, 126–139.
- Hassani A, Plata-Chaves J, Bertrand A and Moonen M (2017). Multi-task wireless sensor network for joint distributed node-specific signal enhancement, LCMV beamforming and DOA estimation. *IEEE Journal of Selected Topics in Signal Processing* **11**, 518–533.
- Roshanaei M, Lucas C and Mehrabian AR (2009). Adaptive beamforming using a novel numerical optimisation algorithm. *IET Microwaves, Antennas & Propagation* **3**, 765–773.
- Al-Ardi EM, Shubair RM and Al-Mualla ME (2004). Performance evaluation of the LMS adaptive beamforming algorithm used in smart antenna systems. *IEEE Midwest Symposium on Circuits & Systems*.
- Rasmussen TK and Krink T (2003). Improved hidden Markov model training for multiple sequence alignment by a particle swarm optimization-evolutionary algorithm hybrid. *Biosystems* **72**, 5–17.
- Wei C and Lu Y (2015). Adaptive beamforming for arbitrary array by particle swarm optimization. *IEEE International Conference on Computational Electromagnetics*.
- Vitale M, Vesentini G, Ahmad NN and Hanzo L (2002). Genetic algorithm assisted adaptive beamforming. *Proceedings IEEE 56th Vehicular Technology Conference*.
- Maina RM, Langat K and Kihato PK (2013). Comparative analysis between use of particle swarm optimization and simulated annealing algorithms in beam null steering a rectangular array system. *International Journal of Innovative Research & Development* **2**, 7.
- Darzi S, Tiong SK, Islam MT, Ismail M and Kibria S (2015). Optimal null steering of minimum variance distortionless response adaptive beamforming using particle swarm optimization and gravitational search algorithm. *IEEE International Symposium on Telecommunication Technologies*.
- Mahmoud KR, El-Adawy M, Ibrahim SMM, Bansal R, Mahmoud KR and Zainud-Deen SH (2008). Performance of circular Yagi-Uda arrays for beamforming applications using particle swarm optimization algorithm. *Journal of Electromagnetic Waves & Applications* **22**, 353–364.
- He L and Huang S (2017). Modified firefly algorithm based multilevel thresholding for color image segmentation. *Neurocomputing* **240**, 152–174.
- Berryman MJ, Allison A and Abbott D (2004). Optimizing genetic algorithm strategies for evolving networks. *Noise in Communication. International Society for Optics and Photonics* **5473**, 122–131.
- Korošec P, Šilc J and Filipič B (2012). The differential ant-stigmergy algorithm. *Information Sciences* **192**, 82–97.
- Liao B and Chan SC (2012). Adaptive beamforming for uniform linear arrays with unknown mutual coupling. *IEEE Antennas and Wireless Propagation Letters* **11**, 464–467.
- Etminaniesfahani A, Ghanbarzadeh A and Marashi Z (2018). Fibonacci indicator algorithm: a novel tool for complex optimization problems. *Engineering Applications of Artificial Intelligence* **74**, 1–9.
- Subasi M, Yildirim N and Yildiz B (2004). An improvement on Fibonacci search method in optimization theory. *Applied Mathematics and Computation* **147**, 893–901.
- Yildiz B and Karaduman E (2003). On Fibonacci search method with k-Lucas numbers. *Applied Mathematics and Computation* **143**, 523–531.
- Omolehin JO, Ibiejugba MA, Onachi AE and Evans DJ (2005). A Fibonacci search technique for a class of multivariable functions and ODEs. *International Journal of Computer Mathematics* **82**, 1505–1524.
- Jayaprakasam S, Abdul Rahim SK, Leow CY, Ting TO (2017). Sidelobe reduction and capacity improvement of open-loop collaborative beamforming in wireless sensor networks. *PLoS ONE* **12**, e0175510.
- Adorio EP and Diliman U (2005). Mvf-multivariate test functions library in c for unconstrained global optimization. Quezon City, Metro Manila, Philippines, pp. 100–104.
- Molga M and Smutnicki C (2005). Test functions for optimization needs. *Test functions for optimization needs*, 101.
- Cui L, Li G, Wang X-Z, Lin Q, Chen J, Lu N, Lu J (2017). A ranking-based adaptive artificial bee colony algorithm for global numerical optimization. *Information Sciences* **417**, 169–185.
- Zhang C, Ning J, Lu S, Ouyang D and Ding T (2009). A novel hybrid differential evolution and particle swarm optimization algorithm for unconstrained optimization. *Operations Research Letters* **37**, 117–122.
- Mallipeddi R, Lie JP, Suganthan PN, Razul SG and See CMS (2011). A differential evolution approach for robust adaptive beamforming based on joint estimation of look direction and array geometry. *Progress in Electromagnetics Research* **2011**, 381–394.
- Banerjee S and Dwivedi VV (2016). Performance analysis of adaptive beamforming using particle swarm optimization. *11th International Conference on Industrial and Information Systems (ICIIS)*. IEEE, pp. 242–246.
- Ismaiel AM, Elsaydy EI, Albagory Y, Atallah HA, Abdel-Rahman AB and Sallam T (2018). Performance improvement of high altitude platform using concentric circular antenna array based on particle swarm optimization. *AEU-International Journal of Electronics and Communications* **91**, 85–90.
- Ruchi R, Nandi A and Basu B (2015). 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* **28**, 508–521.
- Lu Y, Yeo B-K (2000). Adaptive wide null steering for digital beamforming array with the complex coded genetic algorithm[C]// Phased Array Systems and Technology, 2000. Proceedings. 2000 IEEE International Conference on. IEEE.



Haichuan Zhang was born in China in 1991. He received the Master's degree in School of Electronic Countermeasures, National University of Defense Technology (NUDT), Hefei, China, in 2016. Since 2016, he proceeded with a doctoral research in the National University of Defense Technology. His research interests are in the areas of adaptive beamforming and its applications, array signal processing, and digital communication system.



Fangling Zeng is currently a professor at the National University of Defense Technology (NUDT). She obtained her BSc, Master's, and PhD degrees from the University of Science Technology (USTC) of China in 1996, 1999, and 2002, respectively. Her main research interests include precise point positioning phased array signal processing and GNSS/INS.