International Journal of Electrical and Computer Engineering (IJECE)
Vol. 2, No. 1, February 2012, pp. 98~105
ISSN: 2088-8708

Comparative Performance Investigations of Stochastic and Genetic Algorithms Under Fast Dynamically Changing Environment in Smart Antennas

Jafar Ramadhan Mohammed
Departement of Communication Engineering, College of Electronic Engineering, University of Mosul

Article Info

Article history:

Received Oct 19th, 2011
Revised Nov 29th, 2011
Accepted Jan 4th, 2012

Keyword:

Smart Antennas
Adaptive Algorithms
Genetic Algorithm
Optimization;
Tracking;
Dynamic Systems

ABSTRACT

In a mobile communication systems, the number of observation data (snapshots) used for covariance matrix estimation can be insufficient, which often occurs due to fast dynamically changing environment or signal characteristics are rapidly changing. In these situations, the performances of the standard adaptive algorithms such as LMS are known to degrade substantially. In this paper, we propose the use of a Genetic Algorithm (GA) to perform the adaptation control of the system parameters under dynamically changing environments The GA-based beamformer has nearly optimal interference cancellation under dynamic conditions, and makes the output SINR consistently close to the optimal one regardless of the number of snapshot used. Other advantages of the GA are its simplicity and fast convergence provided that the parameters are appropriately chosen, which makes it a practical algorithm for beamforming in smart antenna. Simulation results validate substantial performance improvements relative to other standard adaptive algorithms. Although, the use of GA is not new in smart antenna technology, the performance evaluation of the genetic optimization under fast dynamically changing environment has not been investigated to the best of my knowledge and it is of great practical significance.

Corresponding Author:

Jafar Ramadhan Mohammed,
Departement of Communication Engineering, College of Electronic Engineering,
University of Mosul,
Mosul, IRAQ.
Email: jafarram@yahoo.com

1. INTRODUCTION

There has been an explosive growth in the wireless communication area in recent years. This growth has forced system designers to increase the quality of service, coverage, and bandwidth. Smart antennas are an emerging technology that can be used to tackle the capacity, quality, and coverage problems faced in wireless communication under heavy traffic. Such antennas use a weighted average of the received signals to automatically adjust the beam towards the signal of interest (SOI) to radiate or receive desired signals while nulling the interferers. If the environment is changing dynamically, the complex weights need to be adjusted in order to track the changes. In the stationary case the weights are found and the beam is fixed, but in adaptive systems these weights must be updated every time new information comes in. The adaptive algorithm chosen is very important since the convergence speed, stability, and complexity are important issues in an adaptive system design. The algorithm must satisfy some chosen criteria for the optimization process. Most commonly used techniques are least mean squares (LMS), recursive least squares (RLS) [1], direct matrix inversion (DMI) [2], neural networks [3], conjugate gradients [4], and constant modulus algorithm (CMA) [5]. The performance of the RLS and SMI schemes are not dependent of the eigenvalue spread of covariance matrix, since the covariance matrix is inverted directly. On the other hand, the LMS algorithm suffers from slow convergence in the case of large eigenvalue spread of the sample covariance matrix. However, these classical adaptive processing techniques employ large number of snapshots (observation data) to carry out digital beamforming and thereby make their applications in real life prohibitive as they are computationally too expensive and are unsuitable for a fast dynamically changing environment as they require a latent time to collect the snapshots of data [6]. A snapshot is defined as the set of voltages measured at the terminals of the antennas. On the other hand, when the number of snapshots used for covariance matrix estimation is insufficient. In this situation, the performances of the conventional adaptive algorithms are known to degrade substantially [7, 8]. This undesired behavior results in a reduction of the array output signal-to-interference-plus-noise ratio (SINR).

To mitigate these practical shortcomings, recently a genetic algorithm GA was developed for solving the robust adaptive beamforming optimization problem [9]. Although, the use of genetic optimization is not new in smart antenna technology [e.g. see 14, 15] the developed GA in [9] has been shown to be very effective in smart antenna technology. However, the aim of this paper is to devise a practical and simple scheme that is suitable for a fast dynamically changing environment. We achieve this goal by considering a small number of snapshots to carry out digital beamforming. Since no covariance matrix is needed in GA approach and it operates with small population size, it can be implemented in real time using a modern digital signal processing device. The organization of this paper is as follows. Section II contains the signal model and presents the existing adaptive algorithms. The GA approach for the computation of the adaptive array weights is introduced in Section III. Simulation results are given in section IV and conclusions drawn in Section V.

2. BACKGROUND

Fig. 1 shows a block diagram of an antenna array controlled by adaptive algorithm. The output of such array which consists of N antennas at a time sample k is given by

$$y(k) = \mathbf{w^H x}(k) \tag{1}$$

where k is the time index, $\mathbf{x}(k) = [x_0(k) \quad x_1(k)....x_{N-1}(k)]^T$ is the complex vector of received signal, $\mathbf{w} = [w_0 \quad w_1....w_{N-1}]^T$ is the antenna weight vector, T and H denote transpose and conjugate transpose, respectively. The received signal at time instant k is given by

$$\begin{aligned}\mathbf{x}(k) &= \mathbf{s}(\mathrm{k}) + \mathbf{i}(\mathrm{k}) + \mathbf{n}(\mathrm{k}) \\ &= s(k)\mathbf{a}(\theta_s) + \sum_{j=1}^{I} i_j(k)\mathbf{a}(\theta_j) + n(k)\end{aligned} \tag{2}$$

where I is the number of interference signals. Here, $s(k)$ and $i_j(k)$ are the signal and interference symbol samples. It is understood that all the signal of interest (SOI), interferences, and thermal noise vary as function of time. The SOI and interference angles of arrival (AOA) are θ_s and θ_j, $j = 1,..., I$, respectively, with corresponding steering vectors $\mathbf{a}(\theta_s)$ and $\mathbf{a}(\theta_j)$. Let $\mathbf{R}_{xx}$ denote the $N \times N$ theoretical covariance matrix of the received signal. Assume that $\mathbf{R}_{xx}$ is a positive definite matrix with the following form

$$\mathbf{R}_{xx} = \sigma_s^2 \mathbf{a}(\theta_s)\mathbf{a}(\theta_s)^H + \sum_{j=1}^{I} \sigma_j^2 \mathbf{a}(\theta_j)\mathbf{a}(\theta_j)^H + \mathbf{Q} \tag{3}$$

where σ_s^2 and σ_j^2, $j = 1,..., I$ are the powers of the uncorrelatedd impinging signals $s(k)$ and $i_j(k)$ respectively, and $\mathbf{Q}$ is the noise covariance matrix. The error signal $\varepsilon(k)$, as indicated in Fig.1, is

$$\varepsilon(k) = d(k) - \mathbf{w^H}(k)\mathbf{x}(k) \tag{4}$$

where $d(k)$ is the desired output at sample k . The cost (fitness) function is given as

$$J(\mathbf{w}) = E[d^2(k)] - 2\mathbf{w}^H(k)\mathbf{r}_{dx} + \mathbf{w}^H(k)\mathbf{R}_{xx}\mathbf{w}(k) \tag{5}$$

where $\mathbf{r}_{dx} = d(k)\mathbf{x}^H(k)$ and $\mathbf{R}_{xx} = \mathbf{x}(k)\mathbf{x}^H(k)$. We may employ the gradient method to locate the minimum of (5). Thus

$$\nabla_w (J(\mathbf{w})) = -2\mathbf{r}_{dx} + 2\mathbf{R}_{xx}\mathbf{w}(k) \tag{6}$$

The minimum occurs when the gradient is zero. Thus, the solution for the weights is the optimum Wiener solution as given by

$$\mathbf{w}_{opt} = \mathbf{R}_{xx}^{-1}\mathbf{r}_{dx} \tag{7}$$

It is obvious that computation of the optimum weight requires the knowledge of both the correlation matrix of the input signal and the cross correlation matrix between the input signal and the desired signal. Computing the inverse of the autocorrelation matrix can be costly; using the steepest descent algorithm one can reduce the computation time since the weights are computed in a recursive way. The updated weight vector, which uses the steepest descent optimization method, is written as [1]

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu\varepsilon^{H}(k)\mathbf{x}(k) \tag{8}$$

Here, μ is the step size which controls the rate of convergence of the LMS algorithm. One of the drawbacks of the LMS is that the algorithm must go through many iterations before satisfactory convergence is achieved. If the signal characteristics are rapidly changing, the LMS algorithm may not allow tracking of the desired signal in satisfactory manner. One possible approach to circumventing the relatively slow convergence of the LMS is by use of RLS algorithm [1]. The convergence rate can also be accelerated by use of the conjugate gradient (CG) method [4,10]. The goal of CG is to iteratively search for the optimum solution by choosing perpendicular paths for each new iteration. The detailed information about these methods can be found in the cited references. Moreover, these algorithms are canonical adaptive signal processing algorithms. They are based on the steepest descent algorithm, which is easy to implement but can get stuck in a local minimum. Furthermore, the problem with the adaptive algorithms (see Fig.1) is that they need a receiver at each element to detect the incident signals to form the covariance matrix. The receivers are very expensive and require regular calibration, so the cost of this type of an array is extremely high. In addition, if the number of iteration (snapshots of data) used for covariance matrix estimation is insufficient, the adaptive algorithms will attenuate the desired signal as it were interference.

3. PRINCIPLES OF THE PROPOSED METHOD

3.1. Cost function of GA

A genetic algorithm manipulates the variables of a cost function until the cost is minimized. In this case, the cost function is a linear array with variable amplitude and /or phase weights, and the cost is the total output power. It returns the sum of the magnitude of the array factor at interference angles θ_j, $j = 1,..., I$. Its equation is written as

$$J(w) = 20Log10\left(\min\left[\sum_{j=1}^{I}\sum_{n=1}^{N} w_n e^{j\frac{2\pi}{\lambda}nd\sin(\theta_j)}\right]\right) \tag{9}$$

Where d represents the spacing between antenna elements, λ is wavelength and $w_n = a_n e^{\delta_n}$. Controlling the weights modifies the main beam peak and nulls.

The problem with this cost function formulation is that the desired signal and the interfering signals are mixed together. Minimizing the output power will decrease the desired signal in addition to the interfering signals unless the desired signal is assumed to enter the main beam and the adaptive weights are constrained to small values that cannot place a null in the main beam.

Since the total output power consists of both the desired signal and interference signals, some constraints are needed to insure that only the sidelobes are nulled and not the main beam. This paper shows how the constraints are implemented through using only a fraction of the elements in the antenna array. In other words, only a few of the edge element are given variable amplitude weights. For example, if a twenty element array has amplitude weights as given by the following

$$\mathbf{w} = [\underbrace{w_{-10} \quad w_{-9} \quad w_{-8}}_{chromosome} \quad \underbrace{ones(1,14)}_{cons\tan t} \quad \underbrace{w_8 \quad w_9 \quad w_{10}}_{chromosome}]. \tag{10}$$

Where the three edge elements on both ends of the antenna array have continues variable amplitude settings and the remaining 14 elements in the middle of array have uniform amplitude weight. Here the amplitude weights of the array are assumed symmetric. A continuous variable GA is used in place of the binary GA to do the adaptation. The array is assumed to start with a uniform amplitude distribution. Each chromosome in the GA population represents amplitude setting at each edge element in the array. Adjusting these settings has a small effect on the main beam but can place nulls in the sidelobes. The goal of the GA is to minimize the total output power of the antenna by adjusting these edge elements. Since the algorithm must be fast and a global minimum is not necessary, the GA uses a small population size.

It should be mentioned that the most genetic algorithms have a large population size and low mutation rate. Although these implementations have been successful, they require many function calls to find an acceptable solution. These slow algorithms will not work under fast dynamically changing environment in real time applications like mobile communication systems. There has been strong evidence that genetic algorithms with small population size and high mutation rates find good solutions fast [13], [14].

3.2. Genetic Algorithm

In conventional adaptive processing, it is assumed that a set of weights $\mathbf{w}_n$, $n = 0,1,2,..., N-1$, is connected to each one of the antenna elements. Then, a block of data is generated corresponding to M snapshots (i.e., $\mathbf{x}_n^m$ for m= 0, 1, . .. , M-1 and n= 0, 1,. . . , N-1). Here the superscript m on $\mathbf{x}_n$ denotes that the voltage x_n^m is induced at antenna element n at a specific time instance m. Then, a covariance matrix of this block of data of $(N \times M)$ samples is evaluated and the adaptive weights are given by the Wiener solution, which is related to the inverse of the covariance matrix. The computational load of forming covariance matrix and its inversion is an $\Theta(N^3)$ operation (see equation 7). Hence, it is difficult to implement it in real time. In addition, the procedure assumes that the data is stationary over these (M) samples (i.e., the environment of the SOI, and interference scenarios have not changed over the entire data collection process). Because of these disadvantages of the conventional adaptive processing, a GA with small population size is used to perform the adaptation control of the antenna weights. The structure of the adaptive beamforming controlled by a GA is shown in Fig. 2. The GA performs the adaptation by manipulating the weight vector of the cost function until the total output power is minimized.

Since the GA reduces the total output power of the beamformer, constraints are used to prevent desired signal attenuation in the main beam. In this work, the constraints take the form of using only a few of the edge elements of the array. Because only few of the edge elements are adaptive, the main beam receives limited perturbation. As an example, consider an array of 20 elements that are spaced 0.5λ apart. Three edge elements on both ends of the array have continues variable amplitude settings. There are two interference signals incident at 20 degree and -20 degree. The resulting adapted amplitude weights are given by

$$\mathbf{w} = [0.0949 \quad 0.1685 \quad 0.8188 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 0.8188 \quad 0.1685 \quad 0.0949].$$

The array is assumed to start with a uniform amplitude distribution. The GA uses the following steps for adaptive interference cancellation:

1. An initial population of chromosomes is randomly generated. By this way, the first generation of chromosome is created. The weights of these three edge elements are described by a chromosome, i.e. each chromosome contains three variables.
2. The weights of these edge elements are sending to the beamformer and the output power is measured. In this way, a fitness (cost) value is assigned to each chromosome in the population in order to expressing how well the chromosome meets requirements to the optimized system.
3. Members of the population with high costs are discarded and a new population of chromosome (offspring) is generated by selecting the best existing chromosomes (parents). The parents are combined by crossover and mutation to produce offspring. The offspring replace the discarded chromosomes. This step is iterated M times. This means that M generations of chromosome are created in order to find as good chromosome as possible.
4. The result of the genetic optimization is obtained as the best chromosome at the M iteration.

The flow chart of the GA-based adaptive beamforming is shown in Fig. 3. Each chromosome represents the variable weights of edge elements. These weights are sent to the antenna array and the output power is measured. In this way, every population member has an associated cost. Members of the population with high costs are discarded. The surviving members form a mating pool. The parents are combined by using single point crossover to produce offspring. The offspring replace the discarded chromosomes. The next step randomly mutates a certain percentage of the population. After mutation, the process repeats by measuring the output power associated with the new population.

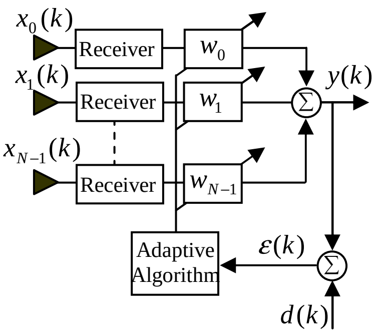


Fig.1 Smart Antenna Based Adaptive Algorithms.

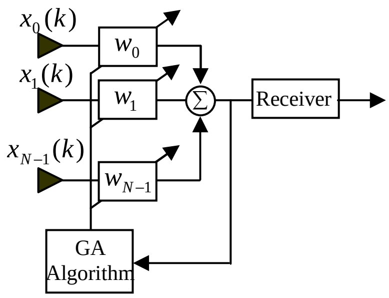


Fig.2 (a) Smart Antenna Based GA.

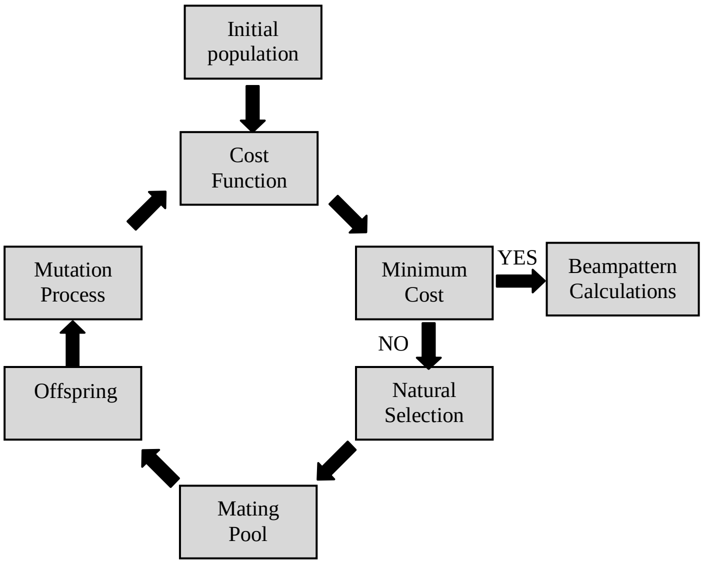


Fig. 2 (b) Flow Chart of Genetic Algorithm.

4. SIMULATION RESULTS

To evaluate the performance of the GA under a fast dynamically changing environment, some computer simulations have been carried out in various scenarios. In the following, we assume a uniform linear array with 20 elements and half-wavelength element spacing. The desired signal with SNR=20dB is assumed to impinge on the array from the direction $\theta_s = 0^o$. Two interferers are assumed to impinge on the array from the directions $\theta_1 = -20^o$ and $\theta_2 = 20^o$, both with interference-to-noise ratio (INR) equal to 20dB. The noise, $n(k)$, is spatially and temporally white and it has a complex Gaussian zero mean distribution with variance $\sigma_n^2 = 0.01$.

In our first example, we study the convergence rate of the GA-based smart antenna beamformer compared to LMS, RLS, and CG for interference cancellation. In this case, each run of Monte Carlo simulation consisting of $M = 100$ samples of $\mathbf{x}(k)$, i.e. 100 iterations or generations are used. The step size parameter of the LMS algorithm is chosen as $\mu = 1/4trace[\hat{\mathbf{R}}_{\mathbf{xx}}]$. The convergence rate of the LMS for interference cancellation is governed by the eignvalue spread of $\hat{\mathbf{R}}_{\mathbf{xx}}$. For RLS, the forgetting factor is chosen to be 0.9. While GA parameters include a population size of 8, a 50% selection rate, roulette wheel selection, uniform crossover, and a 10% mutation rate. The quiescent and resulting adapted beam patterns for all techniques appears in Fig.3(a). From Fig.3(a), we observe that, when the number of snapshots (M) is sufficient, all patterns have deep nulls at the AOAs of the interferences. The cost function (or mean square error) of all techniques as a function of iteration is shown in Fig. 3(b). It is quite clear from Fig.3(b) that the GA quickly converge to optimal interference cancellation. For this case, the CG performs similarly in terms of convergence rate. The LMS did not converge until after 50 iterations.

In the second simulation example, we investigate the effect of the small number of iterations (which can also be viewed as a steering vector error problem [8]) on the performance of the algorithms tested for the same scenario as in the previous example except for a sample size of $M = 20$ and $M = 10$ iterations. Figs.4(a) and 5(a) shows the beam patterns of the tested algorithms. Convergences of the LMS, RLS, CG, and GA algorithms for interference cancellation are shown in Figs. 4(b) and 5(b). In this scenario, the GA technique demonstrates an appropriate operation under this situation. On the other hand, as illustrated in Fig. 5(a), the beam pattern of the RLS allocates a deep null for the desired signal and the interference cancellation (creating nulls) of the LMS algorithm is unsatisfactory. This inadequate operation of the LMS and RLS highly depends on the number of iteration (M) which is used to estimate the covariance matrix $\hat{\mathbf{R}}_{\mathbf{xx}}$. In Fig.6 we show the output SINR of the tested algorithms versus the number of iteration. It is clearly demonstrate that the GA shows better capabilities against the effect of small number of iterations. It works well even when M is as small as $M = 2$. The LMS requires a large number of M. As illustrated in Fig.6, the RLS algorithm has a problem of instability with large number of iterations under a fast dynamically changing environment. This problem of the RLS is well-known in the adaptive control literature [15].

In our last example, we study the impact of constraints, using subset of few edge elements with GA, on the mainbeam perturbation and interference cancellation. Fig.7 show the SINR reduction when various subset of edge elements are used to perform the mainbeam constraint in GA. This reduction depends on the number of edge elements that used for constraint. In our simulation we found that the best result may be obtained when 3 elements on each end of the array were used. Fig.8(a) shows the GA patterns of the various subset of the edge elements. Corresponding convergence rates are shown in Fig. 8(b). From Fig.8, we have verified that the conventional GA (here we refer to conventional GA as unconstraint GA, where all antenna elements are variables, while the proposed one is refered as constraint GA) exhibits little perturbation to the mainbeam peak. On the other hand, the constraint GA performs very well especially when 3 edge elements on both ends of the array are variable.

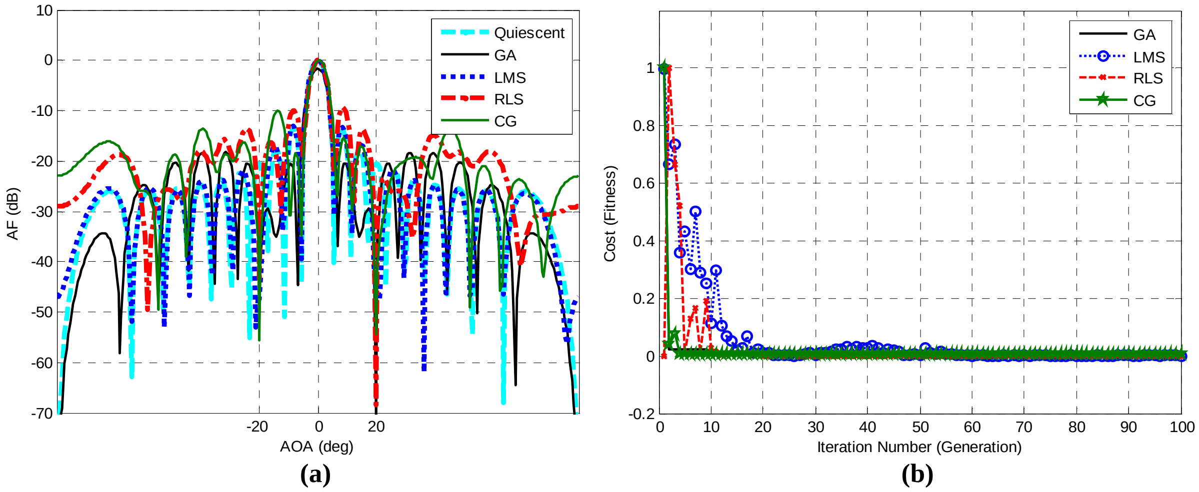


Fig.3 M=100 iterations: (a) Comparison of Patterns, (b) Objective Function versus Number of Iteration.

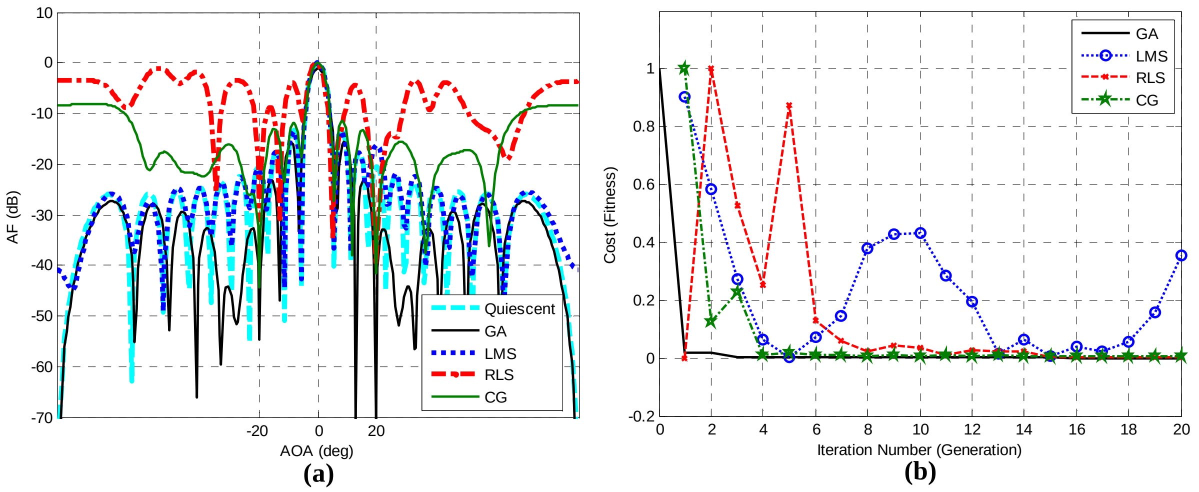
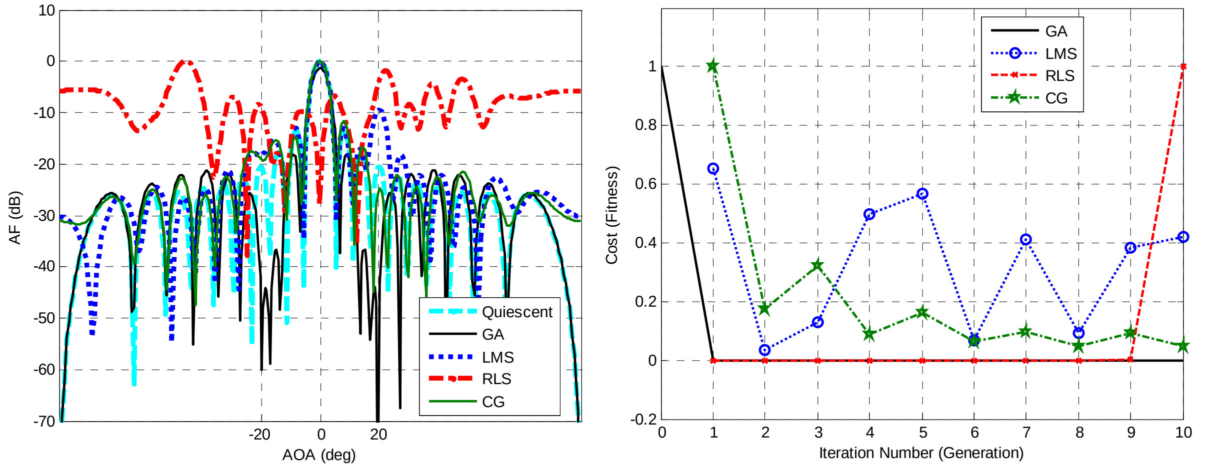


Fig.4 M=20 iterations: (a) Comparison of Patterns, (b) Objective Function versus Number of Iteration.



(a) **(b)**

Fig.5 M=10 iterations: (a) Comparison of Patterns, (b) Objective Function versus Number of Iteration.

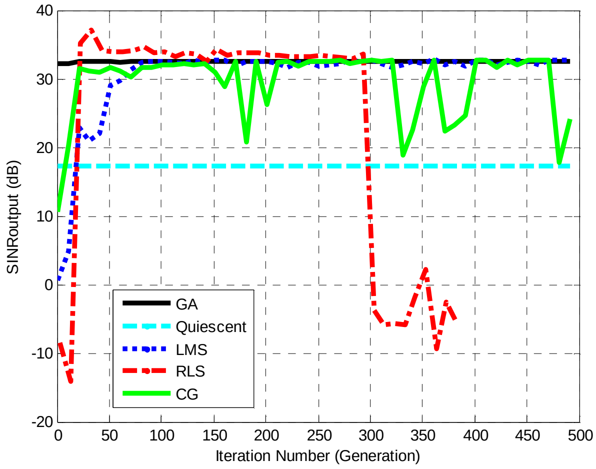


Fig.6 Output SINR Versus Number of Iteration.

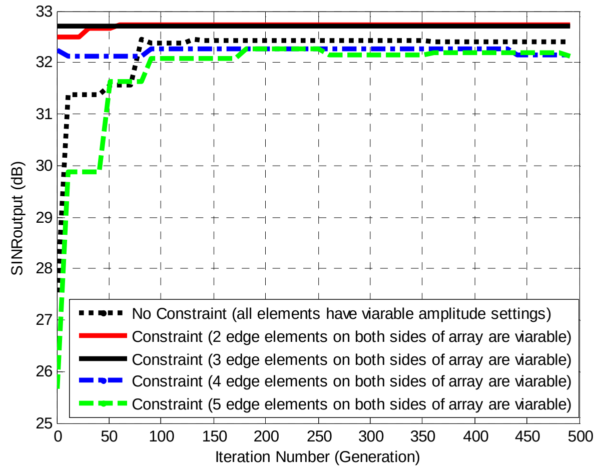
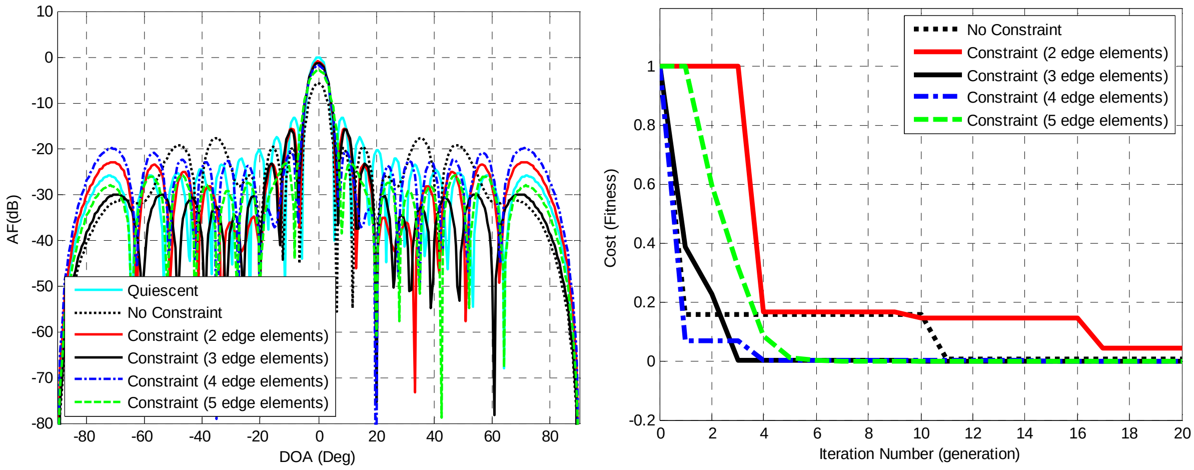


Fig.7 Output SINR Reduction When Various Subset are Used to Perform the Main Beam Constraint in GA.



(a) **(b)**

Fig.8 M=20 iterations: (a) GA Patterns, (b) Objective Function versus Number of Iteration.

5. CONCLUSIONS

For the conventional adaptive algorithms, an inadequate estimation of the covariance matrix (which often occurs under a fast dynamically changing environment) results in adapted antenna patterns with high sidelobes and distorted mainbeams. The GA has been proposed as an alternative to the conventional many iterations-based adaptive algorithms. The design of the corresponding GA was highlighted and its achievable performance was characterized in terms of both the optimal interferences cancellation and the SINR. It was demonstrated that a potentially more attractive SINR is achievable by the proposed GA-based smart antenna beamformer even when the number of available snapshots is scarce. Moreover, fast convergence to optimal solution is achieved by using a small population size and high mutation rate. Furthermore, using subset of the edge elements to form the constraint in the GA reduced the mainbeam perturbation and provides additional control over the sidelobe level.

REFERENCES

[1] Haykin, S. and Kailath, T. "Adaptive Filter Theory" , Fourth Edition, Person Education (Singapore), Indian Branch, India, 2002.

[2] Chang, L., and C. Yeh, ''Performance of DMI and Eigenspace-Based Beamformers,'' *IEEE Trans. On Antennas and Propagation*, Vol. 40, No. 11, November 1992, pp. 1336–1347.

[3] R. Ghayoula R., Fadlallah G., Gharsallah A., and Rammal M., " Phase-only adaptive nulling with neural networks for antenna array synthesis" *IET Microw. Antennas Propag.*, 2009, Vol. 3, Iss. 1, pp. 154–163.

[4] Chen, H., et al., ''Adaptive Spectral Estimation by the Conjugate Gradient Method,'' *IEEE Trans. On Acoustics, Speech and Signal Processing*, Vol. ASSP-34, No. 2, April 1986, pp. 272–284.

[5] Larimore, M., and J. Treichler, ''Convergence Behaviour of the Constant Modulus Algorithm,'' *IEEE International Conference on ICASSP'83*, Vol. 8, April 1983, pp. 13–16.

[6] Sarkar, T. K., M. C. Wicks, and M. Salazar-Palma, Smart Antennas, New York: Wiley, 2003.

[7] M. Wax and Y. Anu, "Performance Analysis of the Minimum Variance Beamformer in the Presence of Steering Vector Errors," *IEEE Trans. Signal Process.*, Vol. 44, No. 4, PP. 938–947, Apr. 1996.

[8] J. Li and P. Stoica, "Robust Adaptive Beamforming", John Wiley and Sons, Inc., Hoboken, New Jersey, 2006.

[9] Weijun Cheng, Xiaoling Ma, Liu Jie, "The Impact of Genetic Algorithm's Parameters on Random Interference Nulling for Circular Array", 7th *IEEE International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM), PP. 1-5, 2011.*

[10] Noori, N. and Razavizadeh, S.M., "Cognitive beamforming using genetic algorithm", IEEE International Symposium on Antennas and Propagation (APSURSI), PP. 1-4, 2010.

[11] Khan, M.R.R. and Tuzlukov, V., "Null Steering Beamforming for Wireless Communication System Using Genetic Algorithm", *IEEE International Conference on Microwave Technology & Computational Electromagnetics (ICMTCE)*, PP. 289 - 292, May, 2011.

[12] Godara, L., Smart Antennas, CRC Press, Boca Raton, FL, 2004.

[13] R.L. Haupt and S.E. Haupt, "Optimum population size and mutation rate for a simple real genetic algorithm that optimizes array factors," *Applied Computational Electromagnetics Society Journal*, Vol. 15, No. 2, PP. 94-102, July 2000.

[14] R.L. Haupt and S.E. Haupt, Practical Genetic Algorithms, 2nd ed. New York: Wiley, 2004.

[15] M. Tham and S. Mansoori, " Covariance Resetting in Recursive Least Squares Estimation, " *Int. Conf. Control*, April 1988, PP. 128-133.

BIBLIOGRAPHY OF AUTHORS

Jafar Ramadhan Mohammed received the B.Sc. and M.Sc. Degrees in Electronic and Communication Engineering from University of Mosul, IRAQ, in 1998, and 2001, respectively, and the Ph.D. degree in Digital Communication Engineering from Panjab University, INDIA in Nov. 2009. He is currently a Senior Lecturer at University of Mosul, IRAQ. His main research interests are in the area of Adaptive Signal Processing and its application, Digital Systems and Wireless Communications.