Hindawi
Wireless Communications and Mobile Computing
Volume 2019, Article ID 9046409, 17 pages
https://doi.org/10.1155/2019/9046409

Research Article

Backtracking Search Optimization Paradigm for Pattern Correction of Faulty Antenna Array in Wireless Mobile Communications

Fawad Zaman,[1] Hammad ul Hassan,[1] Shafqat Ullah Khan,[2] Ata ur Rehman,[1] Muhammad Asif Zahoor Raja,[1] and Shahab Ahmad Niazi[3]

[1]*Department of Electrical and Computer Engineering, COMSATS University Islamabad, Attock Campus, Attock, Pakistan*
[2]*Department of Electronics, The University of Buner, Buner, Khyber Pakhtunkhwa, Pakistan*
[3]*Department of Electronic Engineering, UCE&T, The Islamia University of Bahawalpur, Pakistan*

Correspondence should be addressed to Fawad Zaman; fawad@ciit-attock.edu.pk

Received 19 September 2018; Revised 7 February 2019; Accepted 17 February 2019; Published 11 July 2019

Academic Editor: Zhou Su

The demand for wireless mobile communication is growing exponentially with expectations that in the near future mobile device or user will be available in every corner of the globe. Alternatively, it increases the importance of antenna arrays which are responsible for transmission and reception of information. Every antenna array is projected to generate a desired pattern and, hence, failure of any antenna causes misrepresentation of the overall pattern in terms of increased side lobe levels and displacement of nulls from their original position. The aim of the study is to present viable, simple, and accurate stochastic solver based on backtracking search optimization algorithm (BSA) for the pattern correction of faulty antenna array in mobile communication systems. A fitness function is developed to optimize the weights of the remaining healthy antenna elements in the array. The fitness function consists of two parts: the first part is based on mean square error approach for the reduction of sidelobes level, while, in the second part, steering vectors are used for the repositioning of nulls. Simulation results establish the validity of the BSA from its counterparts based on genetic algorithm and its memetic combination with pattern search technique.

1. Introduction

The interest of researchers in wireless mobile communication systems is growing day by day due to several reasons. First the price for production of antenna arrays is getting cheaper and secondly, their size is getting smaller that ultimately decreases the power consumption due to which battery capacity is increasing. In order to improve the directivity (gain), usually an array is used that consists of two or more sensors or antennas. Antenna array is one of the main components that improve the performance and spectral efficiency of mobile communication systems [1–3]. Every antenna has its own weight that can be configured to get desired excitation level of that antenna. These weights have two parameters, namely, phase and amplitude, that can be adjusted individually. The combined effects of all weights generate the desired pattern having suitable side lobe level, nulls position, and null depths [1]. This process is known as array synthesis which is used for controlling amplitudes, positions, and phases to achieve the desired radiation patterns and eliminate noise and unwanted signals. Nulls can be used to achieve this goal by moving them in the direction of unwanted signals but this is not an easy task as direction of unwanted signal is not known or the signal's direction is changing on regular basis. This is known as nulling operation that can be achieved by adjusting amplitude-only [4], position-only [5], and phase-only [6] of the individual antenna to generate nulls in the direction of unwanted signals.

The antennas used in the array of mobile communication systems are expected to work for longer periods of time. Due to this longer operation of times, wear and tear or failure probability is also increased. For this reason, the antenna array must be able to evaluate its performance and detect possible breakdowns as the desired pattern is destroyed due

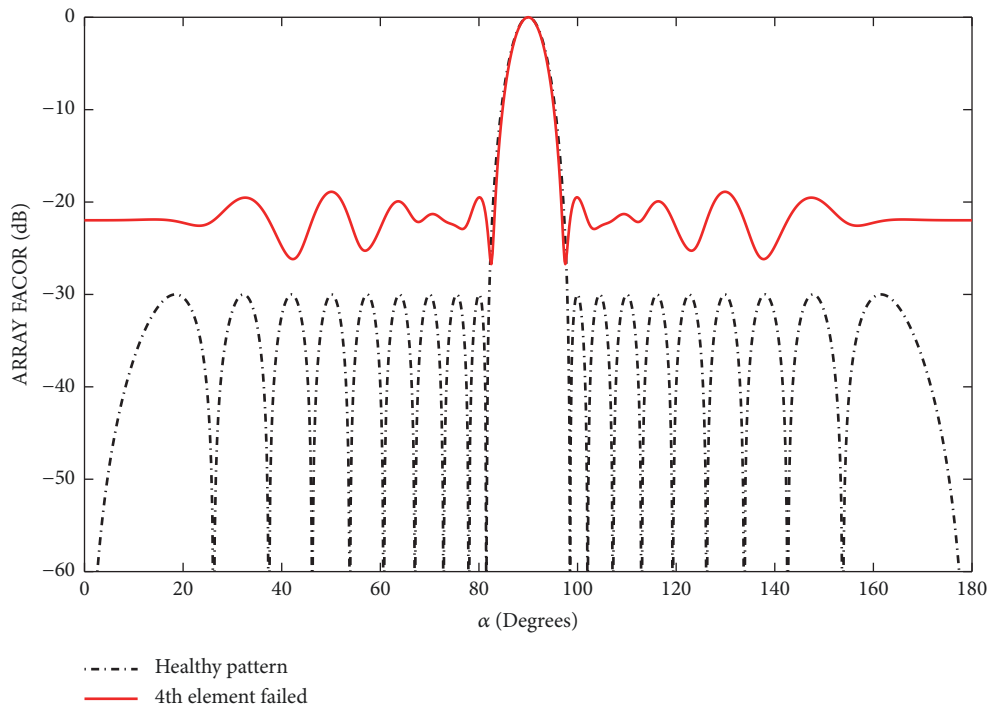

Figure 1: Array factor of 20-element antenna array vs. 4th element failed.

to the failure of single or multiple antennas of an array, which leads to widening the main beam, high side lobe levels, misplacement of nulls from their original position, and diminished null depth. [7, 8]. The effects of antenna failure on the array pattern are shown in Figure 1. An array must be able to compensate for the failure and maintain an error-free operation, as repairing a failed antenna is costly, as well as time consuming.

1.1. Related Work. In this modern and fast growing era of science and technology, meta-heuristic based optimization algorithms have their own importance and effectiveness that got applications in each field of science and engineering. A deterministic model yields a single solution of an experiment that is given some inputs but at the same time meta-heuristic based model gives a distribution of possible outputs with a probability of how likely each is to happen. They are conceptually simple, more flexible, and prominent particularly in the existence of local minima. The various metaheuristic techniques are particle swarm optimization [9], bee's algorithm [10], bacterial foraging algorithm [11], differential evolution [12], genetic algorithm (GA) [13], ant colony optimization [14], backtracking search optimization algorithm (BSA) [15], artificial bee colony algorithm [16], touring ant colony algorithm [17], firefly algorithms [18], etc. These techniques are implemented for many applications such as astrophysics, plasma and atomic physics, nonlinear optics, random matrix theory, electromagnetic, nanotechnology, fluid dynamics, electric machines, bioinformatics, energy, power, signal processing, controls, and communication; see [19–28] and the references cited therein. Among them recently BSA has attracted the attention of researcher due to its simplicity and efficiency. BSA has a single control parameter unlike other metaheuristic algorithms. Moreover, the problem solving capability of BSA is less dependent on the initial value of the parameter. The simple structure of the BSA enables it to solve multimodal functions effectively and quickly. In reported study [15], BSA has been statistically compared with PSO, ABC, self-adaptive differential evolution algorithm (SADE), adaptive differential evolution algorithm (JDE), covariance matrix adaptation evolution strategy (CMAES), and comprehensive learning particle swarm optimizer (CLPSO), and comparison was done using 3 real-world and 75 constrained benchmarks. In general, the simulations and comparisons in [15] show that BSA yielded more success as compared to the other algorithms. These facts motivate authors to explore in BSA for pattern correction of faulty antenna array in wireless mobile communications.

1.2. Innovative Contributions. Failure of any antenna element in the mobile communication systems causes distortion of the overall pattern in terms of increased side lobe levels and displacement of nulls from their original position of the transmitting beam. The exploration and exploitation of metaheuristic is investigated in the present study for pattern correction of faulty antenna array with the following salient features in terms of innovative contributions.

(i) The novel application of evolutionary computational heuristics through backtracking search optimization algorithm is presented for the correction of faulty array for wireless mobile communication systems.

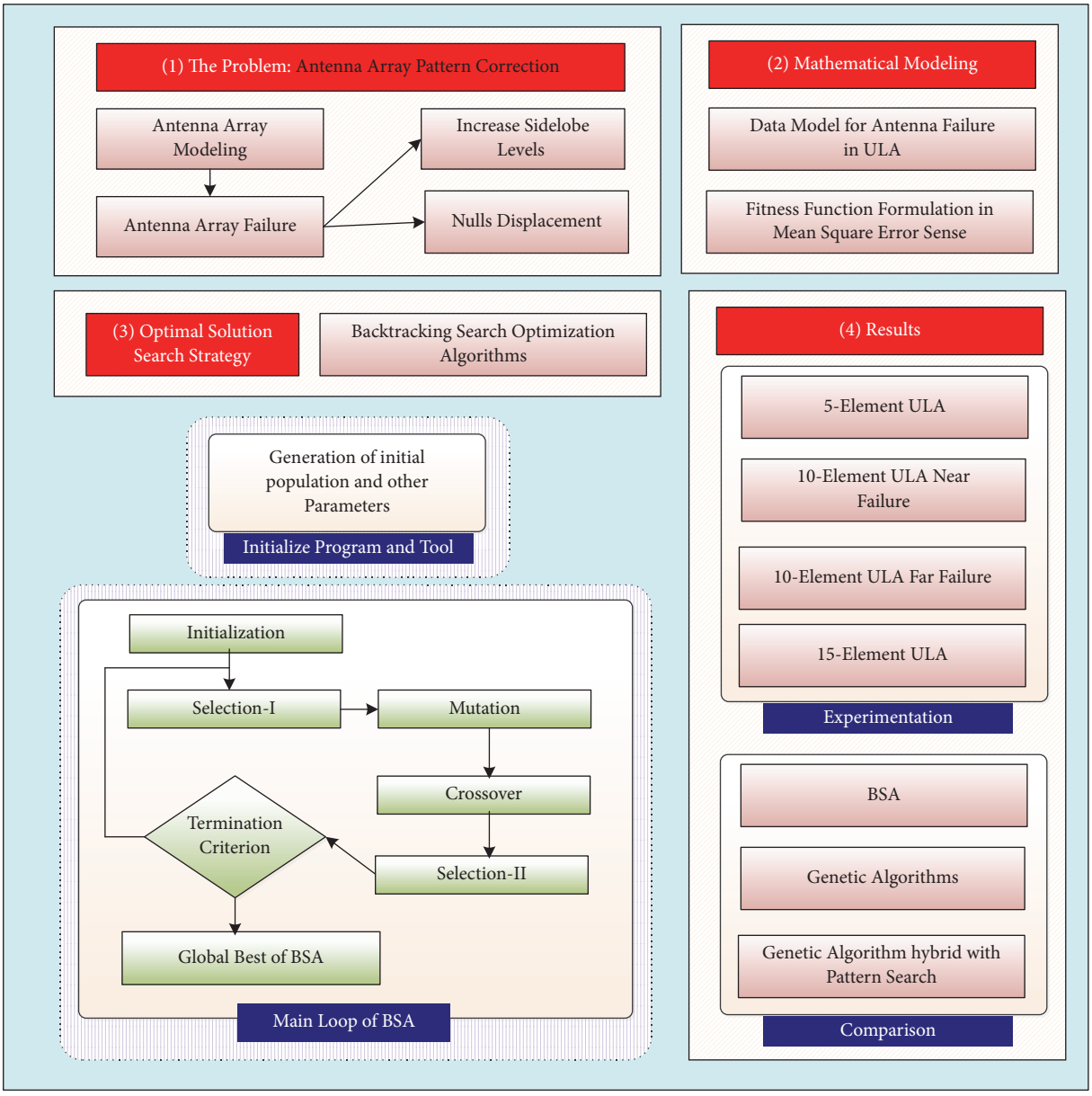

Figure 2: Schematic diagram of the proposed scheme.

(ii) In case of failure of any antenna element, a fitness function is constructed to optimize the weights of the remaining healthy antenna elements in the array. The developed fitness function consists of two parts, i.e., the first part is based on mean square error approach dedicated for the reduction of sidelobe levels while the second part is built on steering vectors used for the repositioning of nulls.

(iii) The simulation results show the validity of the proposed computational paradigm of BSA in terms of decreased sidelobe levels and nulling pattern. The comparison of BSA results from its counterparts based on GA and memetic combination of GA with pattern search (PS) further indorsed its superior performance.

1.3. Organization. The remainder of the paper is divided into four sections. In Section 2.1, the problem formulation is given for the pattern correction of uniform linear array (ULA) used for mobile communication systems. A detailed explanation for BSA is presented in Section 2.2 while Section 3 is dedicated for simulation results. Finally, the conclusions and future work directions are given in Section 4.

2. Design Methodology

The design scheme presented here consists of two parts: in the first part a date model for ULA is provided briefly while in the second part an overview of the optimization procedure based on backtracking search optimization methodology is provided. The schematic of the proposed methodology is shown in Figure 2.

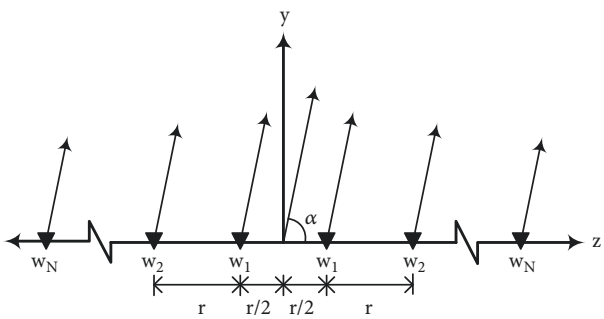

Figure 3: Uniform linear array of even number of sensors.

2.1. Problem Formulation. Let us consider ULA used for mobile communication which is composed of 2N (even) isotropic antenna placed symmetrically along z-axis, as shown in Figure 3. r represents inter-element spacing and N is the number of antennas, α is the elevation angle, while w_i is the complex weight corresponds to i^{th} antenna in the ULA.

The array factor for this ULA can be written as

$$\begin{aligned}(AF)_{2N} = {} & w_N e^{+j((2N-1)/2)kr\cos\alpha} + \cdots + w_2 e^{+j(3/2)kr\cos\alpha} \\ & + w_1 e^{+j(1/2)kr\cos\alpha} + w_1 e^{-j(1/2)kr\cos\alpha} \\ & + w_2 e^{-j(3/2)kr\cos\alpha} + \cdots \\ & + w_N e^{-j((2N-1)/2)kr\cos\alpha}\end{aligned} \tag{1}$$

where k represents the wavenumber of the signal. Simplifying (1) by applying Euler's identity gives

$$(AF)_{2N} = \sum_{i=1}^{N} w_i \cos\left[\frac{(2i-1)}{2} kr\cos\alpha\right] \tag{2}$$

By substituting $\eta = (\pi r\cos\alpha)/\lambda$, we get

$$(AF)_{2N} = \sum_{i=1}^{N} w_i \cos\left[(2i-1)\eta\right] \tag{3}$$

Similarly for odd number of elements, array factor can be calculated as

$$(AF)_{2N+1} = \sum_{i=1}^{N+1} w_i \cos\left[2(i-1)\eta\right] \tag{4}$$

In order to steer nulls and main beam in the desired direction, we need to calculate the weights adaptively.

Weighting Method: research community has proposed different techniques to choose the best suited weight for each antenna element to place the null(s) and/or main beam in a desired direction. For many practical applications the weights of ULA are calculated using Dolph-Tschebyscheff method [25, 26] and so we did in this work. Dolph-Tschebyscheff array was developed by Dolph using Tschebycheff polynomial so it is referred as Dolph-Tschebyscheff array. From (3) and (4), it can be observed that the array factor is the summation of N or 2N cosine terms, largest one being less than total number of antenna elements. These cosine terms can be written as a series of cosine functions as

$$\begin{aligned}
n = 0 &\longrightarrow \cos(n\vartheta) = 1 \\
n = 1 &\longrightarrow \cos(n\vartheta) = \cos\vartheta \\
n = 2 &\longrightarrow \cos(n\vartheta) = \cos(2\vartheta) = 2\cos^2\vartheta - 1 \\
n = 3 &\longrightarrow \cos(n\vartheta) = \cos(3\vartheta) = 4\cos^3\vartheta - 3\cos\vartheta \\
n = 4 &\longrightarrow \cos(n\vartheta) = \cos(4\vartheta) = 8\cos^4\vartheta - 8\cos^2\vartheta + 1 \\
n = 5 &\longrightarrow \cos(n\vartheta) = \cos(5\vartheta) \\
& = 16\cos^5\vartheta - 20\cos^3\vartheta + 5\cos\vartheta
\end{aligned} \tag{5}$$

Using Euler's formula $[e^{j\vartheta}]^n = (\cos\vartheta + j\sin\vartheta)^n = e^{jn\vartheta} = \cos(n\vartheta) + j\sin(n\vartheta)$ and the trigonometric identity $\sin^2\vartheta = 1 - \cos^2\vartheta$ and letting $x = \cos\vartheta$, we get

$$\begin{aligned}
n = 0 &\longrightarrow \cos(n\vartheta) = 1 = T_0(x) \\
n = 1 &\longrightarrow \cos(n\vartheta) = x = T_1(x) \\
n = 2 &\longrightarrow \cos(n\vartheta) = 2x^2 - 1 = T_2(x) \\
n = 3 &\longrightarrow \cos(n\vartheta) = 4x^3 - 3x = T_3(x) \\
n = 4 &\longrightarrow \cos(n\vartheta) = 8x^4 - 8x^2 + 1 = T_4(x) \\
n = 5 &\longrightarrow \cos(n\vartheta) = 16x^5 - 20x^3 + 5x = T_5(x)
\end{aligned} \tag{6}$$

Equation (6) shows that each cosine term is related to a different polynomial $T_q(x)$; it should be noted that these relationships are only valid in the range $-1 \le x \le +1$ because $|\cos(n\vartheta)| \le 1$. The values for the Tschebyscheff polynomial can be found by the recursion formula:

$$T_q(\mathrm{x}) = 2\mathrm{x}T_{q-1}(\mathrm{x}) - T_{q-2}(\mathrm{x}) \tag{7}$$

Equation (7) shows that the next order polynomial can be calculated if the previous two polynomials are known. Numerous research articles have been written for effective calculation of Dolph-Tschebyscheff weights [29, 30].

The critical parameters for a ULA radiation pattern include null positioning, null depth, main beam amplitude, beam width, and side lobe level. In case, one or more elements get faulty due to any reason, like thunder storm and aging, the pattern of ULA is largely affected. If repair is not immediately available there is another possibility to reassign weights to the remaining healthy antennas. In order to recover the side lobe level and null positioning, a metaheuristic algorithm known as BSA is proposed in this work. The formulation of cost function is explained in the subsequent section.

Cost Function: in order to achieve pattern correction, a cost function is formulated. The aim is to obtain an array pattern, from a failed array that is close to that of healthy array by reconfiguring the weights of the remaining healthy elements. The cost function is given by

$$c = c_1 + c_2 \tag{8}$$

```
(1)   Initialization
R epeat
     (2) Selection – I
      Generation of trial population
           (3) Mutation
           (4) Crossover
      end
      (5) Selection – II
Until stopping criteria are met
```

Algorithm 1: Algorithm of BSA.

Table 1: BSA parameter settings.

Parameter	Definition	Value
Popsize	Population size	80
Dim	Dimension	9
DIM_RATE	Dimension rate	1
Epoch	Maximum iteration number	1000

Here c_1 denotes the function which is used to minimize side lobe levels (SLL). It is given by

$$c_1 = \sum \left(|SLL|_d\right)^2 = \sum_{i=1}^{p} \left|AF_H\left(\alpha_i\right) - AF_d\left(\alpha_i\right)\right|^2. \quad (9)$$

Equation (9) represents the mean square error (MSE) in order to minimize the SLL. The reduction is achieved by using MSE between $AF_H(\alpha_i)$, which is healthy level and $AF_d(\alpha_i)$ which is faulty level. Here, p is the limit up to which the side lobe level is to be minimized.

In (8) c_2 denotes the function which is used to generate nulls in the desired locations. Initial theory about null placement was presented by Steyskal et al. [31] in which a steering vector based null positioning is proposed. Using this idea we need the following equation for null generation in the required direction:

$$AF\left(\alpha_i\right) = \mathbf{w}^H \mathbf{v}\left(\alpha_i\right) = 0 \quad (10)$$

where $\mathbf{w}$ is a $N \times 1$ vector, defined as

$$\mathbf{w} = \left[w_{-M}, \cdots, w_{-1}, w_0, w_1, \cdots, w_M\right]^T \quad (11)$$

and $\boldsymbol{v}(\alpha_i)$ is $N \times 1$ vector, used as steering vector

$$\mathbf{v}\left(\alpha_i\right) = \left[e^{-j((N-1)/2)kr\cos\alpha_i}, e^{-j((N-3)/2)kr\cos\alpha_i} \cdots e^{j((N-1)/2)kr\cos\alpha_i}\right]^T_{N\times 1} \quad (12)$$

N is the number of elements. The constraint matrix $\mathbf{A}$ is defined by following equation:

$$\mathbf{A} = \left[\mathbf{v}\left(\alpha_1\right), \mathbf{v}\left(\alpha_2\right), \ldots, \mathbf{v}\left(\alpha_{M0}\right)\right] \quad (13)$$

Here α_i for $i=1,2,\ldots M_o$ represents the direction of the nulls. To get the nulls in a desired direction the below-mentioned squared weighed function should be minimized:

$$c_2 = \left\|\mathbf{w}^H \mathbf{A}\right\|^2 \quad (14)$$

Backtracking search optimization (BSA) is used to minimize c as given in (8), which is the overall cost function.

2.2. Backtracking Search Optimization Algorithm. BSA is a newly developed evolutionary algorithm (EA) presented in 2013 by Pinar Civicioglu [15]. The BSA has an ability to efficiently handle various types of optimization problems. It uses the information from the previously generated population to generate a new population having superior fitness values. BSA is divided into five main steps:

(1) Initialization
(2) Selection-I
(3) Mutation
(4) Crossover
(5) Selection-II

The basic or general working of BSA is shown in Algorithm 1, while the details of the fundamental steps in term of pseudocode are given in Pseudocode 1.

3. Simulation Results

In this section, performance of BSA is validated and several simulations are performed for a single antenna failure in 5, 10, and 15 elements ULA, respectively. The effects of antenna failure near and far from the center are also considered. Throughout the simulations, the inter-element spacing 'r' is taken to be 0.5λ. The performance of BSA is compared to already available techniques for array pattern correction using GA and GA-PS as reported in [1]. The parameter setting for BSA is given in Table 1.

One can observe that failure of element near the center (4th element) causes more damage to the pattern in terms of high side lobe level and displacement of nulls from their original position as compared to failure of element far from center (9th). These effects are shown in Figures 4 and 5 which

Step 1: Initialization:

An initial population P is initialized as

$$\mathbf{P}_{kl} \sim V(lower_l, upper_l) \tag{15}$$

Where $k = 1, 2, \cdots, N$, and $l = 1, 2, \cdots, D$, V is the uniform distribution function, N is population size, and D is problem dimension. P_k represents k^{th} chromosome in the population P. $lower_l$ and $upper_l$ represents the lower limit upper limit of the solution respectively.

Step 2: Selection-I:

In Selection-I, historical population P^{his} is determined, which is used to calculate the direction of search for optimum solution. Initial P^{his} is determined by:

$$\mathbf{P}^{his}{}_{kl} \sim \mathrm{V}(lower_l, upper_l)\ . \tag{16}$$

This process is done through the "if-then" rule as:

$$if\ c < d\ then\ \mathbf{P}^{his}{}_{kl} := \mathbf{P} \mid c, d \sim \mathrm{V}(0, 1) \tag{17}$$

Here $:=$ is defined as update operation, c and d are random numbers in the range [0,1] which help decide if the Phis is selected from the former generation. After this step the order of the chromosomes is shuffled by random shuffling function given by:

$$\mathbf{P}^{his} := \mathrm{permuting}(\mathbf{P}^{his}) \tag{18}$$

Step 3: Mutation:

A mutation is done in $\mathbf{P}^{his}$ to generate a trail population $\mathbf{P}^{mutant}$ using:

$$\mathbf{P}^{mutant} = \mathbf{P} + G(\mathbf{P}^{his} - \mathbf{P}) \tag{19}$$

So, $\mathbf{P}^{mutant}$ is due to the propagation of chromosomes of $\mathbf{P}$ in the direction set by $(\mathbf{P}^{his} - \mathbf{P})$ and G controls amplitude of $(\mathbf{P}^{his} - \mathbf{P})$. Use of $\mathbf{P}^{his}$ gives a partial advantage to BSA of the experiences of previous population in creating an intelligent trial population $\mathbf{P}^{mutant}$.

Step 4: Crossover:

It creates the final trial population $\mathbf{P}^T$, the initial step towards $\mathbf{P}^T$ was $\mathbf{P}^{mutant}$ set by (19). Target population is improved by chromosomes of $\mathbf{P}^T$ having improved fitness. For this crossover uses two procedures. First determines a binary matrix map $\mathbf{M}$ of $N \times D$ order. Its function is to indicate particles of $\mathbf{P}^T$ that are to be changed by using relevant particles of $\mathbf{P}$. The initial value of $\mathbf{M}_{m,n}$ is set as 1, and trial population $\mathbf{P}^T$ is updated as:

$$\mathbf{P}^T{}_{n,m} := \mathbf{P}_{n,m} \tag{20}$$

Here $n \in \{1, 2, \cdots, N\}$ & $m \in \{1, 2, \cdots, D\}$. Crossover strategy of BSA is presented in Algorithm 2. BSA uses a unique crossover strategy as compared to other EAs. The mix-rate parameter controls the number of particles that are mutated in a trial by using $[mixrate \text{ x } rnd \text{ x } D]$, is mentioned in line (3) of Algorithm 2. This function is the main reason that BSA crossover is unique to other EAs

BSA's $\mathbf{M}$ is defined by two predefined strategies being used randomly. The first strategy implements mix-rate (Algorithm 2 line (2) - (4)), and other allows one chromosome to mutate, chosen at random in each trial (Algorithm 2 line (6) - (8)).

The mutation strategy results in overflow of some particles of the PT during crossover procedure. For this, a boundary control process is defined that keeps the individual inside the bounds. Its algorithm is given in Algorithm 3.

Step 5: Selection-II:

In this process the particles of $\mathbf{P}^T$ having improved fitness are updated by their corresponding better particles of $\mathbf{P}$. The global minimum and global minimizer are also updated based on the best fitness so far. The particle having best fitness is called global minimizer denoted by Pbest and its fitness is called global minimum.

Step 6: Termination

Finally, optimization process of the BSA is stopped if one the following criterion is met:

(i) Max number of iteration is reached, or

(ii) The fitness value is below a certain threshold

Pseudocode 1: Pseudocode of BSA with procedural details.

```
Input: P^mutant , mix-rate, N and D
Output: P^T : trial population
M_(1:N,1:D) = 1
If c < d | c, d ~ V(0,1) then
        for k from 1 to N do
                M_k,v(1:[mixrate × rnd × D]) = 0 | v = permuting(<1,2,...,D>)
        end
else
        for k from 1 to N do
                M_k,randi(D) = 0
        end
end
P^T := P^mutant
for k from 1 to N do
        for l from 1 to D do
                if M_k,l = 1 then
                        P^T_k,l := P_k,l
        end
end
```

Algorithm 2: Crossover strategy of BSA.

```
Input : P^T, Search space limits (i.e., lower_l , upper_l )
Output : P^T
for k from 1 to N do
    for l from 1 to D do
        If ( P^T_k,l < lower_l ) or ( P^T_k,l > upper_l ) then
            P^T_k,l = rnd × ( upper_l−lower_l ) + lower_l
        End
    End
End
```

Algorithm 3: Boundary mechanism of BSA.

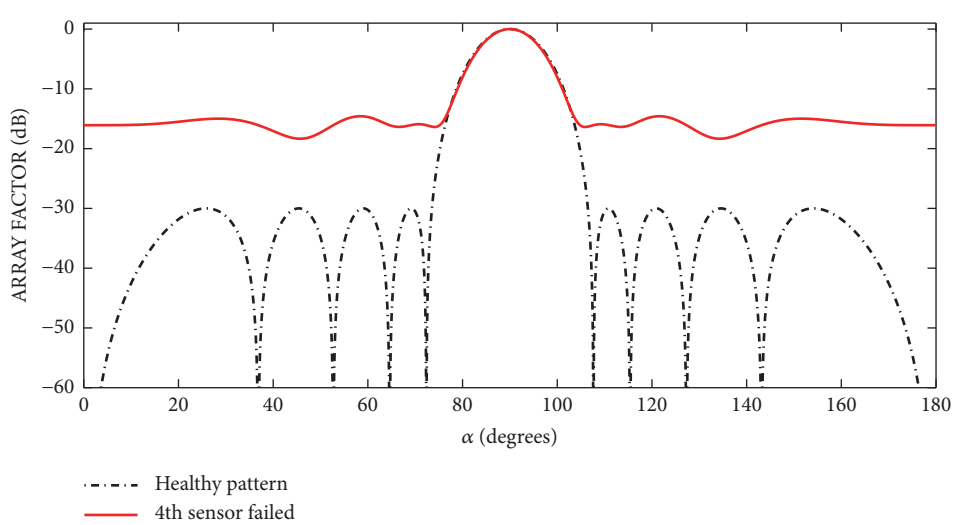

Figure 4: Pattern of 10-element array with 4th element failure.

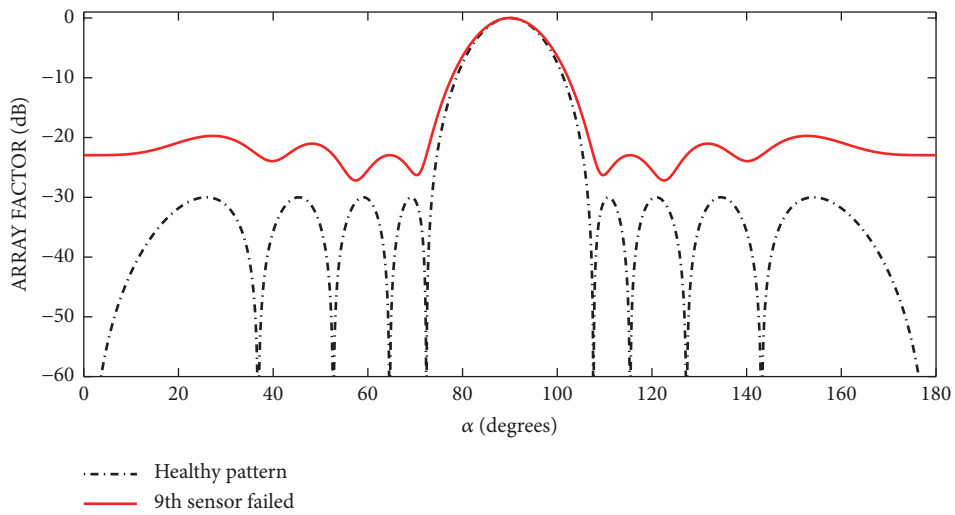

Figure 5: Pattern of 10-element array with 9th element failure.

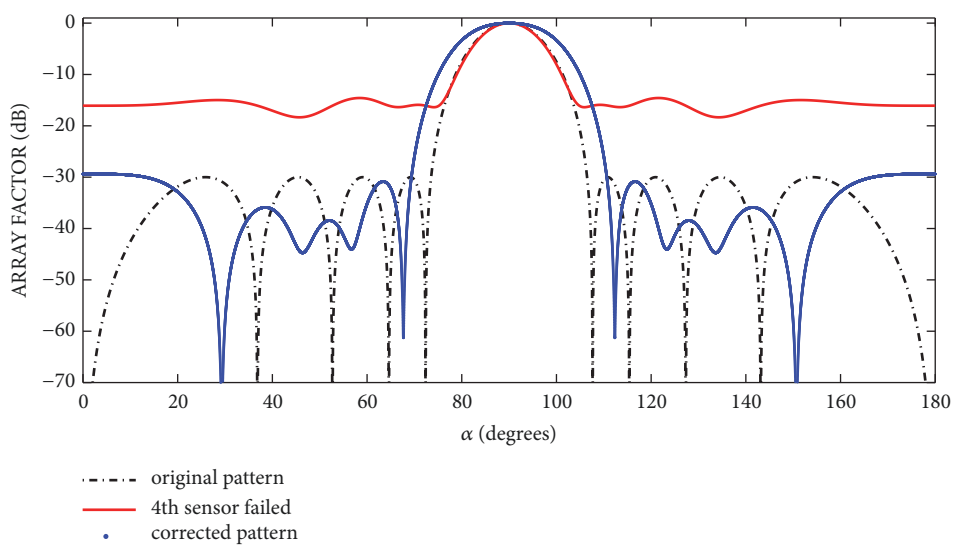

Figure 6: SLL recovered for 10-element 4th sensor failed array.

also verify that BSA is an efficient scheme for solving both cases.

The examples of first group are simulated using a 10-element array with element failure near center. The simulations show the recovery of SLL in Figure 6, recovery of single null in Figures 7 and 8, and recovery of multiple nulls in Figures 9–11.

The second group of examples is simulated using a 10-element array with element failure far from center. The simulations show the recovery of SLL in Figure 12, recovery of single null in Figures 13 and 14, and recovery of multiple nulls in Figures 15 and 16.

The third group of examples is simulated using a 5-element ULA. The simulations show the recovery of SLL in Figure 17 and recovery of single nulls in Figure 18. The fourth group of examples is simulated using a 15-element array with element failure far from center. The simulations show the recovery of SLL in Figure 19 and recovery of multiple nulls in Figures 20–22. Tables 2, 3, 4, and 5 show the comparison between faulty pattern and recovered pattern for each case, respectively.

Performance of BSA is compared in terms of efficiency and accuracy, with an already proven technique in [1]. The results presented in Table 6 show the comparison results. The table shows that BSA is much quicker than both GA and GA-PS due to its structural simplicity, BSA is also more accurate. The simulations for GA and GA-PS were also done on the same computer.

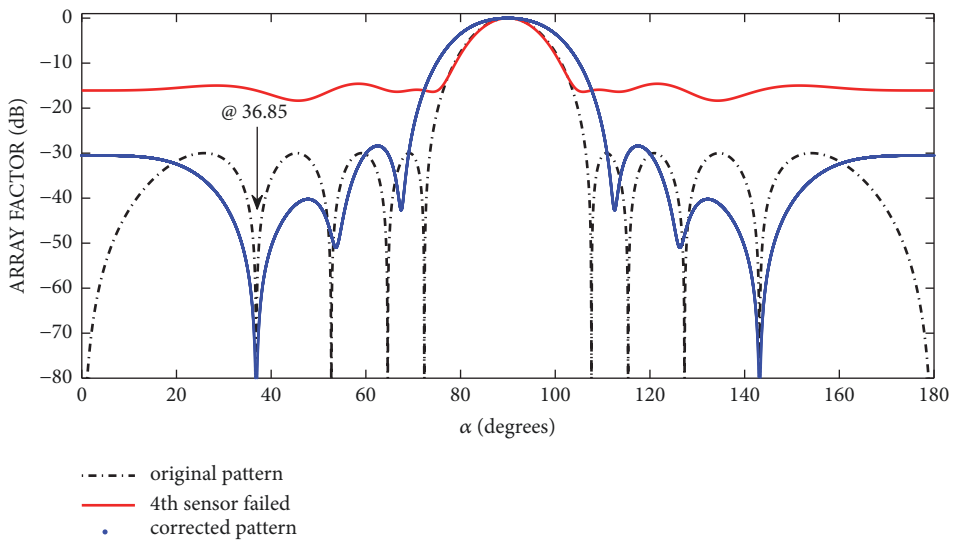

Figure 7: Fourth null recovered for 10-element 4th sensor failed array.

Table 2: Comparison of faulty and corrected pattern for 10-element array with failure near center.

Faulty pattern		Corrected pattern		Recovery of nulls
SLL(dB)	*NDL (dB)*	*SLL(dB)*	*NDL (dB)*	
-16	*-18*	-30	-85	1 null
-16	*-18*	-29	-105	2 nulls
-16	*-16*	-29	-98	
-16	*-18*	-28	-111	3 nulls
-16	*-17*	-28	-108	
-16	*-16*	-28	-122	
-16	*-18*	-30	-68	4 nulls
-16	*-17*	-30	-103	
-16	*-16*	-30	-72	
-16	*-16*	-30	-105	

Table 3: Comparison of faulty and corrected pattern for 10-element ULA with failure far from center.

Faulty pattern		Corrected pattern		Recovery of nulls
SLL(dB)	*NDL (dB)*	*SLL(dB)*	*NDL (dB)*	
-16	*-18*	-31	-108	1 null
-16	*-18*	-28	-100	3 nulls
-16	*-17*	-28	-105	
-16	*-16*	-28	-117	
-16	*-18*	-30	-80	4 nulls
-16	*-17*	-30	-93	
-16	*-16*	-30	-95	
-16	*-16*	-30	-105	

Table 4: : Comparison of faulty and corrected pattern for 5-element array.

Faulty pattern		Corrected pattern		Recovery of nulls
SLL(dB)	*NDL (dB)*	*SLL(dB)*	*NDL (dB)*	
-19	*-18*	-28	-78	2 nulls
-19	*-16*	-30	-81	

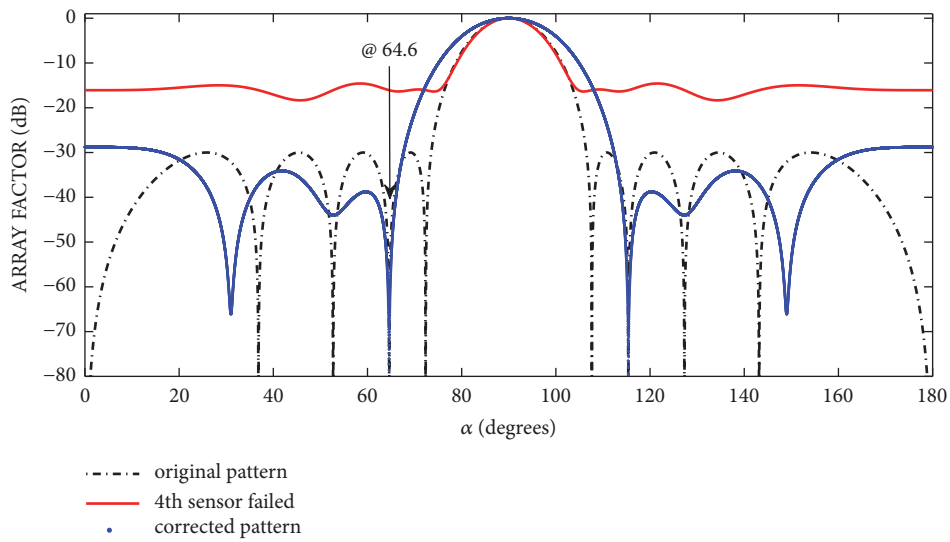

Figure 8: Second null recovered for 10-element 4th sensor failed array.

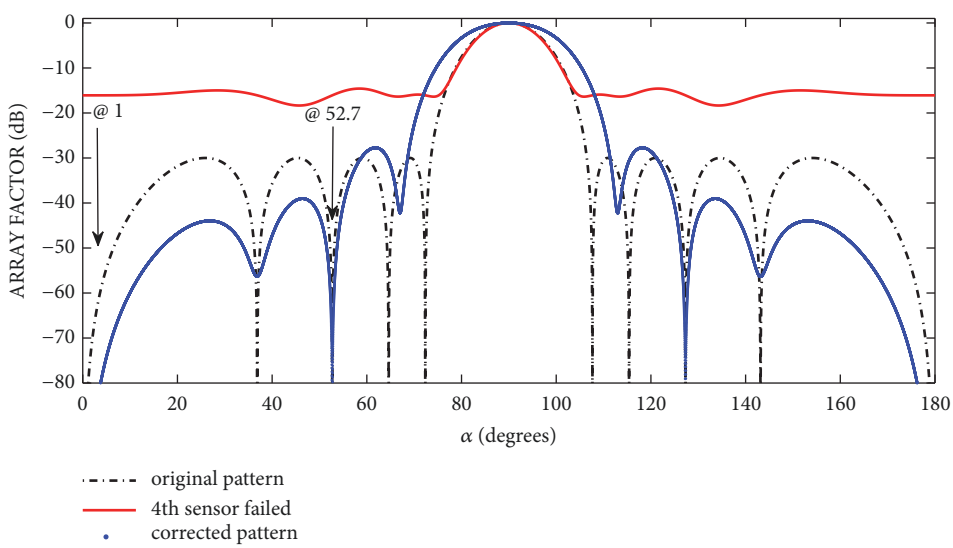

Figure 9: 3rd and 5th null recovered for 10-element ULA 4th sensor failed.

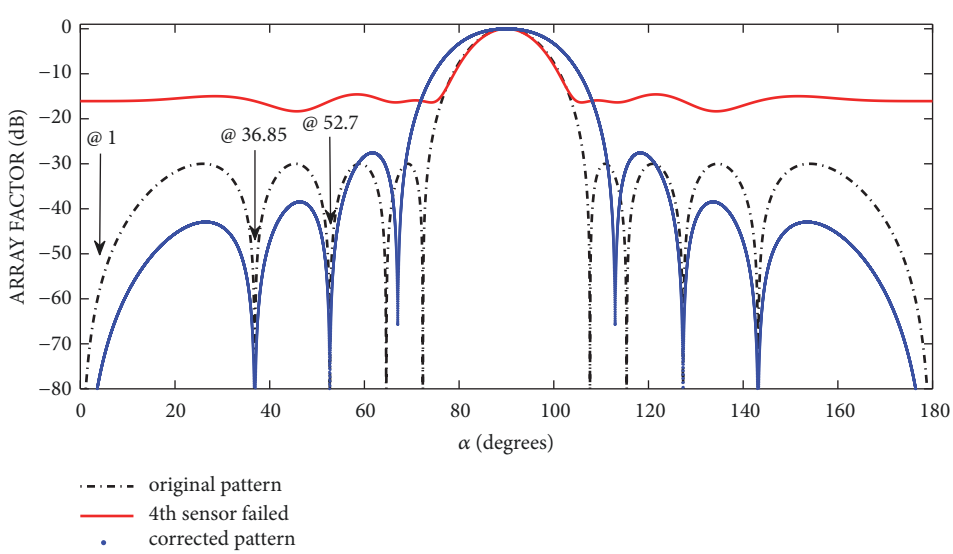

Figure 10: 3rd, 4th, and 5th null recovered for 10-element ULA 4th sensor failed.

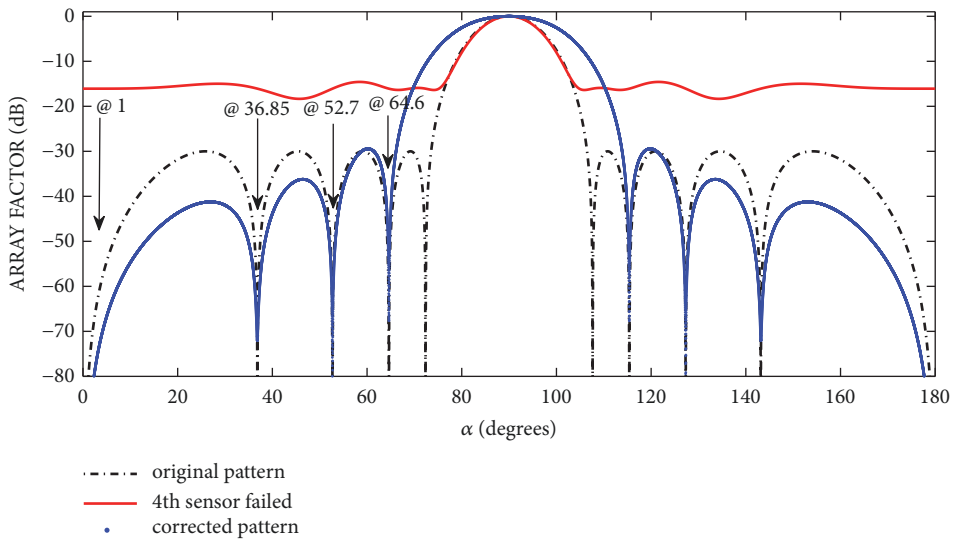

FIGURE 11: 2nd, 3rd, 4th, and 5th null recovered for 10-element ULA 4th sensor failed.

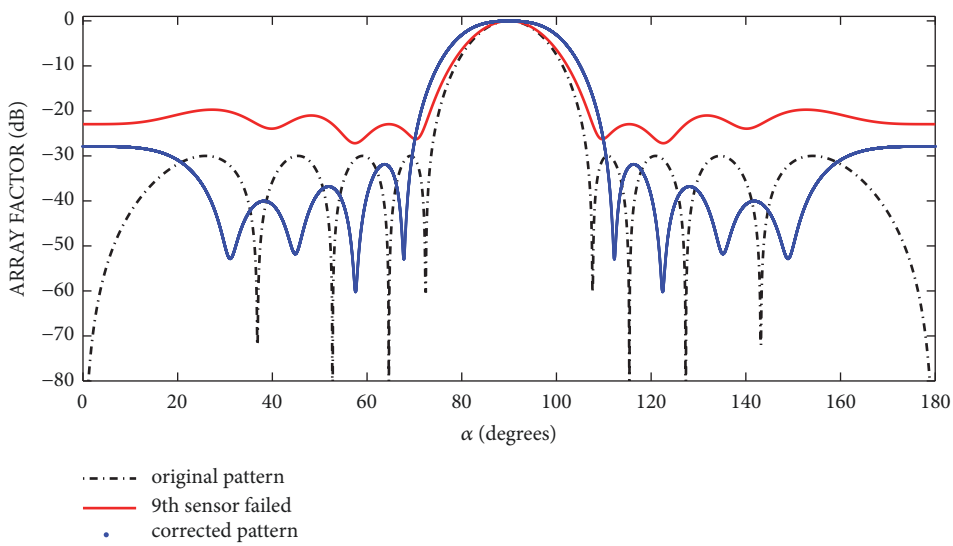

FIGURE 12: SLL recovered for 10-element 9th sensor failed array.

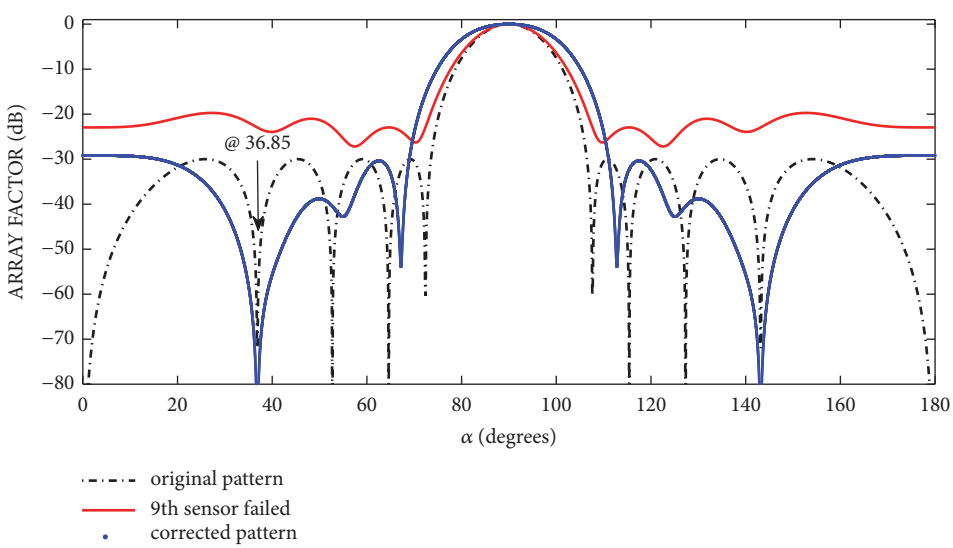

FIGURE 13: 4th null recovered for 10-element ULA 9th sensor failed.

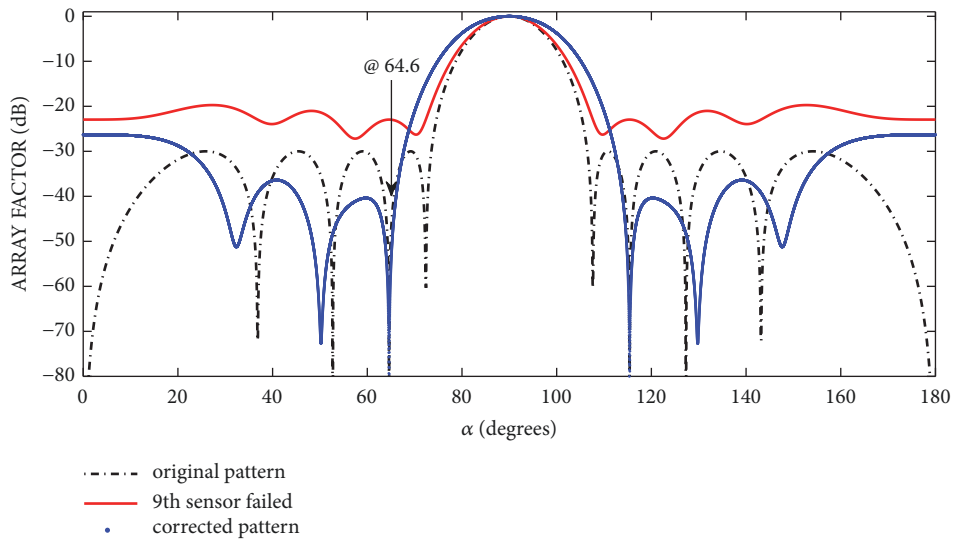

FIGURE 14: 2nd null recovered for 10-element ULA 9th sensor failed.

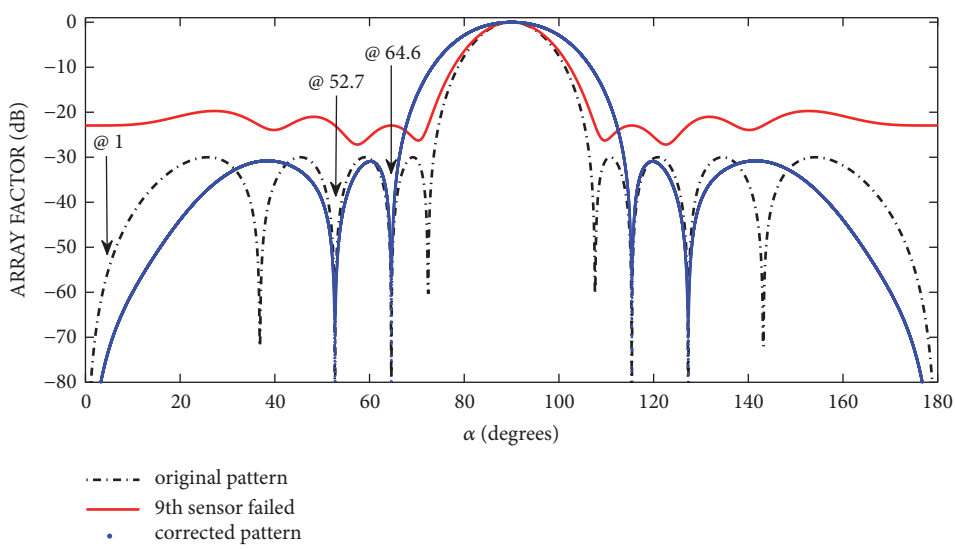

FIGURE 15: 2nd, 3rd, and 5th null recovered for 10-element ULA 9th sensor failed.

TABLE 5: Comparison of faulty and corrected pattern for 15-element array.

Faulty pattern		Corrected pattern		Recovery of nulls
SLL(dB)	*NDL (dB)*	*SLL(dB)*	*NDL (dB)*	
-10	*-16*	-30	-96	2 nulls
-10	*-16*	-30	-111	
-10	*-19*	-29	-61	4 nulls
-10	*-14*	-29	-91	
-10	*-12*	-29	-95	
-10	*-12*	-29	-108	
-10	*-18*	-30	−41	7 nulls
-10	*-15*	-30	-55	
-10	*-14*	-30	-71	
-10	*-19*	-30	-64	
-10	*-12*	-30	-82	
-10	*-10*	-30	-60	
-10	*-14*	-30	-71	

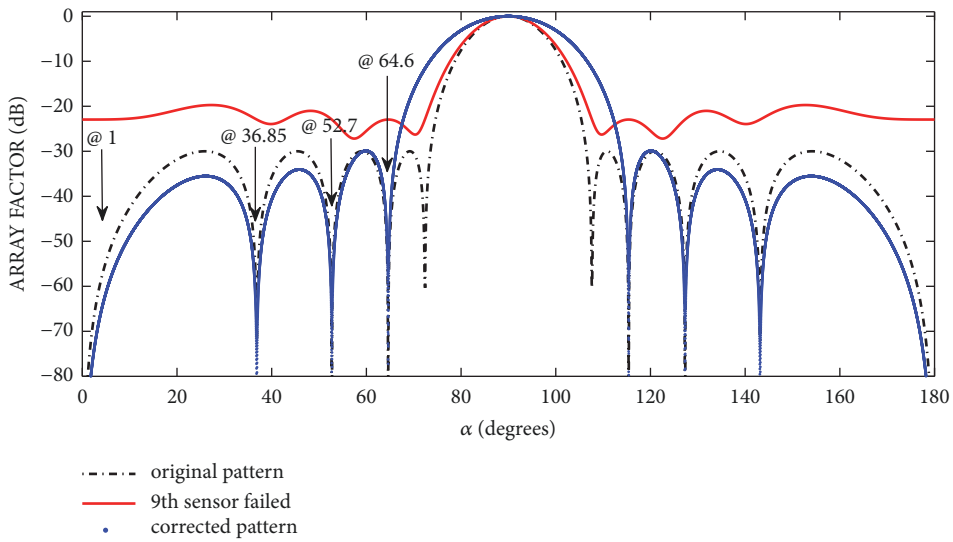

FIGURE 16: 2nd, 3rd, 4th, and 5th null recovered for 10-element ULA 9th sensor failed.

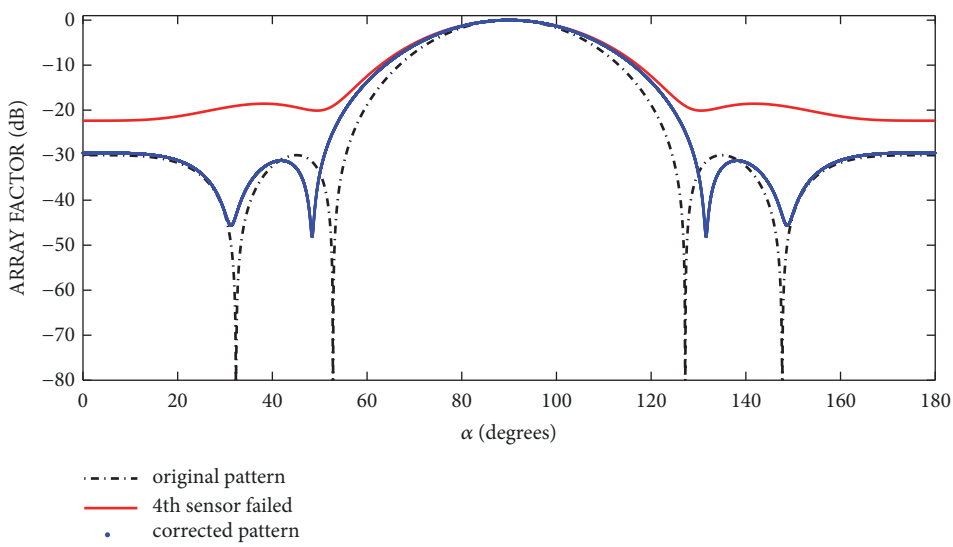

FIGURE 17: SLL recovered for 5-element 4th sensor ULA failed.

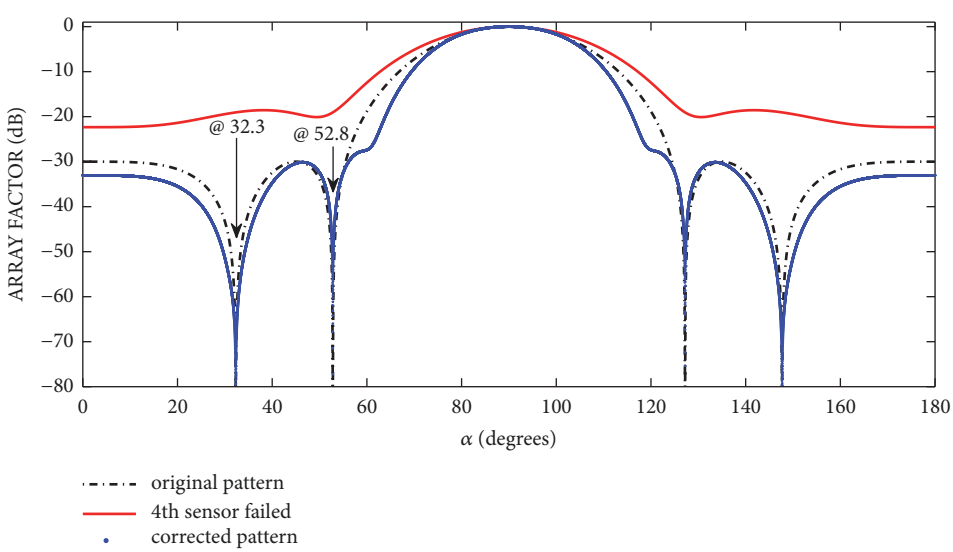

FIGURE 18: 1st and 3rd recovered for 5-element 4th sensor failed array.

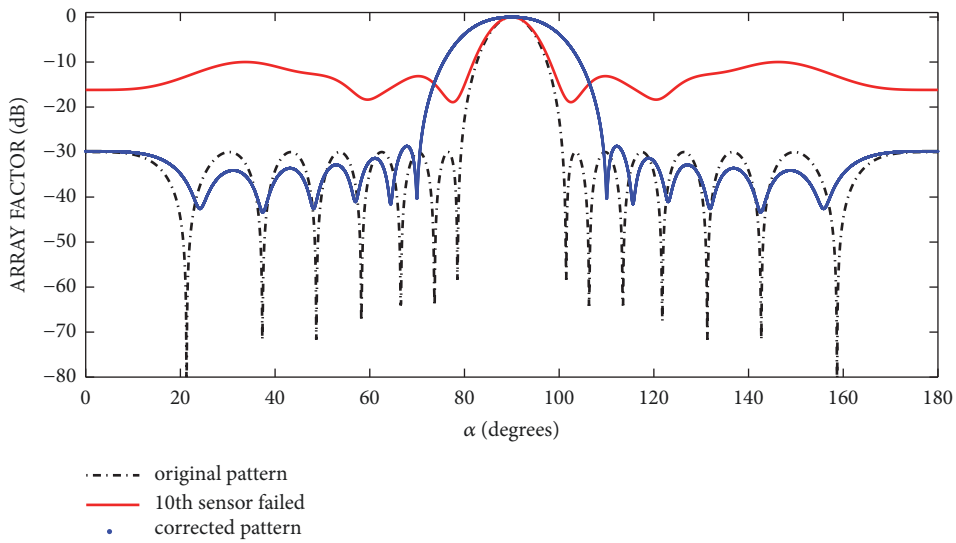

Figure 19: SLL recovered for 15-element ULA 10th sensor failed.

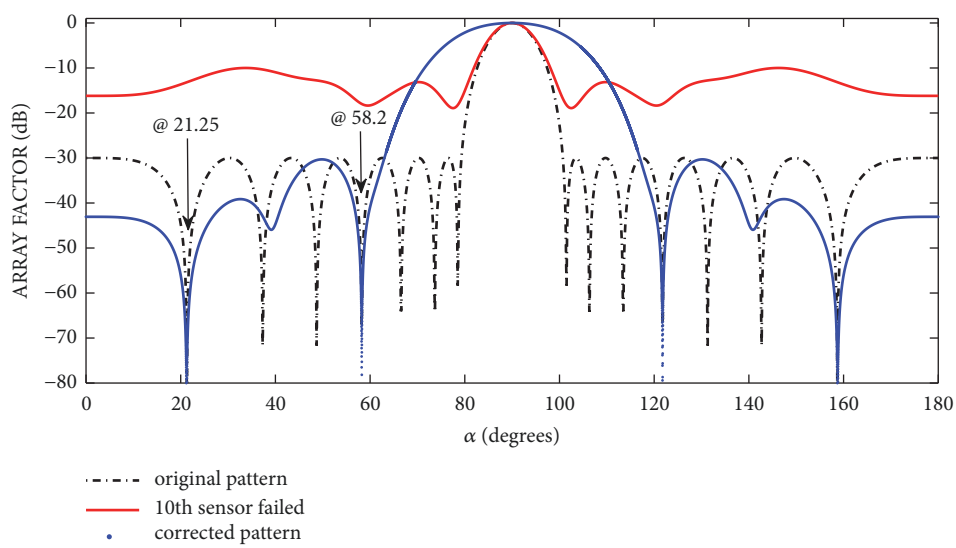

Figure 20: 4th and 7th null recovered for 15-element ULA 10th sensor failed.

Table 6: Comparison of BSA, GA, and GA-PS results.

BSA		GA [1]		GA-PS [1]		No of array elements
Fitness value (10^{-3})	*Time elapse* (s)	*Fitness value* (10^{-3})	*Time elapse* (s)	*Fitness value* (10^{-3})	*Time elapse* (s)	
1.57	49.79	6.24	147.1	5.41	154.3	5-element
4.56	74.15	8.56	204.5	6.43	218.9	10-element near failure
3.27	73.8	7.68	202.7	4.29	217.2	10-element far failure
7.08	94.3	9.86	295.4	8.47	335.8	15-element

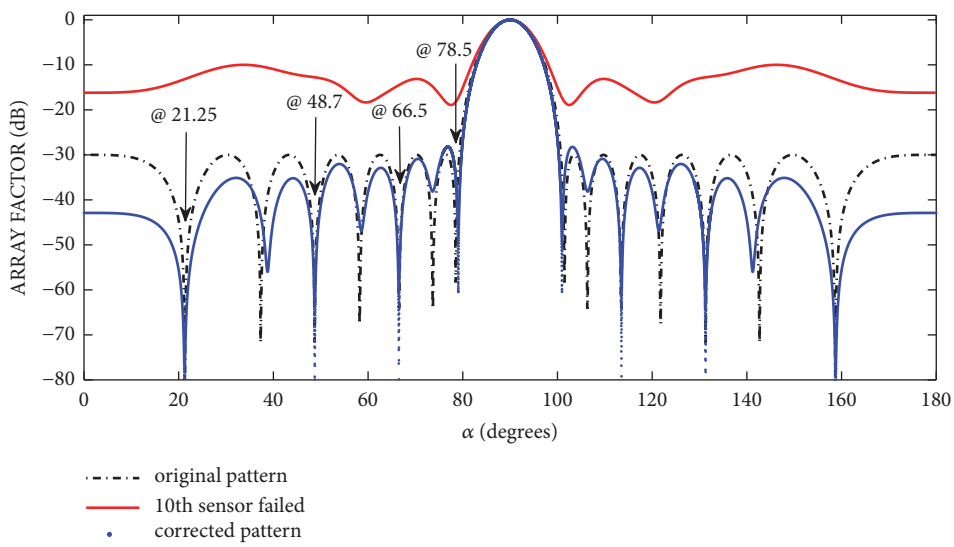

Figure 21: 1st, 3rd, 5th, and 7th null recovered for 15-element ULA 10th sensor failed.

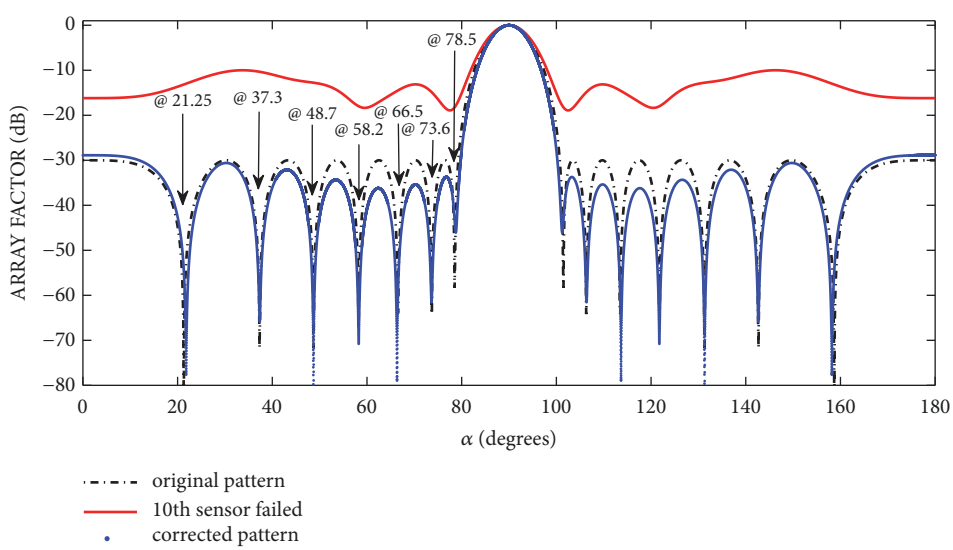

Figure 22: 1st, 2nd, 3rd, 4th, 5th, 6th, and 7th null recovered for 15-element ULA 10th sensor failed.

4. Conclusion and Future Work

In this paper, strength of BSA is used for the correction of faulty antenna array in wireless mobile communication systems. BSA has successfully reduced side lobe levels and repositioned single and multiple nulls at the original positions when any antenna element in the ULA is failed. The comparison of BSA is made with GA and GA-PS and it has been established that BSA produced better results in terms of computational complexity.

In future, one can explore BSA for other geometries of antenna arrays such as L shape, planar, and circular arrays. Similarly, BSA can be successfully applied to other problems in array signal processing such as adaptive beamforming, parameter estimation of signals, and tracking.

Data Availability

The data used to support the findings of this study are included within the article.

Ethical Approval

All the authors of the manuscript declared that there is no research involving human participants and/or animal.

Disclosure

The authors are responsible for any fees, if any, associated with this manuscript.

Conflicts of Interest

All the authors of the manuscript declared that there are no potential conflicts of interest.

Authors' Contributions

Fawad Zaman contributed in conceptualization, supervision, and review and editing, Hammad ul Hassan contributed in formal analysis and writing original draft, Shafqat Ullah Khan contributed in investigation and writing review and editing, Ata ur Rehman, contributed in formal analysis and validation, Muhammad Asif Zahoor Raja contributed in methodology and software, and Shahab Ahmad Niazi contributed in visualization and review and editing.

References

[1] S. U. Khan, I. M. Qureshi, F. Zaman, and A. Naveed, "Null placement and sidelobe suppression in failed array using symmetrical element failure technique and hybrid heuristic computation," *Progress in Electromagnetics Research*, no. 52, pp. 165–184, 2013.

[2] T. E. Tuncer and B. Friedlander, *Classical and Modern Direction-of-Arrival Estimation*, Academic Press, 2009.

[3] S. Akbar, M. A. Z. Raja, F. Zaman, T. Mehmood, and M. A. R. Khan, "Design of bio-inspired heuristic techniques hybridized with sequential quadratic programming for joint parameters estimation of electromagnetic plane waves," *Wireless Personal Communications*, vol. 96, no. 1, pp. 1475–1494, 2017.

[4] T. Pavani, G. S N Raju, and P. V Sridevi, "Pattern synthesis of linear array with position and position-amplitude control using firefly algorithm," *International Journal of Computer Applications*, vol. 99, no. 7, pp. 18–23, 2014.

[5] S. K. Mahto and A. Choubey, "A novel hybrid IWO/WDO algorithm for interference minimization of uniformly excited linear sparse array by position-only control," *IEEE Antennas and Wireless Propagation Letters*, vol. 15, pp. 250–254, 2016.

[6] A. Kumar, R. Mudumbai, S. Dasgupta et al., "A scalable feedback mechanism for distributed nullforming with phase-only adaptation," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 1, no. 1, pp. 58–70, 2015.

[7] W. P. M. N. Keizer, "Amplitude-only low sidelobe synthesis for large thinned circular array antennas," *IEEE Transactions on Antennas and Propagation*, vol. 60, no. 2, pp. 1157–1161, 2012.

[8] Z.-U. Khan, A. N. Malik, F. Zaman, S. A. Hussain, and A.-R. Khan, "Sidelobe suppression with null steering by independent weight control," *International Journal of Antennas and Propagation*, vol. 2015, pp. 1–6, 2015.

[9] J. Kennedy, "Particle swarm optimization," in *Encyclopedia of Machine Learning*, pp. 760–766, Springer, USA, 2011.

[10] B. Yuce, E. Mastrocinque, M. S. Packianather, A. Lambiase, and D. T. Pham, "The bees algorithm and its applications," *Handbook of Research on Artificial Intelligence Techniques and Algorithms*, pp. 122–151, 2014.

[11] K. M. Passino, "Bacterial foraging optimization," in *Innovations and Developments of Swarm Intelligence Applications*, vol. 219, 2012.

[12] V. Feoktistov, *Differential evolution*, Springer, USA, 2006.

[13] D. Whitley, "A genetic algorithm tutorial," *Statistics and Computing*, vol. 4, no. 2, pp. 65–85, 1994.

[14] M. Dorigo and M. Birattari, "Ant colony optimization," in *Encyclopedia of machine learning*, pp. 36–39, Springer, Boston, MA, USA, 2011.

[15] P. Civicioglu, "Backtracking search optimization algorithm for numerical optimization problems," *Applied Mathematics and Computation*, vol. 219, no. 15, pp. 8121–8144, 2013.

[16] W. F. Gao and S. Y. Liu, "A modified artificial bee colony algorithm," *Computers & Operations Research*, vol. 39, no. 3, pp. 687–697, 2012.

[17] D. Karaboga, K. Guney, and A. Akdagli, "Antenna array pattern nulling by controlling both amplitude and phase using modified touring ant colony optimization algorithm," *International Journal of Electronics*, vol. 91, no. 4, pp. 241–251, 2004.

[18] X.-S. Yang, "Firefly algorithms for multimodal optimization," in *International Symposium on Stochastic Algorithms*, pp. 169–178, Springer, Berlin, Heidelberg, Germany, 2009.

[19] M. A. Z. Raja, M. A. Manzar, F. H. Shah, and F. H. Shah, "Intelligent computing for Mathieu's systems for parameter excitation, vertically driven pendulum and dusty plasma models," *Applied Soft Computing*, vol. 62, pp. 359–372, 2018.

[20] Z. Sabir, M. A. Manzar, M. A. Raja, M. Sheraz, and A. M. Wazwaz, "Neuro-heuristics for nonlinear singular Thomas-Fermi systems," *Applied Soft Computing*, vol. 65, pp. 152–169, 2018.

[21] S. Akbar, F. Zaman, M. Asif, A. U. Rehman, and M. A. Z. Raja, "Novel application of FO-DPSO for 2-D parameter estimation of electromagnetic plane waves," *Neural Computing and Applications*, pp. 1–10, 2018.

[22] A. Mehmood, A. Zameer, and M. A. Z. Raja, "Intelligent computing to analyze the dynamics of Magnetohydrodynamic flow over stretchable rotating disk model," *Applied Soft Computing*, vol. 67, pp. 8–28, 2018.

[23] A. Zameer, J. Arshad, A. Khan, and M. A. Z. Raja, "Intelligent and robust prediction of short term wind power using genetic programming based ensemble of neural networks," *Energy Conversion and Management*, vol. 134, pp. 361–372, 2017.

[24] F. Zaman, "Joint angle-amplitude estimation for multiple signals with L-structured arrays using bioinspired computing," *Wireless Communications and Mobile Computing*, vol. 2017, Article ID 9428196, 12 pages, 2017.

[25] M. Pakdaman, A. Ahmadian, S. Effati, S. Salahshour, and D. Baleanu, "Solving differential equations of fractional order using an optimization technique based on training artificial neural network," *Applied Mathematics and Computation*, vol. 293, pp. 81–95, 2017.

[26] A. Jafarian, R. Jafari, A. K. Golmankhaneh, and D. Baleanu, "Solving fully fuzzy polynomials using feed-back neural networks," *International Journal of Computer Mathematics*, vol. 92, no. 4, pp. 742–755, 2015.

[27] M. A. Zahoor Raja, Z. Shah, M. Anwaar Manzar, I. Ahmad, M. Awais, and D. Baleanu, "A new stochastic computing paradigm for nonlinear Painlevé II systems in applications of random matrix theory," *The European Physical Journal Plus*, vol. 133, no. 7, p. 254, 2018.

[28] A. Jafarian, M. Mokhtarpour, and D. Baleanu, "Artificial neural network approach for a class of fractional ordinary differential equation," *Neural Computing and Applications*, vol. 28, no. 4, pp. 765–773, 2017.

[29] A. D. Bresler, "A new algorithm for calculating the current distributions of Dolph- Tschebyscheff arrays," *IEEE Transactions on Antennas and Propagation*, vol. 28, no. 6, pp. 951-952, 1980.

[30] N. Balakrishnan and R. Sethuraman, "Easy generation of Dolph-Tschebyscheff excitation coefficients," *Proceedings of the IEEE*, pp. 1508-1509, 1981.

[31] H. Steyskal, R. A. Shore, and R. L. Haupt, "Methods for null control and their effects on the radiation pattern," *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, pp. 404–409, 1986.