Advanced Research in Electrical and Electronic Engineering
Print ISSN: 2349-5804; Online ISSN: 2349-5812 Volume 2, Number 1 October-December (2014) pp. 95-100
© Krishi Sanskriti Publications
http://www.krishisanskriti.org/areee.html

Comparative Analysis of Smart Antenna with Adaptive Beamforming using BFO Algorithm

Hardeep Singh[1] and Gurwinder Kaur[2]

*[1]M.tech, Electronics and Communication Engg. Yadavindra College of Engineering,
Guru Kashi Campus, Punjabi University Patiala, Talwandi Sabo, Bathinda-151302, India.
[2]Electronics and Communication Engg. Deptt. Yadavindra College of Engineering,
Guru Kashi Campus, Punjabi University Patiala, Talwandi Sabo, Bathinda-151302, India.
E-mail: [1]gillhardeep91@gmail.com, [2]gurwinder.k.garcha@gmail.com*

Abstract: *This paper deals with a scheme to realize a smart and adaptive antenna for the mobile base station using BFO (Bacterial Foraging Algorithm). In order to make a increment in the efficiency and to decrease the signal losses in the system we use bacterial foraging algorithm in place of genetic algorithm to provide training for neural network and fuzzy logic. In this given a desired angle is given as input, the proposed method gives the corresponding antenna parameters as output. The comparative and analytical results prove the performance of proposed method over existing methods. The weights obtained by the above algorithm are then used to steer the antenna array in the direction of interest and thereby enhancing the signal to noise interference ratio. It is found that bacterial foraging optimization algorithm is capable of improving the gain and side lobe level (SLL) for the desired outcomes.*

Keywords: *Adaptive Antenna System, Bacterial Foraging Algorithm, Smart Antenna, Neural Network and Fuzzy Logic Controller.*

1. INTRODUCTION

Beamforming is the technique[11,18] used to create the radiation pattern of an antenna array by adding constructively phases of the signals in the direction of the targets desired and nullifying the pattern of the targets which are undesired.

Various adaptive algorithms were developed for various purposes and tasks. The objective of the algorithm in a Smart antenna system is to modify the received signals so that the necessary signals are extracted once signals are mixed[2,3,4]. Various techniques can be used in the realization of an adaptive algorithm. In this project, the adaptive algorithm is implemented in MATLAB code. A smart algorithm commonly requires more resources than algorithms that are less intelligent. In this work we use bacterial foraging algorithm as explained below.

2. BACTERIAL FORAGING ALGORITHM

Bacterial Foraging Technique is getting importance in the optimization problems because,

1) Philosophy says biology gives highly automated, robust and effective organism.
2) Search strategy of bacteria is salutatory in nature.
3) Bacteria can sense, decide and act to adopt social foraging.

During the foraging of the actual bacteria, locomotion is obtained by a set of stretch flagella. Flagella aid an *E.coli* bacterium to tumble or swim, which are basically two operations performed by a bacterium at the time of foraging. When the flagella are rotated by them in clockwise direction, each flagellum drags on the cell. That results in the moving of flagella independently and at last the bacterium tumbles with smaller number of tumbling whereas in a harmful place it tumbles rarely to find out a nutrient gradient. Moving the flagella in the anticlockwise direction aids the bacterium to swim at a very high rate. In the above algorithm the bacteria goes through chemotaxis, where they like to move in the direction of a nutrient gradient and avoid noxious environment[18].

[Step 1] Initialize parameters *p, S, Nc, Ns, Nre, Ned, Ped, C(i)(i=1,2…S)*, θ^i.
[Step 2] Elimination-dispersal loop: $l=l+1$
[Step 3] Reproduction loop: $k=k+1$
[Step 4] Chemotaxis loop: $j=j+1$
[a] For i =1,2…S take a chemotactic step for bacterium i as follows.
[b] Compute fitness function, $J(i, j, k, l)$.
Let, $J(i, j, k, l) = J(i, j, k, l) + J_{cc}(\theta^i, (j,k,l), P(j, k, l))$ (i.e. add on the cell-to cell attractant–repellant profile to simulate the swarming behavior)
where, J_{cc} is defined in (2).
[c] Let *J last*= $J(i, j, k, l)$ to save this value since we may find a better cost via a run.
[d] Tumble: generate a random vector D(i)Î $\Re^p$ with each element $\Delta_m(i)$ 1,2,..., p, m = 1, 2,….,p, a random number on [-1, 1].
[e] Move: Let

$$\theta^i\ (j+1,k,l)\ =\ \theta^i\ (j,k,l)+\frac{C(i)\ \Delta(i)}{\sqrt{\{\Delta^T(i)\ \Delta(i)\}}}$$

This results in a step of size *C*(*i*) in the direction of the tumble for bacterium *i*.

[f] Compute *J* (*i*, *j* +1, *k*, *l*) and let
J (*i*, *j+1*, *k*, *l*) = *J* (*i*, *j*, *k*, *l*) + J_{cc} (θ^i, $(j+1,k,l)$,*P*(*j+1*, *k*, *l*)).
[g] Swim
i) Let *m*=0 (counter for swim length).
ii) While *m*< *N* (if have not climbed down too long).
• Let *m=m+1*.
• If J (*i*, *j* +1, *k*, *l*) < J*last* (if doing better), let J*last* = J (*i*, *j* +1, *k*, *l*) and let

$$\theta^i\ (j+1,k,l)\ =\ \theta^i\ (j,k,l)+\frac{C(i)\ \Delta(i)}{\sqrt{\{\Delta^T(i)\ \Delta(i)\}}}$$

And use this θ^i (*j*+1, *j*, *k*) to compute the new J (*i*, *j* +1, *k*, *l*) as we did in [f]
• Else, let *m*= N_s . This is the end of the while statement.
[h] Go to next bacterium (*i*+1) if *i* ≠ *S* (i.e., go to [b] to process the next bacterium).
[Step 5] If $j < N_c$, go to step 4. In this case continue chemotaxis since the life of the bacteria is not over.

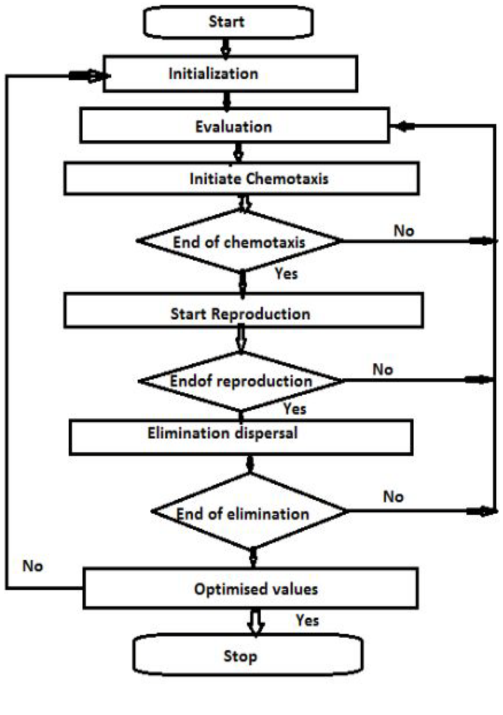

Fig. 1: Flow chart of BFO algorithm

[**Step 6**] Reproduction:
[a] For the given *k* and *l*, and for each *i* = 1,2,..., *S* , let

$$J^i_{health} = \sum_{j=1}^{N_c+1} J(i,j,k,l)\ldots \qquad (3)$$

be the health of the bacterium *i* (a measure of how many nutrients it took over its lifetime and how successful it was at avoiding noxious substances). Sort bacterial and chemotactic parameters *C*(*i*) in order of ascending cost J_{health}(higher cost means lower health).

[b] The S_r bacteria with the highest J_{health} values die and the remaining S_r bacteria with the best values split (this process is performed by the copies that are made are placed at the same location as their parent).

[**Step 7**] If $k < N_{re}$, go to step 3. In this case, we have not reached the number of specified reproduction steps, than we start the next generation of the chemotactic loop.

[**Step 8**] Elimination-dispersal: For *i* = 1,2, 3..., *S* with probability P_{ed} , eliminate and disperse each bacterium (this keeps the number of bacteria in the population constant). To do this, if a bacterium is eliminated, simply disperse other one to a random location on the optimization domain. If *l* is less than N_{ed} , then go to step 2; otherwise end.

3. TRAINING FIS

FIS is used in this for computation of the phase angle and position for a given set of angles. The input given to the fuzzy logic is angle values and the output that we got from FIS is phase angle and position. We can feed one or more than one angles as input to the fuzzy logic. Triangular member function is used in this for training FIS. The fuzzy training consists of three steps namely; fuzzification, generating fuzzy rules and defuzzification. In the fuzzification the input data is converted into the fuzzy data. After fuzzification the next step is to generating fuzzy rules.

Generating fuzzy rules: Generating fuzzy rules is also important steps in fuzzy logic training. The input variable is in the range [θ_{min} , θ_{max}] and the output variable is in the range [x_{min} , x_{max}] and [ϕ_{min} , ϕ_{max}]. The minimum and maximum θ value is 0 and 360, respectively. The input variables are fuzzified by considering this obtained range into three sets namely large, medium and small and the output variables are fuzzified into five sets namely very large, large, medium, small and very small. Fuzzy rules are produced by using these variables. The next step after completion of the fuzzy rules generation, is training FIS. The next process after training FIS is defuzzification. The fuzzy data is converted back into system data in the defuzzification process. FIS is ready for practical application after completion of this training. If angle is given as input to FIS, it gives the position and the phase angle as output data.

4. NN TRAINING FOR BEAMFORMING

NN is used here for computing the phase angle and position for given angles. Basically NN consisting three layers namely, input, hidden and output layer. In our method, input, hidden and output layer consist of m, N and two layers, respectively. Here the number of inputs depends on our requirement. NN back propagation algorithm is used in this for training.

Step 1: Initialize the input weight of each neuron.
Step 2: Apply a training sample to network.
Here θ is the input to the network and x_P and ϕ_p are the outputs of the network.

$$\phi_S = \sum_{r=1}^{n} W_{2r1}\, Y(r) \ldots\ldots\ldots\ldots\ldots \qquad \ldots\ldots\ldots\ldots\ldots(4)$$
$$x_S = \sum_{r=1}^{n} W_{2r2}\, Y(r) \ldots\ldots\ldots\ldots\ldots\ldots \qquad \ldots\ldots\ldots\ldots(5)$$
Where
$$Y(r) = \frac{1}{1+\exp(-W_{11r}(\theta_1+\theta_2+\cdots\ldots+\theta_m))} \cdots \qquad \ldots\ldots\ldots(6)$$

(4) and (5) shows the activation function performed for training NN.
Step 3: Adjust the weights of all the neurons.
Step 4: Determine the value of x_p and ϕ_p using the actual output of the network.
Step 5: Repeat the iteration process till the output reaches its least value.
NN is ready for practical application after completion of training. Therefore, if we give angles as input to NN, it gives corresponding phase angle and position as output

5. RESULTS AND DISCUSSIONS

The results have been obtained using MATLAB 2012. Various outcomes are displayed as in following results.

Results of Beamforming with Genetic algorithm

In this we a comparative analysis at three different angles is done and there parameters are to be considered for each different angles.

Beamforming at 0° with GA

Here, the input angle is given as 0° and as from Fig. 2 it is clear that the main beam is focused towards $0^\circ$and except the main beam, all other beams are side lobes.

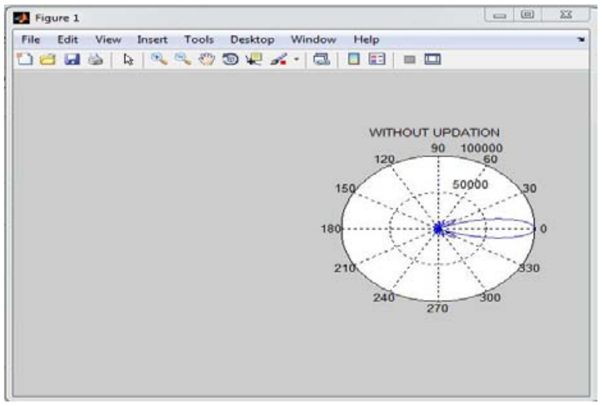

Fig. 2: Beamforming at 0° with GA

The main beam will focus towards the direction of the angle given as input. In this side lobe level is 22.33 db and the gain across this angle is 91.225 db. The Signal to interference ratio measured at 0° is 0.7771 db and the total time for computing outputs by GA is 8.21 seconds.

Beamforming at 0° and 230° with GA

In this the input angle is given as 0° and 230°. From the Fig. 3 it is clear that main beam focused towards 0° and 230° and all other beams are side lobes. In this side lobe level is 23.12 db and the gain across this angle is 97.230 db. The Signal to interference ratio measured at 0° and 230° is 0.7989 db and the total time for computing outputs by GA is 9.12 seconds.

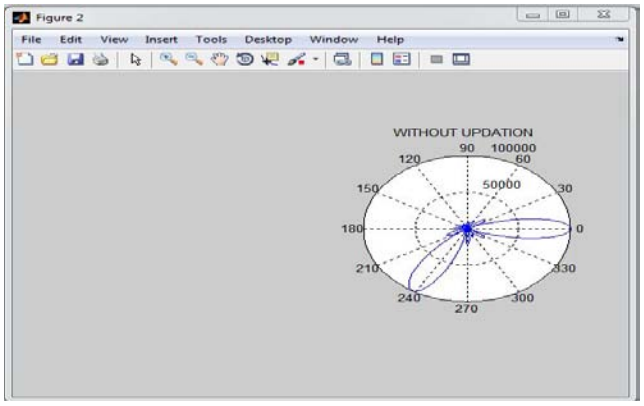

Fig. 3: Beamforming at 0° and 230° with GA

Beamforming at 0° and 180° with GA

In this the angle given as input is 0° and 180° and from the Fig. 4 it is clear that the main beam is focused towards 0° and 180° and all other beams are side lobes. So main beam is focus towards the direction of the angles given as input. . In this side lobe level is 22.12 db and the gain across this angle is 96.112 db. The Signal to interference ratio measured at 0° and 180° is 0.8121 db and the total time for computing outputs by GA is 10.11 seconds.

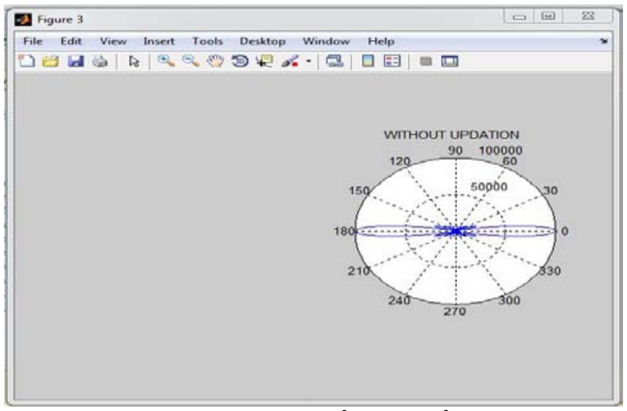

Fig. 4: Beamforming at 0° and 180° with GA

Results of Beamforming with Bacterial Foraging Algorithm :

Beamforming at 0° with BFO

Here, the input angle is given as 0° and as from Fig. 5 it is clear that the main beam is focused towards 0° and except the main beam, all other beams are side lobes. The main beam will focus towards the direction of the angles given as input. In this SLL level is reduced as compared to GA at 0° and therefore signal to interference ratio also improved. The value of SLL is 18.4245 db. Here Gain is 91.45 db which is more by 0.225 db as compared to GA. The value Signal to Interference ratio is also improved by 0.0202 db. Here computation time is also less as compared to GA i.e. 8 seconds.

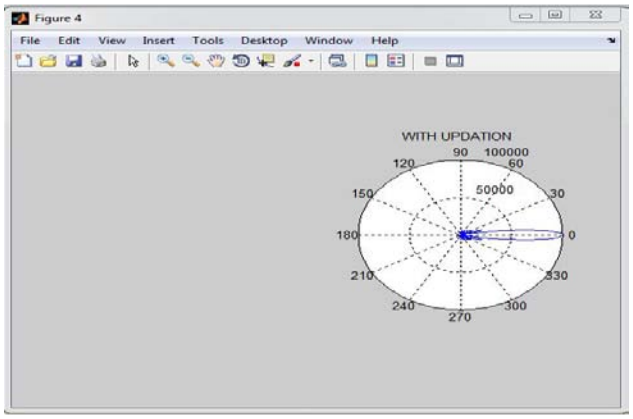

Fig. 5: Beamforming at 0° with BFO

Beamforming at 0° and 230° with BFO

In this the input angle is given as 0° and 230°.

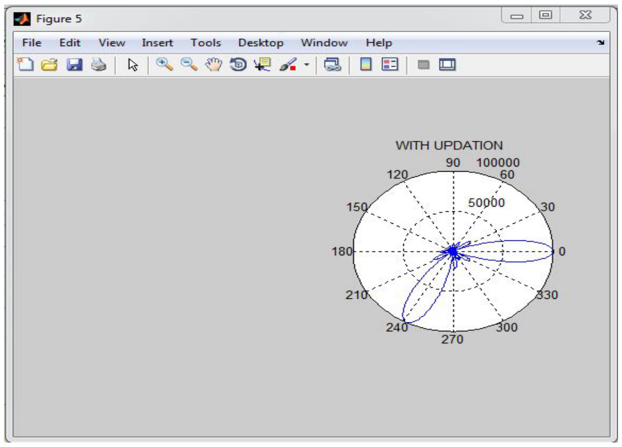

Fig. 6: Beamforming at 0° and 230° with BFO

Beamforming at 0° and 180° with BFO

From the Fig. 6 it is clear that main beam focused towards 0° and 230° and all other beams are side lobes. . In this SLL level is reduced as compared to GA at 0° and 230° and therefore signal to interference ratio also improved.

The value of SLL is 21.2145 db. Here Gain is 97.415 db which is increased by 0.185 db as compared to GA. The value Signal to Interference ratio is also improved by 0.0023 db. Here computation time is also less as compared to GA i.e. 8.97 seconds.

In this the angle given as input is 0° and 180° and from the Fig. 7 it is clear that the main beam is focused towards 0° and 180° and all other beams are side lobes.

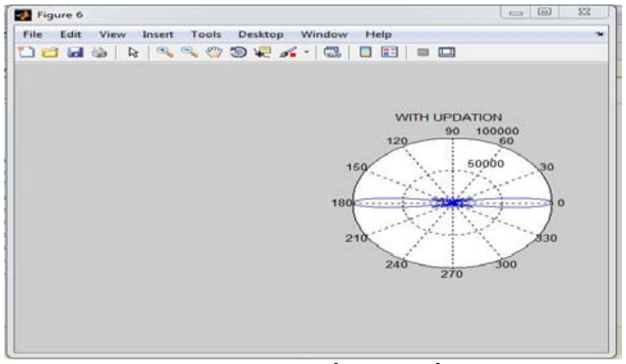

Fig. 7: Beamforming at 0° and 180° with BFO

Graph of Bit Error Rate versus SNR

In this SLL level is reduced as compared to GA at 0° and 230°

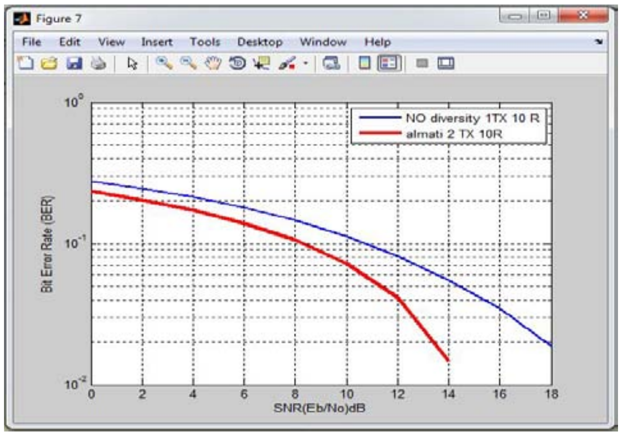

Fig. 8: Bit error rate against SNR graph with GA

and therefore signal to interference ratio also improved. The value of SLL is 21.2145 db. Here Gain is 96.28 db which is increased by 0.168 db as compared to GA. The value Signal to Interference ratio is also improved by 0.0097 db. Here computation time is also less as compared to GA i.e. 9.10 seconds. In this section BER against signal to noise ratio plots are plotted using GA and BFO as shown in Fig. 8 and Fig. 9. From these figures we see that bit error rate is decreased with updating and SNR increases with the use of BFO. In Fig. 8 the value of BER in graph is from 0.01 to 0.2 approximately which small but by using BFO, Bit Error Rate goes to ranges from 0.000001 to 0.0001 which is very small , so BER gets reduced and also the SNR increases from 14db to 18 db which is improved.

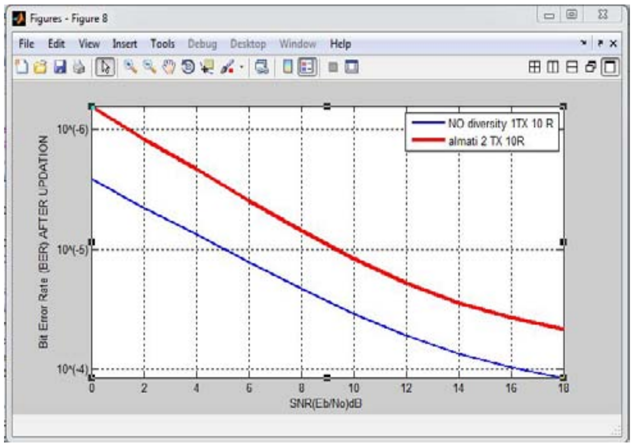

Fig. 9: Bit error rate against SNR graph with BFO

Results table of these values is as shown below:

Table 1: Performance of beamforming training based on GA

Angle (degree)	SLL (db)	Gain (db)	SIR (db)	Time (s)
0°	22.33	91.225	0.7771	8.21
0°- 180°	22.12	96.112	0.8121	10.11
$0^{\circ} - 230^{\circ}$	23.12	97.230	0.7989	9.12

Table 2: Performance of beamforming training based on BFO

Angle (degree)	SLL (db)	Gain (db)	SIR (db)	Time (s)
0°	18.424	91.45	0.7973	8.00
0°- 180°	20.214	96.28	0.8218	9.10
$0^{\circ} - 230^{\circ}$	21.214	97.41	0.8012	8.97

When we compared the BFO algorithm with other heuristic search algorithms such as GA, PSO and SA than BFO out performs in major circumstances with enough positive performance deviation. BFO algorithm may be inefficient under some circumstances but in terms of performance measure such as gain and signal to interference ratio BFO shines well. This asserts that for beamforming, it would be better if we utilize BFO for generating a training database.

6. CONCLUSION

In this study, fuzzy Inference System and Neural Network are used for beamforming in smart antenna. In the proposed method bacterial foraging optimization algorithm is used for generating the training dataset. Then using this generated dataset, fuzzy logic and neural network is trained. The proposed method is implemented in MATLAB 2012 and tested by giving different angles as input to the system. The computational time and gain of the proposed method (BFO) is compared with beamforming with genetic algorithm and from the results obtained it clear that the proposed method is better than the method of beamforming with GA. It is also observed that using Bacterial Foraging Algorithm side lobe level is reduced as comparison to other methods. The interference of the system is also decreased with reduction in side lobe level.

REFERENCES

[1] Liu.J., Gershman A.B.: 'Adaptive Beamforming With Sidelobe Control: A Second-Order Cone Programming Approach', IEEE Signal Proc., 10, (11), 2003, pp. 331-334

[2] Rugamba J., Snyman L.W., Kurien A., Chatelain D.: 'Viability of using intelligent (smart) antenna system in GSM cellular networks'. Proc. IEEE Conf., South Africa, 2004, pp. 124–130

[3] Bellofiore S., Balanis C.A., Foutz J., Spanias A.S.: 'Smart-Antenna Systems for Mobile Communication Networks Part I: Overview and Antenna Design', IEEE Antenna's Propag. Mag., 44(3), 2002

[4] Jootar J., Zeidler J.R., Proakis J.G.: 'Performance of Alamouti Space-Time Code in Time-Varying Channels with Noisy Channel Estimates', IEEE Commun. Society / WCNC, 2005, pp. 498-503

[5] Awan H., Abdullah K., Faryad M.: 'Implementing smart antenna system using genetic algorithm and artificial immune system', National Uni. Sci. Technol., Rwp, 2006

[6] Mani V.V., Bose R.: 'Genetic algorithm based smart antenna design for UWB beamforming', Proc. IEEE Conf., Delhi, 2007, pp. 442-446

[7] Gaudes C.C., Vía J.: 'Robust Array Beamforming With Sidelobe Control Using Support Vector Machines', IEEE Trans. Signal Proc., 55, (2), 2007, pp. 574-584

[8] Yoon B.J., Tashev I., Acero A.: 'Robust Adaptive Beamforming Algorithm UsingInstantaneous Direction of Arrival with Enhanced Noise Suppression Capability', IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), 2009, pp. 133-136

[9] Datta T., Misra I.S.: 'Improved adaptive bacteria foraging algorithm in optimization of antenna array for faster convergence', Electromagn. Research C, 2008, 1, pp. 143-157

[10] Wang Y., Liao G.S., Ye Z.: 'Combines Beamforming with Alamauti Coding using Double Antenna Array Groups for Multiuser Interference Cancellation', Progress Electromagn. Research, 88, 2008, pp. 213-226

[11] Datta T., Misra I.S.: 'A comparative study of optimization techniques in adaptive antenna array processing : The bacterial-foraging algorithm and particle-swarm optimization', IEEE Antennas Propag. Mag., 2009, 51, (6), pp. 69-79

[12] Gotsis A.K., Siakavara K.: 'On the direction of arrival (DoA) estimation for a switched-beam antenna system using neural networks', IEEE Trans. Antennas Propag., 2009, 57, (5), pp. 1399-1411

[13] Jeyanthi K.M., Kabilan A.P.: 'A simple adaptive beamforming algorithm with interference suppression', Int. J., Dindigul, 2009

[14] Santhi Rani Ch., Subbaiah P.V., Chennakesava Reddy K., Sudha Rani S.: 'LMS and RLS algorithms for smart antennas in a WCDMA mobile communication environment', ARPN J. Eng. Appl. Sci., 2009, 4, (6), pp. 78–88

[15] Shaukat S.F., Hassan M.U., Farooq R., Saeed H.U., Saleem Z.: 'Sequential Studies of Beamforming Algorithms for Smart Antenna Systems', World App. Sci. J., 6, (6), 2009, pp. 754-758

[16] Hussain M.A., Suresh Varma P., Satya Rajesh K., Pathan H.B., Sarraju L.M.: 'Use of smart antennas in AD HOC networks', Int. J. Comput. Sci. Inf. Tech., 2010, 2, (6), pp. 48–54

[17] Shivam Prasad A., Vasudevan s., R.S., Ram K.S., G.S., Narayanan B.S.: ' Analysis of adaptive algorithms for digital beamforming in smart antennas',IEEE Int. Conf., Chennai, 2011, pp. 64-68

[18] Basha Ghouse T.S., Sridevi P.V., Giri Prasad M.N.: 'Enhancement in Gain and Interference of Smart Antennas Using Two Stage Genetic Algorithm by Implementing it on Beam Forming', Int. J. Electron. Eng., 3 (2), 2011, pp. 265– 269

[19] R.M.S., Venkateswaran N.: 'Optimization of linear array antenna pattern synthesis using bacterial foraging algorithm', Proc.IEEE Int. Conf., Kalavakkam, 2012, pp. 130-134

[20] Bose S., Prabu K., Kumar D.S.: 'Array signal processing and optimization using algorithms in nature', Int. Conf. Info. Net. Techno., 2012, 37, pp. 139-144

[21] Choudhury B., Acharya O.P., Patnaik A.: 'Bacteria foraging optimization in antenna engineering: An application of array fault finding', Wiley Periodicals, Inc., 2012, pp. 141-147

[22] S. Hardeep, K. Gurwinder, "Comparative Analysis of Smart Antenna with Adaptive Beamforming using BFO Algorithm", Int. J. of Adv. Eng. and Research Develop., print-ISSN: 2348-6406, Issue 6, vol. 1, Jun. 2014.