World Applied Sciences Journal 26 (2): 232-238, 2013
ISSN 1818-4952

DOI: 10.5829/idosi.wasj.2013.26.02.1387

Application of Firefly Algorithm to Fault Finding in Linear Arrays Antenna

[1]S.U. Khan, [2,4]I.M. Qureshi, [3]F. Zaman, [3]A. Basit and [3]W. Khan

[1]School of Engineering and Applied Sciences, ISRA University, Islamabad, Pakistan
[2]Department of Electrical, Air University, Islamabad, Pakistan
[3]Department of Electronic Engineering, IIU, H-10, Islamabad, Pakistan
[4]Institute of Signals, Systems and Soft Computing (ISSS), Islamabad, Pakistan

Submitted: Aug 25, 2013; **Accepted:** Oct 27, 2013; **Published:** Nov 12, 2013

Abstract: Detection of faulty elements in an array of sensors is a practical issue which has applications in radar, satellite and mobile communication. Due to element failure, the radiation pattern disturb in terms of sidelobe level, damage of nulls and increase of bandwidth. In this paper, we develop a new technique based on Firefly Algorithm (FA) to locate the position of faulty elements in a linear array. The FA is a global optimization method and come under the umbrella of swarm optimization. The cost function is used as a fitness evaluation function which defines an error between the degraded far field power pattern and the estimated one. The proposed algorithm is used successfully for the detection of complete, as well as, for partial faulty elements position. Various simulation results are evaluated for 34 elements Chebyshev array of specific Side Lobe Level (SLL), to validate and test the performance of the proposed algorithm.

Key words: Array Antenna · Firefly Algorithm · Fault Finding

INTRODUCTION

Fault finding in antenna array is a hot problem which has direct application in radar, satellite and mobile communication. The antenna array of such applications have large number of radiating elements and the possibility of getting failure of one or more elements increases due to unforeseen reasons. The malfunction of one or more elements increases the Side Lobe Level (SLL), displacement of nulls from their unique positions and increases the bandwidth of the power pattern. Array antenna has the advantage that the weights of the active elements can be re-adjusted to achieve the required radiation pattern. R. J. Mailloux [1] used digitally beamformed array for the correction of array element failure. In the literature different techniques are developed for this compensation that numerically finding a set of weights of the active elements that minimized the fitness function [2-7]. But before using these compensation methods one has to locate the faulty elements. Authors in their previous work [8] have used the symmetrical element failure technique to achieve the required null depth level and first null beamwidth. In [9] J. A. Rodríguez *et al.* used genetic algorithm to find the defective element's position in planar array while in [10] the same author used a simple and rapid technique for finding the defective elements in antenna arrays. In [11] artificial neural network is used for finding the three faulty elements in a small array of 16-elements. Bucci *et al.* [12] studies the uncertainty of the solution in the continuous and the discrete on-off cases using amplitude-only pattern and then propose an adapted genetic algorithm to solve the problem in the discrete case. Nan Xu *et al.* [13] used machine learning optimization for the detection failure of antenna array elements.

In this paper, we develop a new technique based on Firefly Algorithm (FA) to locate the position of faulty elements in a linear array. The FA is a global optimization method and come under the umbrella of swarm optimization. The cost function is used as a fitness evaluation function which defines an error between the degraded far field power pattern and the estimated one. The minimum of cost function will give us the location of faulty element. The proposed algorithm is used successfully for the detection of complete, as well as, for partial faulty elements position. Simulation results are

Corresponding Author: S.U. Khan, School of Engineering and Applied Sciences, ISRA University, Islamabad, Pakistan.

evaluated for 34 elements Chebyshev array of specific SLL, to authenticate and test the performance of the proposed algorithm. The rest of the article is organized as follows. The problem formulation is discussed in section 2, while in section 3 the FA is provided. Section 4 describes the simulations and the results while section 5 concludes the paper and proposes some future work.

Problem Formulation: Consider a linear array of 34 elements and the elements are placed along z-axis, equally spaced, non-uniform and the progressive phase excitation is given by the array factor [14, 15].

$$AF(\theta_i) = \sum_{n=1}^{N} w_n \cos\left[kd\cos\theta_i(2n-1)/2 + \alpha\right] \quad (1)$$

where w_n is the weight vector of the nth element, k is the wave number and α is the progressive phase shift. Now suppose that w_m is failed in the array, by putting their weight w_m equal to zero in equation (1). The damage array factor can be expressed as

$$AF_m(\theta_i) = \sum_{\substack{n=1 \\ n\neq m}}^{N} w_n \cos\left[kd\cos\theta_i(2n-1)/2 + \alpha\right] \quad (2)$$

The far field pattern of the damage element can be calculated in dB's by the following expression as:

$$P_d(\theta_i) = 10.\log_{10}\left(\sum_{\substack{n=1 \\ n\neq m}}^{N} w_n \cos\left[kd\cos\theta_i\left(2n-1\right)/2 + \alpha\right]\right)^2 \quad (3)$$

The defected array pattern can be obtained from equation (1) by making the weight equals to zero to represent a complete faulty element. The partial fault element (50%) is equivalent to assume that their relative weight equal to half of the original weight.

The cost function in equation (4) is minimized using FA and to find the weights which generate the radiation pattern that is close to the measured one.

$$I = \sum_{k=1}^{K}\left[\left|P_d(\theta_k) - P_o(\theta_k)\right|\right]^2 \quad (4)$$

Where K the number of samples used in the comparison is, $P_d(\theta_k)$ is the desired degraded array pattern and $P_o(\theta_k)$ is the pattern obtained from FA in K directions. This cost function compares the measured radiation pattern with the given arrangement of failed elements and the minimal of this cost function will give the location of complete, as well as partially faulty element.

Firefly Algorithm (FA): FA is new swarm optimization algorithm motivated by behavior and motion of fireflies, developed by Xin-She Yang [16-17] in 2007. It is a population-based meta-heuristic algorithm which uses swarm intelligence. It is similar to other meta-heuristic algorithm like GA, PSO and DEA but FA is found to have improved performance in many cases [18]. FA is based on the flashing light of fireflies. The flashing light of fireflies is a remarkable sight in the tropical and temperature region. The FA has three rules which are based on (i) all fireflies are unisex and they will move towards more attractive and brighter ones. (ii) The attractiveness of a firefly is proportional to its brightness which decreases as the distance from the other firefly increases. If there is not a brighter or more attractive firefly than a particular one, then it will move randomly in the space. (iii) The brightness of a firefly is determined by the value of the fitness function. For minimization problems, the light intensity is inversely proportional to the value of the fitness function. The flow chart for FA is shown in Fig. 1 while the important steps are summarized below.

Step 1 Initialization: The number of fireflies in the population space is K. The position of the *n*th firefly is denoted by a vector w_n, where each measurement indicates the weight of an array element.

$$\mathbf{w}_n = \left(w_n^1, w_n^2, w_n^3, \ldots w_n^t, \ldots w_n^N\right) \quad (5)$$

Where n=1, 2, 3…K and t=1, 2, 3…N. To initialize the location of K fireflies in N-dimensional search space, which are randomly selected with in the search boundary by the equation (6)

$$w_n^t = w_L^t + \left(x_H^t - x_L^t\right) \times rand(\,) \quad (6)$$

where w_L^t and w_H^t represents the lower and upper values of the t-th variable in the population respectively and rand ( ) is a uniform random variable with values ranging from 0 to 1.

Step 2 Fitness Function: Calculate the fitness for each firefly position in the population and sort the population from brightest to lightest. The brightness of each firefly is calculated at current generation by the fitness function at

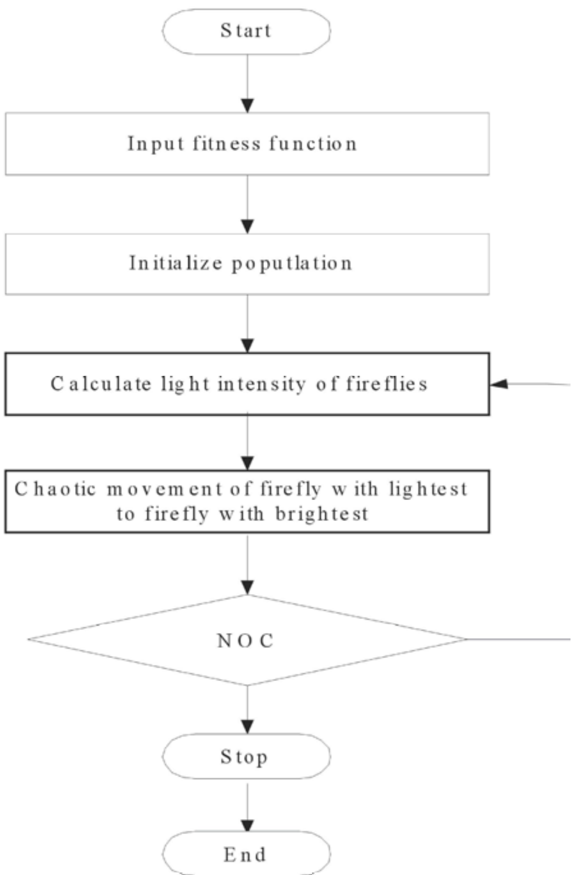

Fig. 1: The flow diagram of firefly algorithm

their current location. The brightest or light intensity is inversely proportional to cost function of individual firefly for minimization problem.

Step 3 Update the Location of Fireflies: The location of each firefly in the population depending on the attractiveness and each firefly in the population will move toward the adjacent firefly with more light intensity and its position is updated for the next generation. The firefly i (less intensity) will move toward the other fireflies j that are brighter. There are two important issues in the FA, the deviation of brightness or light intensity and formation of the attractiveness. The attractiveness of a firefly is calculated by its brightness or light intensity which is directly associated with the cost function. The brightness of the nth firefly B_n is given by the equation,

$$B_n = f_{fitness}(w_n) \tag{7}$$

The attractiveness between the i-th and j-th firefly is given by

$$\beta_{ij} = \beta_{\circ} \exp(-\gamma r_{ij}^2) \tag{8}$$

Where β is a constant whose value is 1, γ is dynamic range of search space and r_{ij} is a distance between w_i and w_j given by the equation

$$r_{ij} = \left\| w_j - w_i \right\| = \sqrt{\sum_{n=1}^{N} (w_j^t - w_i^t)^2} \tag{9}$$

The position of firefly is updated in each iterative step. If the intensity/brightness of j-th firefly is greater than the brightness of the i-th firefly, then the ith firefly moves towards the j-th firefly. The motion of the fireflies is denoted by the following equation

$$w_n = w_n + \beta_{ij}\left(w_j - w_i\right) + \alpha\varepsilon_n \tag{10}$$

Where α is a constant whose value depends on the dynamic range of the solution space and rand ( ) is random number between 0 and 1.

Step 4 Ranking and Computation of Global Best: On the basis of their brightness, the fireflies are ranked in the current generation and the brightness of each firefly is compared with all other fireflies and the location of the brightest firefly in the population is taken as current global best and in this way, for the brightest firefly we received a best fitness value in the recent generation.

Step 5 Termination of Program: When fitness function achieves a certain prescribed value, or when maximum number of cycles (NOC) is reached, the program terminates and stores the best value, otherwise it goes back to step 2 to 4. The location of the best firefly gives the optimum solution and the corresponding brightness of firefly gives the optimum fitness value of the fitness function.

Simulation and Results: In this section, in order to implement the FA, we set the values of the parameters, i.e the population size P, the light absorption co-efficient γ, the parameter α, the attractiveness β, the lower value w_L^t and the upper value w_H^t are taken as 30, 1, 0.25, 0.2, 0 and 1 respectively. Consider a Classical Dolph-Chebyshev linear array of 34 elements with $\lambda/2$ inter-element spacing is used as the test antenna. The linear array design is a -30 dB constant SLL. Analytical techniques are used to find out the non-uniform excitations for Classical Dolph-Chebyshev array.

The original Chebyshev array pattern for 34 elements is shown in Fig. 2 and its normalized weight distribution is shown in Fig.3. The Chebyshev weight obtained for 34 element linear array by analytical method are given in Table 1. The complete as well as partial faults were created by making the Chebyshev weights either equal to zero or some fraction of the original weight respectively. The cost function in equation (4) is minimized using FA w.r.t to the weights of the element. Various combinations of faults are discussed, i.e single fault, more than one single fault and combination of partial as well as complete fault are tested.

At first instant we consider that 5th element failed (100%) and 10th element (25%). Fig. 4 shows the damage pattern of the 5th (100%) and 10th (25%) element and Fig. 5 depicts the weight distribution of the damage pattern. From Fig. 4 it is clear that after failure its pattern become disturb completely and from the damage pattern it will be very difficult to detect the faulty element position. Now we run FA to locate the faulty element position as well as the grade of failure. Fig.6 shows the weights of the damage pattern and the weights obtained by FA to locate the faulty elements position. From this comparison one can clearly observe the defective element position, as well as, the grade of failure of damage elements. The weights obtained by FA, which shows the location of defective elements are given in Table 1.

Now the algorithm is tested for different types of fault and simulation results show the performance of the proposed method. We assumed that the elements located at 6th (100%) and 15th (50%) positions are failed. We received the damage pattern for this faulty configuration, which are depicted in Fig. 7 and the damage weight distribution is shown in Fig. 8. Now the FA is run to locate the positions of the faulty elements. Fig. 9 shows the comparison of the weights of the damaged pattern with the weights obtained by the FA is given in Table 1. This comparison, clearly indicate the position as well as the grade of faulty elements.

Now the proposed algorithm is tested for the fault located at 6th (100%), 15th (50%) and 21th (100%). The damage array pattern and weight distribution are shown in Fig. 10 and Fig. 11. In this case, the algorithm also able to locate the complete, as well as, partial faults successfully. Fig. 12 shows the comparison of the damage pattern and the weights obtained by FA, which clearly marks the position of faulty element. The weights obtained by FA, which indicates the positions of faulty elements are given in Table 1.

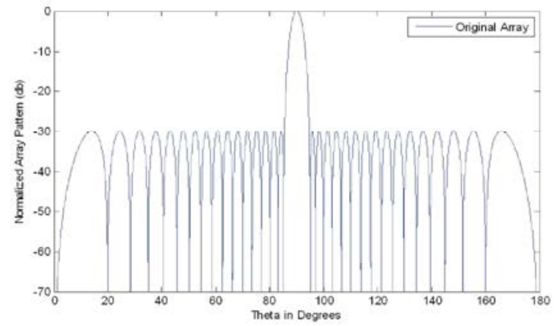

Fig. 2: The Original -30-dB Chebyshev array of 34 elements.

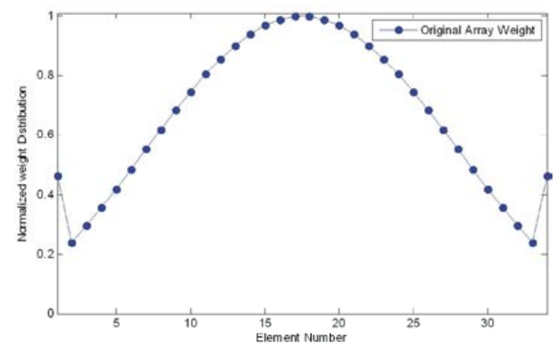

Fig. 3: The Original Chebyshev array weight distribution of 34 elements.

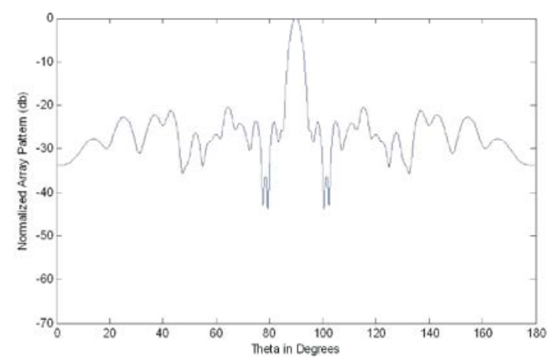

Fig. 4: The Defected pattern with fault at 5th (100%) and 10th (25%) element.

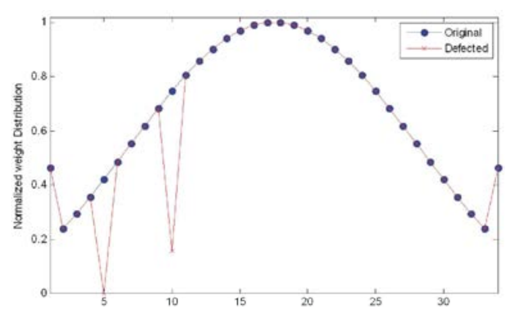

Fig. 5: The weight distribution of Original, with fault at 5th (100%) and 10th (25%) element.

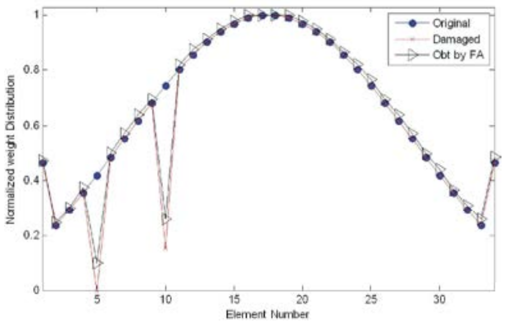

Fig. 6: Performance of Firefly Algorithm with fault at 5th (100%) and 10th (25%) element.

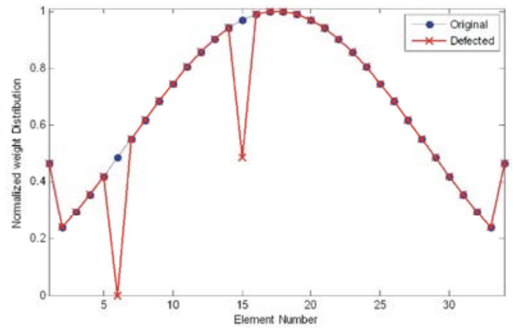

Fig. 8: The weight distribution of Original and with fault at 6th and 15th (50%) element.

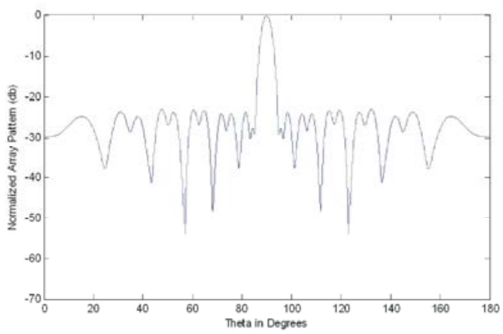

Fig. 7: The Defected pattern with fault at 6th and 15th (50%) element.

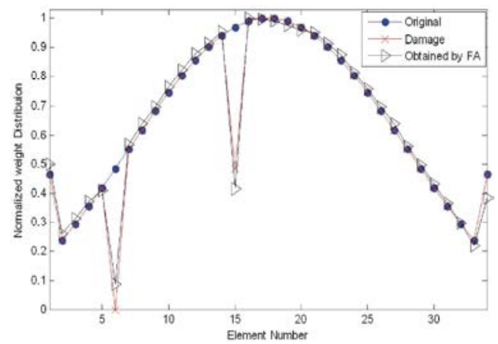

Fig. 9: Performance of Firefly Algorithm with fault at 6th and 15th (50%) element.

Table 1: Original Chebyshev weights and the weights obtained by Firefly Algorithm

Element Number	Chebyshev Weights	5th (100%) and 10th (25%) Damaged	6th 100%) and 15th (50%) Damage	6th (100%), 15th (50%) and 21th (100%) Damage
1	0.4645	0.4764	0.4783	0.4698
2	0.2395	0.2487	0.2465	0.2389
3	0.2956	0.3015	0.2989	0.3041
4	0.3559	0.3672	0.3713	0.3735
5	0.4195	0.0159	0.4387	0.4416
6	0.4854	0.4913	0.0027	0.0135
7	0.5523	0.5637	0.5743	0.5741
8	0.6190	0.6203	0.6298	0.6420
9	0.6841	0.6795	0.6968	0.6976
10	0.7464	0.1978	0.7648	0.7673
11	0.8044	0.8137	0.8193	0.8375
12	0.8569	0.8635	0.8598	0.8643
13	0.9027	0.9165	0.9312	0.9318
14	0.9407	0.9531	0.9621	0.9503
15	0.9700	0.9862	0.3987	0.4026
16	0.9899	0.9987	0.9976	0.9917
17	1.0000	0.9989	0.9897	1.0000
18	1.0000	1.0000	0.9986	0.9869
19	0.9899	0.9968	0.9987	0.9975
20	0.9700	0.9877	0.9865	0.9864
21	0.9407	0.9513	0.9586	0.0147

Table 1: Continue

Element Number	Chebyshev Weights	5th (100%) and 10th (25%) Damaged	6th 100%) and 15th (50%) Damage	6th (100%), 15th (50%) and 21th (100%) Damage
22	0.9027	0.9128	0.9587	0.9356
23	0.8569	0.8659	0.8674	0.8771
24	0.8044	0.8132	0.8192	0.8198
25	0.7464	0.7534	0.7643	0.7589
26	0.6841	0.6935	0.6978	0.6984
27	0.619	0.6257	0.6361	0.6474
28	0.5523	0.5651	0.5713	0.5657
29	0.4854	0.4972	0.4967	0.4971
30	0.4195	0.4386	0.4391	0.4376
31	0.3559	0.3787	0.3743	0.3784
32	0.2956	0.3325	0.3015	0.3125
33	0.2395	0.2450	0.2395	0.2712
34	0.4645	0.4856	0.4645	0.4835

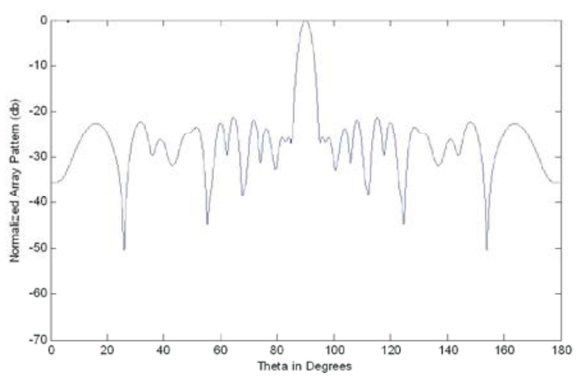

Fig. 10: The Defected pattern with fault at 6th and 15th (50%) and 21th element.

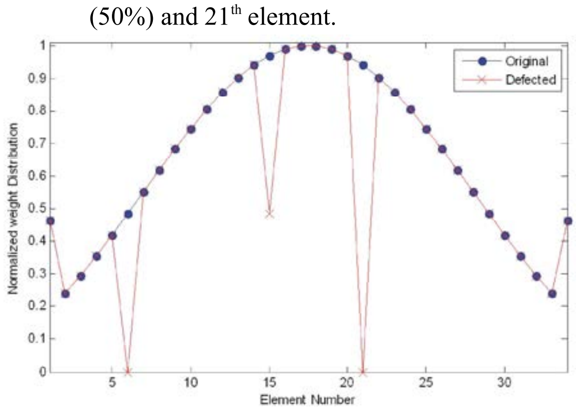

Fig. 11: The weight distribution of Original and with fault at 6th, 15th (50%) and 21th element.

Table 1 shows the weights obtained by FA for different cases. Comparison of the weights obtained by FA with that of the damage array show the positions and grade of failure of the defective elements.

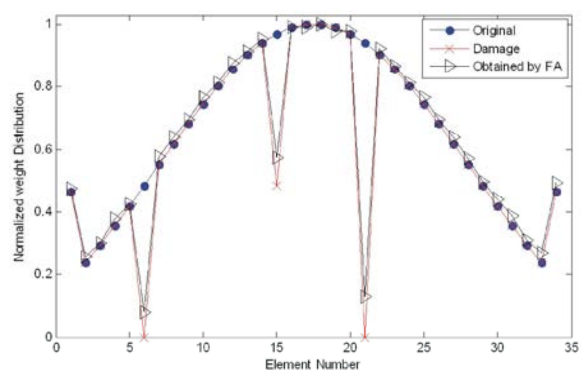

Fig. 12: Performance of Firefly Algorithm with fault at 6th, 15th (50%) and 21th element.

CONCLUSION

We have developed FA for fault finding in linear arrays. This meta-heuristic computing method is used to find the weights of the damaged array of the far field pattern and then compared the weights with the damage Chebyshev array weights to find the location and grade of failure of the defected array. Complete as well as partial fault were consider at different position and trace out successfully. FA is used first time for fault finding in antenna arrays. This method can be extended to planar arrays.

REFERENCES

1. Mailloux, R.J., 1996. "Array failure correction with a digitally beamformed array," IEEE Trans Antennas Propag. 44: 1542-1550.

2. Levitas, M., D.A. Horton and T.C. Cheston, 1999. “Practical failure compensation in active phased arrays,” IEEE Trans. Antennas Propag. 47(3): 524-535.
3. Yeo, B. and Y. Lu, 1999. “Array failure correction with a genetic algorithm,” IEEE Trans. Antennas Propag. 47(5): 823-828.
4. Steyskal, H. and R.J. Mailloux, 1998. “Generalization of an array failure correction method,” Proc. Inst. Elect. Eng. Microw. Antennas Propag. 145(4): 332-336.
5. Peters, T.J., 1991. “A conjugate gradient-based algorithm to minimize the sidelobe level of planar arrays with element failures,” IEEE Trans. Antennas Propag. 39(10): 1497-1504.
6. Mailloux, R.J., 2005. Phased Array Antennas Handbook, 2nd ed. Norwood, MA: Artech House.
7. Rodríguez, J.A. and F. Ares, 1998. “Optimization of the performance of arrays with failed elements using the simulated annealing technique,” J. Electromagn. Waves Appl., 12: 1625-1637.
8. Shafqat U. Khan, Ijaz M. Qureshi, Fawad Zaman and Aqdas Naveed, 2013. "Null Placement and Sidelobe Suppression in Failed Array Using Symmetrical Element Failure Technique and Hybrid Heuristic Computation," Progress In Electromagnetics Research B, 52: 165-184.
9. Rodríguez, J.A., F. Ares, H. Palacios and J. Vassal’lo, 2000. “Finding defective elements in planar arrays using genetic algorithms,” Progr. Electromagn. Res. PIER, 29: 25-37.
10. Rodríguez, J.A., F. Ares, H. Palacios, M.F. Delgado, R. Iglesias and S. Barrow, 2009. " Rapid method for finding faulty elements in antenna arrays,” IEEE Trans Antennas Propagat, 57: 1679-1683.
11. Patnaik, A., B. Choudhury, P. Pradhan, R. K. Mishra and C. Christodoulou," 2007. An ANN application for fault finding in antenna arrays," IEEE Trans Antennas Propagat, 55: 775-777.
12. Bucci, O.M., A. Capozzoli and G. D’Elia, 2000. “Diagnosis of array faults from far-field amplitude-only data,” IEEE Trans. Antennas Propag. 48(5): 647-652.
13. Nan Xu, C.G. Christodulou, S.E. Barbin and M. Martin-Ramon, 2007. Detecting failure of antenna array elements using machine learning optimization, 2007 IEEE Antennas and Propagation Social International Symposium, Honolulu, HI, USA, pp: 5753-5756.
14. Wolf, I., 1937. Determination of the radiating system which will produce a specified directional characteristic. Proc. IRE, 25: 630-643.
15. Balanis, C.A., 2005. "Antenna theory: analysis and design," John Wiley and Sons, Inc.
16. Yang, X.S., 2009. Firefly algorithms for multimodal optimization, in O.Watanabe and T. Zeugmann, Eds. Stochastic Algorithms: Foundations and Appplications, Vol. 5792 of SAGA 2009, Lecture Notes in Computer Science, Springer, Berlin, Germany, pp: 169-178.
17. Yang, X.S., 2010. Firefly algorithm, stochastic test functions and design optimization, Int J Bio-Inspired Comput, 2: 78-84.
18. Basu, B. and G.K. Mahanti, 2011. “Firefly and artificial bees colony algorithm for synthesis of scanned and broadside linear array antenna,” Progress in Electromagnetic Research B, 32: 169-190.