Concentric Circular Antenna Array Synthesis Using a Differential Invasive Weed Optimization Algorithm

Siddharth Pal[1],Aniruddha Basak[1], Swagatam Das[1], Ajith Abraham[2] and Ivan Zelinka[3]

[1] Dept. of Electronics and Telecommunication Engg ,
Jadavpur University, Kolkata 700 032, India
[2] Machine Intelligence Resarch Labs (MIR Labs), USA
[3] FEI, VSB-TU Ostrava, Czech Republic
E-mail: ajith.abraham@ieee.org, ivan.zelinka@vsb.cz

Abstract: Concentric Circular Antenna Array (CCAA) has several interesting features that makes it indispensable in mobile and communication applications. Here we have considered a uniform arrangement of elements where the inter-element spacing is kept half a wavelength. The main aim is to reduce the sidelobe levels and the primary lobe beamwidth as much as possible. Central to our design is a hybridization of two prominent metaheuristics of current interest namely the Invasive Weed Optimization (IWO) and Differential Evolution (DE). The results of the DIWO algorithm have been shown to perform better then other state-of-the-art metaheuristics like the Particle Swarm Optimization (PSO), DE and IWO.

Keywords: Antenna array, concentric circular arrays, invasive weed optimization, differential evolution, metaheuristics.

1. Introduction

A circular array is a planar array with elements lying on a circle. If we have several circular arrays with different radii then the antenna structure is called a concentric circular antenna array (CCAA). CCAA provides great flexibility in array pattern synthesis and design both in narrowband and broadband applications. It is also favoured in direction of arrival (DOA) applications since it provides almost invariant azimuth angle coverage.

Lot of research has gone into optimizing antenna structures such that the radiation pattern has low sidelobe level. Different types of antenna array CCAA [1-3] have become very popular in mobile and wireless communications. This very fact has driven researchers to optimize the CCAA design [13-14]. Genetic Algorithm (GA) has been used in [6] to optimize the element placement in CCAA. Particle Swarm Optimization (PSO) [5] has been applied for the optimized synthesis of thinned CCAA. Efficient sidelobe reduction techniques have been discussed in [4]. In this paper we will try to optimize the element excitations of a uniform CCAA (UCCAA) such that the sidelobe levels are minimized and the width of the primary lobe is as narrow as possible.

In this article, we propose to use an improved variant of one recently developed and very powerful metaheuristic algorithm, called the Invasive Weed Optimization (IWO) [7], for designing CCAA with optimized performance with respect to SLL and primary lobe beamwidth in a scanning range of $[0^o, 180^o]$. Since its inception, IWO has found successful applications in many electromagnetic optimization problems like design of E-shaped MIMO Antenna [8], design of compact U-array MIMO antenna design [9], aperiodic thinned array antennas [10], time modulated antenna array synthesis [11] etc. In this paper, a modified IWO, whose standard deviation is modulated with individual fitness, has been hybridized with DE [12] and the resulting hybrid algorithm used (for the first time, to the best of our knowledge) to optimize the amplitude excitation of a concentric circular antenna array to produce a radiation pattern with optimal performances. Two numerical instantiations of the design problem have been used to illustrate the application of the algorithm. Comparison with the results obtained with other best known real-parameter optimizers like PSO, DE and IWO reflect the superiority of the hybrid Differential IWO (DIWO) in a statistically meaningful way

The rest of the paper is organized in the following way. A formulation of the array pattern synthesis as an optimization task has been discussed in Section 2. Section 3 provides a comprehensive overview of the DIWO algorithm. Experimental results have been presented in Section 4. Section 5 finally concludes the paper and unfolds a few future research issues.

2. Formulation of the Design Problem

UCCAA consists of antenna elements arranged in multiple concentric circular rings which differ in radius and number of elements. This leads to different radiation patterns for different configurations and parameters of UCCAA [4].

Figure 1 shows the configuration of multiple concentric circular arrays in XY plane which consists of M concentric circular rings. The m^{th} ring has a radius r_m and N_m number of isotropic elements where m=1,2,...,M. Since here we are considering a uniform CCAA the elements are considered to be equally spaced along a common circle.

The far field pattern in free space is given by:

$$E(\theta,\phi)=\sum_{m=1}^{M}\sum_{n=1}^{N_m} I_{mn} e^{j2\pi r_m \sin\theta\cos(\phi-\phi_{mn})} \quad (1)$$

Normalized power pattern $P(\theta,\phi)$ in dB can be expressed as,

$$P(\theta,\phi)=10\log_{10}\left[\frac{|E(\theta,\phi)|}{|E(\theta,\phi)|_{\max}}\right]^2 \quad (2)$$

where,

r_m =radius of m^{th} ring = $N_m d_m/2\pi$

d_m =interelement arc spacing of m^{th} circle

$\phi_{mn}=2\pi n/N_m$ = angular position of n^{th} element of m^{th} ring

ϕ = azimuth angle, k = wave number= $2\pi/\lambda$

I_{mn} = excitation amplitude of n^{th} element of m^{th} ring

For a $\lambda/2$ UCCAA we will have element increment $n_i=N_{m+1}-N_m=6$, normalized ring-radial separation $d_m=0.4775$ and normalized element-arc separation $a_m=0.5$. The total number of elements in the array will be $N_t=MN_1+\sum_{k=1}^{M-1}6k$. Thus in a $\lambda/2$ UCCAA the important design parameters determining the array pattern are- No. of rings (M) and No. of elements in the 1st ring(N_1). Because we have constant element increment of 6 to maintain nearest half wave radial separation, N_1 represents an important parameter in design.

3. Proposed Hybrid Algorithm (DIWO)

A. The Invasive Weed Optimization (IWO) Algorithm

Invasive Weed Optimization (IWO) is a meta-heuristic algorithm that mimics the colonizing behavior of weeds. The basic characteristic of a weed is that it grows its population entirely or predominantly in a geographically specified area which can be substantially large or small. There are four steps of the algorithm as described below:

1) Initialization: A certain number of weeds are randomly spread over the entire search space (*D* dimensional). This initial population of each generation will be termed as $\boldsymbol{X}=\{\vec{X}_1,\vec{X}_2,...,\vec{X}_m\}$.

2) Reproduction: Each member of the population *X* is allowed to produce seeds within a specified region centered at its own position. The number of seeds produced by $\vec{X}_i, i\in\{1,2,...,m\}$ depends on its relative fitness in the population with respect to the best and worst fitness. The number of seeds produced any weed varies linearly from *min_seed* to *max_seed* with *min_seed* for the worst member and *max_seed* for the best member in the population. The seed production profile is shown in Figure 2.

3) Spatial distribution: The generated seeds are being randomly distributed over the d-dimensional search space by normally distributed random numbers with zero mean and variance σ^2.

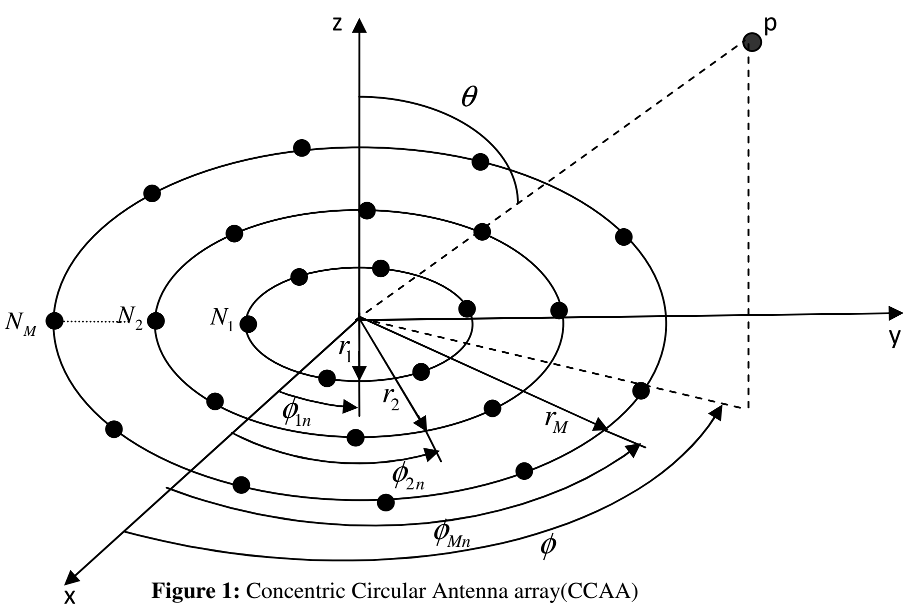

Figure 1: Concentric Circular Antenna array(CCAA)

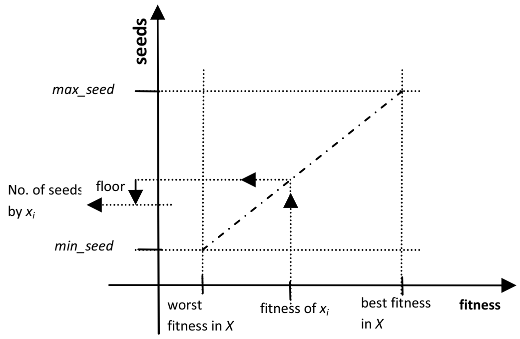

Figure 2: Seed production procedure in the population

This step ensures that the produced seeds will be generated around the parent weed, leading to a local search around its parent. However, the standard deviation σ is made to decrease over the generations so that the algorithm gradually moves from exploration to exploitation with increasing generations. If $\sigma_{\max}$ and $\sigma_{\min}$ are the maximum and minimum standard deviation, then the standard deviation in particular generation (or iteration) is given by,

$$\sigma_t = \sigma_{\min} + \left(\frac{iter_{\max} - iter}{iter_{\max}} \right)^{n\_m\_i} \left(\sigma_{\max} - \sigma_{\min} \right) \qquad (3)$$

This step ensures that the probability of dropping a seed in a distant area decreases nonlinearly with iterations, which results in grouping fitter plants and elimination of inappropriate plants.

4) Competitive Exclusion: If a plant leaves no offspring then it would go extinct, otherwise they would take over the world. Thus, there is a need of some kind of competition between plants to limit the maximum number of plants in a population. Initially, the plants in a colony will reproduce fast and all the produced weeds will be included in the colony, until the number of plants reaches a maximum value of pop_max. From then on, only the fittest plants, among the existing ones and the reproduced ones; are taken in the colony and the steps 1 to 4 are repeated until the maximum number of iterations (or function evaluations) have been reached. So, in every generation the population size must be less than or equal to pop_max. This method is known as competitive exclusion and is a selection procedure of IWO.

B. Differential Evolution

DE starts with a population of *NP* *D*-dimensional parameter vectors representing the candidate solutions. For each decision variable of the problem, there may be a user-specified range within which value of the variable should lie for more accurate search results at less computational cost. The initial population (at iteration = 0) should cover the entire search space as much as possible by uniformly randomizing individuals within the search space constrained by the prescribed minimum and maximum

bounds: $\vec{X}_{\min} = \{x_{1,\min}, x_{2,\min}, ..., x_{D,\min}\}$ and

$\vec{X}_{\max} = \{x_{1,\max}, x_{2,\max}, ..., x_{D,\max}\}$.

a) Mutation: After initialization, DE creates a *donor* vector $\vec{V}_{i,t}$ corresponding to each population member or *target* vector $\vec{X}_{i,t}$ in the current generation through mutation. Five most frequently referred mutation strategies implemented in the public-domain DE codes available online *at http://www.icsi.berkeley.edu/~storn/code.html* are listed below:

"DE/rand/1": $\vec{Y}_{i,t} = \vec{X}_{\alpha_1^i,t} + F \cdot (\vec{X}_{\alpha_2^i,t} - \vec{X}_{\alpha_3^i,t}).$ (4)

"DE/best/1": $\vec{Y}_{i,t} = \vec{X}_{best,t} + F \cdot (\vec{X}_{\alpha_1^i,t} - \vec{X}_{\alpha_2^i,t}).$ (5)

"DE/rand/2"

$$\vec{Y}_{i,t} = \vec{X}_{\alpha_1^i,t} + F \cdot (\vec{X}_{\alpha_2^i,t} - \vec{X}_{\alpha_3^i,t}) + F \cdot (\vec{X}_{\alpha_4^i,t} - \vec{X}_{\alpha_5^i,t}). \qquad (6)$$

The indices $\alpha_1^i, \alpha_2^i, \alpha_3^i, \alpha_4^i$, and α_5^i are mutually exclusive integers randomly chosen from the range [1, *NP*], and all are different from the base index *i*. These indices are randomly generated once for each donor vector. The scaling factor *F* is a positive control parameter for scaling the difference vectors. $\vec{X}_{best,t}$ is the best individual vector with the best fitness (i.e. lowest objective function value for minimization problem) in the population at iteration *t*.

b) Crossover: To enhance the potential diversity of the population, a crossover operation comes into play after generating the donor vector through mutation. The donor vector exchanges its components with the target vector $\vec{X}_{i,t}$ under this operation to form the *trial* vector $\vec{U}_{i,t}$. The DE family of algorithms can use two kinds of crossover methods - *exponential* (or two-point modulo) and *binomial* (or uniform). In this article we focus on the widely used binomial crossover that is performed on each of the *D* variables. The scheme may be outlined as:

$$u_{j,i,t} = \begin{cases} y_{j,i,t}, & \text{if } rand_{i,j}[0,1) \le Cr \text{ or } j = j_{rand} \\ x_{j,i,t}, & \text{otherwise,} \end{cases} \qquad (7)$$

where, as before, $rand_{i,j}[0,1)$ is a uniformly distributed random number, which is called anew for each *j*-th component of the *i*-th parameter vector. $j_{rand} \in [1,2,....,D]$ is a randomly chosen index, which ensures that $\vec{U}_{i,t}$ gets at least one component from $\vec{V}_{i,t}$.

c) Selection: To keep the population size constant over subsequent generations, the next step of the algorithm calls for *selection*. This operation determines which one of the target and the trial vector survives to the next generation i.e. at $t = t+1$. The selection operation may be outlined as:

$$\begin{aligned}\vec{X}_{i,t+1} &= \vec{U}_{i,t}, \text{ if} f\left(\vec{U}_{i,t}\right) \le f\left(\vec{X}_{i,t}\right) \\ &= \vec{X}_{i,t}, \text{ if} f\left(\vec{U}_{i,t}\right) > f\left(\vec{X}_{i,t}\right)\end{aligned} \quad (8)$$

where $f(\vec{X})$ is the function to be minimized. So if the new trial vector yields a lower value of the objective function, it replaces the corresponding target vector in the next generation; otherwise the target is retained in the population. Hence the population either gets better (w .r. t. the minimization of the objective function) or remains constant, but never deteriorates.

C. Modifications of IWO algorithm

Modulation of Standard Deviation with Fitness*:* In IWO, creation of the new solutions is mainly dependent on the standard deviation profile. So the variation of σ_t with iterations cast a huge effect on the performance of the algorithm. One problem with original IWO is σ_t is same for all the weeds in the population in a particular iteration. So the spread of all the seeds around its parent weed is same for all the parent weeds in the population. So the weeds, far from an optimum may not get much opportunity to be attracted towards that point; and some weeds nearer to the global optimum often get trapped in the nearby local optima.

We introduced a novel approach of modulating the standard deviation of the weeds according to their fitness values. So in one generation the values of $\sigma_{i,t}$ ($i \in$ [1, *NP*]), with which the produced seeds are dispersed, are not same for all the weeds in the population rather it is made to vary for all the weeds. Considering σ_t of eqn. (3) as an average value that we assign to the median element of the sorted population (sorted according to fitness values) of the weeds at the *t*-th iteration and consider its fitness value to be $F_{median,t}$. The weeds having objective function value smaller than $F_{median,t}$ (i.e. greater fitness value for minimization) will have $\sigma_{i,t}$ less than σ_t and the others with higher objective function value will have higher $\sigma_{i,t}$ compared to σ_t. This is implemented as,

$$\sigma_{i,t} = \begin{cases} \left(1 + 0.5 * \dfrac{f\left(\vec{X}_{i,t}\right) - F_{median,t}}{F_{worst,t} - F_{median,t}}\right).\sigma_t & \text{if} f\left(\vec{X}_{i,t}\right) \ge F_{median,t}, \\ \left(1 - 0.5 * \dfrac{F_{median,t} - f\left(\vec{X}_{i,t}\right)}{F_{median,t} - F_{best,t}}\right).\sigma_t & \text{if} f\left(\vec{X}_{i,t}\right) < F_{median,t}, \end{cases}$$

(9)

where $F_{worst,iter}$ and $F_{best,iter}$ denote the worst and best fitness in the population respectively at a particular iteration.

D. Hybridization of the M-IWO algorithm with DE (DIWO)

To enhance the explorative power of M-IWO algorithm we incorporate the mutation and crossover schemes of the DE algorithm to produce new seeds from the weeds created by M-IWO.

We have incorporated the features of "DE/best/1/bin" variant with our M-IWO algorithm. Here we describe our whole algorithm:

A. Initialization: A certain number of weeds are randomly spread over the entire search space (D dimensional). This population at the t-th iteration will be termed as $X_t = \left\{\vec{X}_{1,t}, \vec{X}_{2,t}, ..., \vec{X}_{m,t}\right\}$.

B. Seeds production: Each member of the population $\boldsymbol{X}_t$ is allowed to produce seeds is allowed to generate seeds according to classical IWO algorithm. Number of seeds to be generated by $\vec{X}_{i,t}$, $i \in [1, m]$ is decided by,

$$s_i = floor\left(\frac{F_{\max} - f\left(\vec{X}_i\right)}{F_{\max} - F_{\min}}.s_{\max}\right), \quad (10)$$

where s_{max} is the same as *max_seed* and *min_seed* is taken to be zero. The standard deviation, with which the seeds of each weed will be distributed in the search space, $\sigma_{i,t}$ is calculated by using equation (9).

So for each $\vec{X}_{i,t}$ we get s_i seeds, generated using standard deviation $\sigma_{i,t}$. These seeds along with their parent weeds create a new intermediate population $\boldsymbol{V}_t$ *of* size *n*. Then the mutation and crossover operation is applied on each member $\vec{V}_{i,t}$, $i \in [1, n]$ to create another population $\boldsymbol{Z}_t$ as,

$$\vec{Y}_{i,t} = \vec{V}_{\alpha_1^i,t} + F(\vec{V}_{\alpha_2^i,t} - \vec{V}_{\alpha_3^i,t}) \ , \quad (11)$$

and subsequently

$$z_{j,i,t} = y_{j,i,t} \quad , \text{if } \left(rand(0,1] \le p_{Cr}\right) \cup \left(j = j_{rand}\right)$$
$$= v_{j,i,t} \quad , \text{if } rand(0,1] > p_{Cr}, \quad j \in [1,2,..,D],$$

(12)

where the indices α_1^i, α_2^i, and α_3^i are mutually exclusive integers randomly picked up from the range [1, n] for each index i, p_{Cr} is equivalent to the crossover rate Cr in DE and lies between 0 and 1, and like in (12), $j_{rand} \in [1,2,....,D]$ is a randomly chosen index, which ensures that $\vec{Z}_{i,t}$ gets at least one component from $\vec{Y}_{i,t}$.

C. Partial Selection using DE: Each element in $\boldsymbol{Z}$, $\vec{Z}_i$, is compared with its corresponding parent $\vec{V}_i$ and the fitter solution is chosen to survive. Using this selection procedure a population of n weeds is created and this is named as $\boldsymbol{P}$. So each weed in $\boldsymbol{P}$ can be described by,

$$\vec{P}_i = \begin{cases} \vec{Z}_i \ , & \text{if } f\left(\vec{Z}_i\right) \le f\left(\vec{V}_i\right) \\ \vec{V}_i \ , & \text{if } f\left(\vec{Z}_i\right) > f\left(\vec{V}_i\right) \end{cases} \quad (13)$$

D. Final Selection of weeds: The members of $\boldsymbol{P}$ are arranged according to increasing fitness (from best to worst) and the only the first m weeds are allowed to pass to the next generation.

Steps B, C, D are repeated until the maximum function evaluations are reached. From the above description of the algorithm it is clear that our algorithm is a tightly coupled hybrid algorithm and here from we will call this as DIWO algorithm.

4. Simulation Results

In our problem the optimization parameters are the element excitations. The objective is to minimize the maximum sidelobe level (MSLL) and also minimize the beamwidth between first nulls (BWFN). We use the weighted sum approach to cater to both the objectives.

$$F = w_1 MSLL + w_2 BWFN \quad (14)$$

F given in equation (14) is the objective function. In this section we compare our algorithm with three standard single objective algorithms like PSO, DE and IWO. We consider two test cases for comparative purposes. In our simulations we choose $w_1 = 1$ and $w_2 = 1$.

Case 1: N1=8, M=9

Table 1: Design Objectives achieved (Case 1)

	DIWO	DE	PSO	IWO
BW(Degrees)	**16.15**	16.25	17.45	16.25
MSLL	**-26.01**	-23.40	-23.45	-24.63
Fitness	**-9.86**	-7.15	-6.00	-8.38

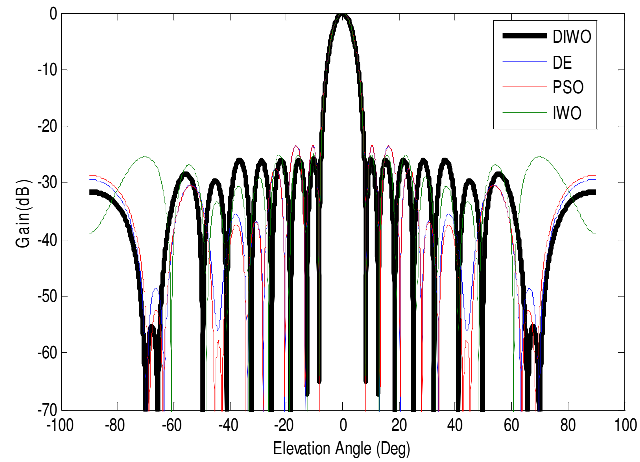

Figure 3. Array pattern for $N_1 = 8$ and $M = 9$

Table 1 shows that DIWO obtains better values for both BWFN and MSLL. This is also supported by Figure 3 which shows that better array pattern is achieved with DIWO. The MSLL of DIWO is clearly at a lower level than the patterns produced by the competitor algorithms.

Case 2: N1=5, M=11

From Table 2 we notice that lower fitness function values are achieved for DIWO. Also we observe that DIWO achieved best values of beamwidth and MSLL among all the competitor algorithms. Figure 4 shows the array pattern obtained by the various algorithms.

Table 2: Design achieved (Case 2)

	DIWO	DE	PSO	IWO
BW(Degrees)	**13.81**	14.28	13.92	13.84
MSLL	**-25.12**	-23.81	-21.08	-23.37
Fitness	**-11.31**	-9.53	-7.16	-9.53

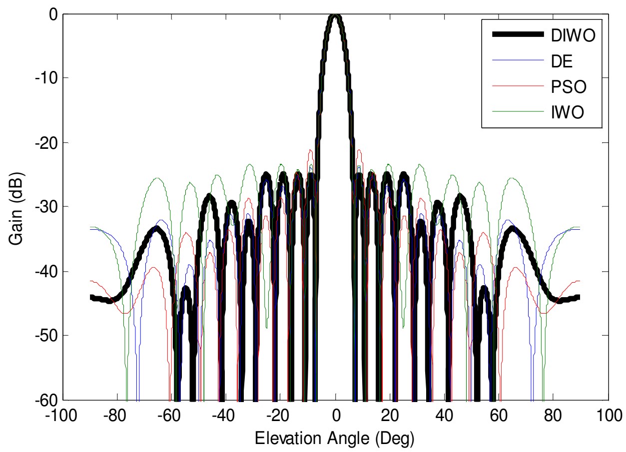

Figure 4: Array pattern for $N_1 = 5$ and $M = 11$

5. Conclusions

This paper illustrates the use of a hybrid evolutionary algorithm DIWO in the synthesis of concentric circular antenna arrays for the purpose of suppressed sidelobes and thin primary lobe. DIWO successfully outperformed three other state-of-the-art optimization techniques like PSO,DE and IWO making the hybridization meaningful and necessary. Future research works may focus on application of the hybrid algorithm to optimizing other antenna structures. As a metaheuristic algorithm DIWO is bound to attract researchers from the electromagnetics and antenna community.

References

[1] P. Fletcher, P. Darwood, "Beamforming for circular and semicircular array antennas for low-cost wireless lan data communications systems", *IEE Proc. Microwaves, Antennas and Propagation*, Vol.145, No.2, 153–158, Apr.1998.

[2] L. Bogdan, C. Comsa, "Analysis of circular arrays as smart antennas for cellular networks," *IEE Proc. Microwaves, Antennas and Propagation*, Vol.145, No.2, 153–158, Apr. 1998.

[3] S.C. Chan, C.K.S.Pun, "On the design of digital broadband beamformer for uniform circular array with frequency invariant characteristics," *Proc. IEEE ISCAS'02*, Vol.1, I-693–I-696, Arizona, USA, May 2002.

[4] M. Dessouky, H. Sharshar,Y. Albagory, " Efficient Sidelobe Reduction Technique for small-sized concentric circular arrays," *Progress In Electromagnetics Research*, PIER 65, 187–200,2006.

[5] N. Pathak, G.K. Mahanti, S.K. Singh, J.K. Mishra, A. Chakraborty, "Sythesis of thinned planar circular array antenna using modified particle swarm optimization," *Progress In Electromagnetics Research Letters*, Vol. 12, 87, 2009.

[6] R.L. Haupt, " Optimized Element pacing for Low Sidelobe Concentric Ring Arrays," *IEEE Transactions on Antenna and Propagation*, Vol.56, No. 1, 2008.

[7] A. R. Mehrabian, C. Lucas, "A novel numerical optimization algorithm inspired from weed colonization," *Ecological Informatics*, vol. 1, pp. 355–366, 2006.

[8] A. R. Mallahzadeh, S. Es'haghi, A. Alipour, "Design of an E-Shaped Mimo Antenna Using IWO Algorithm for Wireless Application at 5.8 GHz", *Progress In Electromagnetics Research,* PIER 90, 187 - 203, 2009.

[9] A. R. Mallahzadeh, S. Es'haghi, H. R. Hassani, "Compact U-array MIMO antenna designs using IWO algorithm," *International Journal of RF and Microwave Computer-Aided Engineering*, Wiley-InterSscience, DOI: 10.1002/mmce.20379, Jul, 2009.

[10] S. Karimkashi, A. A. Kishk, "Invasive weed optimization and its features in electromagnetics," *IEEE Transactions on Antennas and Propagation*, Vol. 58, Issue 4, pp.1269 – 1278, April, 2010.

[11] A. Basak, S. Pal, S. Das, A. Abraham and V. Snasel, A Modified Invasive Weed Optimization Algorithm for Time-Modulated Linear Antenna Array Synthesis, IEEE World Congress on Computational Intelligence (WCCI 2010), Barcelona, IEEE, USA, ISBN 978-1-4244-8126-2, pp. 3137-3144, 2010.

[12] K. Price, R. Storn, and J. Lampinen, *Differential Evolution - A Practical Approach to Global Optimization*, Springer, Berlin, 2005.

[13] A. Basak, S. Pal, S. Das and A. Abraham, Circular Antenna Array Synthesis with a Differential Invasive Weed Optimization Algorithm, Tenth International Conference on Hybrid Intelligent Systems, USA, IEEE, ISBN 978-1-4244-7364-9, pp. 153-158, 2010.

[14] S. Pal, A. Basak, S. Das and A. Abraham, Linear Antenna Array Synthesis with Invasive Weed Optimization Algorithm, International Conference on Softcomputing and Pattern Recognition (SoCPaR 2009), IEEE Computer Society, USA, ISBN 978-0-7695-3879-2, pp. 161-166, 2009.