International Journal of Microwave and Wireless Technologies, 2014, 6(2), 181–194. © Cambridge University Press and the European Microwave Association, 2013
doi:10.1017/S175907871300086X

RESEARCH PAPER

Optimized hyper beamforming of receiving linear antenna arrays using Firefly algorithm

GOPI RAM[1], DURBADAL MANDAL[1], RAJIB KAR[1] AND SAKTI PRASAD GHOSHAL[2]

In this paper, an optimized hyper beamforming method is presented based on a hyper beam exponent parameter for receiving linear antenna arrays using a new meta-heuristic search method based on the Firefly algorithm (FFA). A hyper beam is derived from the sum and difference beam patterns of the array, each raised to the power of a hyper beam exponent parameter. As compared to the conventional hyper beamforming of the linear antenna array, FFA applied to the hyper beam of the same array can achieve much more reduction in sidelobe level (SLL) and improved first null beam width (FNBW), keeping the same value of the hyper beam exponent. As compared to the uniformly excited linear antenna array with inter-element spacing of λ/2, conventional non-optimized hyper beamforming and optimal hyper beamforming of the same obtained by real-coded genetic algorithm, particle swarm optimization and Differential evolution, FFA applied to the hyper beam of the same array can achieve much greater reduction in SLL and same or less FNBW, keeping the same value of the hyper beam exponent parameter. The whole experiment has been performed for 10-, 14-, and 20-element linear antenna arrays.

Keywords: Hyper beam, Linear antenna arrays, RGA, PSO, DE, FFA, SLL, FNBW

Received 15 March 2013; Revised 2 September 2013; first published online 14 October 2013

I. INTRODUCTION

Beamforming is a signal processing technique used to control the directionality of the transmission and reception of the radio signals [1]. This is achieved by distributing the elements of the array in such a way that signals at a particular angle experience constructive interference, whereas others experience destructive interference. Beamforming can be used at both transmitting and receiving ends in order to achieve spatial selectivity. Hyper beamforming [2] refers to the spatial processing algorithm used to focus an array of spatially distributed elements (called sensors) to increase the signal-to-interference plus noise ratio (SINR) at the receiver. This beamforming processing improves significantly the gain of the wireless link over a conventional technology, thereby increasing range, rate, and penetration [3–5]. It has found numerous applications in radar, sonar, seismology, wireless communication, radio astronomy, acoustics, and biomedicine [6]. It is generally classified as either conventional (switched and fixed) beamforming or adaptive beamforming. Switched beamforming system [7, 8] is a system that can choose one pattern from many predefined patterns in order to enhance the received signals. Fixed beamforming uses a fixed set of weights and time delays (or phasing) to combine the signals received from the sensors in the array, primarily using only information about the locations of the sensors in space and the wave direction of interest [2]. Adaptive beamforming or phased array is based on the desired signal maximization mode and interference signal minimization mode [9–11]. It is able to place the desired signal at the maximum of the main lobe. Hyper beamforming/any other beamforming offers high detection performance such as beamwidth, target-bearing estimation and reduces false alarm, sidelobe suppression. A new optimized hyper beamforming technique is presented in this paper, and the Firefly algorithm (FFA) is applied to obtain optimal hyper beam patterns of linear antenna arrays.

The classical gradient-based optimization methods are not suitable for optimal design of hyper beamforming of linear antenna arrays because of the following reasons: (i) highly sensitive to the starting points when the number of solution variables and hence the size of the solution space increase, (ii) frequent convergence to local optimum solution or divergence or revisiting the same suboptimal solution, (iii) requirement of continuous and differentiable objective function, (iv) requirement of the piecewise linear cost approximation (linear programming), and (v) problem of convergence and algorithm complexity (non-linear programming). Hence, evolutionary methods have been employed for the optimal design of hyper beamforming of linear antenna arrays with better parameter control.

Different evolutionary optimization algorithms such as simulated annealing algorithm [12], genetic algorithm (GA) [13–17] etc. have been widely used for the synthesis of design methods capable of satisfying the constraints. When considering the global optimization methods for antenna

[1]Department of Electronics and Communication Engineering, National Institute of Technology, Durgapur, India
[2]Department of Electrical Engineering, National Institute of Technology, Durgapur, India
Corresponding author:
D. Mandal
Email: durbadal.bittu@gmail.com

arrays design, GA seems to be the promising one. Although standard GA [here referred to as real-coded genetic algorithm (RGA)] has a good performance for finding the promising regions of the search space, however, finally, the RGA is prone to revisiting the same suboptimal solutions.

Particle swarm optimization (PSO) is an evolutionary algorithm developed by Kennedy and Eberhart [18]. PSO is simple to implement and its convergence may be controlled via a few parameters [19–27]. The limitations of the conventional PSO are that it may be influenced by premature convergence and stagnation problem.

The Differential evolution (DE) algorithm was first introduced by Storn and Price in 1995 [28]. Like the RGA, it is a randomized stochastic search technique enriched with the operations of crossover, mutation and selection [29–39] and prone to premature convergence and stagnation. Hence, to enhance the performance of the optimization algorithms in global search (exploration stage) as well as local search (exploitation stage), an alternative technique such as FFA [40–44] has been suggested for the optimization of hyper beamforming in this paper.

The rest of the paper is arranged as follows. In Section II, the design equations of hyper beamforming of linear antenna array are formulated. Section III briefly discusses the evolutionary techniques RGA, PSO, DE, and FFA employed for the designs of linear antenna arrays. Section IV describes the simulation results obtained by using the techniques. Finally, Section V concludes the paper.

II. DESIGN EQUATIONS

In hyper beamforming for linear antenna array the inter-element spacing of $\lambda/2$ in either direction is considered. The sum beam can be created by summation of the absolute values of complex left and right half beams, as shown in Fig. 1. The difference beam is the absolute magnitude of the difference of complex right beam half beam and left half beam signals. Furthermore, the difference beam has a minimum in the direction of the sum beam at zero degree as shown in Fig. 2. The resulting hyper beam is obtained by subtraction of the sum and the difference beams, each raised to the power of the exponent u.

Consider a broadside linear array of N equally spaced isotropic elements as shown in Fig. 3. The array is symmetric in both geometry and excitation with respect to the array center [2].

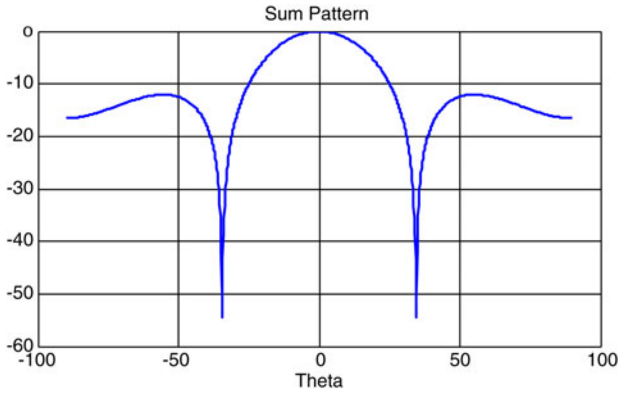

Fig. 1. Sum beam pattern for the 10-element linear array for $u = 1$.

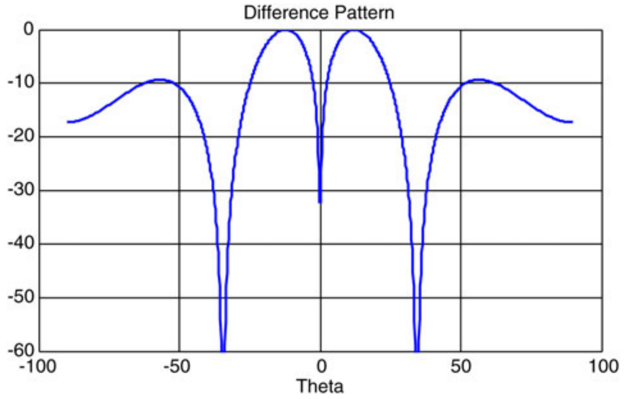

Fig. 2. Difference beam pattern for the 10-element linear array for $u = 1$.

For broadside beams, the array factor is given in (1) [7].

$$AF(\theta) = \sum_{n=1}^{N} I_n e^{j(n-1)K_d[\sin\theta\cos\varphi - \sin\theta_o\cos\varphi_o]}, \tag{1}$$

where θ is the angle of radiation of electromagnetic plane wave; θ_o is the angle where the highest maximum of the angle is attained in $\theta \in [-\pi/2, \pi/2]$; d is the spacing between elements; K is the propagation constant; N is thetotal number of elements in the array; and I_n is the excitation amplitude of the nth element.

In this case, for the linear antenna arrays $\phi_o = 0$. The equations for the creation of the sum, difference and simple hyper beam patterns in terms of two half beams are as follows (2) (3) [2]:

$$\text{Sum pattern, } Sum(\theta) = |R_L| + |R_R|; \tag{2}$$

$$\text{Difference pattern, } Diff(\theta) = |R_L - R_R|; \tag{3}$$

where

$$R_L = \sum_{n=1}^{N/2} I_n e^{j(n-1)K_d[\sin\theta\cos\varphi - \sin\theta_o\cos\varphi_o]},$$

$$R_R = \sum_{n=N/2+1}^{N} I_n e^{j(n-1)K_d[\sin\theta\cos\varphi - \sin\theta_o\sin\varphi_o]}.$$

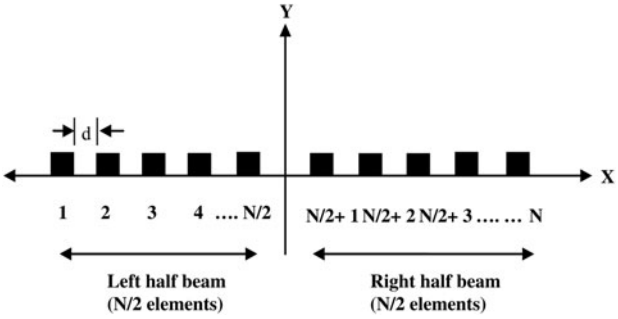

Fig. 3. Geometry of an N-element linear array along the x-axis.

https://doi.org/10.1017/S175907871300086X Published online by Cambridge University Press

The hyper beam is obtained by subtraction of the sum and difference beams, each raised to the power of the exponent u; the general equation of the hyper beam is a function of the hyper beam exponent u given in (4) [2].

$$AF_{Hyper}(\theta) = \{(|R_L| + |R_R|)^u - (|R_L - R_R|)^u\}^{1/u}, \qquad (4)$$

where u ranges from 0.2 to 1. If u lies below 0.2, the hyper beam pattern will contain a large spike height at the peak of the main beam without changing the overall hyper beam pattern. If u is more than 1, the sidelobes of the hyper beam will be more as compared to those of the conventional radiation pattern.

All of the antenna elements are assumed to be isotropic. Only amplitude excitations and inter-element spacing are used to change the antenna radiation pattern. The cost function *(J)* for improving the sidelobe level (SLL) of the radiation pattern of the hyper beam linear antenna arrays is given in (5).

$$\begin{aligned} J = Max|AF_{Hyper}(\theta_{msl1}, I_n)|/|AF_{Hyper}(\theta_0, I_n)| \\ + Max|AF_{Hyper}(\theta_{msl2}, I_n)|/|AF_{Hyper}(\theta_0, I_n)|. \end{aligned} \qquad (5)$$

θ_{msl1} is the angle where the maximum sidelobe $AF_{Hyper}(\theta_{msl1}, I_n)$ is attained in the lower band of the hyper beam pattern. θ_{msl2} is the angle where the maximum sidelobe $AF_{Hyper}(\theta_{msl2}, I_n)$ is attained in the upper side band of the hyper beam pattern. In J, both the numerator and the denominator are in absolute magnitude. Minimization of J means maximum reduction of SLL. RGA, PSO, DE, and FFA are employed individually for minimization of J by optimizing the current excitation weights of the elements and inter-element spacing. Results of the minimization of J and SLL are described in Section IV.

III. OPTIMIZATION TECHNIQUES EMPLOYED

A) Real coded genetic algorithm

RGA is mainly a probabilistic search technique, based on the principles of natural selection and evolution. At each generation, it maintains a population of individuals where each individual is a coded form of a possible solution of the problem at hand called chromosome. Chromosomes are constructed over some particular alphabet, e.g., the binary alphabet {0, 1}, so that chromosomes' values are uniquely mapped onto the real decision variable domain. Each chromosome is evaluated by a function known as cost function, which is usually the objective function of the corresponding optimization problem [13–17]. The basic steps of RGA are shown in Table 1.

Table 1. Steps for the RGA.

Step 1	Initialize the real chromosome strings of the n_p population, each consisting of a set of coefficients of current excitation weights and inter-element spacing J
Step 2	Decoding the strings and evaluation of each string
Step 3	Selection of elite strings in order of increasing J values from the minimum value
Step 4	Copying the elite strings over the non-selected strings
Step 5	Crossover and mutation generate the off-springs
Step 6	Genetic cycle updation
Step 7	The iteration stops when the maximum number of cycles is reached. The grand minimum J and its corresponding chromosome string or the desired solution of coefficients of optimal current excitation weights and optimal inter-element spacing are finally obtained

B) Particle swarm optimization

PSO is a flexible, robust population-based stochastic search or optimization technique with implicit parallelism, which can be easily handled with non-differential objective functions, unlike traditional gradient-based optimization methods. PSO is less susceptible to getting trapped on local optima unlike GA, simulated annealing, etc. Kennedy and Eberhart [18] developed a PSO concept similar to the behavior of a swarm of birds [19–27]. PSO is developed through a simulation of bird flocking and fish schooling in multi-dimensional space. Bird flocking optimizes a certain objective function. Each particle knows its best value so far (*pbest*). This information corresponds to the personal experiences of each particle. Moreover, each particle knows the best value so far in the group (*gbest*) among all of the pbests. Namely, each particle tries to modify its position using the following information:

- The distance between the current position and the *pbest*.
- The distance between the current position and the *gbest*.

Mathematically, the velocities of the vectors are modified according to the following equation:

$$\begin{aligned} V_i^{k+1} = CFa \times (w^{k+1} * V_i^k + C_1 * rand_1 * (pbest_i - S_i^k) \\ + C_2 * rand_2 * (gbest^k - S_i^k)), \end{aligned} \qquad (6)$$

where V_i^k is the velocity of vector i at iteration k; w is the weighting function; C_1 and C_2 are called social and cognitive constants, respectively; $rand_i$ is the random number between 0 and 1; S_i^k is the current position of vector i at iteration k; $pbest_i$ is the *pbest* of vector i; $gbest^k$ is the *gbest* of the group of vectors at iteration k. The first term of (6) is the previous velocity of the vector. The second and third terms are used to change the velocity of the vector. Without the second and third terms, the vector will keep "flying" in the same direction until it hits the boundary. The parameter w corresponds to a kind of inertia and tries to explore new areas. Here, the vector is termed for the string of real current excitation weight coefficients (N number) and uniform inter-element spacing (01 number). Total variables $= nvar = N + 1$ in each vector. Normally, $C_1 = C_2 = 1.5$–2.05 and the Constriction Factor ($CF\alpha$)is given in (7).

$$CFa = \frac{2}{\left|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}\right|}, \qquad (7)$$

where

$$\varphi = C_1 + C_2 \quad \text{and} \quad \varphi > 4. \qquad (8)$$

For $C_1 = C_2 = 2.05$, the computed value of $CF\alpha = 0.73$.

https://doi.org/10.1017/S175907871300086X Published online by Cambridge University Press

Table 2. Steps for PSO.

Step 1	Initialization: Population (swarm size) of particle vectors, $n_P = 120$; maximum iteration cycles = 100; N number of current excitation weights and one number uniform inter-element spacing, total optimizing coefficients equal $nvar = N + 1$; fixing values of C_1, C_2 as 1.5; minimum and maximum values of current excitation coefficients, $I_{min} = 0$, $I_{max} = 1$; minimum and maximum values of inter-element spacing, $d_{min} = 0.5\lambda$, $d_{max} = \lambda$; initialization of the velocities of all of the particle vectors
Step 2	Generation of initial particle vectors, each vector consisting of current excitation weights and uniform inter-element spacing randomly with limits; computation of initial J values of the total population, n_P
Step 3	Computation of population based minimum J value and computation of the personal best solution vectors (*pbest*), group best solution vector (*gbest*)
Step 4	Updating the velocities as per (6); updating the particle vectors as per (10) and checking against the limits of current excitation weights coefficients and one number uniform inter-element spacing; finally, computation of the updated J values of the particle vectors and population-based minimum J value
Step 5	Updating the *pbest* vectors, *gbest* vector; reuse of the updated particle vectors as initial particle vectors for step 4
Step 6	Iteration continues from step 4 till the maximum iteration cycles or the convergence of minimum J values; finally, *gbest* is the vector of optimal current excitation weights (N number) and uniform inter-element spacing (01 number)

The best values of C_1, C_2, and $CF\alpha$ are found to vary with the design sets (Table 2).

Inertia weight (w^{k+1}) at $(k+1)$th cycle is given in (9).

$$w^{k+1} = w_{max} - \frac{w_{max} - w_{min}}{k_{max}} \times (k+1), \tag{9}$$

where $w_{max} = 1.0$; $w_{min} = 0.4$; k_{max} = maximum number of iteration cycles. The searching point/updated vector in the solution space can be modified by (10).

$$S_i^{k+1} = S_i^k + V_i^{k+1}. \tag{10}$$

C) DE algorithm

The crucial idea behind the DE algorithm [28–39] is a scheme for generating trial parameter vectors and adds the weighted difference between two population vectors to a third one. Like any other evolutionary algorithm, the DE algorithm aims at evolving a population of N_P, D-dimensional parameter vectors, so-called individuals, which encode the candidate solutions, i.e.,

$$\overrightarrow{x_{i,g}} = \{x_{1,i,g}, x_{2,i,g}, \ldots, x_{D,i,g}\}, \tag{11}$$

where $i = 1, 2, 3,\ldots, N_P$. The initial population (at $g = 0$) should cover the entire search space as much as possible by uniformly randomizing the individuals within the search constrained by the prescribed minimum and maximum parameter bounds: $\vec{x}_{min} = \{x_{1,min}, \ldots, x_{D,min}\}$ and $\vec{x}_{max} = \{x_{1,max}, \ldots, x_{D,max}\}$.

For example, the initial value of the jth parameter of the ith vector is

$$x_{j,i,0} = x_{j,min} + rand(0,1) * \left(x_{j,\max} - x_{j,min}\right), \tag{12}$$

where $j = 1, 2, 3,\ldots, D$.

The random number generator, *rand* (0,1), returns a uniformly distributed random number from within the range [0,1]. After initialization, DE enters a loop of evolutionary operations: mutation, crossover, and selection.

1) MUTATION

Once initialized, DE mutates and recombines the population to produce new population. For each trial vector $x_{i,\ g}$ at generation g, its associated mutant vector $\vec{v}_{i,g} = \{v_{1,i,g}, v_{2,i,g}, \ldots, v_{D,i,g}\}$ can be generated via a certain mutation strategy. Five most frequently used mutation strategies in the DE codes are listed as follows:

$$\text{"}DE/rand/1\text{"}{:}\vec{v}_{i,g} = \vec{x}_{r'_1,g} + F\left(\vec{x}_{r'_2,g} - \vec{x}_{r'_3,g}\right), \tag{13}$$

$$\text{"}DE/best/1\text{"}{:}\vec{v}_{i,g} = \vec{x}_{best,g} + F\left(\vec{x}_{r'_1,g} - \vec{x}_{r'_2,g}\right), \tag{14}$$

$$\begin{aligned}\text{"}DE/rand-to-best/1\text{"}{:}\vec{v}_{i,g} = \vec{x}_{i,g} + F\left(\vec{x}_{best,g} - \vec{x}_{i,g}\right)\\ + F\left(\vec{x}_{r'_1,g} - \vec{x}_{r'_2,g}\right),\end{aligned} \tag{15}$$

$$\begin{aligned}\text{"}DE/best/2\text{"}{:}\vec{v}_{i,g} = \vec{x}_{best,g} + F\left(\vec{x}_{r'_1,g} - \vec{x}_{r'_2,g}\right)\\ + F\left(\vec{x}_{r'_3,g} - \vec{x}_{r'_4,g}\right),\end{aligned} \tag{16}$$

$$\begin{aligned}\text{"}DE/rand/2\text{"}{:}\overrightarrow{v}_{i,g} = \vec{x}_{r'_1,g} + F\left(\vec{x}_{r'_2,g} - \vec{x}_{r'_3,g}\right)\\ + F\left(\vec{x}_{r'_4,g} - \vec{x}_{r'_5,g}\right).\end{aligned} \tag{17}$$

The indexes r'_1, r'_2, r'_3, r'_4, and r'_5 are mutually exclusive integers randomly chosen from the range [1, N_P], and all are different from the base index i. These indexes are randomly generated once for each mutant vector. The scaling factor F is a positive control parameter for scaling the difference vector. $x_{best,g}$ is the best individual vector with the best fitness value in the population at generation "g". In the present work, (15) has been used.

2) CROSSOVER

To complement the differential mutation search strategy, a crossover operation is applied to increase the potential diversity of the population. The mutant vector $v_{i,g}$ exchanges its components with the target vector $x_{i,g}$ to generate a trial vector:

$$\vec{u}_{i,g} = \{u_{1,i,g}, u_{2,i,g}, \ldots, u_{D,i,g}\}. \tag{18}$$

https://doi.org/10.1017/S175907871300086X Published online by Cambridge University Press

In the basic version, DE employs the binomial (uniform) crossover defined as

$$u_{j,i,g} = \begin{cases} v_{j,i,g}, & \text{if } (rand_{i,j}\,(0,\,1) \le C_r \text{ or } j = j_{rand}), \\ x_{i,j,g}, & \text{otherwise,} \end{cases} \tag{19}$$

where $j = 1, 2, \ldots, D$.

The crossover rate C_r is user-specified constant within the range (1,0), which controls the fraction of parameter values copied from the mutant vector. j_{rand} is a randomly chosen integer in the range [1,D]. The binomial crossover operator copies the jth parameter of the mutant vector $\vec{v}_{i,g}$ to the corresponding element in the trial vector $\vec{u}_{i,g}$ if $rand_{i,j}(0,1) \le C_r$ or $j = j_{rand}$. Otherwise, it is copied from the corresponding target vector $\vec{x}_{i,g}$.

3) Selection

To keep the population size constant over subsequent generations, the next step of the algorithm calls for selection to determine whether the target or the trial vector survive to the next generation, i.e., at $g = g + 1$. The selection operation is described in (20):

$$\vec{x}_{i,g+1} = \begin{cases} \vec{u}_{i,g}, & \text{if } f(\vec{u}_{i,g}) \le f(\vec{x}_{i,g}), \\ \vec{x}_{i,g}, & \text{otherwise,} \end{cases} \tag{20}$$

where $f(x)$ is the J (in this work) to be minimized. Hence, if the new vector yields an equal or lower value of J, it replaces the corresponding target vector in the next generation; otherwise, the target is retained in the population. Hence, the population either gets better (with respect to the minimization of the cost function) or remains the same in fitness status, but never deteriorates.

The above three steps are repeated generation after generation until some specific termination criteria are satisfied.

4) Control parameter selection of DE

Proper selection of control parameters is very important for the success and performance of an algorithm. The optimal control parameters are problem-specific. Therefore, the set of control parameters that best fit each problem have to be chosen carefully. Values of F lower than 0.3 may result in premature convergence, while values greater than 1 tend to slow down the convergence speed. Large populations help maintain diverse individuals, but also slow down convergence speed. In order to avoid premature convergence, F or N_P should be increased or C_r should be decreased. Larger values of F result in larger perturbations and better probabilities to escape from the local optima, while lower C_r preserves more diversity in the population, thus avoiding the local optima.

5) Algorithmic description of DE

Step 1. *Generation of initial population*: Set the generation counter $g = 0$ and randomly initialize the D-dimensional N_p individuals (parameter vectors/target vectors), $\vec{x}_{i,g} = \{x_{1,i,g}, x_{2,i,g}, \ldots, x_{D,i,g}\}$, where $i = 1,2,3,\ldots,N_P$. The initial population (at $g = 0$) should cover the entire search space as much as possible by uniformly randomizing the individuals within the search constrained by the prescribed minimum and maximum parameter bounds: $\vec{x}_{min} = \{x_{1,min}, \ldots, x_{D,min}\}$ and $\vec{x}_{max} = \{x_{1,max}, \ldots, x_{D,max}\}$.

Step 2. *Mutation*: For $i = 1$ to N_P, generate a mutated vector, $\vec{v}_{i,g} = \{v_{1,i,g}, v_{2,i,g}, \ldots, v_{D,i,g}\}$ corresponding to the target vector $\vec{x}_{i,g}$ via mutation strategy (15).

Step 3. *Crossover*: Generation of a trial vector $\vec{u}_{i,g}$ for each target vector $\vec{x}_{i,g}$, where $\vec{u}_{i,g} = \{u_{1,i,g}, u_{2,i,g}, \ldots, u_{D,i,g}\}$.
for $i = 1$ to N_P; $j_{rand} = [rand(0,1)*D]$; for $j = 1$ to D.

$$u_{j,i,g} = \begin{cases} v_{j,i,g}, & \text{if } (rand_{i,j}\,(0,\,1) \le C_r \text{ or } j = j_{rand}), \\ x_{i,j,g}, & \text{otherwise.} \end{cases}$$

Step 4. *Selection*: for $i = 1$ to N_P,

$$\vec{x}_{i,g+1} = \begin{cases} \vec{u}_{i,g}, & \text{iff}(\vec{u}_{i,g}) \le f(\vec{x}_{i,g}), \\ \vec{x}_{i,g}, & \text{otherwise.} \end{cases}$$

Increment the generation count $g = g + 1$.

D) Firefly algorithm

FFA, developed by Yang [40], is inspired by the flash pattern and characteristics of fireflies. The basic rules for the FFA are:

- All of the fireflies are unisex so that one firefly will be attracted to other fireflies regardless of their sex.
- Attractiveness is proportional to their brightness, thus for any two flashing fireflies, the less bright one will move toward the brighter one, and the brightness decreases as their distance increases. If there is no brighter one than a particular firefly, it will move randomly.
- The brightness of a firefly is affected or determined by the landscape of the cost function. For a minimization problem, the brightness can simply be inversely proportional to the value of the cost function. In this work, the cost function is J.

In the simplest case for the minimization optimization problems, the brightness B of a firefly at a particular location x can be chosen as $B(x) = 1/f(x)$, where $f(x)$ is J in this work. However, the attractiveness β is relative; it should be seen in the eyes of the beholder or judged by the other fireflies. Thus, it will vary with the distance r_{ij} between firefly i and firefly j. For a given medium with a fixed light absorption coefficient,γ, the light intensity varies with the distance r. That is

$$B = B_0 e^{-\gamma r}, \tag{21}$$

where B_0 is the original light intensity; r is the Euclidean distance between the fireflies. As a firefly's attractiveness is proportional to the light intensity seen by adjacent fireflies, the attractiveness/repulsiveness β of a firefly can be defined by

$$\beta = \beta_0 e^{-\gamma r^2}, \tag{22}$$

where β_0 is the attractiveness (positive sign)/repulsiveness (negative sign) at $r = 0$.

The distance between any two fireflies i and j at x_i and x_j, respectively, is the Euclidean distance.

$$r_{ij} = \|x_i - x_j\| = \sqrt{\sum_{k=1}^{D} (x_{i,k} - x_{j,k})^2}, \tag{23}$$

https://doi.org/10.1017/S175907871300086X Published online by Cambridge University Press

where $x_{i,k}$ is the kth component of the special coordinate x_i of the ith firefly; D is the dimension of each x_i and x_j.

The movement of a firefly i is attracted by another more attractive (brighter) firefly j *or repelled by more repulsive (less bright)* firefly j and is determined by

$$x_i = x_i + \beta_o e^{-\gamma r_{ij}^2}(x_j - x_i) + \alpha\left(rand - \frac{1}{2}\right) \qquad (24)$$

where the second term is due to the attraction or repulsion. The third term is randomized with a control parameter α, which makes the exploration of search space more efficient. Usually, $\beta_o = 1$, $\alpha \in [0,1]$ for most applications. By adjusting the parameters γ, α and β_o, the performance of the algorithm can be improved.

Steps of FFA are as follows:

Step 1: Generate initial firefly vectors $xi = (xi1,\ldots,xiD)$ ($i = 1,\ldots,120$), where $D = N + 1$ (N element excitations $I \in [0,1]$ plus the common inter-element spacing $d \in [\lambda/2,\lambda]$). Set the maximum allowed number of iterations to 100. $\beta_o = 0.6$, $\gamma = 0.2$, and $\alpha = 0.01$ (these values, and the population size, 120, were determined as optimal in a series of 30 preliminary trials.

Step 2: Computation of initial J of the total population.

Step 3: Computation of the initial population based best solution (*gbest*) vector corresponding to the historical population best and least J value.

Step 4: Update the firefly positions:

(a) Compute the square root (rsqrt) of the Euclidean distance between the first particle vector and the second particle vector as per (23).

(b) Compute β with the help of β_o as per (22).

(c) If J of second particle is $<J$ of first particle, then, update the first particle as per (24) with $+\beta_o$ (case of attraction), otherwise with $-\beta_o$ (case of repulsion).

IV. NUMERICAL RESULTS

Each algorithm was run 100 times to get its best solutions. The best results are reported in this work. Population size chosen for RGA, PSO, DE, and FFA is 120. Best control parameters for the RGA are: Crossover rate = 0.8; Crossover taken is two point crossover; mutation rate = 0.05; type of mutation is Gaussian mutation; selection, probability = Roulette wheel, 1/3. Best control parameters for PSO are: C_1, C_2 = 1.5, 1.5; v_i^{min}, v_i^{max} = 0.01, 1.0; w_{max}, w_{min} = 1.0, 0.4. Best parameters for DE are: $C_r = 0.3$; $F = 0.5$. Best parameters for FFA are: α,γ, β_o = 0.01, 0.2, 0.6. These best parameters have been determined after 30 trial runs of each algorithm.

A) Analysis of radiation patterns of hyper beam without optimization

This section gives the experimental results for various hyper beams of non-optimized linear antenna array designs. Three linear antenna array designs considered are of 10-, 14-, and 20-element sets, each maintaining uniform inter-element spacing. Reduction of main beam width [first null beam width (FNBW)] and SLL can be controlled by varying the hyper beam exponent value u, thereby obtaining different hyper beam patterns. The results show that the SLL reduction increases as the exponent value u decreases. For 10-, 14-, and 20-element linear arrays, with $u = 1$, SLL reductions are −19.91, −20.10, and −20.20 dB, respectively, whereas with $u = 0.5$, the SLL reduces to −32.78, −33.02, and −33.20 dB, respectively, as shown in Figs 4–9 and Table 3. The uniform linear array shows the respective SLL values as −12.97, −13.11, and −13.20 dB. Therefore, the optimization technique applied to the hyper beam yields much more reduction of SLL in comparison to that of the uniform linear array and non-optimized hyperbeam case. Main beam width (FNBW) remains unaltered or has been improved for all of the cases of FFA unlike RGA, DE, and PSO.

B) Analysis of radiation patterns of hyper beam with optimization by RGA, PSO, DE, and FFA

This section gives the experimental results for various optimized hyper beam antenna array designs obtained by the RGA, PSO, DE, and FFA techniques. The parameters of the RGA, PSO, DE, and FFA are set after many trial runs. It is found that the best results are obtained for the initial population (n_p) of 120 chromosomes and maximum number of generations, Nm as 100. With the RGA, for selection operation, the method of natural selection is chosen with a selection probability of 0.3. Crossover is randomly selected as a dual point. The Crossover ratio is 0.8. Mutation probability is 0.05. Each RGA, PSO, DE, and FFA technique individually generates a set of optimized, non-uniform current excitation weights, and optimal uniform inter-element spacing for same three sets of linear antenna arrays. Tables 4 and 5 show the SLL, FNBW, and optimal current excitation weights with the hyper beam exponent values $u = 0.5$, and $u = 1$, respectively, for the optimally excited hyper beam linear antenna array with optimized uniform inter-element spacing ($d \in [\lambda/2,\lambda]$) using RGA, PSO, DE, and FFA. Figures 4–9 depict the radiation patterns of linear antenna arrays with the exponent values $u = 0.5$ and $u = 1$ for sets of 10, 14, and 20 number of elements, respectively, with optimized non-uniform excitations and optimized fixed inter-element spacing, as obtained by the techniques. Figures clearly show improvement of SLL and FNBW by optimization of hyper beam.

1) Analysis of radiation patterns of hyper beam with $u = 0.5$ and optimization by RGA, PSO, DE, and FFA

The following observations are made from Table 4, in which the exponent value $u = 0.5$. For the 10-element array, RGA, PSO, DE, and FFA yield SLL values −100.6, −117.2, −151.9, and −168.5 dB, respectively, of the optimized hyper beam pattern against the SLL of −32.78 dB of the non-optimized hyper beam pattern. Similarly, for the same array, RGA, PSO, DE, and FFA yield FNBW values 41.04°, 39.60°, 34.56°, and 26.64°, respectively, of the optimized hyper beam pattern against the FNBW of 33.12° of the non-optimized hyper beam pattern. For the 14-element array, RGA, PSO, DE, and FFA yield SLL values −96.21, −113, −125.8, and −149.7 dB, respectively, of the optimized hyper beam pattern against the SLL of −33.02 dB of the non-optimized hyper beam pattern. Similarly, for the same array, RGA, PSO, DE, and FFA yield FNBW values 25.92°, 25.20°, 23.04°, and 20.16°, respectively, of the optimized hyper beam pattern against the FNBW of 23.04° of the non-

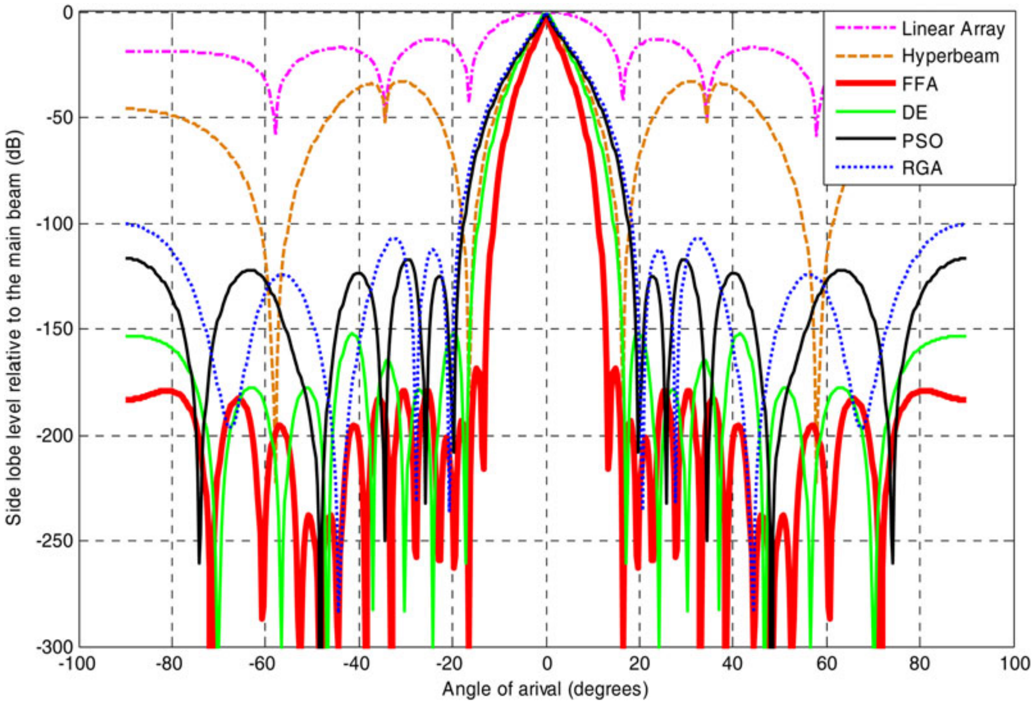

Fig. 4. Best array pattern found by the FFA for the 10-element array with improved SLL and FNBW at $u = 0.5$.

optimized hyper beam pattern. For the 20-element array, RGA, PSO, DE, and FFA yield SLL values −83.69, −88.71, −101.9, and −115.1 dB, respectively, of the optimized hyper beam pattern against the SLL of −33.20 dB of the non-optimized hyper beam pattern. Similarly, for the same array, RGA, PSO, DE, and FFA yield FNBW values 19.44°, 18.72° and 18°, 11.52°, respectively, of the optimized hyper beam pattern against the FNBW of 16.56° of the non-optimized hyper beam pattern. The figures as well as Tables clearly show improvement of the SLL and FNBW by FFA-based optimization.

2) Analysis of radiation patterns of hyper beam with $U = 1$ and optimization by RGA, PSO, DE, and FFA

The following observations are made from Table 5, in which the exponent value $u = 1$. For the 10-element

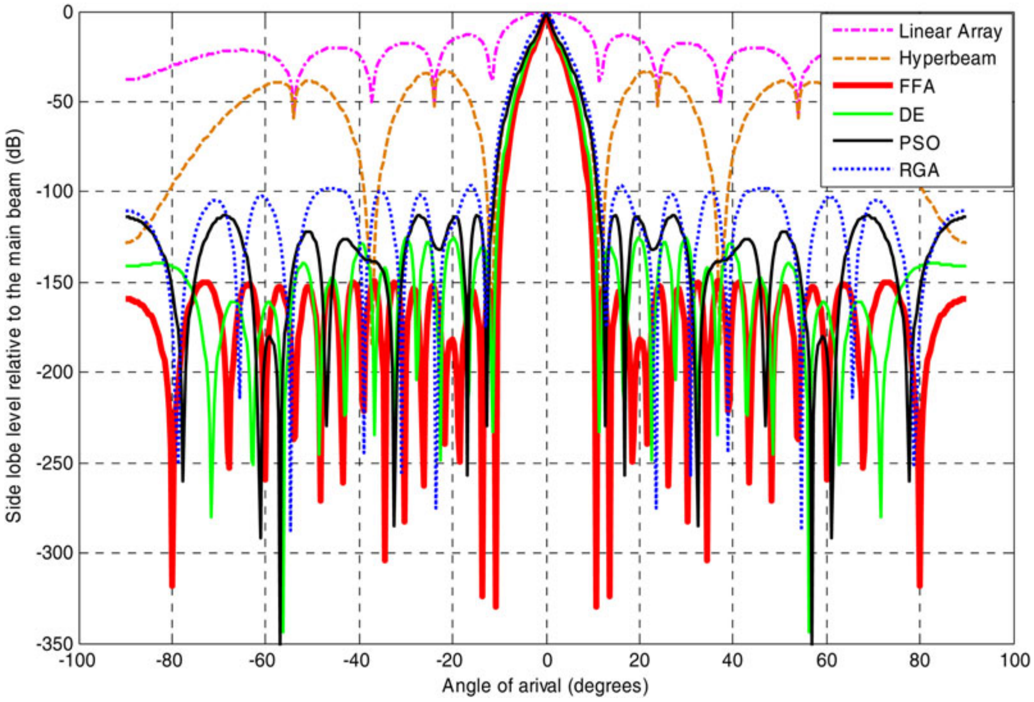

Fig. 5. Best array pattern found by the FFA for the 14-element array with improved SLL and FNBW at $u = 0.5$.

https://doi.org/10.1017/S175907871300086X Published online by Cambridge University Press

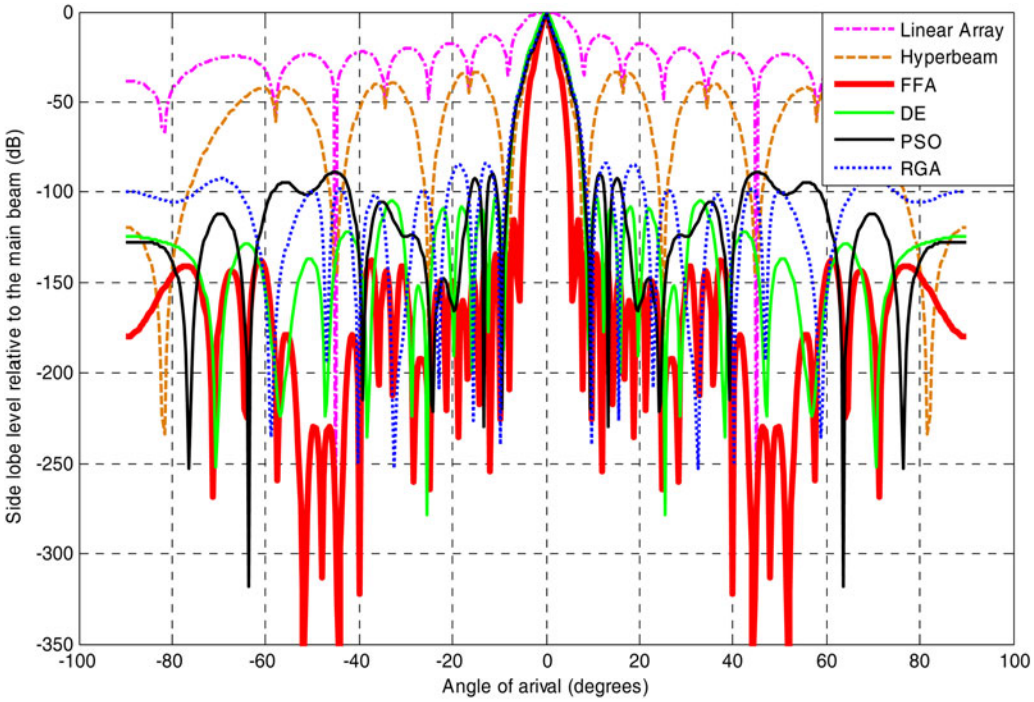

Fig. 6. Best array pattern found by the FFA for the 20-element array with improved SLL and FNBW at $u = 0.5$.

array, RGA, PSO, DE, and FFA yield SLL values −46.76, −58.88, −64.57, and −89.33 dB, respectively, of the optimized hyper beam pattern against the SLL of −19.91 dB of the non-optimized hyper beam pattern. Similarly, for the same array, RGA, PSO, DE, and FFA yield FNBW values 36.72, 35.28, 34.56, and 30.96°, respectively, of the optimized hyper beam pattern against the FNBW of 33.12° of the non-optimized hyper beam pattern. For the 14-element array, RGA, PSO, DE, and FFA yield SLL values −46.76, −51.4, −61.71, and −72.93 dB, respectively, of the optimized hyper beam pattern against the SLL of −20.10 dB of the non-optimized hyper beam pattern. Similarly, for the same array, RGA, PSO, DE, and FFA yield FNBW values 25.20°, 24.48°, 23.76°, and 19.44°, respectively, of the optimized hyper beam pattern against the FNBW of 23.04° of the non-optimized hyper

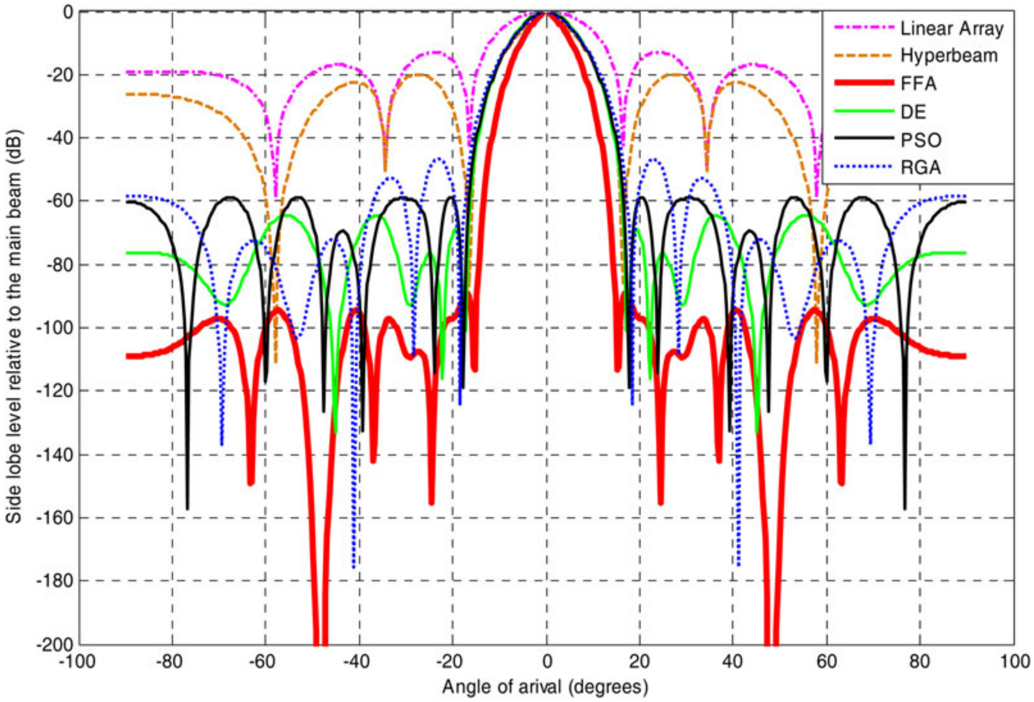

Fig. 7. Best array pattern found by the FFA for the 10-element array with improved SLL and FNBW at $u = 1$.

https://doi.org/10.1017/S175907871300086X Published online by Cambridge University Press

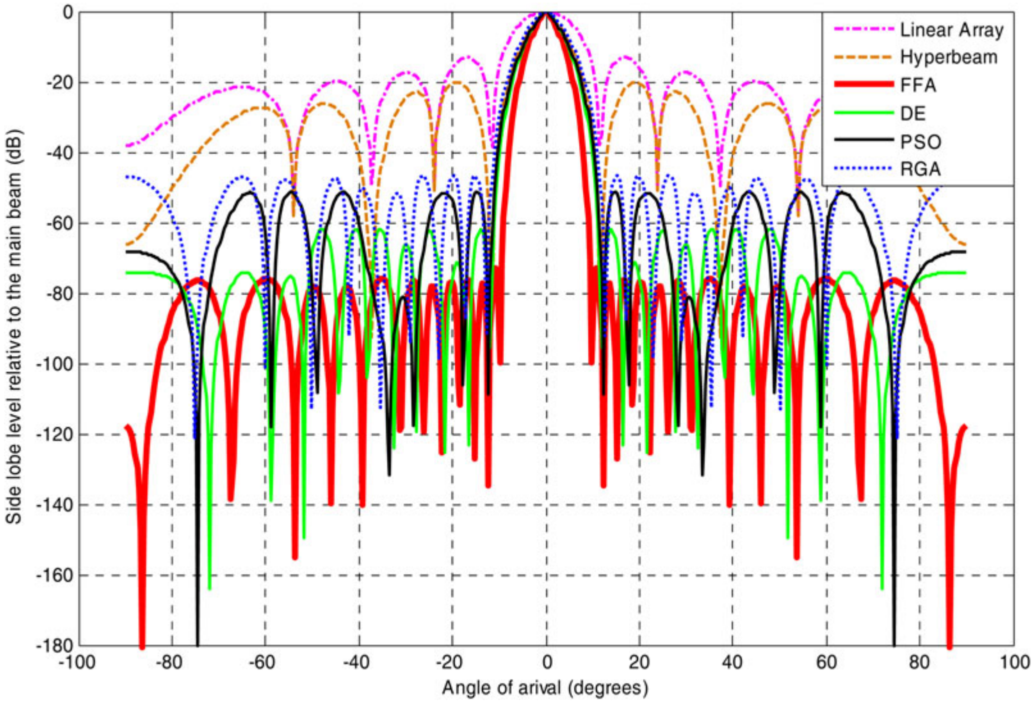

Fig. 8. Best array pattern found by the FFA for the 14-element array with improved SLL and FNBW at $u = 1$.

beam pattern. For the 20-element array, RGA, PSO, DE, and FFA yield SLL values −42.85, −52.97, −61.19, and −73.24 dB, respectively, of the optimized hyper beam pattern against the SLL of −20.20 dB of the non-optimized hyper beam pattern. Similarly, for the same array, RGA, PSO, DE, and FFA yield FNBW values 18.72°, 18°, 17.28°, and 13.68°, respectively, of the optimized hyper beam pattern against the FNBW of 16.56° of the non-optimized hyper beam pattern. The figures as well as Tables clearly depict improvement of the SLL and FNBW by FFA-based optimization.

The FFA efficiently computes N number of near global optimal current excitation weights and one number optimal uniform inter-element separation for each hyper beam linear antenna array to have maximum SLL reduction and improved FNBW.

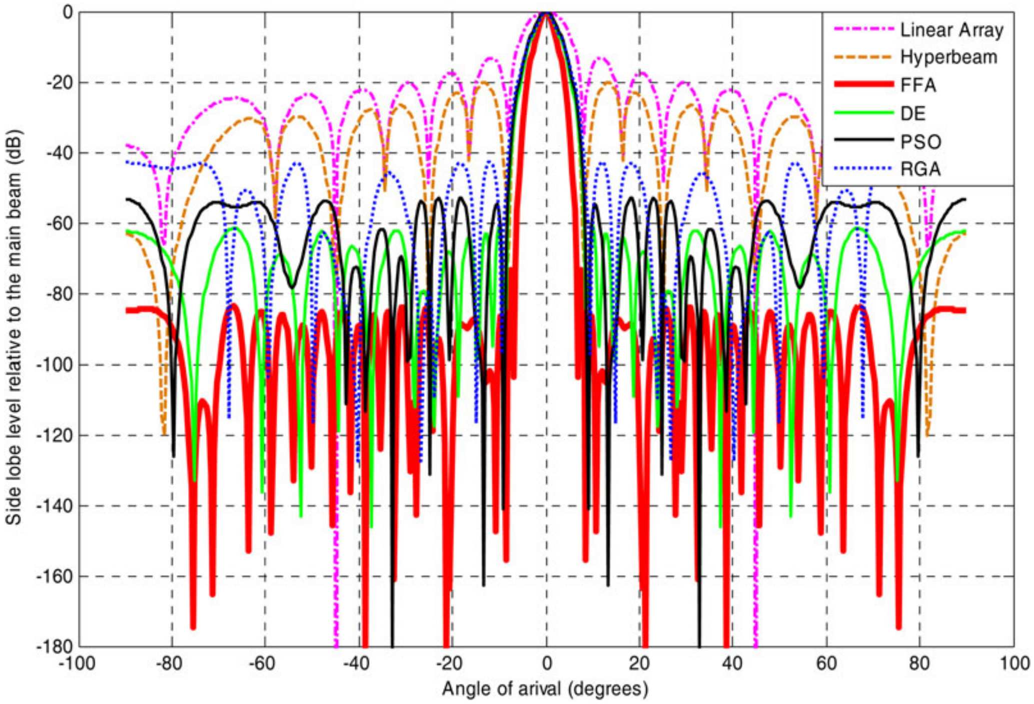

Fig. 9. Best array pattern found by the FFA for the 20-element array with improved SLL and FNBW at $u = 1$.

https://doi.org/10.1017/S175907871300086X Published online by Cambridge University Press

Table 3. Initial values of SLL and FNBW for uniform linear array having uniform excitation ($I_n = 1$) and $d = \lambda/2$ inter-element spacing.

N	SLL (dB) for uniform linear array with ($I_n = 1$) and $d = \lambda/2$	SLL of hyper beam non-optimized (dB) at $u = 0.5$, ($I_n = 1$) and $d = \lambda/2$	SLL of hyper beam non-optimized (dB) at $u = 1$, ($I_n = 1$) and $d = \lambda/2$	FNBW (deg) for uniform linear array with ($I_n = 1$) and $d = \lambda/2$	FNBW of hyper beam non-optimized (deg) at $u = 0.5$, ($I_n = 1$) and $d = \lambda/2$	FNBW of hyper beam non-optimized (deg) at $u = 1$, ($I_n = 1$) and $d = \lambda/2$
10	−12.97	−32.78	−19.91	33.12	33.12	33.12
14	−13.11	−33.02	−20.10	23.04	23.04	23.04
20	−13.20	−33.20	−20.20	16.56	16.56	16.56

Table 4. SLL, FNBW, optimal current excitation weights, and optimal inter-element spacing for hyper beam pattern of linear array with hyper beam exponent ($u = 0.5$), obtained by RGA, PSO, DE, and FFA for different sets of arrays.

N	Algorithms	Optimized current excitation weights and [I_1, I_2, I_3, I_4......I_N]				Optimal inter-element spacingIn (λ)	SLL of hyper beam with optimization (dB)	FNBW of hyper beam with optimization (deg)
10	GA	0.2844	0.5240	0.8813	0.9032	0.5441	−100.6	41.04
		0.4231	0.8425	0.4564	0.6402			
		0.3414	0.3853					
	PSO	0.2398	0.6414	0.9123	0.9722	0.5717	−117.2	39.60
		0.4312	0.9502	0.4327	0.6582			
		0.3571	0.3982					
	DE	0.1029	0.3802	0.6258	0.9394	0.8470	−151.9	34.56
		0.7907	1.0000	0.5211	0.5538			
		0.2118	0.2156					
	FFA	0.2490	0.4210	0.1819	0.6099	0.9460	−168.5	26.64
		0.8929	0.5685	0.6123	0.6745			
		0.3250	0.1579					
14	GA	0.3631	0.2555	0.4905	0.0043	0.5878	−96.21	25.92
		0.6114	0.5778	0.8634	0.5042			
		0.5782	0.5913	0.7502	0.5545			
		0.2878	0.3431					
	PSO	0.2319	0.1857	0.6027	0.5089	0.6036	−113	25.20
		0.7906	0.4163	0.6275	0.7212			
		0.9097	0.2907	0.2525	0.2755			
		0.5506	0.3615					
	DE	0.2297	0.3701	0.3080	0.2229	0.7949	−125.8	23.04
		0.6599	0.9495	0.6941	0.8597			
		0.4157	0.7559	0.7305	0.2389			
		0.3759	0.0982					
	FFA	0.3124	0.3709	0.5444	0.3167	0.8760	−149.7	20.16
		0.8885	0.9115	0.9869	0.9916			
		0.9243	0.6967	0.9560	0.4002			
		0.1592	0.1990					
20	GA	0.2505	0.3933	0.4881	0.4829	0.5361	−83.69	19.44
		0.3027	0.6697	0.3436	0.9551			
		0.5974	0.8952	0.5252	0.9773			
		0.4056	0.6612	1.0000	0.1577			
		0.8144	0.3284	0	0.5558			
	PSO	0.1675	0.2453	0.2113	0.5168	0.5353	−88.71	18.72
		0.6011	0.5661	0.7962	0.2148			
		0.8279	0.2476	0.9888	0.3429			
		0.8064	0.1836	0.2281	0.1792			
		0.4317	0.6579	0.2244	0.3467			
	DE	0.1567	0.1345	0.5561	0.4817	0.5852	−101.9	18
		0.9529	0.7651	0.9420	0.7511			
		0.6736	0.5927	0.9889	0.8862			
		0.4313	0.4025	0.2891	0.3316			
		0.4286	0.4649	0.4306	0.3195			
	FFA	0.0737	0.3285	0.4033	0.5217	0.9520	−115.1	11.52
		0.8113	0.9979	0.9665	0.7002			
		0.7260	0.7769	0.9957	1.0000			
		0.9467	0.4586	0.2044	0.2364			
		0.5734	0.6510	0.4416	0.2468			

https://doi.org/10.1017/S175907871300086X Published online by Cambridge University Press

Table 5. SLL, FNBW, optimal current excitation weights, and optimal inter-element spacing for hyper beam pattern of linear array with hyper beam exponent ($u = 1$), obtained by RGA, PSO, DE, and FFA for different sets of arrays.

N	Algorithms	Optimized current excitation weights and $[I_1, I_2, I_3, I_4 \ldots\ldots I_N]$					Optimal inter-element spacingIn (λ)	SLL of hyper beam with optimization (dB)	FNBW of hyper beam with optimization (deg)
10	GA	0.1339	0.1010	0.4353	0.3657		0.6503	−46.76	36.72
		0.6166	0.5295	0.6264	0.4194				
		0.3935	0.2296						
	PSO	0.3889	0.4254	0.2096	0.7456		0.6436	−58.88	35.28
		0.7961	0.4382	0.3525	0.5002				
		0.1764	0.1603						
	DE	0.2057	0.4820	0.9658	0.9686		0.9974	−64.57	34.56
		1.0000	0.9881	0.5582	0	0			
		0.0086							
	FFA	0.1431	0.3331	0.7904	0.7837		0.6822	−89.33	30.96
		0.8057	0.9664	0.7236	0.2823				
		0.3939	0.2024						
14	GA	0	0.4146	0.6005	0.7859	0.7903	0.5824	−46.76	25.20
		0.7755	0.4159	0.9358	0.2159				
		0.3125	0.4533	0	0.6501	0.1934			
	PSO	0.1011	0.2588	0.3020	0.5343		0.6698	−51.4	24.48
		0.6365	0.6937	0.5245	0.8198				
		0.3813	0.4761	0.3815	0.4803				
		0.1301	0.3374						
	DE	0.1822	0.4092	0.3052	0.3611		0.7436	−61.71	23.76
		0.5660	0.8365	0.6771	0.7047				
		0.4664	0.6376	0.5091	0.3593				
		0.0822	0.1951						
	FFA	0.0000	0.0559	0.3976	0.6918		0.9828	−72.93	19.44
		0.9234	0.9165	0.9799	0.9966				
		0.9171	0.6335	0.5352	0.4790				
		0.4668	0.2407						
20	GA	0.2739	0.0772	0.4652	0.3369		0.5587	−42.85	18.72
		0.4341	0.6162	0.5613	0.8008				
		0.4211	0.7082	0.6840	0.8283				
		0.3579	0.4822	0.3872	0.7091				
		0.3145	0.3415	0.1838	0.4675				
	PSO	0.5918	0.0903	0.4110	0.0131		0.5961	−52.97	18
		0.6447	0.1519	0.7800	0	0.8548			
		0.8593	0.6530	0.7593	0.9763				
		0.9991	0.7571	0.8972	0.5175				
		0.7424	0.2818	0.2433					
	DE	0.0938	0.1489	0.2668	0.5640		0.6638	−61.19	17.28
		0.6043	0.8496	0.8847	0.7082				
		0.6143	0.7392	0.8163	0.8767				
		0.7024	0.3612	0.2068	0.3691				
		0.3646	0.5144	0.3815	0.1665				
	FFA	0.1513	0.3224	0.4109	0.5525		0.9869	−73.24	13.68
		0.3320	0.4109	0.5534	0.9734				
		0.8671	0.9796	0.9428	0.9907				
		0.7807	0.9722	0.9538	0.7113				
		0.3323	0.2590	0.2029	0.0035				

V. COMPARATIVE EFFECTIVENESS AND CONVERGENCE PROFILE OF RGA, PSO, DE, AND FFA

The algorithms can be compared in terms of the J values, Figs 10 and 11 show the convergences of the $\log_{10}(J)$ values obtained as RGA, PSO, DE, and FFA are employed, respectively. As compared to RGA, PSO, and DE which yield suboptimal higher values of J, the FFA converges to the least minimum J values in finding the near-global optimal current excitation weights and optimal inter-element spacing of the hyper beam of the antenna arrays. With a view to the above fact, it may finally be inferred that the performance of the FFA algorithm is the best among the algorithms. Figures 10 and 11 portray the convergence profiles of minimum J for the 10-element array sets, for $u = 0.5$ and 1, respectively. Table 6 shows the execution times of RGA, PSO, DE, and FFA. From the same table it is clear that the execution times of the FFA are all less than those of RGA,

https://doi.org/10.1017/S175907871300086X Published online by Cambridge University Press

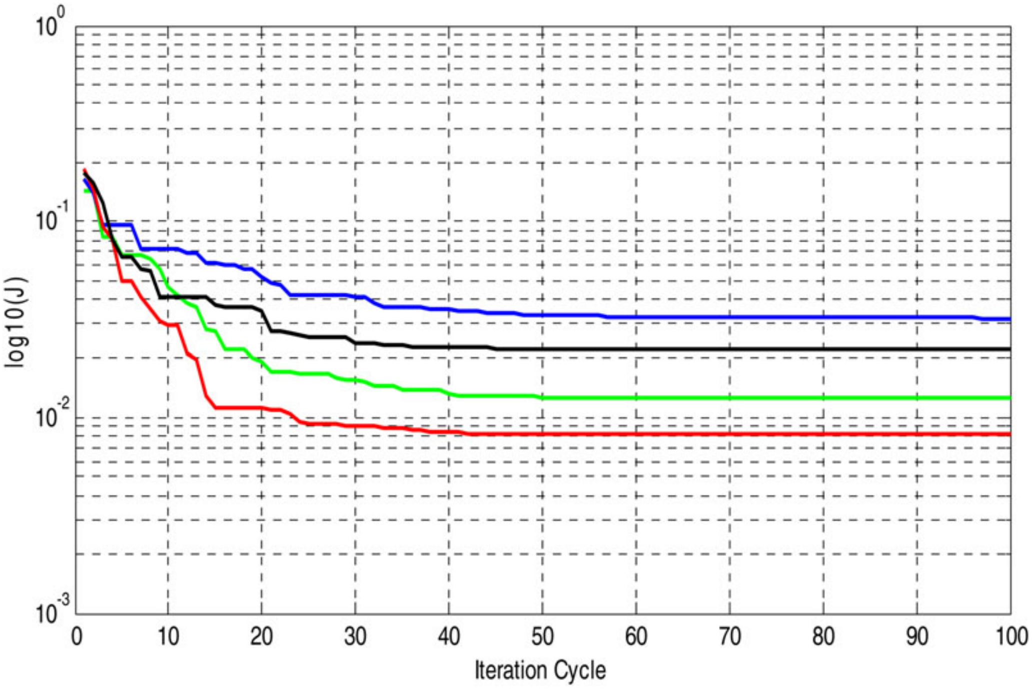

Fig. 10. Convergence profile of the FFA in case of 10-element linear antenna array at $u = 0.5$.

PSO, and DE. The simulation programming was performed in MATLAB language using MATLAB 7.5 on a dual core (TM) processor, 2.88 GHz with 2 GB RAM.

VI. CONCLUSIONS

In this paper, a novel FFA is used for finding the optimal sets of non-uniformly excited ($0 < I_n \leq 1$) hyper beamforming of receiving linear antenna arrays, each with optimal uniform inter-element spacing ($\lambda/2 \leq d < \lambda$).Three broad cases of arrays are considered in the study. The first two cases are: (i) conventional uniformly excited ($I_n = 1$) linear antenna arrays with inter-element spacing, $d = \lambda/2$ and (ii) non-optimized uniformly excited ($I_n = 1$) hyper beamforming of linear antenna arrays with inter-element spacing, $d = \lambda/2$. The last one is of actual concern, which is the hyper beamforming of linear antenna arrays with optimized inter-element spacing ($\lambda/2 \leq d < \lambda$) along with optimized non-uniform

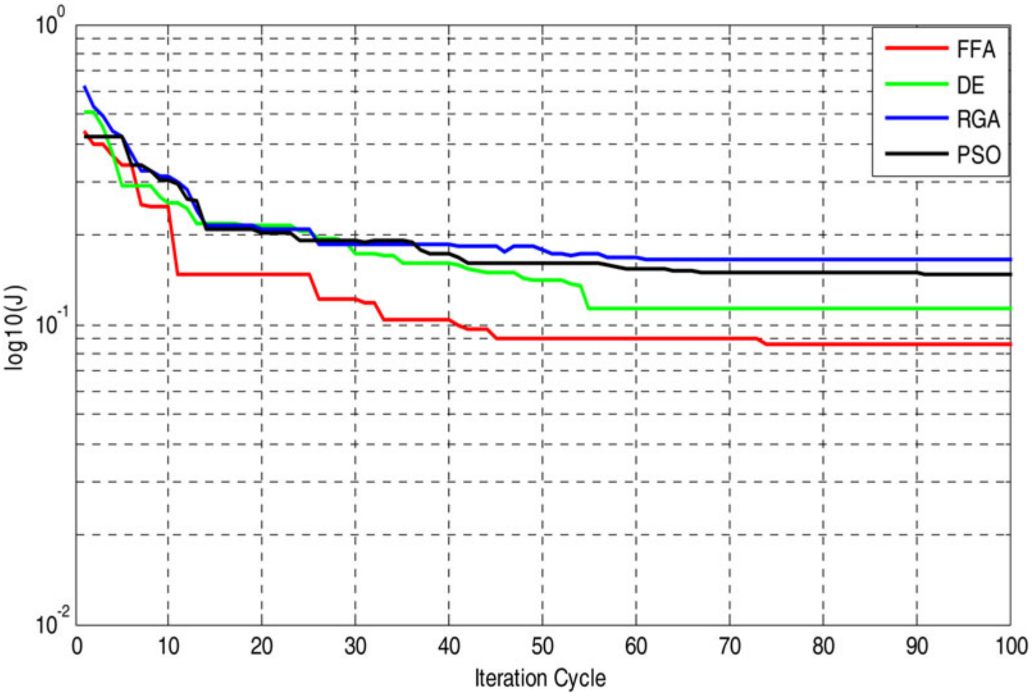

Fig. 11. Convergence profile of the FFA in case of 10-element linear antenna array at $u = 1$.

https://doi.org/10.1017/S175907871300086X Published online by Cambridge University Press

Table 6. Comparison of execution times for different algorithms for different sets of elements.

No. of elements	Execution times for different algorithms			
	RGA (s)	DE (s)	PSO (s)	FFA (s)
10	385	311	244	136
14	408	376	289	152
20	538	484	320	185

excitations ($0 < I_n \leq 1$). The optimization algorithms considered are RGA, PSO, DE and the proposed FFA. The experimental results reveal the following observations regarding SLL reductions: (a) FFA-based optimal design of hyper beamforming (hyperbeam exponent, $u = 0.5$) of 10-, 14-, and 20-element linear antenna arrays offer considerable respective reductions of 155.53, 136.59, and 111.90 dB in SLL as compared to the corresponding case (i). Similar considerable SLL reductions have occurred for $u = 1$ also; (b) the same FFA-based designs for the same hyper beam exponent yield respective SLL reductions of 135.72, 116.68, and 81.90 dB with respect to the corresponding case (ii). Similar considerable SLL reductions have occurred for $u = 1$ also. Regarding FNBW, the following observations are: (a) the same FFA-based designs for $u = 0.5$ yield FNBW improvements of 6.48°, 2.88°, and 5.04°, respectively, for the arrays, as compared to both the corresponding cases (i) and (ii). Similar FNBW improvement occurs for the other hyper beam exponent also. The above-mentioned results of the FFA are the best grand optimal results as compared to those of RGA, PSO, and DE. It is also found that the proposed FFA-based technique takes the least execution times for finding all of the optimal hyper beamforming designs of the linear antenna arrays. The other algorithms are entrapped in sub-optimal solutions and corresponding sub-optimal designs in higher execution times.

REFERENCES

[1] Schlieter, H.; Eigenbrod, H.: Method for the formation of radiated beams in direction finder systems, Patent US 6021096 A, February 1, 2000.

[2] Anita, V.; Sri Jaya Lakshmi, S.; Sreedevi, I.; Khan, H.; Sarat Kumar, K.; Ramakrishna, P.: An adaptive processing of linear array for target detection improve. Int. J. Comput. Appl. (0975–8887), **42** (4) (2012), 33–36.

[3] Isernia, T.; Ares Pena, F.J.; Bucci, O.M.; Durso, M.; Gomez, J.F.; Rodriguez, J.A.: A hybrid approach for the optimal synthesis of pencil beams through array antennas. IEEE Trans. Antennas Propag., **52** (11) (2004), 2912–2918.

[4] Walkar, R.S.: Bearing accuracy and resolution bound of high-resolution beam formers, in Proc. IEEE ICASSP '85, Tampa, FL.

[5] Takao, K.; Komiyama, K.: An adaptive antenna array under directional constraint. IEEE Trans. Antennas Propag., **AI 24** (1976), 662–669.

[6] Schilter, H.: Method for three-dimensional beam forming in direction finding systems, Patent US 6178140, January 23, 2001.

[7] Balanis, C.A.: Antenna Theory Analysis and Design. John Wiley & Sons, New York, 1997.

[8] Krous, J.D.: Antenna. Mc GRAW-HILL, New York, 1950.

[9] Mailloux, R.J.: Phased array architecture for millimetric active arrays, IEEE Antennas Propag. Soc. Newslett., February 1986, **28**, 4–7.

[10] Schrank, H.E.: Low sidelobe phased array antennas. IEEE Antennas Propag. Soc. Newslett., **25** (2) (1983), 4–9.

[11] Applebaum, S.P.; Chapman, D.J.: Adaptive arrays with main beam constraints. IEEE Trans. Antennas Propag., **AI-24** (1976), 650–662.

[12] Chen, S.: IIR model identification using batch-recursive adaptive simulated annealing algorithm, in 6th Annual Chinese Automation and Computer Science Conf., 2000, 151–155.

[13] Haupt, R.L.: Phase-only adaptive nulling with a genetic algorithm. IEEE Trans. Antennas Propag., **45** (6) (1997), 1009–1015.

[14] Haupt, R.L.; Werner, D.H.: Genetic Algorithms in Electromagnetics. IEEE Press, Wiley–Interscience, John Wiley & Sons, Inc., Publication, USA, 2007.

[15] Chung, Y.C.; Haupt, R.L.: Adaptive nulling with spherical arrays using a genetic algorithm. Proc. IEEE AP-S Int. Symp. Digest, **3** (1999), 2000–2003.

[16] Hardel, G.R.; Yalapragada, N.T.; Mandal, D.; Bhattacharjee, A.K.: Introducing dipper nulls in time modulated linear symmetric antenna array using real coded genetic algorithm, in Symp. Computers and Informatics, March 2011, 249–254.

[17] Eberhart, R.; Shi, Y.: Comparison between genetic algorithm and particle swarm optimization, in Evolutionary Programming VII, Springer, 1998, 611–616.

[18] Kennedy, J.; Eberhart, R.: Particle swarm optimization, in Proc. IEEE Int. Conf. Neural Network, vol. 4, 1995, 1942–1948.

[19] Mandal, D.; Yallaparagada, N.T.; Ghoshal, S.P.; Bhattacharjee, A.K.: Wide null control of linear antenna arrays using particle swarm optimization, in IEEE INDICON, Kolkata, India, December 2010, 1–4.

[20] Hao, Z.F.; Guo, G.H.; Huang, H.: A particle swarm optimization algorithm with differential evolution, in Int. Conf. Machine Learning and Cybernetics, vol. 2 (2007), 1031–1035.

[21] Luitel, B.; Venayagamoorthy, G.K.: Differential evolution particle swarm optimization for digital filter design, in IEEE Congress on Evolutionary Computation, CEC, 2008, 3954–3961.

[22] Eberhart, R.C.; Shi, Y.: Particle swarm optimization: developments, applications and resources, evolutionary computation, in Proc. Congress on Evolutionary Computation, 2001, 81–86.

[23] Liang, J.J.; Qin, A.K.; Suganthan, P.N.; Baskar, S.: Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. IEEE Trans. Evol. Comput., **10** (3) (2006), 281–295.

[24] Luitel, B.; Venayagamoorthy, G.K.: Particle swarm optimization with quantum infusion for system identification. Eng. Appl. Artif. Intell., **23** (2010), 635–649.

[25] Panda, G.; Mohanty, D.; Majhi, B.; Sahoo, G.: Identification of nonlinear systems using particle swarm optimization technique, in IEEE Congress on Evolutionary Computation, 2007, 3253–3257.

[26] Durmus, B.; Gun, A.: Parameter identification using particle swarm optimization, in 6th Int. Advanced Technologies Symp., IATS'11, Turkey, May 2011, 188–192.

[27] Van den Bergh, F.; Engelbrecht, A.P.: Cooperative learning in neural network using particle swarm optimizers. South African Comput. J., **26** (2000), 84–90.

[28] Storn, R.; Price, K.: Differential evolution- a simple and efficient adaptive scheme for global optimization over continuous spaces, Technical Report, International Computer Science Institute, Berkley, TR-95-012, 1995.

https://doi.org/10.1017/S175907871300086X Published online by Cambridge University Press

[29] Storn, R.; Price, K.V.: Minimizing the real functions of the ICEC 1996 contest by differential evolution, in Proc. 1996 IEEE Int. Conf. Evolutionary Computation, Nagoya, Japan, IEEE Press, New York, 1996, 842–844.

[30] Price, K.; Storn, R.; Lampinen, J.: Differential evolution – A Practical Approach to Global Optimization. Springer, Berlin, 2005.

[31] Das, S.; Mandal, D.; Kar, R.; Ghoshal, S.P.: Application of differential evolution with best of random mutation strategy on asymetric location only synthesis of broadside circular antenna array, in *SEMCCO 2012*, Bhubaneswar, Odisha, India, December 2012.

[32] Liu, J.; Lampinen, J.; Matousek, R.; Osmera, P.: Adaptive parameter control of differential evolution, in Proc. Mendel, 8-th Int. Conf. Soft Computing, 2002, 19–26.

[33] Lin, C.; Quing, A.: Synthesis of unequally spaced antenna arrays by a new differential evolution algorithm. Int. J. Commun. Netw. Inf. Secur. (IJCNIS), **1** (1) (2009), 20–25.

[34] Roscca, P.; Oliveri, G.; Massa, A.: Differential evolution as applied to electromagnetics. IEEE Antennas Propag. Mag., **53** (1) (2011), 38–49.

[35] Kurup, D.G.; Himdi, M.; Rydberg, A.: Synthesis of uniform amplitude unequally spaced antenna arrays using the differential evolution algorithm. IEEE Trans. Antennas Propag., **51** (9) (2003), 2210–2217.

[36] Lin, C.; Qing, A.; Feng, Q.: Synthesis of unequally spaced antenna arrays by using differential evolution. IEEE Trans. Antennas Propag., **58** (2010), 2553–2561.

[37] Zhang, X.; Liu, S.: Differential evolution without the scale factor F. Acta Eelctron. Sin., **36** (2009), 1318–1323.

[38] Qin, A.K.; Huang, V.L.; Suganthan, P.N.: Differential evolution algorithm with strategy adaptation for global numerical optimization. IEEE Trans. Evol. Comput., **13** (2) (2009), 398–417.

[39] Mandal, S.; Ghoshal, S.P.; Kar, R.; Mandal, D.: Differential evolution with wavelet mutation in digital FIR filter design. J. Optim. Theory Appl., **155** (1) (2012), 315–324.

[40] Yang, X.S.: Firefly algorithms for multimodal optimization, In Proc. 5th Int. Conf. Stochastic Algorithms: Foundations and Applications, SAGA 2009, LNCS-Springer, vol. 5792, 2009, 169–178.

[41] Basu, B.; Mahanti, G.K.: Firefly and artificial bees colony algorithm for synthesis of scanned and broadside linear array antenna. Prog. Electromagn. Res. B, **32** (2011), 169–190.

[42] Yang, X.S.; Hosseini, S.S.; Gandomi, A.H.: Firefly algorithm for solving non-convex economic dispatch problems with valve loading effect. Appl. Soft Comput., **12** (3) (2012), 1180–1186.

[43] Yang, X.S.; Deb, S.: Eagle strategy using Levy walk and firefly algorithms for stochastic optimization, Nature Inspired Cooperative Strategies for Optimization (NICSO). Stud. Comput. Intell., **284** (2010), 101–111.

[44] Yang, X.S.: Multiobjective firefly algorithm for continuous optimization. Eng. Comput., **29** (2) (2013), 175–184.

Gopi Ram passed his B.E. degree in Electronics and Telecommunication Engineering, from Government Engineering College, Jagdalpur, Chhattisgarh, India in the year 2007. He received the M.Tech. degree from the National Institute of Technology, Durgapur, West Bengal, India in the year 2011. At present, he is attached with the National Institute of Technology, Durgapur, West Bengal, India, as Institute Ph.D. Research Scholar in the Department of Electronics and Communication Engineering. His research interest includes Array Antenna design via Evolutionary Computing Techniques.

Durbadal Mandal passed his B.E. degree in Electronics and Communication Engineering, from Regional Engineering College, Durgapur, West Bengal, India in the year 1996. He received the M. Tech. and Ph.D. degrees from the National Institute of Technology, Durgapur, West Bengal, India in the years 2008 and 2011, respectively. Presently, he is attached with the National Institute of Technology, Durgapur, West Bengal, India, as Assistant Professor in the Department of Electronics and Communication Engineering. His research interest includes Array Antenna design and filter Optimization via Evolutionary Computing Techniques. He has published more than 180 research papers in International Journals and Conferences.

Rajib Kar passed his B.E. degree in Electronics and Communication Engineering, from Regional Engineering College, Durgapur, West Bengal, India in the year 2001. He received the M. Tech. and Ph.D. degrees from the National Institute of Technology, Durgapur, West Bengal, India in the years 2008 and 2011, respectively. At present, he is attached with the National Institute of Technology, Durgapur, West Bengal, India, as an Assistant Professor in the Department of Electronics and Communication Engineering. His research interest includes VLSI signal Processing and Filter optimization via Evolutionary Computing Techniques. He has published more than 190 research papers in International Journals and Conferences.

Sakti Prasad Ghoshal passed his B.Sc. and B.Tech., degrees in 1973 and 1977, respectively, from Calcutta University, West Bengal, India. He received his M.Tech. degree from I.I.T (Kharagpur) in 1979. He received his Ph.D. degree from Jadavpur University, Kolkata, West Bengal, India in 1992. At present he is Acting Professor of the Electrical Engineering Department of N.I.T. Durgapur, West Bengal, India. His research interest areas are: Application of Evolutionary Computing Techniques to Electrical Power systems, Digital Signal Processing, Array antenna optimization and VLSI. He has published more than 225 research papers in International Journals and Conferences.

https://doi.org/10.1017/S175907871300086X Published online by Cambridge University Press