Circular Antenna Array Design Using Novel Perturbation Based Artificial Bee Colony Algorithm

Digbalay Bose[1], Souvik Kundu[1], Subhodip Biswas[1], and Swagatam Das[2]

[1] Faculty of Engineering & Technology, Jadavpur University, Kolkata 700 032, India
[2] Indian Statistical Institute, Kolkata – 700 108, India
{digbose92,sk210892,subho.opto.placis91}@gmail.com,
swagatam.das@isical.ac.in

Abstract. The field of optimization is abundant with algorithms, which are inspired from nature based phenomena. The increasing popularity of such algorithms stems from their applications in real life situations. Here in this article a real life problem in the form of the design of circular antenna array has been discussed. The design of the antenna array is based on the application of a novel variant of Artificial Bee Colony Algorithm using selective neighborhood called sNABC**.** We use a neighborhood based perturbation on the basis of Euclidean distance and fitness of individuals are used for obtaining minimum side lobe levels, maximum directivity and appropriate null control. To illustrate the effectiveness of our design procedure, the results have been compared with several existing algorithms like DE, ABC and PSO**.**

1 Introduction

In this modern era where long distance communication is highly prevalent, antennas possessing greater directive radiation nature are desirable. For achieving such a directive radiation pattern antenna arrays [1-3] are used in place of a single antenna, whose elements of the are combined in specific electrical and geometrical combinations. The main utility of the antenna array is to obtain a set of positions for the antenna elements such its radiation pattern matches the desired one. The consideration of the design of antenna array as a specific problem has attracted the interests of the researchers particularly in the field of electromagnetic optimization. Earlier the approach was primarily based on applying numerical techniques but their inadequacies have forced the researchers to implement nature inspired metaheuristics [4]. Notable works have been reported in this regard, most importantly that of Panduro *et.al.* [5], [9-11] in which Differential Evolution [6], Genetic Algorithm [7] and Particle Swarm Optimization [8], have been applied to obtain the optimal design of scanned antenna arrays. Gurel and Ergul [10] applied GA for designing circular array antenna where every element is a log periodic antenna.

This paper has been organized into 6 sections. Section 2 describes the design problem in detail .Section 3 gives a detailed analysis of the Artificial Bee Colony algorithm. In Section 4 our proposed sNABC algorithm have been illustrated in detail. Section 5 specifies the experimental design section of the three possible cases and

B.K. Panigrahi et al. (Eds.): SEMCCO 2012, LNCS 7677, pp. 459–466, 2012.

discusses the results obtained. Lastly the paper is concluded in Section 6 with future research scope being specified.

2 Design Problem

Circular antenna array problems consist of N antenna elements distributed on a circle of radius **r.** Arrangement of the circular elements is given below

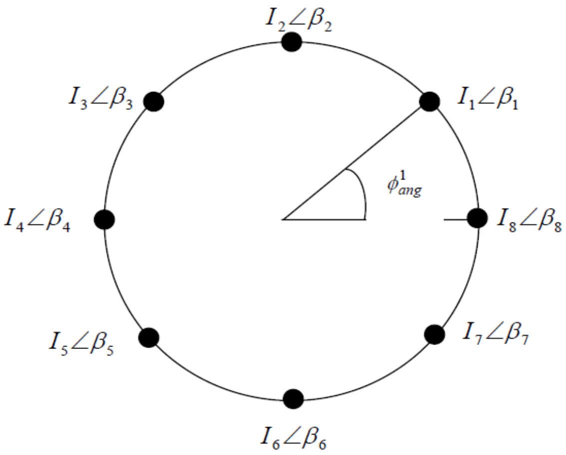

Fig. 1. Geometrical arrangement of circular antenna array

The Array Factor for the circular antenna array is calculated as :-

$$AF(\phi)=\sum_{n=1}^{N} I_n \exp\left[jkr\left(\cos\left(\phi-\phi^n_{ang}\right)-\cos\left(\phi_0-\phi^n_{ang}\right)\right)+\beta_n\right] \quad (1)$$

The array factor includes the following terms:-

(i) $\boldsymbol{I_n}$ is the current excitation associated with the nth element.
(ii) β_n is the phase excitation associated with the nth element.
(iii) $\emptyset$ is the angle of the incident plane wave.

In the array factor $\emptyset^n_{ang}$ is the angular position of the n^{th} element of the antenna array in x-y plane. It is calculated as:-

$$\emptyset^n_{avg} = 2\pi(\frac{n-1}{N}) \quad (2)$$

Here in case of circular antenna array, if k is the wavenumber and r is the radius of the circle defined by circular array then the relation ***kr=Nd*** holds. $\emptyset_0$ is the direction of maximum radiationThe current and the phase excitations of the antenna elements are varied so as to suppress the sidelobes,minimize the beamwidths and obtain null control in the directions as desired. Symmetrical excitation of the circular antenna array is considered in which the following relations hold

$$I_{n/2+1}\angle\beta_{n/2+1} = conj(I_1\angle\beta_1),$$
$$I_{n/2+2}\angle\beta_{n/2+2} = conj(I_1\angle\beta_1),.....$$
$$I_n\angle\beta_n = conj(I_{n/2}\angle\beta_{n/2}) \quad (3)$$

The objective function considered for the design problem is given as

$$OF = \left|AR(\varphi_{sll},\bar{I},\bar{\beta},\varphi_0)\right| / \left|AR(\varphi_{max},\bar{I},\bar{\beta},\varphi_0)\right| + 1/DIR(\varphi_0,\bar{I},\bar{\beta}) + |\varphi_0 - \varphi_{des}| + \sum_{k=1}^{num}\left|AR(\varphi_k,\bar{I},\bar{\beta},\varphi_0)\right| \quad (4)$$

The objective function can be minimised by separately considering the four components of the objective function.The first component attempts to minimize the level of the sidelobes, where $\emptyset_{sll}$ is the angle at which the maximum level of the side lobe is attained. The second component's task is to maximize the directivity of the array configuration. Directivity, which measures the directive gain of the antenna configuration has emerged as a key figure of merit for comparing various antenna patterns. The third component of the objective function attempts to drive the array configuration towards the desired maxima i.e. $\emptyset_{des}$. The fourth component deals with the nulls and penalizes the objective function if the null control of desired proportion is not achieved. ***nl*** is the number of null control directions and $\emptyset_k$ is the the kth null control direction.

3 Artificial Bee Colony Algorithm

The Artificial Bee Colony Algorithm intoduced by Karaboga [9], simulates the foraging behavior of the bees in solving intricate computational problems. The ABC algorithm allows the division of labour scheme, in which the entire bee colony is subdivided into three groups:-***employed bees,onlooker bees, scout bees.*** The basic steps of the ABC algorithm is elucidated below as:-

1 Foragers which are unemployed search the entire search space for potential food. Once food source is found it is promoted to an Employer Bee and carries local exploitation of the source, analyzing food sites based on their nectar content.
2 The employed bee chooses a fit source to load nectar, memorizes its position and returns to hive where, the employer unloads their nectar and in the process communicate the presence of a fit source to waiting onlookers via waggle dance.
3 Onlooker probabilistically selects one of the sources advertized to it,which is exploited by onlooker and again the steps performed by Employer bees are repeated.A food source is abandoned by a forager when it has exhausted its nectar content and it becomes a scout bee performing random walks in search space.

4 The *sNABC* Algorithm

The *sNABC* algorithm is a novel variant of the original Artificial Bee Colony Algorithm in which a novel mutation based on selective neighborhood has been introduced, which makes it superior in comparison with the original ABC algorithm. The main steps of the algorithm are detailed as below:-

(i) Initialization of Food Sources: The algorithm starts by initializing the food sources randomly in the search space .The food sources are initialized in the following manner:-

$$Food^{j}_{i} = Food^{j}_{min} + r. (Food^{j}_{max} - Food^{j}_{min}) \quad (5)$$

Food $^{j}_{max}$ and Food $^{j}_{min}$ are the upper and the lower limits of the $\boldsymbol{j^{th}}$ dimension of the $\boldsymbol{i^{th}}$ Food source, respectively. Here, **j Є [1, D]** for a D-dimensional problem, i belongs to **[1, FN]** for FN number of Food Sources and ***r*** is a random number uniformly generated in the range [0, 1] and later designated as ***rand***.

(ii) Employed Phase: Foragers are mainly responsible for carrying out the exploitation of the food sources. They are responsible for constant searching of the food sources. In order that the search process of the employed phase is balanced between exploration and exploitation. A perturbation scheme is applied taking into account randomly selected *k* parameters and perturbing them with a food source selected from a list of neighbouring members sorted in order of increasing Euclidean distance. The value of *k* is randomly varied but truncated to the range [1, *FN*/2].A random number is generated between 0 and 1 and if it is less than $\boldsymbol{Prb}_{nbd}$ (neighborhood selection probability) which is set to 0.5, then the perturbation is done with respect to k nearest neighbors of the current individual .Otherwise the farthest k number of neighbors of the current individuals are selected.

$$\vec{V_i} = \begin{cases} \overrightarrow{Food_i} + \emptyset_i \times (\overrightarrow{Food_i} - \overrightarrow{Food_k}^{N} & if\ rand \le Prb_{nbd} \\ \overrightarrow{Food_i} + \emptyset_i \times (\overrightarrow{Food_i} - \overrightarrow{Food_k}^{F} & Otherwise \end{cases} \quad (6)$$

The selection factor S determines the frequency of perturbation and ideally ranges from 0.3 to 0.6 for competitive results. We have chosen S to be 0.6 since it provides a good tradeoff between exploration and exploitation. Positional perturbation of the forager's position is enacted if a randomly generated number *rand*, in the range (0, 1), is less than the selection factor. The positional modification is as follows:-

$$\vec{U}_i^{\,j} = \begin{cases} \vec{V}_i^{\,j} & if\ rand \le S \\ \overrightarrow{Food_i}^{\,j} & Otherwise \end{cases} \quad (7)$$

This is followed by a greedy (fitness-based) selection by which the forager replaces it's currently employed Food Source by a fitter one recently discovered.The fitness is calculated using the following formula :-

$$fit(\overrightarrow{Food}) = \begin{cases} \dfrac{1}{1 + f(\overrightarrow{Food})} & f(\overrightarrow{Food}) \le 0 \\ 1 + |f(\overrightarrow{Food})| & f(\overrightarrow{Food}) > 0 \end{cases} \quad (8)$$

In the above mentioned formula, *fit(Food)* is the fitness value and *f(Food)* is the objective functional value of the food source respectively.

(iii) Onlooker Phase: In the Onlooker Phase, an waiting onlooker bee selects a food source by means of probabilistic selection method and determines an adjacent food source for exploitation to get a better solution. The onlooker bees select food sources if the randomly generated number in the range [0, 1] is less than $\boldsymbol{prob_i}$.

$$prob_i = 0.9 \times \frac{fit_i}{\max(fit_i)} + 0.1 \quad (9)$$

(iv) Scout Phase: After rejecting a food source, the concerned employee bee becomes a scout bee for searching a better food source in the the unexplored region. A trial counter is set which counts the number of cycles for which the food source does not improve upon its previous fitness value. As the trial counter reaches a predefined limit this food source is reinitialized.

5 Simulation Results

Three instances of the problem are considered here. The cases involve design of optimal array pattern possessing no null control, having null at 50^o and nulls at both 50^0 and 120^0. The results of our algorithm *sNABC* has been compared with the following algorithms using the parametric settings as inferred from literature.

(i) Differential evolution(DE). [6]
(ii) Particle swarm optimization(PSO). [8]
(iii) Artificial bee colony algorithm (ABC). [9-10]

The simulation of the benchmark suite have been performed on an Intel dual-core machine with 2 GB RAM 2.36 GHz speed using MATLAB 7.5. The results for each algorithm have been tabulated based on the data obtained from 25 trial runs. The mean and standard deviation for the objective function have been reported.

5.1 Parametric Settings

For the algorithm ***sNABC*** the parameters used are identical to that of classical ABC:

(i) Colony Size (CS)=100
(ii) limit=100
(iii) K (number of nearest individuals)=10

5.2 Results

Case I: 12 element array having no null control

Table 1. Results for the median of 25 trials(*Case I*)

Algorithm	SLL(dB)	Directivity(dB)
sNABC	**-20.16**	**11.53**
ABC	-18.47	11.27
PSO	-18.64	11.31
DE	-18.61	11.29

Table 2. Mean function values and standard deviations over 25 test runs (*Case I*)

Algorithm	Mean obj. func value	Standard deviation
sNABC	**0.1878**	**0.0032**
ABC	0.2154	0.0048
PSO	0.2241	0.0432
DE	0.2176	0.0311

The optimal array pattern thus obtained is shown in figure 2. The results clearly indicate that *sNABC* finds the best possible array pattern with minimized side lobes and greater directivity.

Case II: 12 element array having null at 50^0

Table 3. Results for the median of 25 trials (Case II)

Algorithm	SLL(dB)	Directivity(dB)	Array factor (dB) at 50^0
sNABC	**-18.87**	**11.34**	**-70.44**
ABC	-14.57	10.65	-68.45
PSO	-17.83	10.87	-69.43
DE	-18.13	11.08	-47.44

Table 4. Mean function values and standard deviations over 25 test runs (*Case II*)

Algorithm	Mean obj.func value	Standard deviation
sNABC	**0.2194**	**0.0093**
ABC	0.2791	0.0194
PSO	0.2243	0.0487
DE	0.2178	0.0318

From the obtained results it is visible that sNABC clearly outperforms other competing algorithms by suppressing the array pattern to -70.44 dB, at the required null direction of 50^0.The array pattern thus obtained is shown in figure 3.

Case III*:* 12 element array having null at 50^0 and 120^0

Table 5. Results for the median of 25 test runs (Case III)

Algorithm	SLL(dB)	Directivity(dB)	Array factor (dB) at 50^0	Array factor (dB) at 120^o
sNABC	**-18.76**	**11.21**	**-68.13**	**-70.35**
ABC	-13.73	10.68	-41.42	-52.17
PSO	-11.42	10.34	-42.92	-54.15
DE	-18.05	10.96	-52.53	-48.47

Table 6. Mean function values and standard deviations over 25 test runs (Case III)

Algorithm	Mean	Standard deviation
sNABC	**0.2017**	**0.0089**
ABC	0.2558	0.0115
PSO	0.3865	0.0287
DE	0.2211	0.0345

The results given above clearly indicate that our algorithm has provided the best possible minimized sidelobe levels in both 50^o and 120^0 directions. The corresponding array pattern is thus shown in figure 4.

5.3 Computational Effort and Complexity

To do away with the platform hardware specifications, we state the computational time as the ration of the time taken by sNABC to that of basic ABC to optimize the problem. The average value was found out to be 1.021 which is suitable for real world applications. Similar ratios of 1.011 and 1.042 were obtained with respect to PSO and DE. Due to computation of the distance matrix the worst case complexity is $O(\mathrm{CS}^2)$. But due to nature of the benchmark problem the difference in time is not pronounced. Usually the average case complexity is $O(\mathrm{CS}\ log\ \mathrm{CS})$ owing to symmetric nature of distance matrix which saves repeatative calculation.

6 Conclusion

In the current scenario, the design of such an optimal antenna array pattern, where side lobes have to be minimized apart from maximizing the directivity is quite challenging task. In this paper we have focused on the application of a novel variant of the Artificial Bee Colony Algorithm, where the employed phase is modified by incorporating a trade-off scheme between exploitation and exploration, which has proved to be superior in comparison with the other contender algorithms. The optimization task considered here is in the form of a cost function that considers the average levels of the side lobes, null control, and circumference of the array. On all the three cases considered above, our algorithm has significantly outperformed the competing state of art algorithms regarding all the possible criteria considered (SLL, directivity etc).Future research works will stress on exploring the possibility of application of our algorithm in case of other array geometries. Further research work is possible by considering the components of the cost function as a multi objective problem but suitable problem specific knowledge is required in order to point out the best solution from the Pareto-optimal set.

References

[1] Godara, L.C.: Handbook of Antennas in Wireless Communications, Boca Raton, FL (2002)

[2] Balanis, C.A.: Antenna Theory-Analysis and Design. Wiley, India (2005)

[3] Chandran, S.: Adaptive Antenna Arrays:Trends and Applications. Springer (2004)

[4] Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence: From Natural to Artificial Systems. Oxford University Press, New York (1999)

[5] Panduro, M.A., Brizuela, C.A., Balderas, L.I., Acosta, D.A.: A comparison of genetic algorithms, particle swarm optimization and the differential evolution method for the design of scannable circular antenna arrays. Progress In Electromagnetics Research B 1, 171–186 (2009)

[6] Storn, R., Price, K.: Differential Evolution- A Simple Efficient heuristic Strategy for Global Optimization over Continuous Spaces. Journal of Global Optimization 11, 341–359 (1997)

[7] Goldberg, D.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Reading (1989)

[8] Eberhart, R.C., Kennedy, J.: A new optimizer using particle swarm theory. In: Proc. 6th Int. Symp. Micromach. Human Sci., pp. 39–43 (1995)

[9] Karaboga, D., Basturk, B.: A powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm. Journal of Global Optimization 39(3) (2007)

[10] Karaboga, D., Basturk, B.: On the performance of artificial bee colony (ABC) algorithm. Applied Soft Computing 8, 687–697 (2008)

[11] Mandal, A., Zafar, H., Das, S., Vasilakos, A.V.: Efficient Circular Array Synthesis with a mementic Differential Evolution Algorithm. Progress in Electromagnetics Research, 367–385 (2012)
[12] Gurel, L., Ergul, O.: Design and simulation of circular arrays of trapezoidal-tooth log-periodic antennas via genetic optimization. Progress In Electromagnetics Research (PIER) 85, 243–260 (2008)

Supplementary Attachment: Optimal Array Pattern for the cases discussed above

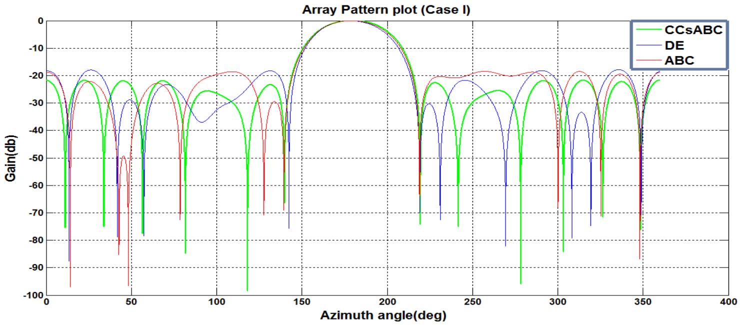

Fig. 2. Antenna array pattern for 12 element array having no null control

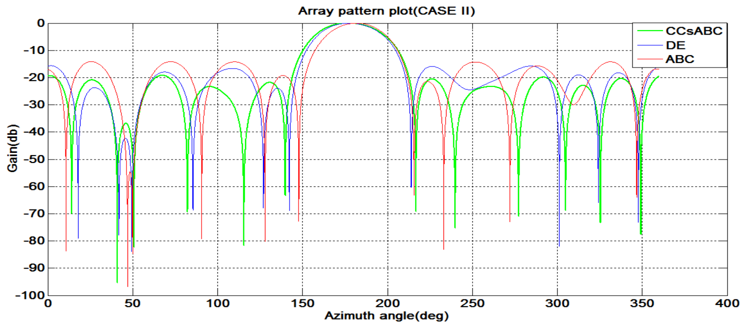

Fig. 3. Antenna array pattern for 12 element array having null at 50^0

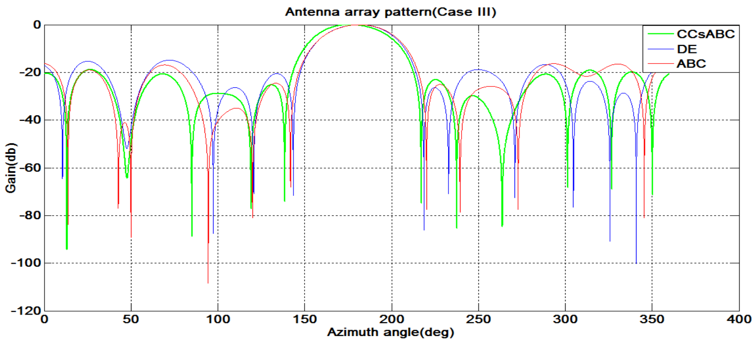

Fig. 4. Antenna array pattern for 12 element array having null at 50^0 and 120^0