



## Detecting faulty sensors in an array using symmetrical structure and cultural algorithm hybridized with differential evolution \*

Shafqat Ullah KHAN<sup>†1</sup>, Ijaz Mansoor QURESHI<sup>2</sup>, Fawad ZAMAN<sup>3</sup>, Wasim KHAN<sup>4</sup>

<sup>(1)</sup>School of Engineering & Applied Sciences, ISRA University, Islamabad 44000, Pakistan)

<sup>(2)</sup>Electrical Department, Air University, Islamabad 44000, Pakistan)

<sup>(3)</sup>Electrical Department, COMSAT Institute of Information Technology, Attock 44000, Pakistan)

<sup>(4)</sup>Electronic Engineering Department, International Islamic University, H-10, Islamabad 44000, Pakistan)

<sup>†</sup>E-mail: shafqatphy@yahoo.com

Received Sept. 26, 2015; Revision accepted Jan. 9, 2016; Crosschecked Jan. 3, 2017

**Abstract:** The detection of fully and partially defective sensors in a linear array composed of  $N$  sensors is addressed. First, the symmetrical structure of a linear array is proposed. Second, a hybrid technique based on the cultural algorithm with differential evolution is developed. The symmetrical structure has two advantages: (1) Instead of finding all damaged patterns, only  $(N-1)/2$  patterns are needed; (2) We are required to scan the region from  $0^\circ$  to  $90^\circ$  instead of from  $0^\circ$  to  $180^\circ$ . Obviously, the computational complexity can be reduced. Monte Carlo simulations were carried out to validate the performance of the proposed scheme, compared with existing methods in terms of computational time and mean square error.

**Key words:** Cultural algorithm; Differential evolution; Linear symmetrical sensor array

<http://dx.doi.org/10.1631/FITEE.1500315>

**CLC number:** TN929

### 1 Introduction

In the literature, there have been several techniques that address the issue of detecting defective sensors in an array antenna. Rodríguez-González *et al.* (2000; 2009) diagnosed the defective sensor using the genetic algorithm (GA), where the fitness function is used to compare the measured radiation pattern with the given configuration of failed/unfailed sensors. Patnaik *et al.* (2007) used a neural network (NN) approach to detect a maximum of three defective sensors in a small array composed of 16 sensors. Bucci *et al.* (2000) considered the ambiguity of the result in continuous and discrete on-off cases. Xu *et al.* (2007) used the support vector machine (SVM) to diagnose the defective sensors in a small array

composed of four sensors. However, this technique is not applicable to large arrays, where the possible number of combinations boosts. Moreover, the available techniques are computationally expensive, as they not only require to store the patterns of all defective sensors in the array, but also need to scan the entire region from  $0^\circ$  to  $180^\circ$ . Oliveri *et al.* (2009) presented a linear thinned array with predictable and well-behaved sidelobes, in which element placement is based on almost difference sets. The array power pattern is forced to pass through uniformly spaced values. Oliveri *et al.* (2010) further proposed an analytical technique based on almost difference sets for thinning planar arrays with well controlled sidelobes. Khan *et al.* (2015) used the compressed sensing technique hybridized with the genetic algorithm for the detection of faulty sensors, while Mailloux (1996), Yeo and Lu (1999), and Khan *et al.* (2013; 2014) developed different algorithms for failure correction.

\* Project supported by the Higher Education Commission of Pakistan

ORCID: Shafqat Ullah KHAN, <http://orcid.org/0000-0003-1969-1289>

© Zhejiang University and Springer-Verlag Berlin Heidelberg 2017

Today, biologically inspired techniques, especially differential evolution (DE) and the cultural algorithm (CA), are considered efficient and reliable optimization methods (Zaman *et al.*, 2012a; 2012b).

CA and DE are optimization techniques that include domain knowledge obtained during the evolutionary process. For many optimization problems, both CA and DE have successfully overcome the shortcomings of conventional optimization techniques due to their suppleness and effectiveness (Reynolds and Chung, 1996a; 1996b; Jin and Reynolds, 1999; Reynolds and Peng, 2005; Becerra and Coello, 2006). DE and CA are stochastic-based search algorithms, in which function parameters are programmed as floating-point variables. They are simple in structure, converge fast, and are robust against noise. Fonollosa *et al.* (2013) developed a more reliable electronic nose (e-nose) and a robust system, in which machine learning based on multiple kernels was generated to overcome sensor failures. The outcome confirms that multi-kernel models are more robust to sensor failures when the sub-kernel models are trained with small sets of sensors.

In this paper, the detection of fully and partially defective sensors in a linear array composed of  $N$  sensors is addressed. First, the symmetrical structure of a linear array is proposed. Second, a hybrid technique based on CA and DE is developed. In this hybrid process, the results achieved through CA are further tuned using DE. The mean squared error (MSE) is used as an objective evaluation function that defines the error between the responses of the desired and estimated patterns. The symmetrical structure has two advantages: (1) Instead of finding all damaged patterns, only  $(N-1)/2$  patterns are needed; (2) We are required to scan the region from  $0^\circ$  to  $90^\circ$  instead of from  $0^\circ$  to  $180^\circ$ . Obviously, the computational complexity can be reduced. The proposed method outperforms the conventional one proposed by Choudhury *et al.* (2013) in terms of computational time and MSE. Monte Carlo simulations are carried out to validate the performance of the proposed scheme, compared with the existing methods in terms of computational time and MSE.

## 2 Problem formulation

Consider a uniform linear array (ULA) composed of  $N=2M+1$  sensors along its  $x$ -axis with

respect to the original one. The far-field array factor (AF) for a healthy setup of equally spaced sensors of nonuniform amplitudes and progressive phase excitations can be given as (Wolff, 1937)

$$AF(\theta_i) = \sum_{n=-M}^M w_n \exp[jn(kd \cos \theta_i + \alpha)], \quad (1)$$

where  $w_n$  is the nonuniform weight of the  $n$ th sensor,  $d$  is the spacing between the adjacent sensors,  $\theta$  is the angle from broadside,  $k=2\pi/\lambda$  is the wave number with wavelength  $\lambda$ , and  $\alpha=-kdcos \theta_s$  is the progressive phase shift, where  $\theta_s$  is the steering angle for the main beam. For an unhealthy setup (Fig. 1), AF can be written as

$$AF(\theta_i) = \sum_{\substack{n=-M \\ n \neq m}}^M w_n \exp[jn(kd \cos \theta_i + \alpha)]. \quad (2)$$

If either  $w_m$  or  $w_{-m}$  is damaged, i.e., by putting  $w_m$  or  $w_{-m}$  equal to zero, the array factor of the  $m$ th damaged sensor in a noisy environment is given by

$$AF(\theta_i) = \sum_{\substack{n=-M \\ n \neq m}}^M w_n \exp[jn(kd \cos \theta_i + \alpha)] + \eta_i, \quad (3)$$

where  $\eta_i$  is the additive zero mean complex Gaussian noise with variance  $\sigma$  at the  $n$ th sensor.  $AF(\theta_i)$  is the pattern when  $w_m$  or  $w_{-m}$  is fully faulty (Fig. 1). Mathematically, the measurement noise (in dB) can be expressed as

$$SNR = \frac{\sum_{i=1}^K |AF(\theta_i)|^2}{\sum_{i=1}^K |AF(\eta_i)|^2}. \quad (4)$$

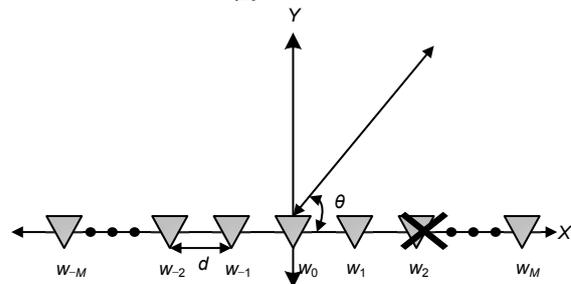
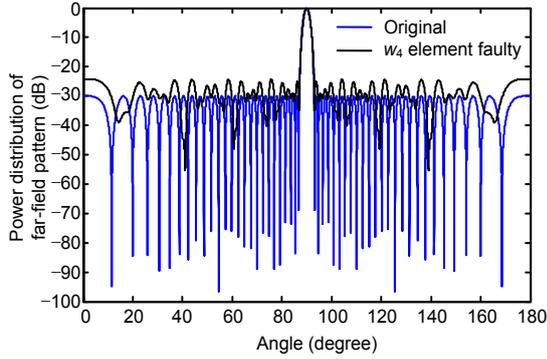


Fig. 1 Nonuniform amplitude array composed of  $2M+1$  sensors with sensor  $w_2$  defective

Assume that sensor  $w_4$  fails in the array. The method of locating a faulty element in a linear array starts with the measurement of several samples of the faulty pattern. The damaged array pattern for sensor  $w_4$  is shown in Fig. 2, where one can clearly observe that the pattern is symmetrical about  $\theta=90^\circ$ .



**Fig. 2 Original Chebyshev array and  $w_4$  sensor damage pattern**

### 3 Proposed methodology

In this section, we develop a method based on faulty patterns that are symmetric about  $\theta=90^\circ$ . Due to the symmetric structure, no matter  $w_m$  or  $w_{-m}$  is damaged, the patterns are the same. The failure of  $w_4$  or  $w_{-4}$  gives the same pattern (Figs. 2 and 3). To detect the faulty sensor, we tabulate only half the number of faulty patterns; i.e., only  $(N-1)/2$  faulty patterns are required. The other advantage is that we need only to scan the damage pattern from  $0^\circ$  to  $90^\circ$ , as the damage pattern is symmetrical about line  $\theta=90^\circ$ . The location of the faulty sensor can be found as

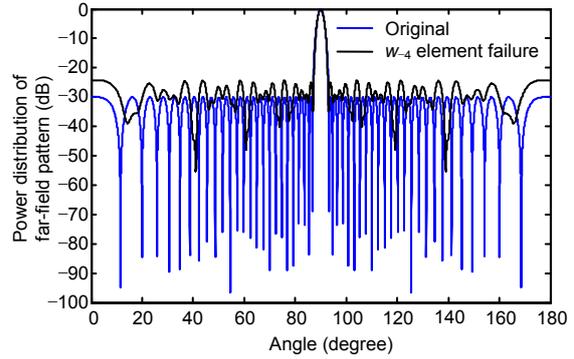
$$C_m = \sum_{i=1^\circ}^{90^\circ} |P_F(\theta_i) - P_m(\theta_i)|^2, \quad (5)$$

where  $P_F(\theta_i)$  is the faulty pattern and  $P_m(\theta_i)$  is the pattern when  $w_m$  or  $w_{-m}$  is fully faulty ( $1 \leq m \leq (N-1)/2$ ). In Eq. (5) the faulty patterns are compared with a given configuration of a fully faulty sensor, and its minimum result will give us the location of a faulty sensor. Then, based on another fitness function, we will decide whether the sensor is fully or partially defective. The value of the threshold has

been found on the basis of MSE. If the lowest error is not larger than  $E_{th}$  (which is set as 0.5), the weight  $w_m$  or  $w_{-m}$  is fully faulty. If the lowest error is larger than  $E_{th}$  ( $E_{th}=0.5$ ), then the weight is partially faulty. We use the cultural algorithm with differential evolution (CADE) technique to find the weights for a partially defective sensor. The fitness function is given by

$$G = \sum_{i=1^\circ}^{90^\circ} |P_F(\theta_i) - P_{CADE}(\theta_i)|^2, \quad (6)$$

where  $P_F(\theta_i)$  is the desired response, and  $P_{CADE}(\theta_i)$  is the value of the pattern obtained by using the CADE technique. The proposed method starts with tabulating both the faulty patterns  $\{F_1(\theta_i), F_2(\theta_i), \dots, F_{(N-1)/2}(\theta_i)\}$  and the single defective pattern  $P_m(\theta_i)$  evaluated in the range of  $0^\circ$  to  $90^\circ$ . Then  $C_m$  in Eq. (5) is calculated by finding the faulty sensor that minimizes  $C_m$  between one faulty pattern and one defective pattern when  $w_m$  or  $w_{-m}$  is fully faulty.



**Fig. 3 Original Chebyshev array and  $w_{-4}$  sensor failure pattern**

The proposed method is computationally efficient, as we require half the number of samples  $\{\theta_0^\circ, \theta_2^\circ, \dots, \theta_{90^\circ}\}$ ; also, we have to tabulate  $(N-1)/2$  faulty patterns to detect faulty sensors. The method of finding the faulty sensor in an array starts with measuring faulty patterns. If the defective sensor fails but radiates some power (i.e., the defective array pattern can be obtained from Eq. (2)), the defective weight is a fraction of the original one.

#### 3.1 Differential evolution

DE, developed by Storn and Price (1997), is used to solve real-valued optimization problems. DE is a stochastic-based search algorithm that has a

simple structure, fast convergence, and robustness against noise. It has shown good results for multi-model and non-differential fitness functions. DE is based on a mutation operator, which adds an amount obtained by the difference between two randomly chosen individuals of the present population. Thus, it has found tremendous applications (Rogalsky *et al.*, 2000; Das and Konar, 2006). The basic steps are given in the form of pseudo-codes as follows:

Step 1 (Initialization): First we randomly initialize  $Q$  chromosomes, each with a length of  $1 \times P$ . The  $P$  genes in each chromosome represent the weights of the array antenna, given as

$$S = \begin{pmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,p} \\ w_{2,1} & w_{2,2} & \dots & w_{2,p} \\ \vdots & \vdots & \dots & \vdots \\ w_{Q,1} & w_{Q,2} & \dots & w_{Q,p} \end{pmatrix},$$

$$w_{i,k} \in \mathbb{R}, l_b \leq w_{i,k} \leq u_b, \forall i = 1, 2, \dots, Q, k = 1, 2, \dots, P,$$

where  $l_b$  and  $u_b$  are the lower and upper bounds of  $w_{i,k}$ , respectively.

Step 2 (Update): All the chromosomes from 1 to  $Q$  of the current generation are updated. Choose  $d_h^{q,g_e}$  from the matrix, where  $g_e$  and  $h$  represent the particular generation and length of chromosome, respectively. Our main task is to find the chromosome of the next generation, i.e.,  $e^{q,g_{e+1}}$ , by using mutation, crossover, and selection operations.

Mutation: To perform the mutation process, we select randomly three different chromosomes from matrix  $S$ :

$$f^{q,g_e} = d^{c_1,g_e} + F(d^{c_2,g_e} - d^{c_3,g_e}), \quad (7)$$

$$0.5 \leq F \leq 1, 1 \leq c_1, c_2, c_3 \leq Q, c_1 \neq c_2 \neq c_3 \neq j.$$

Crossover: Crossover is performed using

$$r_k^{q,g_e} = \begin{cases} f_h^{q,g_e}, & \text{rand}() \leq CR \text{ or } h = h_{\text{rand}}, \\ d_h^{q,g_e}, & \text{otherwise,} \end{cases} \quad (8)$$

where  $0.5 \leq CR \leq 1$ , and  $h_{\text{rand}}$  is chosen randomly.

Selection: The next-generation chromosome is generated by

$$d^{q,g_{e+1}} = \begin{cases} r^{q,g_e}, & \text{error}(r^{q,g_e}) \leq \text{error}(d^{q,g_e})CR \text{ or } j = j_{\text{rand}}, \\ d^{q,g_e}, & \text{otherwise,} \end{cases}$$

where  $\text{error}(r^{q,g_e})$  and  $\text{error}(d^{q,g_e})$  are defined in matrix  $S$ .

Step 3 (Stopping criterion): The stopping criterion is based on the following condition:

If  $\text{error}(d^{q,g_{e+1}}) < \varepsilon$ , the maximum number of iterations is reached.

### 3.2 Cultural algorithm

CA was developed by Reynolds (1994) to model the evolution of the cultural component of an evolutionary computational system over time. The main idea behind CA is to clearly attain the problem-solving knowledge from the growing population, and apply this knowledge to guide the search space (Reynolds and Chung, 1996b). CA uses culture as a van for storing information available to the entire population over many generations. The parameter setting is given in Table 1. The flow diagram for CA is shown in Fig. 4. CA consists of three components: a populated space, a belief space, and a communication protocol. The first one contains the population to evolve and the mechanisms for its estimate. The population space consists of a set of possible solutions to the problem. In this study, the population space is DE. The second one is a belief space which represents the bias that has been acquired by the population during its problem-solving process. The belief space is the information depository in which the individuals can store their experiences for other individuals to learn ultimately. These two spaces are connected to each other through the communication protocol composed of two functions, i.e., acceptance and influence. The acceptance function is used to accept the experience of the best individuals from the population space, and store them in the belief space. Then the knowledge in the belief space can be updated through the update function. The influence function can guide the search space. In the present work, the belief space is divided into two knowledge components, i.e., situational knowledge and normative knowledge. CA is

**Table 1 Parameters used in the cultural algorithm with differential evolution**

Parameter	Setting
Population size	500
Number of generation	500
Value of $F$	0.5
Value of $CR$	$0.5 < CR < 1$

used as an optimization algorithm, whose belief space is stored and updated with the population space during each generation. The knowledge in the belief space is used to guide the search space toward the required solution.

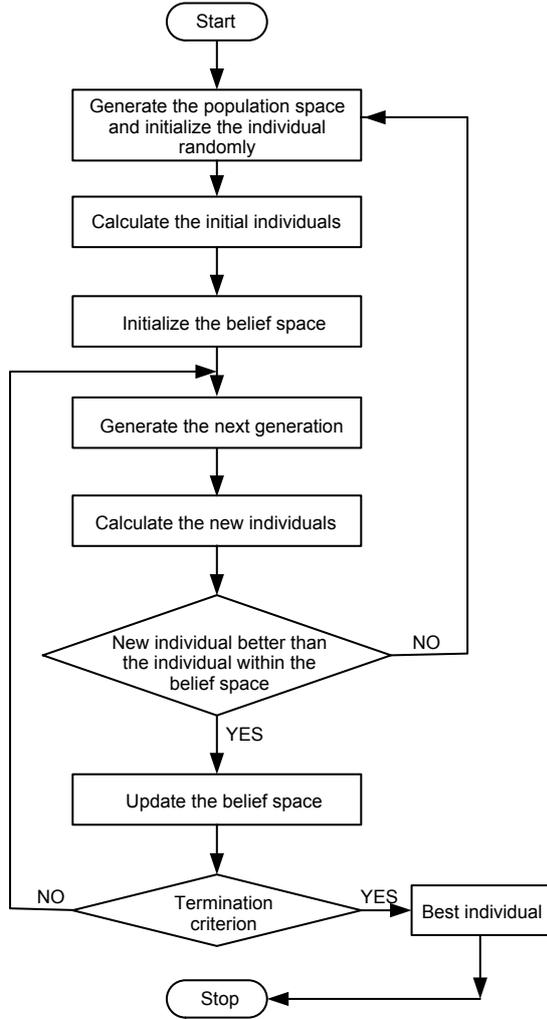


Fig. 4 Generic flow diagram of the cultural algorithm

The differential evolution variation operators are influenced in the following way:

$$y^{i,g} = Q_i + F(w^{n_2,g} - w^{n_3,g}), \quad (9)$$

where  $Q_i$  is the  $i$ th component of the individuals stored in the situational knowledge.

The normative knowledge includes a scaling factor  $ds_i$  to influence the mutation operator adopted in DE. The following expression shows the influence of the normative knowledge of the variation operators:

$$z_j^{i,g} = \begin{cases} w^{n_1,g} + F(w^{n_2,g} - w^{n_3,g}), & w^{n_1,g} < l_i, \\ w^{n_1,g} - F(w^{n_2,g} - w^{n_3,g}), & w^{n_1,g} > u_i, \\ w^{n_1,g} + \frac{u_i - l_i}{ds_i} \cdot F(w^{n_2,g} - w^{n_3,g}), & \text{otherwise,} \end{cases} \quad (10)$$

where  $l_i$  and  $u_i$  are the lower and upper bounds for the  $i$ th decision variable, respectively, and  $w^{n_j,g}$  represents the  $j$ th component of the  $i$ th individual selected from the  $g$ th generation by the acceptance function  $i$  ( $i=1, 2, \dots, n_{\text{accepted}}$ , where  $n_{\text{accepted}}$  is the number of best individuals at the  $g$ th generation).  $\{ds_i\}$  is updated with the difference  $(w^{n_2,g} - w^{n_3,g})$  of the variation operators of the prior generation. Normative knowledge leads the individuals to jump into the good range if they are not there. Normative knowledge is updated as follows. Consider  $\{x_{a_1}, x_{a_2}, \dots, x_{a_{n_{\text{accepted}}}}\}$  as the accepted individuals in the current generation, where  $\{a_1, a_2, \dots, a_{n_{\text{accepted}}}\}$  is the series of accepted individuals. Thus, we have

$$u_i = \begin{cases} w_{i,\max_i}, & w_{i,\max_i} > u_i \text{ or } f(w_{\max_i}) < U_i, \\ u_i, & \text{otherwise,} \end{cases} \quad (11)$$

$$l_i = \begin{cases} w_{i,\min_i}, & w_{i,\min_i} < l_i \text{ or } f(w_{\min_i}) < L_i, \\ l_i, & \text{otherwise,} \end{cases} \quad (12)$$

where  $w_{\min_i}$  and  $w_{\max_i}$  are the minimum and maximum values for parameter  $i$ , respectively. If  $l_i$  and  $u_i$  are updated, the values of  $L_i$  and  $U_i$  will be updated in the same way.  $\{ds_i\}$  is updated with the largest difference of  $|w_{i,r_1} - w_{i,r_2}|$  found during the variation operators at the previous generation.

#### 4 Simulation results and discussions

In this section, we discuss several cases based on different numbers of defective sensors in an array.

**Case 1** Consider a Chebyshev linear array composed of 51 sensors with a  $\lambda/2$  intersensor spacing as the test antenna. The array of sensors was placed symmetrically along the  $x$ -axis, and excited around the center of the array. An analytical technique was used to find the nonuniform weights for a  $-30$  dB constant sidelobe level (SLL) in the Chebyshev array.

To diagnose the faulty sensor in the linear symmetrical array, the radiation patterns for the fully and partially faulty sensors were generated. The samples were taken from the patterns in the region of  $0^\circ$  to  $90^\circ$ . Fifteen samples were taken from each pattern at an interval of  $6^\circ$  to scan the region of  $0^\circ$  to  $90^\circ$  to detect fully and partially faulty sensors. It is clear from Figs. 2 and 3 that the failure of either  $w_4$  or  $w_{-4}$  gave the same pattern, which is symmetrical about  $\theta=90^\circ$ . For the detection of the faulty sensor, we we have to tabulate both the faulty patterns  $\{F_1(\theta_i), F_2(\theta_i), \dots, F_{25}(\theta_i)\}$  and the single fault pattern  $P_m(\theta_i)$ , if either  $w_m$  or  $w_{-m}$  is damaged ( $m=0, 1, \dots, 25$ ). The cost function in Eq. (5) is minimized for a given sample  $\{\theta_{0^\circ}, \theta_{1^\circ}, \dots, \theta_{90^\circ}\}$ . The decision for the detection of a faulty sensor will be made based on the cost function. The minimization of cost functions in Eq. (5) will give us the location of a faulty sensor. If  $C_m \leq E_{th}=0.5$ , the weight  $w_m$  is fully faulty. If  $C_m > E_{th}=0.5$ , then the weight is partially faulty, and for the partial fault we will use the CADE technique (Eq. (6)) to find the weights. Simulation results for full and partial faults have been checked, showing the validity of the proposed method.

Fig. 5 shows the behavior of the diagnostic errors for different values of the signal-to-noise ratio (SNR). We examined different values of SNR versus MSE. As one can see from Fig. 5, MSE decreases as the value of SNR increases.

**Case 2** Consider that a sensor fails partially and radiates some power; i.e., its weight is not zero, but a fraction of the original one. First we consider that sensor  $w_2$  is 50% damaged and this damage pattern is created by making the weight of the sensor half of its original weight in the original Chebyshev array. Now

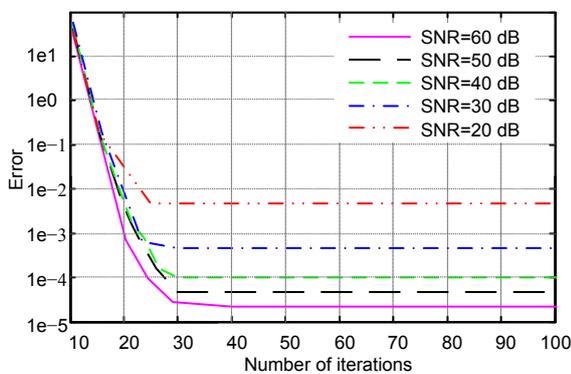


Fig. 5 Mean-squared error versus the number of iterations

the CADE technique is used to locate its position. This array factor was obtained by making the weight of sensor  $w_2$  equal to half of the original weight in Eq. (1), represented by  $P_f(\theta_i)$  in the fitness function of Eq. (6). Then the CADE technique is used to minimize the fitness function, which in turn gives the weight of the defected sensor. MSE is used as a fitness function, given by Eq. (6). To check the performance of the CADE technique, these weights are compared to those obtained by the Chebyshev method in which the weight of the sensor equals half of the original one. The weight obtained by CADE is given in Table 2, and the pattern recovered using the CADE technique is shown in Fig. 6.

Now we suppose that the sensor is 50% faulty by making the weight of  $w_5$  half of the original value. The CADE technique was used to locate its position. The array factor was obtained by making the weight of sensor  $w_5$  equal to half of its original value in Eq. (1). Then the CADE technique was used to minimize the fitness function in Eq. (6), which in turn gave the weight of the faulty array. The weights obtained by using the CADE technique for partial failure are given in Table 2. To check the validity, the weights obtained by the CADE technique were compared to the weights obtained from the Chebyshev method of damage patterns for 50% fault. The comparison of the weights of the defective array with those obtained by the CADE technique is given in Table 2. The original weight distributions, partially faulty weight distributions, and the weight distributions obtained by CADE are depicted in Fig. 7. From this comparison, the partial fault can be clearly identified. Comparison of

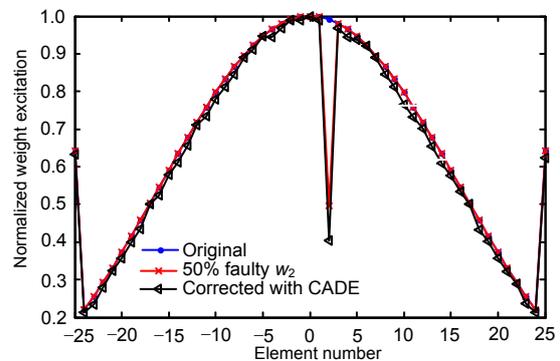
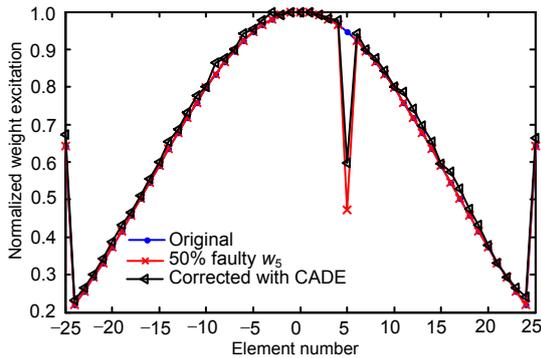


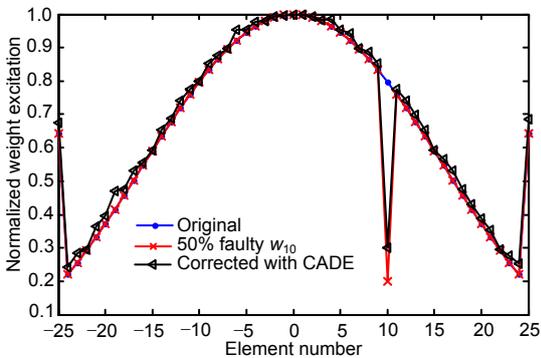
Fig. 6 Weight distributions of the original,  $w_2$  defected, and that obtained by the cultural algorithm with differential evolution (CADE: cultural algorithm with differential evolution)

the weights obtained by CADE with those of the defected array shows the level of the partial fault. The weights for the original Chebyshev array and the weights obtained for different cases by the CADE technique are given in Table 2. Now we assume that sensor  $w_{10}$  is 25% failed. The CADE algorithm is run to locate the partial fault of the sensor. The weights obtained by CADE for detection of partial faults (25%) for the sensor are given in Table 2 and shown in Fig. 8.

**Case 3** Consider a linear Chebyshev array composed of 24 sensors taken as the reference antenna to execute the method of fault diagnosis developed by Choudhury *et al.* (2013). The proposed method was compared with the conventional method. The fully and partially faulty patterns were generated by making their weights either equal to zero or some fractions of the original weights. Assume that the 4th, 10th (50%), and 17th (100%) sensors in the array



**Fig. 7** Weight distributions of the original,  $w_5$  defected, and that obtained by the cultural algorithm with differential evolution (CADE: cultural algorithm with differential evolution)



**Fig. 8.** Weight distributions of the original,  $w_{10}$  defected, and that corrected by the cultural algorithm with differential evolution (CADE: cultural algorithm with differential evolution)

**Table 2** Chebyshev and normalized weights obtained from the cultural algorithm with those of the differential evolution algorithm

Sensor	Chebyshev weight	Nonuniform weight		
		Sensor $w_2$ 50% failed	Sensor $w_5$ 50% failed	Sensor $w_{10}$ 25% failed
$w_{-25}$	0.6426	0.6475	0.6437	0.6453
$w_{-24}$	0.2200	0.2234	0.2252	0.2317
$w_{-23}$	0.2554	0.2593	0.2584	0.2643
$w_{-22}$	0.2928	0.2959	0.2969	0.2975
$w_{-21}$	0.3321	0.3402	0.3451	0.3429
$w_{-20}$	0.3730	0.3754	0.3758	0.3813
$w_{-19}$	0.4152	0.4197	0.4247	0.4341
$w_{-18}$	0.4584	0.4610	0.4635	0.4618
$w_{-17}$	0.5023	0.5076	0.5085	0.5163
$w_{-16}$	0.5465	0.5485	0.5496	0.5538
$w_{-15}$	0.5907	0.5934	0.5952	0.5916
$w_{-14}$	0.6344	0.6431	0.6417	0.6384
$w_{-13}$	0.6773	0.6796	0.6815	0.6794
$w_{-12}$	0.7190	0.7203	0.7351	0.7265
$w_{-11}$	0.7591	0.7612	0.7652	0.7631
$w_{-10}$	0.7973	0.7995	0.7987	<b>0.1987</b>
$w_{-9}$	0.8330	0.8413	0.8418	0.8491
$w_{-8}$	0.8661	0.8676	0.8673	0.8752
$w_{-7}$	0.8961	0.8987	0.8984	0.8987
$w_{-6}$	0.9228	0.9264	0.9378	0.9376
$w_{-5}$	0.9459	0.9476	<b>0.4857</b>	0.9518
$w_{-4}$	0.9651	0.9685	0.9721	0.9687
$w_{-3}$	0.9802	0.9843	0.9873	0.9885
$w_{-2}$	0.9912	<b>0.4879</b>	0.9992	0.9932
$w_{-1}$	0.9978	0.9986	0.9983	0.9986
$w_0$	1.0000	0.9994	0.9987	0.9975
$w_1$	0.9978	0.9995	0.9993	0.9996
$w_2$	0.9912	0.9945	0.9928	0.9934
$w_3$	0.9802	0.9835	0.9834	0.9827
$w_4$	0.9651	0.9676	0.9676	0.9675
$w_5$	0.9459	0.9513	0.9473	0.9486
$w_6$	0.9228	0.9301	0.9362	0.9264
$w_7$	0.8961	0.8978	0.8988	0.8975
$w_8$	0.8661	0.8751	0.8676	0.8676
$w_9$	0.8330	0.8421	0.8414	0.8451
$w_{10}$	0.7973	0.7985	0.7993	0.7998
$w_{11}$	0.7591	0.7584	0.7632	0.7685
$w_{12}$	0.7190	0.7542	0.7531	0.7326
$w_{13}$	0.6773	0.6833	0.6795	0.6891
$w_{14}$	0.6344	0.6461	0.6384	0.6454
$w_{15}$	0.5907	0.5932	0.5951	0.5937
$w_{16}$	0.5465	0.5496	0.5485	0.5579
$w_{17}$	0.5023	0.5159	0.5046	0.5163
$w_{18}$	0.4584	0.4613	0.4597	0.4627
$w_{19}$	0.4152	0.4271	0.4174	0.4256
$w_{20}$	0.3730	0.3742	0.3819	0.3828
$w_{21}$	0.3321	0.3364	0.3357	0.3482
$w_{22}$	0.2928	0.2953	0.2961	0.2952
$w_{23}$	0.2554	0.2571	0.2589	0.2617
$w_{24}$	0.2200	0.2262	0.2317	0.2356
$w_{25}$	0.6426	0.6432	0.6478	0.6513

Bold numbers represent the faulty sensors

have become faulty. The faulty pattern and symmetrical counterpart failure of the 4th, 10th (50%), and 17th (100%) sensors are shown in Fig. 9a and 9b, respectively. It is clear from Figs. 9a and 9b that the failures give the same power pattern. Therefore, we have to tabulate half of the faulty patterns. The second advantage of using a symmetrical linear array is that the faulty pattern is symmetrical about  $\theta=90^\circ$ ; i.e., we need half the number of samples to scan the pattern. First we simulated the pattern with 1–3 faulty elements. We used a set of 1162 patterns. In each case, the faulty pattern contains  $M$  samples as the input to check the diagnosis of fault. In this case, we take 18 samples, 35 samples, or a random number of samples to validate the performance of the proposed method. Assume there are a maximum of three defective sensors, which yields a total of  $\sum_{f=1}^3 \frac{N!}{f!(N-f)!} = 2324$  patterns by the conventional method, while 1162 patterns by our proposed method. To locate the faulty sensors in an array of antennas, the weight of each sensor was considered as the optimizing parameter for the bacteria foraging optimization (BFO) and CADE algorithms. CADE would converge to the optimum solution. In an array diagnosis, 35 samples were taken in the range of  $0^\circ$  to  $180^\circ$  at an interval of  $5^\circ$ . We supposed that the 4th and 10th sensors were partially faulty and that the 17th sensor was fully faulty. We ran BFO and the proposed method to diagnose the faulty sensors. Fig. 10a shows the faulty pattern with the positions of 35 samples. The fault diagnosed by the conventional method is shown in Fig. 10b. Then the same process was repeated using the proposed method (Figs. 11a and 11b).

By the proposed method, we diagnosed the fault by half the number of samples. Similarly, the fault was repeated for 18 samples and some other random number of samples for the conventional and proposed methods, respectively. The results obtained by the conventional and proposed methods for 18 samples and some random number of samples are shown in Figs. 12–13. By the proposed method, one can detect the faulty sensor accurately even with fewer sample points. Fig. 14 shows the MSE plots for conventional and proposed methods.

For a few initial iterations, the value of MSE was high, but after some iterations it went down. BFO and

the proposed method were run to detect the location of faults for six random cases and find the average time of the faulty sensors for various scenarios. The results are given in Tables 3–6. From the simulation results, it is clear that the computation time increases as the number of faulty sensors increases.

## 5 Conclusions

We proposed a computationally efficient technique to find fully and partially defective sensors in a linear array. Using the approach of a symmetrical

**Table 3 Time comparison for six configurations of one defective sensor**

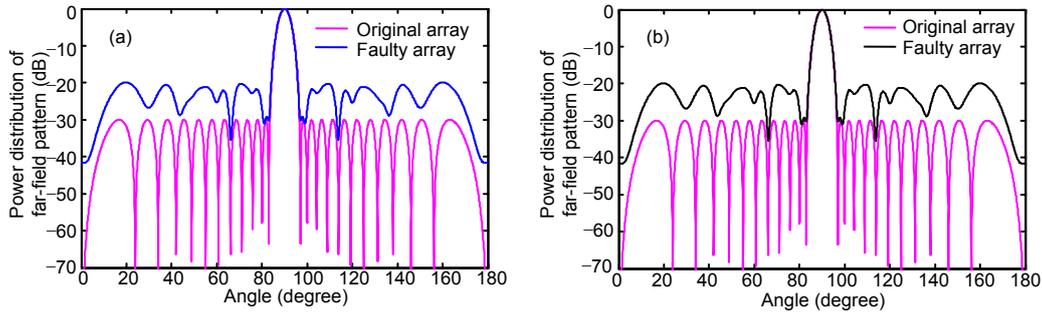
Fault location	Detection time (s)			
	Proposed method		Conventional method	
	18 samples	35 samples	18 samples	35 samples
1	56.72	117.46	108.42	201.71
2	58.53	121.69	113.73	206.50
5	59.65	115.78	114.56	213.31
10	60.87	109.95	117.35	205.37
20	54.47	108.60	107.50	203.95
23	52.89	107.95	105.45	204.87
Average	57.18	113.57	111.17	205.95

**Table 4 Time comparison for six configurations of two defective sensors**

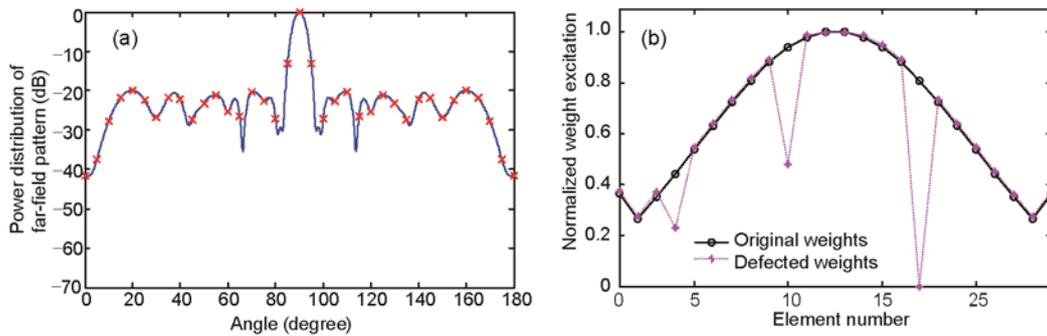
Fault location	Detection time (s)			
	Proposed method		Conventional method	
	18 samples	35 samples	18 samples	35 samples
5, 7	67.70	139.70	123.50	254.30
20, 23	66.40	141.40	121.70	237.40
18, 20	72.30	153.60	137.10	238.75
11, 21	74.80	155.10	139.30	241.50
7, 10	76.20	161.90	143.60	231.94
8, 12	78.90	165.20	144.10	235.50
Average	72.72	152.82	134.88	239.90

**Table 5 Time comparison for six random configurations of three defective sensors**

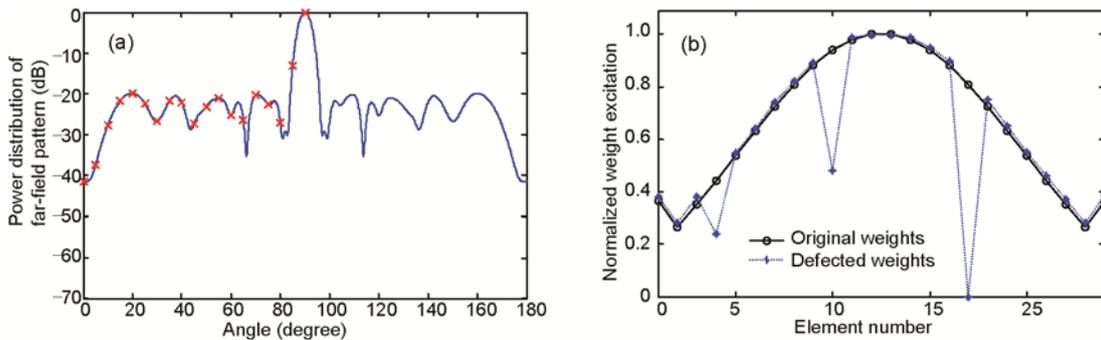
Fault location	Detection time (s)			
	Proposed method		Conventional method	
	18 samples	35 samples	18 samples	35 samples
5, 7, 12	81.70	169.40	144.10	231.40
1, 7, 12	85.30	173.70	157.30	274.10
1, 9, 11	79.30	167.30	142.90	230.90
13, 17, 24	82.60	169.90	153.50	227.20
15, 19, 22	85.50	175.50	151.30	237.60
14, 21, 24	77.30	164.80	145.60	241.70
Average	81.95	170.10	149.12	240.48



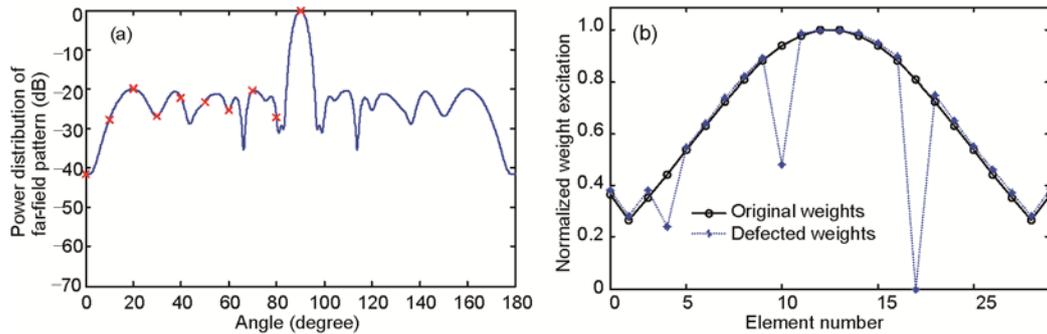
**Fig. 9** Patterns of the Chebyshev array and the 4th, 10th (50%), and 17th (100%) sensors faulty array (a) and Chebyshev array and the symmetrical counterpart of the 4th, 10th (50%), and 17th (100%) sensors faulty array (b)



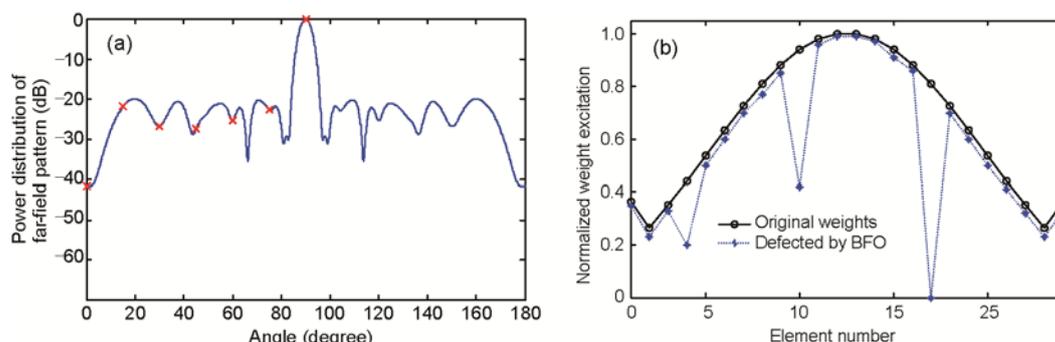
**Fig. 10** Defective array pattern with fault at the 4th, 10th (50%), and 17th (100%) sensors with 35 sample points (a) and its fault diagnosed by the conventional method (Choudhury *et al.*, 2013) (b)



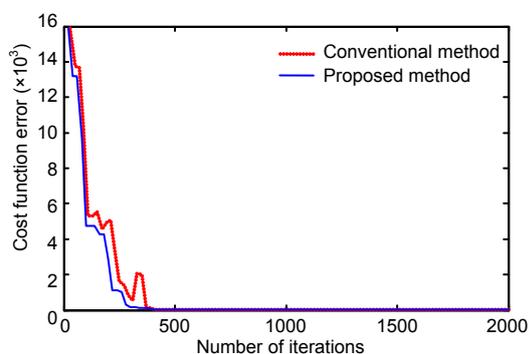
**Fig. 11** Defective array pattern with fault at the 4th, 10th (50%), and 17th (100%) sensors with 19 sample points (a) and its fault diagnosed by the proposed method (b)



**Fig. 12** Defective array pattern with fault at the 4th, 10th (50%), and 17th (100%) sensors with 10 sample points (a) and its fault diagnosed by the proposed method (b)



**Fig. 13** Defective array pattern with fault at the 4th, 10th (50%), and 17th (100%) sensors with a random number of sample points (a) and its fault diagnosed by the proposed method (b) (BFO: bacteria foraging optimization)



**Fig. 14** Mean square error performance of the conventional (Choudhury et al., 2013) and the proposed methods

**Table 6** Time comparison for six random configurations of fully and partially faulty sensors

Fault locations	Detection time (s)			
	Proposed method		Conventional method	
	18 samples	35 samples	18 samples	35 samples
8, 12 (100%), 2 (50%)	85.70	182.60	141.50	241.20
13, 18 (50%), 24 (100%)	84.40	171.60	152.90	251.70
14, 20 (100%), 15 (50%)	82.80	175.80	147.70	249.10
1, 8 (100%), 2 (50%)	80.90	179.70	143.80	252.40
15, 23 (100%), 24 (50%)	86.70	181.40	151.30	246.80
4, 10 (50%), 17 (100%)	85.10	176.30	152.30	258.90
Average	84.27	177.90	148.25	250.02

linear array brings two advantages. First, the failure of  $w_m$  or  $w_{-m}$  gives the same pattern; i.e., we require  $(N-1)/2$  patterns instead of finding all damaged patterns. Second, we need to scan half of the damage patterns  $\{\theta_{0^\circ}, \theta_{1^\circ}, \dots, \theta_{90^\circ}\}$ , as the patterns are symmetrical about  $\theta=90^\circ$ . The decision of the fully or

partially faulty sensor is made based on the cost function. If  $C_m > 0.5$ , the sensor is fully faulty; if  $C_m \leq 0.5$ , the sensor is partially faulty. For partial faults we used the CADE technique to locate the defective sensors. This method can be extended to planar arrays and L-type arrays.

## References

- Becerra, R.L., Coello, C.A.C., 2006. Cultured differential evolution for constrained optimization. *Comput. Method Appl. Mech. Eng.*, **195**(33-36):4303-4322. <http://dx.doi.org/10.1016/j.cma.2005.09.006>
- Bucci, O.M., Capozzoli, A., de Elia, G., 2000. Diagnosis of array faults from far-field amplitude-only data. *IEEE Trans. Antennas Propag.*, **48**(5):647-652. <http://dx.doi.org/10.1109/8.855482>
- Choudhury, B., Acharya, O.P., Patnaik, A., 2013. Bacteria foraging optimization in antenna engineering: an application to array fault finding. *Int. J. RF Microw. Comput.-Aid. Eng.*, **23**(2):141-148. <http://dx.doi.org/10.1002/mmce.20659>
- Das, S., Konar, A., 2006. Two-dimensional IIR filter design with modern search heuristics: a comparative study. *Int. J. Comput. Intell. Appl.*, **6**(3):329-355. <http://dx.doi.org/10.1142/S1469026806001848>
- Fonollosa, J., Vergara, A., Huerta, R., 2013. Algorithmic mitigation of sensor failure: is sensor replacement really necessary? *Sens. Actuat. B.*, **183**:211-221. <http://dx.doi.org/10.1016/j.snb.2013.03.034>
- Jin, X.D., Reynolds, R.G., 1999. Using knowledge-based evolutionary computation to solve nonlinear constraint optimization problems: a cultural algorithm approach. *Proc. Congress on Evolutionary Computation*, p.1672-1678. <http://dx.doi.org/10.1109/CEC.1999.785475>
- Khan, S.U., Qureshi, I.M., Zaman, F., et al., 2013. Null placement and sidelobe suppression in failed array using symmetrical element failure technique and hybrid heuristic computation. *Prog. Electromagn. Res. B*, **52**:165-184. <http://dx.doi.org/10.2528/PIERB13032712>

- Khan, S.U., Qureshi, I.M., Zaman, F., et al., 2014. Correction of faulty sensors in phased array radars using symmetrical sensor failure technique and cultural algorithm with differential evolution. *Sci. World J.*, **2014**:852539. <http://dx.doi.org/10.1155/2014/852539>
- Khan, S.U., Qureshi, I.M., Naveed, A., et al., 2015. Detection of defective sensors in phased array using compressed sensing and hybrid genetic algorithm. *J. Sens.*, **501**: 718914. <http://dx.doi.org/10.1155/2016/6139802>
- Mailloux, R.J., 1996. Array failure correction with a digitally beamformed array. *IEEE Trans. Antennas Propag.*, **44**(12):1543-1550. <http://dx.doi.org/10.1109/8.546240>
- Oliveri, G., Donelli, M., Massa, A., 2009. Linear array thinning exploiting almost difference sets. *IEEE Trans. Antennas Propag.*, **57**(12):3800-3812. <http://dx.doi.org/10.1109/TAP.2009.2027243>
- Oliveri, G., Manica, L., Massa, A., 2010. ADS-based guidelines for thinned planar arrays. *IEEE Trans. Antennas Propag.*, **58**(6):1935-1948. <http://dx.doi.org/10.1109/TAP.2010.2046858>
- Patnaik, A., Choudhury, B., Pradhan, P., et al., 2007. An ANN application for fault finding in antenna arrays. *IEEE Trans. Antennas Propag.*, **55**(3):775-777. <http://dx.doi.org/10.1109/TAP.2007.891557>
- Reynolds, R.G., 1994. An introduction to cultural algorithms. Proc. 3rd Annual Conf. on Evolutionary Programming, p.131-139.
- Reynolds, R.G., Chung, C.J., 1996a. A self-adaptive approach to representation shifts in cultural algorithms. Proc. IEEE Int. Conf. on Evolutionary Computation, p.94-99. <http://dx.doi.org/10.1109/ICEC.1996.542340>
- Reynolds, R.G., Chung, C.J., 1996b. The use of cultural algorithms to evolve multiagent cooperation. Proc. Micro-Robot World Cup Soccer Tournament, p.53-56.
- Reynolds, R.G., Peng, B., 2005. Knowledge learning and social swarms in cultural systems. *J. Math. Sociol.*, **29**(2): 115-132. <http://dx.doi.org/10.1080/00222500590920851>
- Rodríguez-González, J.A., Ares-Pena, F., Palacios, H., et al., 2000. Finding defective elements in planar arrays using genetic algorithms. *Prog. Electromagn. Res.*, **29**:25-37. <http://dx.doi.org/10.2528/PIER00011401>
- Rodríguez-González, J.A., Ares-Pena, F., Fernández-Delgado, M., et al., 2009. Rapid method for finding faulty elements in antenna arrays using far field pattern samples. *IEEE Trans. Antennas Propag.*, **57**(6):1679-1683. <http://dx.doi.org/10.1109/TAP.2009.2019915>
- Rogalsky, T., Kocabiyik, S., Derksen, R.W., 2000. Differential evolution in aerodynamic optimization. *Can. Aeronaut. Space J.*, **46**(4):183-190.
- Storn, R., Price, K., 1997. Differential evolution: a simple and efficient adaptive scheme for global optimization over continuous spaces. *J. Glob. Optim.*, **11**(4):341-359. <http://dx.doi.org/10.1023/A:1008202821328>
- Wolff, I., 1937. Determination of the radiating system which will produce a specified directional characteristic. *Proc. Inst. Radio Eng.*, **25**(5):630-643. <http://dx.doi.org/10.1109/JRPROC.1937.228158>
- Xu, N., Christodoulou, C.G., Barbin, S.E., et al., 2007. Detecting failure of antenna array elements using machine learning optimization. IEEE Antennas and Propagation Society Int. Symp., p.5753-5756. <http://dx.doi.org/10.1109/APS.2007.4396858>
- Yeo, B.K., Lu, Y.L., 1999. Array failure correction with a genetic algorithm. *IEEE Trans. Antennas Propag.*, **47**(5):823-828. <http://dx.doi.org/10.1109/8.774136>
- Zaman, F., Qureshi, I.M., Naveed, A., et al., 2012a. Amplitude and directional of arrival estimation: comparison between different techniques. *Prog. Electromagn. Res. B*, **39**:319-335. <http://dx.doi.org/10.2528/PIERB12022109>
- Zaman, F., Qureshi, I.M., Naveed, A., et al., 2012b. Joint estimation of amplitude, direction of arrival and range of near field sources using memetic computing. *Prog. Electromagn. Res. C*, **31**:199-213. <http://dx.doi.org/10.2528/PIERC12052811>