

Research Article

QPSO versus BSA for Failure Correction of Linear Array of Mutually Coupled Parallel Dipole Antennas with Fixed Side Lobe Level and VSWR

R. Muralidharan,¹ Vallavaraj Athinarayanan,¹ G. K. Mahanti,² and Ananya Mahanti²

¹ Department of Electronic and Computer Engineering, Caledonian College of Engineering, Oman

² Department of Electronics and Communication Engineering, National Institute of Technology, Durgapur 713209, India

Correspondence should be addressed to G. K. Mahanti; gautammahanti@yahoo.com

Received 11 July 2014; Accepted 8 September 2014; Published 28 September 2014

Academic Editor: Sarod Yatawatta

Copyright © 2014 R. Muralidharan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Evolutionary algorithms play an important role in synthesizing linear array antenna. In this paper, the authors have compared quantum particle swarm optimization (QPSO) and backtracking search algorithms (BSA) in failure correction of linear antenna arrays constructed using half wavelength uniformly spaced dipoles. The QPSO algorithm is a combination of classical PSO and quantum mechanics principles to enhance the performance of PSO and BSA is considered as a modernized PSO using historical populations. These two algorithms are applied to obtain the voltage excitations of the nondefective elements in the failed antenna array such that the necessary objectives, namely, the minimization of parameters like side lobe level (SLL) and voltage standing wave ratio (VSWR), are achieved leading to their values matching closely the desired parameter values. The results of both algorithms are compared in terms of parameters used in the objective function along their statistical parameters. Moreover, in order to reduce the processing time, inverse fast Fourier transform (IFFT) is used to obtain the array factor. In this paper, an example is presented for the application of the above two algorithms for a linear array of 30 parallel half wavelength dipole antennas failed with 4 elements and they clearly show the effectiveness of both the QPSO and BSA algorithms in terms of optimized parameters, statistical values, and processing time.

1. Introduction

Any failure in the elements of the antenna array [1] will end up in corruption of radiation pattern and lead to deviation of the radiation pattern parameters, namely, side lobe level, first null beam width (FNBW), half power beam width (HPBW), and so forth, thus leading to degradation in the performance of the communication systems. Researches in the past have clearly depicted various ways of recovery [2–4]. A simple way to recover the radiation pattern is to manually replace the failed antenna elements, which is not quite a better solution when the deformation occurs in applications like satellites, radar, and so forth. Literature survey has revealed that the idea of adjusting the beam weights of the remaining nondefective elements in the array results in production of radiation pattern approaching the original pattern in terms of the radiation pattern parameters being extensively used.

Mailloux [2] utilized the idea of replacing the signals in a digitally beam formed array in a multiple Signal environment without laying any restrictions on correlation properties of signals. A method [3] based on evolutionary algorithm, namely, genetic algorithm [3–5], has been utilized for failure correction by considering the beam weights as an array of complex vector numbers. This algorithm has also been used by Rodriguez et al. [4], successfully in such a way that the adjustments of excitations for minimum number of elements were needed to recover the radiation pattern. Thus, genetic algorithm has established itself as one of the primitive algorithms which enjoyed its tenure in the past and the primary reason was due to its different mating schemes, decimal crossover principles, and so forth. Following it, particle swarm optimization [6] came into picture which dealt with the intelligence and cooperation rules followed by a certain set of birds. Till then, evolutionary algorithms

started occupying the prime position of providing adjustment of beam weights for the correction of radiation patterns. The beam weights can refer to only amplitude or both amplitude and phase excitations of the individual elements. Amplitude only control avoids the unnecessary use of phase shifters, thus reducing the overall complexity in feed circuitry.

Researches also have shown the validity of algorithms [7] superseding over others in terms of the values of optimized outputs, convergence time, and statistical values like mean of the fitness value, standard deviation, and so forth. Moreover, literature survey also reveals that inverse fast Fourier transform [5, 8, 9] can be used to speed up the whole process of failure correction. Here, the relationship of inverse fast Fourier transform (IFFT) with the current excitations of the antenna leading to the array factor is utilized and has been used for synthesis of low side lobe radiation patterns [9]. This relationship is primarily used here for the sole purpose of reducing the overall computing time. Wang et al. [5] incorporated the principles of IFFT in obtaining the array factor computation in a short time and verified successfully the proposed idea for both the linear and planar arrays.

In this paper, mutual coupling [8, 10–12] between the elements is taken into consideration as, in practice, mutual coupling deteriorates not only the antenna radiation pattern but also the matching characteristics. In other words, it reduces the antenna efficiency and performance of antennas in both transmit and receive mode.

Here, the problem is to correct the radiation pattern of 30-element linear antenna array corrupted with 4 elements with the objective of making the SLL and VSWR approach close to the expected values. Two algorithms, namely, QPSO [13, 14] and BSA [15, 16], have been used for this purpose. Moreover, SLL refers to the ratio of the amplitude of the peak lobe to that of the side lobe of a radiation pattern. It is usually expressed in decibels. The second parameter that is used in this paper for correction, namely, VSWR, is a measure of the reflected power in decibels. It also describes how well the antenna is matched to the transmission line to which it is connected.

The rest of the paper is organized as follows. Section 2 covers the methodology used which also includes the radiation pattern equation of linear antenna with mutual coupling effects and the fitness function that has been used in this problem. Sections 3 and 4 detail the quantum particle swarm optimization and backtracking search optimization evolutionary algorithm principles. Section 5 depicts the simulated results followed by conclusion in Section 6 and references.

2. Methodology

The radiation pattern $F(\phi)$ in x - y plane of a linear array [1, 10] of parallel dipole antennas uniformly spaced at a distance d apart along the x -axis is given by

$$F(\phi) = AF(\phi) \times EP(\phi) = \left[\sum_{n=1}^N I_n e^{j(n-1)kd \cos \phi} \right] \times EP(\phi), \quad (1)$$

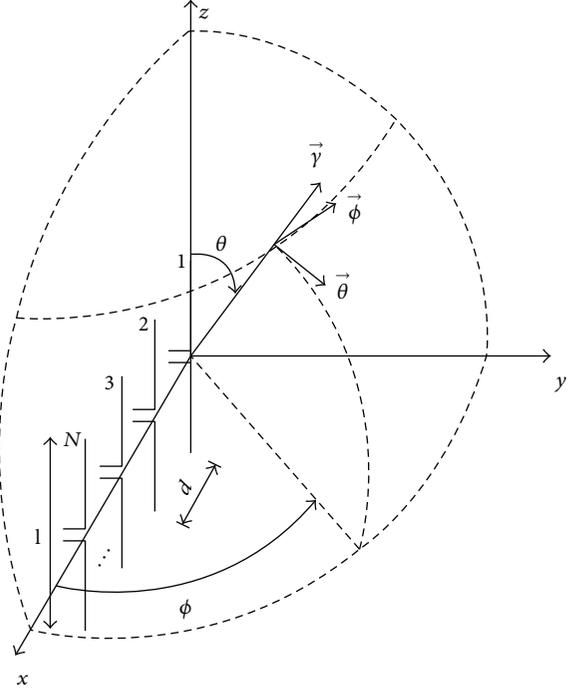


FIGURE 1: Linear array of parallel half wavelength dipole antenna along x -axis.

where AF is array factor, EP is element pattern of a vertical half wavelength dipole antenna, which is assumed to be omnidirectional, N is total number of elements, n is the element number, d is spacing between individual elements in the array, k is the wave number, λ is the wavelength of operation, ϕ is azimuth angle as shown in Figure 1, and I_n refers to the amplitude of current of n th dipole element. Substituting $u = \cos \phi$ in (1) will lead to far field pattern in u -domain.

Moreover, the current amplitudes are obtained by dividing the input voltage excitations of the array by the mutual impedance components stored in a mutual impedance matrix Z [10, 11]. The terms in this matrix Z , namely, self-impedances and mutual impedances of Z , are calculated considering the distributions of the currents on the dipole to be sinusoidal. The array pattern has been directly computed through a 4096-point IFFT [5, 8, 9] operation on the current excitation amplitudes. This reduces the time taken for completion of the operation of array factor drastically.

A measure of the voltage [10, 11] across the n th dipole can be given by

$$V_n = I_n Z_{nn} + \sum_{m \neq n} I_m Z_{nm}, \quad (2)$$

where Z_{nn} is the self-impedance of dipole n and Z_{nm} is the mutual impedance between dipoles n and m .

The active impedance [10, 11] of dipole n , Z_n^A , is given by

$$Z_n^A = \frac{V_n}{I_n} = Z_{nm} + \sum_{m \neq n} \left(\frac{I_m}{I_n} \right) Z_{nm}, \quad (3)$$

$$\text{VSWR}_{\max} = \frac{1 + |\Gamma|_{\max}}{1 - |\Gamma|_{\max}},$$

where Γ is the reflection coefficient across the n th dipole and is equal to $(Z_n^A - Z_o)/(Z_n^A + Z_o)$ and Z_o refers to the characteristic impedance and its value used in this paper is 50Ω . The smaller the value of VSWR, the better the matching condition of antenna. Or in other words, it can be concluded as a good match between the antenna and the feed line connected to it. Failure elements are not considered for calculation of VSWR as their value of active impedance is zero. And moreover, failure of a dipole element refers to the zero value of the voltage excitation at its input. However, current flowing through such faulty element is not zero because of mutual coupling. The faulty element acts as a parasitic radiator. The fitness function F has been written in order to obtain new excitation voltage amplitudes of the remaining unfailed elements (excluding the failed elements). The algorithms that are used here are QPSO and BSA ones

$$F = w_1 * F_1^2 + w_2 * \text{error}, \quad (4)$$

where

$$F_1 = \begin{cases} \text{SLL}_o - \text{SLL}_d, & \text{if } \text{SLL}_o > \text{SLL}_d \\ 0, & \text{if } \text{SLL}_o \leq \text{SLL}_d, \end{cases} \quad (5)$$

if $\text{VSWR}_{\max} \leq 1.4$, error = 0;
else error = $\text{abs}(\text{VSWR}_{\max} - 1.4)$.

The coefficients w_1 and w_2 refer to the weights given to their corresponding terms associated and are made equal to unity here. SLL_d and SLL_o are desired (-25 dB) and obtained values of SLL, respectively.

3. Quantum Particle Swarm Optimization

Various versions of the classical PSO [6] with Newtonian rules introduced by Kennedy and Eberhart in 1995 have proved its superiority over other algorithms like genetic algorithm in the past. The versions included the provisions of adding or tuning of parameters to guarantee the convergence, to improve convergence, and to increase in validity success of algorithms. This resulted in introduction of QPSO [13, 14] in 2004 using quantum mechanics principles which was found to be similar to PSO with less usage of control parameters and it paved its success over classical one by testing using numerous standard benchmark functions. The other main reason for its success is fast global convergence characteristics.

In classical PSO, convergence to a particular location is made possible by careful tuning of the social and cognitive

factors of the algorithm and in order to confine all the particles in the population within a space or boundary as required by the objective function, proper choice of maximum velocity in each dimension is required. And moreover, the state or condition of any particle in the population is defined by its velocity and position. An additional parameter is also introduced to elevate the convergence speed. Still, the global convergence is not guaranteed.

Unlike the above, in QPSO, the particle's state is defined by a wave function instead of position and velocity. There are no cognitive and social factors in the algorithm. This certainly reduced the total number of control parameters in this algorithm. The steps involved in this algorithm [14] are summarized as follows.

- (i) Based on dynamic range, initialize randomly the positions of all swarm in population.
- (ii) Evaluate the particles and the obtained personal best (pb) of all the particles are compared with their current fitness value obtained from the objective function. Their original values are replaced with the current one, if current value is better than the original one.
- (iii) The overall mean best (mb) position of all P particles is obtained using

$$\text{mb} = \frac{1}{P} \sum_{i=1}^P \text{pb}_i. \quad (6)$$

- (iv) The above procedure is adopted for whole population and current value is checked again for best fitness and designated as global best (gb), if it is found to be better than original fitness value.
- (v) The next step is to obtain the particle's vector local focus using (7) as

$$x_{iw}^k = \text{rand}1_{iw}^k * (\text{pb}_{iw}) + (1 - \text{rand}1_{iw}^k) * (\text{gb}). \quad (7)$$

- (vi) Update position of the w th dimension of i th particle using (8) and repeat steps (ii) to (vi) till global best is obtained, which is regarded as the final optimized solution. One has

$$X_{iw}^k = x_{iw}^k + (-1)^{\text{ceil}(0.5 + \text{rand}2_{iw}^k * \alpha * |\text{mb} - X_{iw}^{k-1}| * \log(1/\text{rand}3_{iw}^k))}. \quad (8)$$

If $X_{iw}^k < X_{mn}^k$, then

$$X_{iw}^k = X_{mn}^k + 0.25 * \text{rand}4_{iw}^k * (X_{mx}^k - X_{mn}^k). \quad (9)$$

If $X_{iw}^k > X_{mx}^k$,

$$X_{iw}^k = X_{mx}^k - 0.25 * \text{rand}5_{iw}^k * (X_{mx}^k - X_{mn}^k), \quad (10)$$

where $\text{rand}1$, $\text{rand}2$, $\text{rand}3$, $\text{rand}4$, and $\text{rand}5$ denote the uniform random numbers between 0 and 1, the desired minimum and maximum limits are X_{mn}^k and X_{mx}^k , and x refers to the particle's local attractor with $\alpha = 0.75$ being the coefficient dealing with the convergence. Equations (9) and (10) are used to maintain the position in a particular limit, thus avoiding particles getting exploded.

```

 $\Lambda = \text{zeros}(P, D);$ 
if rand < rand,
    for  $n = 1 : P, t = \text{randperm}(D); \Lambda(n, t(1 : \text{ceil}(\text{DIM\_RATE} * \text{rand} * D))) = 1; \text{end}$ 
else
    for  $n = 1 : P, \Lambda(n, \text{randi}(D)) = 1; \text{end}$ 
end

```

ALGORITHM 1

4. Backtracking Search Algorithm

Backtracking search algorithm (BSA), an evolutionary algorithm, has been proposed by Civicioglu [15, 16]. Backtracking search optimization algorithm (BSA) is a population-based iterative stochastic search evolutionary algorithm that is widely used to solve nonlinear, nondifferentiable, and complex numerical optimization problems.

Unlike many search algorithms, BSA has a single control parameter called *dim_rate*. Moreover, BSA's problem-solving performance is not oversensitive to the initial value of this parameter.

BSA uses three basic genetic operators such as selection, mutation, and crossover to generate trial individuals. BSA uses a nonuniform crossover strategy that is more complex than the crossover strategies used in many genetic algorithms. BSA has a random mutation strategy that uses only one direction individual for each target individual.

Step 1. The algorithm is run for N times. At the start of every run, randomly generate an initial population ($P \times D$) of P individuals with D dimensions of every individual within the variable constraint range. Also, generate another set of population of size ($P \times D$) within the variable constraint range, called historical population. Calculate the fitness value of initial population. The initial population (IP) and historical population (HP) are different in every other run. Consider

$$\text{IP}_{i,j} = U(\text{low}_j, \text{up}_j), \quad \text{HP} = U(\text{low}_j, \text{up}_j), \quad (11)$$

where $i = 1, 2, 3, \dots, P, \quad j = 1, 2, 3, \dots, D,$

where U is uniform random numbers between 0 and 1.

Step 2. Start iteration loop. The iteration number is fixed for every run. Provisions are also available for making historical population equal to initial population [15]. The order of individual in historical population (HP) is randomly changed through a random shuffling function.

Step 3. New offspring $T_{i,j}$ are generated from the combination of initial and historical population through mutation and crossover strategy as follows:

$$T_{i,j} = \text{IP} + (\Lambda * F) * (\text{HP} - \text{IP}), \quad (12)$$

where $i = 1, 2, 3, \dots, P, \quad j = 1, 2, 3, \dots, D,$

where Λ is binary integer-valued matrix of size ($P \times D$). It controls the number of elements of individuals that will

mutate in a trial. F is a scale factor which is equal to $3 * \text{rand}n$, with $\text{rand}n$ being standard normal distribution.

Two predefined strategies are randomly used to define BSA's Λ . The first strategy uses *dim_rate* parameter and the second strategy allows only one randomly chosen element of an individual to mutate in each trial. BSA's crossover process is more complex than the processes used in Differential evolution.

As shown in Algorithm 1, *randperm*(D) function randomly permutes all integer numbers between 1 and D .

Some offspring obtained at the end of BSA's crossover process and mutation strategy can be out of bound and therefore out of the allowed search space limits. The individuals beyond the search-space limits are restricted to lie within the upper and lower bound of every element of any individual. This is done in the following way:

$$T_{i,j} = \begin{cases} \text{rand} * (\text{up}_j - \text{low}_j) + \text{low}_j, & \text{if } \rightarrow T_{i,j} < \text{low}_j \\ \text{rand} * (\text{up}_j - \text{low}_j) + \text{low}_j, & \text{if } \rightarrow T_{i,j} > \text{up}_j. \end{cases} \quad (13)$$

Step 4. Calculate the fitness value of offspring.

Compare the fitness value of every offspring with the fitness value of initial population. If the fitness value of any offspring is better, then assign the fitness value of offspring to initial population and assign the current coordinates of offspring to initial population coordinates. Thus, a new set of population is generated. This new set of population becomes initial population in the next iteration. One has

$$\left. \begin{array}{l} \text{fitness IP}_i = \text{fitness } T_i \\ \text{IP}_i = T_i \end{array} \right\} \quad \text{if } \rightarrow \text{fitness } T_i < \text{fitness IP}_i. \quad (14)$$

Step 5. Determine the current best fitness value in the whole newly generated population and its coordinates. This will give global best value and its coordinates in the current iteration.

Step 6. Repeat Steps 2–5 until a stopping criterion, such as a maximum number of iterations being completed, is satisfied. The best scoring individual in the population is stored at the end of maximum number of iteration in every run. This will complete iteration loop in one run.

Step 7. Repeat Steps 1–6 until a stopping criterion, such as a sufficiently good solution being discovered or a maximum number of runs being completed, is satisfied. The best scoring individual among N gbest individuals (one gbest individual

TABLE 1: Voltage amplitude distribution for original and corrected patterns.

Element number	Original pattern	Corrected pattern		Element number	Original pattern	Corrected pattern	
		QPSO	BSA			QPSO	BSA
1	0.2766	0.0563	0.0700	16	0.8300	0.7611	0.6549
2	0.2456	0.1627	0.2717	17	0.6931	0.8722	0.9331
3	0.1485	0.0000	0.0000	18	0.7520	0.7207	0.6330
4	0.4343	0.1953	0.2654	19	0.9573	0.7747	0.9004
5	0.2666	0.2701	0.4432	20	0.8130	0.0000	0.0000
6	0.5048	0.0000	0.0000	21	0.7879	0.7581	0.7960
7	0.5010	0.4821	0.6664	22	0.3656	0.5205	0.5042
8	0.4498	0.3314	0.3976	23	0.8085	0.5717	0.6534
9	0.5969	0.5435	0.7057	24	0.3853	0.4186	0.4441
10	0.5295	0.3504	0.4351	25	0.4461	0.4129	0.4574
11	0.5183	0.7543	0.8206	26	0.4322	0.2631	0.3286
12	0.8032	0.6261	0.6572	27	0.3077	0.4083	0.3357
13	0.8796	0.6012	0.7785	28	0.2245	0.0000	0.0000
14	0.6779	0.8560	0.8408	29	0.3390	0.1570	0.1470
15	0.7857	0.6352	0.8312	30	0.2010	0.1507	0.1309

TABLE 2: Comparative results.

Design parameters	Case 1 (original pattern)	Case 2 (damaged pattern)	Case 3 (corrected pattern)	
			QPSO	BSA
SLL (dB)	-25.7022	-20.4538	-24.9824	-24.9797
VSWR	1.3970	1.6121	1.4806	1.5281
Mean fitness value of five runs			0.1127	0.1737
Standard deviation			0.0318	0.0283
Processing time in seconds			772	660
Global best fitness value among five runs			0.0808	0.1286

in every run) considering all the runs is taken as the final answer. The mean and standard deviation of all N gbest values is then calculated.

5. Results

In this paper, the authors used a linear antenna array of 30 half wavelength parallel dipoles placed along x -axis with uniform spacing of $\lambda/2$ between individual elements. The requirement is a broadside pattern in azimuthal plane with a SLL of -25 dB and VSWR of 1.4. The dipole element's radius used here is 0.005λ . Four elements, namely, 3, 6, 20, and 28, are chosen randomly to be failed elements ($V = 0$).

Three cases are discussed here.

Case 1. For generation of original pattern without any failures, QPSO is run for 100 iterations with a population of 60. A set of excitation voltage amplitudes is obtained.

Case 2. In this case, damaged pattern is obtained by considering the dipole elements 3, 6, 20, and 28 to be failed ones (randomly chosen). This is done by setting V (3, 6, 20, and 28) to be equal to zero in the voltage excitations obtained using Case 1.

Case 3. For obtaining the corrected pattern, QPSO is run 5 times, each for 1000 iterations. A set of new excitation voltage amplitudes is obtained for nondefective elements at the end of five runs.

The best fitness value among five runs is taken as the final answer and regarded as global best value.

The above three cases are repeated for BSA algorithm also.

MATLAB program is written and run for the above in a PC with Intel Core Duo CPU E8400 at 2.99 GHz and 1.94 GB of RAM. Table 1 shows the voltage excitations for original, damaged, and corrected patterns (using QPSO and BSA) and Table 2 summarizes comparative results of both algorithms. Figure 2 shows the fitness values versus number of iterations for both algorithms and Figures 3 and 4 show the radiation patterns obtained.

6. Conclusions

This paper presented a comparative analysis between QPSO and BSA algorithms in terms of the expected design parameters as well as statistical ones. The tables depicting the results show that even though both algorithms suited well for correction of the damaged pattern, the processing time taken for QPSO is slightly more than BSA. The expected

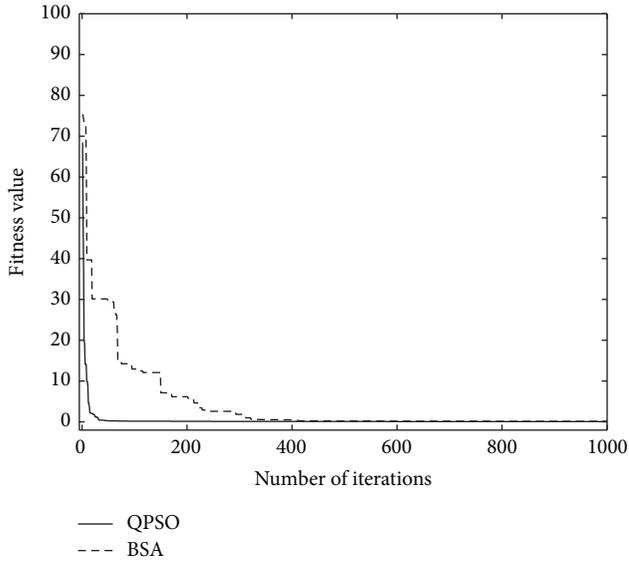


FIGURE 2: Fitness value versus number of iterations for the best run.

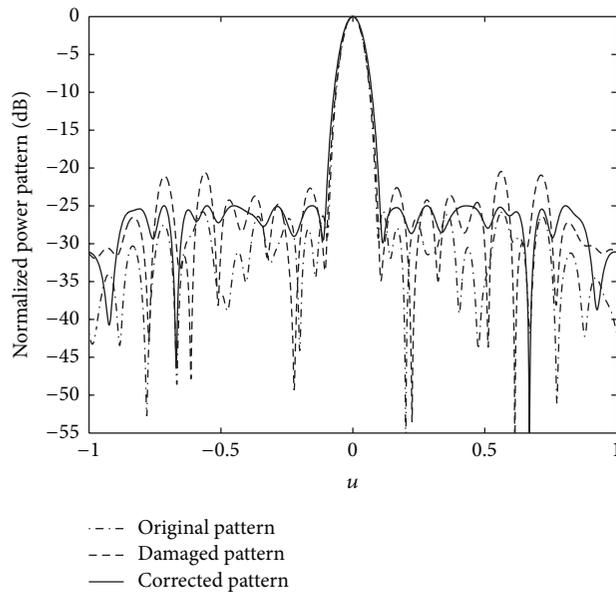


FIGURE 3: Normalized original, damaged, and corrected power patterns in dB using QPSO.

values of antenna parameters, namely, SLL and VSWR, match very closely each other, with QPSO slightly edging over BSA in providing better VSWR value. The same is the case with the statistical parameters where both algorithms match very closely each other, with QPSO showing slightly better results than BSA in global best fitness value and mean. The processing time is lower in BSA. This comparative analysis can be extended to other antenna array configurations also.

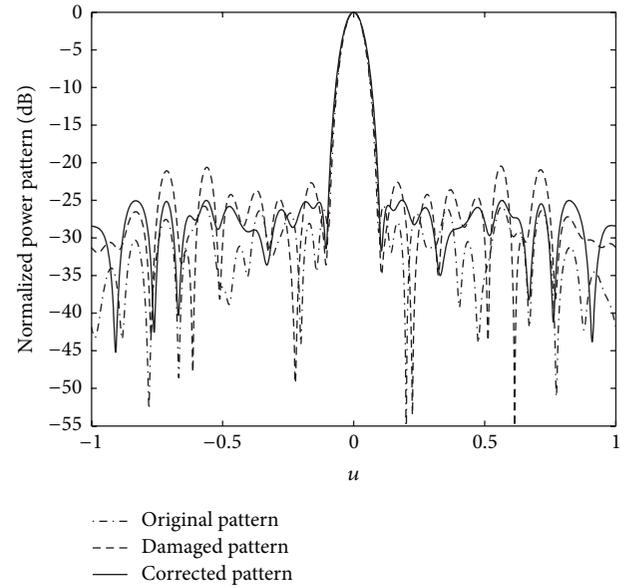


FIGURE 4: Normalized original, damaged, and corrected power patterns in dB using BSA.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

- [1] C. A. Balanis, *Antenna Theory: Analysis and Design*, John Wiley & Sons, New York, NY, USA, 2nd edition, 2003.
- [2] R. J. Mailloux, "Array failure correction with a digitally beam-formed array," *IEEE Transactions on Antennas and Propagation*, vol. 44, no. 12, pp. 1543–1550, 1996.
- [3] B.-K. Yeo and Y. Lu, "Array failure correction with a genetic algorithm," *IEEE Transactions on Antennas and Propagation*, vol. 47, no. 5, pp. 823–828, 1999.
- [4] J. A. Rodríguez, F. Ares, E. Moreno, and G. Franceschetti, "Genetic algorithm procedure for linear array failure correction," *Electronics Letters*, vol. 36, no. 3, pp. 196–198, 2000.
- [5] L. L. Wang, D. G. Fang, and W. X. Sheng, "Combination of genetic algorithm (GA) and fast Fourier transform (FFT) for synthesis of arrays," *Microwave and Optical Technology Letters*, vol. 37, no. 1, pp. 56–59, 2003.
- [6] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, December 1995.
- [7] D. W. Boeringer and D. H. Werner, "Particle swarm optimization versus genetic algorithms for phased array synthesis," *IEEE Transactions on Antennas and Propagation*, vol. 52, no. 3, pp. 771–779, 2004.
- [8] N. Pathak, B. Basu, and G. K. Mahanti, "Combination of inverse fast Fourier transform and modified particle swarm optimization for synthesis of thinned mutually coupled linear array of parallel half-wave length dipole antennas," *PIER: Progress In Electromagnetics Research M*, vol. 16, pp. 105–115, 2011.

- [9] W. P. M. N. Keizer, "Low-sidelobe pattern synthesis using iterative fourier techniques coded in MATLAB," *IEEE Antennas and Propagation Magazine*, vol. 51, no. 2, pp. 137–150, 2009.
- [10] R. S. Elliott, *Antenna Theory and Design*, Prentice-Hall, New York, NY, USA, 1981.
- [11] J. A. Rodriguez, F. Ares, and E. Moreno, "Feeding in-phase dipole arrays: a tutorial and a MATLAB program," *IEEE Antennas and Propagation Magazine*, vol. 47, no. 5, pp. 169–173, 2005.
- [12] M. Thevenot, C. Menudier, A. El Sayed Ahmad et al., "Synthesis of antenna arrays and parasitic antenna arrays with mutual couplings," *International Journal of Antennas and Propagation*, vol. 2012, Article ID 309728, 22 pages, 2012.
- [13] J. Sun, B. Feng, and W. Xu, "Particle swarm optimization with particles having quantum behavior," in *Proceedings of the Congress on Evolutionary Computation (CEC '04)*, pp. 325–331, Portland, Ore, USA, June 2004.
- [14] J. Sun, W. Xu, and B. Feng, "A global search strategy of quantum behaved Particle swarm optimization," in *Proceedings of the IEEE Conference on Cybernetics and Intelligent Systems*, pp. 111–116, Singapore, December 2004.
- [15] P. Civicioglu, "Backtracking search optimization algorithm for numerical optimization problems," *Applied Mathematics and Computation*, vol. 219, no. 15, pp. 8121–8144, 2013.
- [16] P. Civicioglu, "Circular antenna array design by using evolutionary search algorithms," *Progress in Electromagnetics Research B*, vol. 54, pp. 265–284, 2013.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

