APPENDIX-‘A’: OPTIMIZATION TECHNIQUES

The search strategies for this research work have been implemented using GA and PSO algorithms. The fundamental reason for selecting a technique is based on its simplicity, ease of implementation, and adaptability for the problem at hand.

A1. Genetic Algorithm (GA)

A1.1 Basic GA:

The concept of genetic algorithm (GA) was introduced for the first time by Holland [Holland (1975)]. GA is a “global” numerical optimization method, patterned on the basis of natural process of genetic recombination and evaluation. It encodes each parameter into a binary sequence, called a gene. A set of genes is called a chromosome. These chromosomes undergo natural selection mating and mutation to arrive at the final optimal solution. The algorithm begins with a large list of random chromosomes. The fitness functions are evaluated for each chromosome. The fitness of this chromosome is the maximum relative SLL of its associated far-field pattern which is minimized during the search. Chromosomes are ranked from most fit to least fit according to their respective fitness values. Unacceptable chromosomes are discarded. Genes that survive become parents and by swapping some of their genetic properties, two new offspring can be produced. The total number of chromosomes remains constant after each iteration. Random mutations are performed for a small percentage, typically of the order of 1% of bits in the list of chromosomes mutating per iteration. Mutation does not occur in the final iteration. The implementation of the algorithm has been done using MATLAB ver.8 which involves the following steps:

- Initial population of N number of chromosomes are randomly generated which consists of strings of binary digits for thinned antenna array and variable inter-element spacings over the range 0.5λ - 1.5λ for non-uniformly or randomly spaced array.
- Fitness or cost functions for all the chromosomes are computed and they are ranked from most fit to least fit according to their respective fitness or cost values. Unacceptable 50% chromosomes are discarded in the current population.
- Chromosomes that survive become parents and by swapping some of their genetic properties, two new offsprings are produced. Random mutations are performed for small percentage of genes (e.g. 1% of the genes).
- The above steps were repeated multiple times to obtain the optimal configurations of thinned/non-uniformly spaced antenna arrays corresponding to the lowest possible PSLL.

A1.2 Modified Binary Coded GA (MBC-GA)

The basic GA was developed by Holland [Holland (1975)], and further extended to functional optimization by many researchers in this framework including Goldberg [Goldberg *et al.* (1989)] and Davis [Devis (1991)]. This algorithm or other global optimization techniques, such as particle swarm optimization (PSO), differential evolution (DE) algorithm, and ant colony optimization (ACO) are very common in behaviour, and simply employing them to any optimization problem would not bring forth the desired solution. In GA, size of solution space, the type of crossover, and the mutation rate have significant effect on search performance. Towards this end, many researchers including Goldberg *et al.* [Goldberg *et al.* (1989)] suggested that a solution space that is 2 to 3 times larger than the total chromosome size (total number of bits) should be preferred, while others, such as Weile *et al.* [Weile *et al.* (1997)] suggested that large population should be employed. Therefore, a wise selection of algorithm control parameters and operators, the ones which highly conform to specific problem, is really necessary for efficacious execution [Goldberg *et al.* (1989), and Goldberg *et al.* (1991)]. Taking the aforesaid aspects into consideration and inferring the motivation from the affirmative characteristics of classical GA, a modified binary coded GA (MBC-GA) is investigated by the author. Particularly, the MBC-GA is investigated to accomplish following improvements, which would make it a more efficient synthesis tool compared to state-of-the-art thinning techniques.

- An innovative robust randomization strategy for crossover operation which ameliorates convergence rate by allowing for adequate amount of randomization and diversity to the solution space
- An unexampled systematic mutation strategy that leads the MBC-GA much dynamism, soundness, and rapidity in attaining desired solution
- Higher computational efficiency

The novel advancements as described below in Sub-sections are introduced to enhance the exploration capability and tractability of the GA in attaining a fairly good estimation of PSLL at less computational complexity.

A1.2.1 Crossover and Mutation Operations for the Synthesis of TLA Arrays

The crossover operation is performed to interchange the gene content amongst chromosomes, which permits us to ameliorate the new generation (offspring production) and convergence rate of optimization process [Devis (1991), and Goldberg *et al.* (1991)]. Several strategies are reported in the literature to execute crossover operation. Mutation is carried out to bring forth some random changes in the solution space to prevent local trap for search algorithm [Goldberg *et al.* (1991), Man *et al.* (1999)]. This is simply practiced by exchanging the binary bit at the mutating position. It is to be mentioned that with the proper utilization of these basic genetic operators, it is possible to ameliorate the search competency of optimization technique [Devis (1991)].

Towards this perspective, an innovative robust randomization strategy for crossover operation and an unexampled systematic mutation strategy as described below in detail are employed to enhance the solution quality in terms of PSLL reduction, computational complexity, and stability.

A mathematical relation given in Eq. (a.1) is formulated which generates two random numbers within the given array size.

$$C_{nTLA} = \text{random (I, 2)}, \quad \left(\frac{N_{TLA}}{2} < I \leq N_{TLA} - 1\right) \tag{a.1}$$

where C_{nTLA} is the random variable used to generate two random numbers and N_{TLA} is total number of genes (columns) in the parent chromosome. During operation, genes within these randomly generated points are exchanged in selected parent chromosomes.

It was assumed that P_{iTLA} and P_{iTLA+1} were two parent chromosomes, which took take part in generation of new chromosomes called offspring. After performing crossover operation, the offspring O_{iTLA} and O_{iTLA+1} are produced through the expressions listed in Eqs. (a.2) and (a.3) as:

$$O_{iTLA} = \{(P_{iTLA})_{1\le n< c_{nTLA}(1)}, (P_{iTLA+1})_{c_{nTLA}(1)\le n\le c_{nTLA}(2)}, (P_{iTLA})_{c_{nTLA}(2)<n\le N_{TLA}}\} \quad (a.2)$$

$$O_{iTLA+1} = \{(P_{iTLA+1})_{1\le n< c_{nTLA}(1)}, (P_{iTLA})_{c_{nTLA}(1)\le n\le c_{nTLA}(2)}, (P_{iTLA+1})_{c_{nTLA}(2)<n\le N_{TLA}}\} \quad (a.3)$$

where n=1, 2, 3...... N_{TLA}.

The purpose of mutation operation is that of restoring the left out or unexplored genetic material in the solution space to avoid the untimely convergence. It indiscriminately changes one or more gene of selected chromosome, thereby enhancing the operational changeability of the solution space. To achieve this, the author investigated an unexampled systematic mutation strategy as explained below in detail.

The mutation probability is expressed by equation given below in Eq. (a.4)

$$C_{XTLA} = C_{TTLA} \times 0.0025 \quad (a.4)$$

C_{TTLA} is total number of newly generated chromosomes and C_{XTLA} is the number of offsprings which get selected with a probability of 0.0025 to undergo mutation operation. The chromosomes $\left((O_{TLA})_{1\le x\le C_{XTLA}}\right)$ which get selected from C_{TTLA} are determined by the formula given in Eq. (a.5).

$$(O_{TLA})_{1\le x\le C_{xTLA}} = \text{random}\ (j, C_{XTLA}), (1 < j \le C_{TTLA}) \quad (a.5)$$

In the $(O_{TLA})_{1\le x\le C_{xTLA}}$ offsprings undergoing mutation operation, the gene which is randomly mutated is determined by using Eq. (a.6)

$$\mathrm{M_{PTLA}} = \mathrm{random}\{(\mathrm{O_{TLA}})_{1\le x\le C_{xTLA}}\,(\mathrm{k},1)\}, \left(\frac{\mathrm{N_{TLA}}}{2} \le \mathrm{k} \le \mathrm{N_{TLA}}\right) \qquad \text{(a.6)}$$

A1.2.2 Crossover and Mutation Operations for the Synthesis of TPA Arrays

In this case, extending the concepts of crossover operation as described before in sub-section A1.2.1 for TLA arrays, a random variable ($\mathrm{C_{nTPA}}$) used to generate two random numbers to execute the crossover operation in TPA arrays is given as

$$\mathrm{C_{nTPA}} = \mathrm{random}\ (\mathrm{L}, 2), \quad \left(\frac{\mathrm{N_{TPA}}}{2} < L \le \mathrm{N_{TPA}} - 1\right), \forall\ \mathrm{M_{TPA}} \qquad \text{(a.7)}$$

The generation of new chromosomes called offspring are given as

$$\mathrm{O_{iTPA}} = \left\{ \begin{array}{r} (\mathrm{P_{iTPA}})_{\forall\, \mathrm{M_{TPA}}, 1\le d< c_{nTPA}(1)}, (\mathrm{P_{iTPA+1}})_{\forall\, \mathrm{M_{TPA}},\ c_{nTPA}(1)\le d\le c_n(2)} \\ , (\mathrm{P_{iTPA}})_{\forall\, \mathrm{M_{TPA}}, c_{nTPA}(2)<d\le \mathrm{N_{TPA}}} \end{array} \right\} \qquad \text{(a.8)}$$

$$\mathrm{O_{iTPA+1}} = \left\{ \begin{array}{r} (\mathrm{P_{iTPA+1}})_{\forall\, \mathrm{M_{TPA}}, 1\le d< c_{nTPA}(1)}, (\mathrm{P_{iTPA}})_{\forall\, \mathrm{M_{TPA}},\ c_{nTPA}(1)\le d\le c_{nTPA}(2)}, \\ (\mathrm{P_{iTPA+1}})_{\forall\, \mathrm{M_{TPA}}, c_{nTPA}(2)<d\le \mathrm{N_{TPA}}} \end{array} \right\} \qquad \text{(a.9)}$$

where $\mathrm{M_{TPA}}$ is total number of rows and $\mathrm{N_{TPA}}$ is total number of columns in UE-TPA array. The mutation is carried out using the expression given below in Eq. (a.10)

$$\mathrm{C_{XTPA}} = \mathrm{C_{TTPA}} \times 0.025 \qquad \text{(a.10)}$$

where $\mathrm{C_{XTPA}}$ is the number of chromosomes that get selected from the entire newly generated offspring .i.e. $\mathrm{C_{TTPA}}$ which has to undergo mutation. The chromosomes, which get selected are given as:

$$(\mathrm{O_{TPA}})_{1\le x\le C_{xTPA}} = \mathrm{random}\ (\mathrm{S}, \mathrm{C_{XTPA}}), \quad (1 < \mathrm{S} \le \mathrm{C_{TTPA}}) \qquad \text{(a.11)}$$

From each of the selected chromosomes, the gene that has to be mutated is found out through the point of intersection of the randomly selected row and column as

$$M_{PTPA} = \{\text{random}(q,1), \text{random}\ (r,1)\},\ \left(\frac{M_{TPA}}{2} < q \leq M_{TPA}\right), \left(\frac{N_{TPA}}{2} < r \leq N_{TPA}\right) \quad (a.12)$$

A1.2.3 Implementation of the Proposed MBC-GA

The execution of the MBC-GA algorithm is done using MATLAB ver. 8.0, which performs following steps as empathized from the flowchart depicted in Fig. a-1.

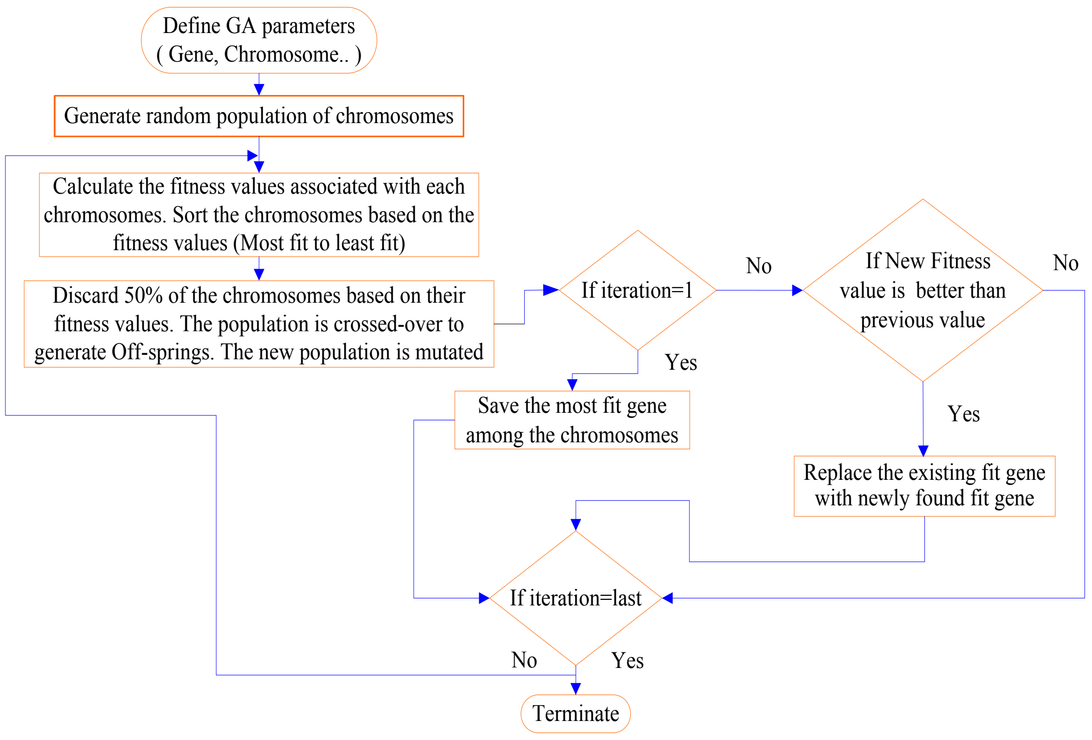

Define GA parameters (Gene, Chromosome..)
Generate random population of chromosomes
Calculate the fitness values associated with each chromosomes. Sort the chromosomes based on the fitness values (Most fit to least fit)
Discard 50% of the chromosomes based on their fitness values. The population is crossed-over to generate Off-springs. The new population is mutated
If iteration=1
No
If New Fitness value is better than previous value
No
Yes
Save the most fit gene among the chromosomes
Yes
Replace the existing fit gene with newly found fit gene
If iteration=last
No
Yes
Terminate

Fig. a-1: Flow chart of MBC-GA.

- The commencing population of chromosomes is randomly generated wherein each chromosome consists of N- and M×N-genes in linear and planar arrays respectively. Each gene is related to amplitude excitation of individual elements in an array i.e. gene having value ‘1’ means that the element is ‘ON’ or ‘active’ and ‘0’ means that is ‘OFF’ or ‘inactive’.

- Fitness functions for all the chromosomes are calculated as defined for problem at hand and they are graded from most appropriate to least appropriate in accordance with their respective fitness values. The roulette wheel selection is employed where the more appropriate chromosomes have more opportunity to get opted and permitted to mate.
- The opted chromosomes are regarded as parents to generate new chromosomes, known as offspring by swapping some of their genetic properties using crossover process as explained in above sections A1.2.1 and A1.2.2
- The repopulated offspring are mutated where mutation process reported before in sections A1.2.1 and A1.2.2 is followed to ensure that the algorithm is not trapped in the same solution space.
- The foregoing steps are repeated multiple times to determine the optimal configurations of thinned/non-uniformly spaced linear or planar antenna arrays for single trials. Such few trials are performed to examine the stability.

A1.3 Improved Binary Coded GA (IBc-GA)

GA offers several advantages over other optimization techniques which have made it immensely popular to solve many electromagnetic problems. These include its simple implementation, its practicability when the population is quite large and large number of variables is associated, and randomizations of operators which contribute diversity to the solution space ensuring ameliorate exploration [Man *et al.* (1999)]. Keeping aforementioned aspects in view and deducing the motivation from the favorable properties of standard GA, an improved binary coded GA designated as IBc-GA is investigated by the author. Novel advancements as described below are introduced to enhance the exploration capability and tractability of the GA in attaining fairly good estimation of PSLL at affordable computational complexity. It is felt that through the utilization of such advancements, IBc-GA can be made a more efficient synthesis tool as compared to standard GA as well as other synthesis techniques.

Towards this objective, we investigated an improved genetic algorithm based optimization technique designating as IBc-GA in which following advancements are introduced to make it more efficient synthesis tool as compared to standard GA and other synthesis techniques available in literature.

- A novel multi-segment chromosome arrangement which enables the IBc-GA with more degree of freedom and less complexity during optimization through the reducing number of variables for optimization
- A new robust randomization approach for crossover operation, which improves the offspring production and convergence rate of maximization process by rendering sufficient amount of randomization and diversity to the population
- A novel systematic mutation approach that makes the IBc-GA more vigorous, efficacious, and rapid in attaining good quality solution

A1.3.1 Novel Arrangement of Chromosomes

A multi-segment based chromosome arrangement consisting of $N \times M$-genes as shown in Fig. a-2 is considered for the present study. A portion of chromosomes, which do not take part in optimization, is indicated with black color genes while the segment of chromosomes, which are involved in optimization, is shown with gray color genes. This implies that the thinning operation happens only on the elements near the periphery without disturbing the core of the antenna array. It is considered due to the fact that for better control of PSLL, central region of array should have uniform element density and the edges of the array should have variable density. In accordance with this, application of optimization process is concerned with the elements in the outer variable density region of the array. The mathematical representation of such chromosomes' arrangement can be formulated by the expression given below in Eq. (a.13).

$$C_i = \{ [\text{fixed ones} (T_{px}, T_{py})], \text{random}(0,1), N - T_{px}, M - T_{py}] \} \qquad (a.13)$$

where C_i is the i^th chromosome, T_{px} is thinning point along X-direction, T_{py} is thinning point along Y-direction. N and M are respectively the total number of elements along X- and Y-axes of the array. Such thinning points are used to separate two segments in X-and Y-directions. It can be interpreted that upto thinning points from centre, elements' density is fixed while beyond thinning points towards edges, elements' density varies and thinning operation is applied only in this region.

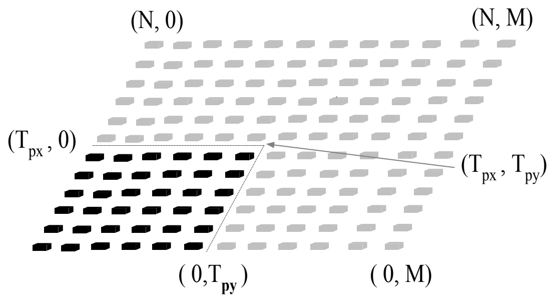


Fig. a-2: A multi-segments based $N \times M$ −genes chromosome arrangement representing one quadrant (X-positive and Y-positive plane) of the 2N×2M-elements TPA array.

It was observed after several numerical simulations that depending upon the array size, the best solutions are obtained when the thinning point values (T_{px} and T_{py}) lie in the ranges $0.3N \le T_{px} \le 0.8N$ and $0.3M \le T_{py} \le 0.8M$. For smaller size arrays, these values tend towards minimum value of aforementioned limit, however, for larger size arrays , they tend towards the maximum value of limit. This is probably due to the fact that if the array size is smaller, optimization algorithm has got less degrees of freedom to lower the PSLL and as the array size increases, the complexity increases for lowering the PSLL. Therefore, through the implementation of such chromosome arrangement in the proposed IBc-GA optimizer, it is possible to reduce the number of elements involved in optimization and successively the associated computational complexity.

A1.3.2 Crossover and Mutation Operations

It is well known that the crossover operator is mainly accountable to search for an optimum solution during optimization. It involves exchange of genetic material between parents chromosomes to generates offspring i.e. child chromosomes and takes into account betterment in offspring production and convergence rate of the optimizer. Towards this end, a new robust randomization approach for crossover operation was employed, which is described below in details.

It is assumed that $\mathrm{M_i}$, $\mathrm{M_{i+1}}$ and $\mathrm{M_{i+2}}$ are three parents, which are selected to mate and produce new generations referred to as children. Variable $\mathrm{E_n}$ was used to generate two random numbers for performing the crossover operation given as

$$\mathrm{E_n} = \text{random}\ (\mathrm{x}, 2)\,,\ \left\{\left(\frac{\mathrm{N}}{2} < x \leq \mathrm{N}\right)\right\} \qquad (\mathrm{a.14})$$

The selected parents undergo crossover operation as per formulations listed below in Eqs. (a.15), (a.16) and (a.17) to generate new chromosomes called children.

$$\mathrm{Child_i} = \left\{(\mathrm{M_i})_{1\leq \mathrm{r}<\mathrm{E_n}(1), \forall\,\mathrm{Q}}\,, (\mathrm{M_{i+1}})_{\mathrm{E_n}(1)\leq \mathrm{r} \leq \mathrm{E_n}(2),\ \forall\,\mathrm{Q}}\,, (\mathrm{M_{i+2}})_{\mathrm{E_n}(2)<r\leq \mathrm{P},\ \forall\,\mathrm{Q}}\right\} \quad (\mathrm{a.15})$$

$$\mathrm{Child_{i+1}} = \left\{(\mathrm{M_{i+1}})_{1\leq \mathrm{r}<\mathrm{E_n}(1), \forall\,\mathrm{Q}}\,, (\mathrm{M_{i+2}})_{\mathrm{E_n}(1)\leq \mathrm{r} \leq \mathrm{E_n}(2), \forall\,\mathrm{Q}}\,, (\mathrm{M_{i}})_{\mathrm{E_n}(2)<r\leq \mathrm{P}, \forall\,\mathrm{Q}}\right\} \quad (\mathrm{a.16})$$

$$\mathrm{Child_{i+2}} = \left\{(\mathrm{M_{i+2}})_{1\leq \mathrm{r}<\mathrm{E_n}(1),\ \forall\,\mathrm{Q}}\,, (\mathrm{M_{i}})_{\mathrm{E_n}(1)\leq \mathrm{r} \leq \mathrm{E_n}(2),\ \forall\,\mathrm{Q}}\,, (\mathrm{M_{i+1}})_{\mathrm{E_n}(2)<r\leq \mathrm{P},\ \forall\,\mathrm{Q}}\right\} \quad (\mathrm{a.17})$$

where ‘r’ is a random variable, which represents the rows being selected to retain from the parent chromosomes. The newly generated offspring are again sorted in descending order based on their fitness values. Then 75% of the newly generated offspring having low fitness are discarded and the rest 25% are appended to the 50% parent population having the best fitness values.

The purpose of mutation operation is that of restoring lost or unexplored genetic material in the solution space to avoid the premature convergence of GA. It arbitrarily alters one or more component of selected chromosome in order to increase the structural unevenness of the population. In this study, a novel systematic mutation operation as explained below in detail was introduced to meet the aforementioned requirements.

The mutation operation is carried out using the probability given below in Eq. (a.18).

$$\mathrm{M_{mp}} = \mathrm{T_c} * 0.0025 \tag{a.18}$$

where $\mathrm{M_{mp}}$ is the number of chromosomes that get selected from the newly generated children chromosomes, and $\mathrm{T_c}$ is the total number of chromosomes consisting of best fit newly generated children chromosomes and best fit previously stored parent chromosomes. A chromosome is selected at random from the newly generated population to undergo mutation operation which is given by Eq. (a.19)

$$\{\mathrm{Chr_{j+2}}\}_{1\leq \mathrm{j} \leq \mathrm{M_{mp}}} = \mathrm{random}\,(\mathrm{Y}, \mathrm{M_{mp}})_{1<Y\leq \mathrm{T_c}} \tag{a.19}$$

Since the chromosomes are arranged in descending order based on fitness values, the top two chromosomes having the best fit values are not considered for mutation. The rest of the chromosomes in the population participate in the mutation process. From each of the selected chromosomes (Chr), mutation is performed on the gene found out by the point of intersection of the randomly selected row and column by

$$\mathrm{M_{tp}} = \{\mathrm{random}(\mathrm{w}, 1), \mathrm{random}(\mathrm{z}, 1)\}, \begin{cases} \left(\frac{\mathrm{M}}{2} < w \leq \mathrm{M}\right) \\ \left(\frac{\mathrm{N}}{2} < z \leq \mathrm{N}\right) \end{cases} \quad (\mathrm{a}.20)$$

where $\mathrm{M_{tp}}$ represents the mutation point.

The mutation point or the gene that has to be mutated is selected randomly after each iteration i.e. after every iteration, a new chromosome and a new gene in that chromosome are chosen which is mutated during optimization process. Therefore, by implementing this systematic mutation operation, it is possible to make proposed IBc-GA more vigorous, statistically effectual and speedy in convergence. The IBc-GA is implemented using MATLAB ver. 8.0 which involves the following steps as deduced from the flowchart shown in Fig. a-3 below.

- To commence with the algorithm, a population of few chromosomes is generated randomly, where every chromosome consists of M×N number of genes. Each gene possesses an amplitude excitation in an array i.e. gene having excitation ‘1’ means that the element is ‘made capable of radiating’ and ‘0’ means that it is ‘made incapable of radiating’.
- For each individual chromosome, fitness values, using fitness function defined for the problem at hand are calculated, and chromosomes are arranged from most fit to least fit according to their individual fitness values. The roulette wheel selection is used where the better fit chromosomes are chosen over lesser fit chromosomes for further operations.
- The selected chromosomes are called as parents, which generate new chromosomes, known as offsprings or children by replacing some of their genetic properties with

each other using crossover process as explained before in Eqs. (a.15), (a.16), and (a.17).

- The repopulated offsprings undergo mutation process as described in Eqs. (a.18), (a.19), and (a.20). This is followed to ensure that the algorithm does not get stuck in the local minima. Hence, it improves the global search ability of the algorithm.
- The aforementioned steps are iterated multiple times in each trial and few such independent trials are used to obtain the desired solution.

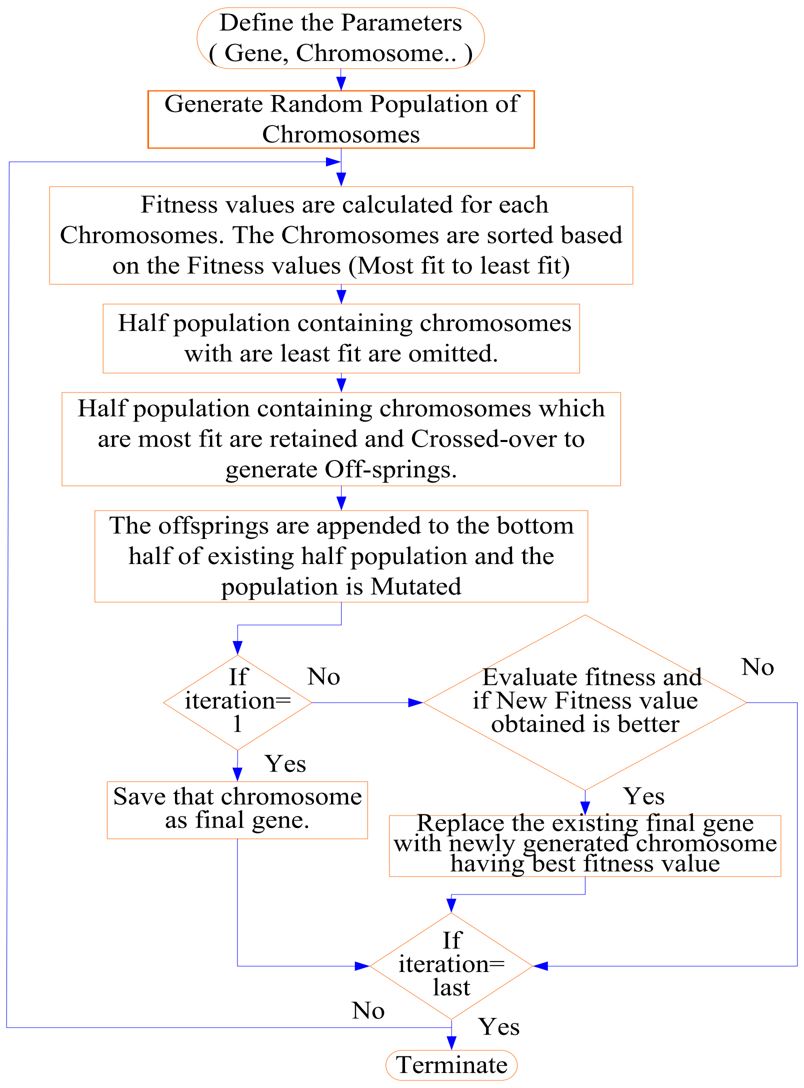

Define the Parameters (Gene, Chromosome..)
Generate Random Population of Chromosomes
Fitness values are calculated for each Chromosomes. The Chromosomes are sorted based on the Fitness values (Most fit to least fit)
Half population containing chromosomes with are least fit are omitted.
Half population containing chromosomes which are most fit are retained and Crossed-over to generate Off-springs.
The offsprings are appended to the bottom half of existing half population and the population is Mutated
If iteration= 1
No
Yes
Save that chromosome as final gene.
Evaluate fitness and if New Fitness value obtained is better
No
Yes
Replace the existing final gene with newly generated chromosome having best fitness value
If iteration= last
No
Yes
Terminate

Fig. a-3: Flow chart of the proposed IBc-GA.

A2. Particle Swarm Optimization (PSO)

A2.1 Basic PSO

The particle swarm optimization (PSO) is a nature based probabilistic optimization technique, which emulates the social behaviour of the birds called as particles [Kennedy and Eberhart (1995)]. Entire collection of particles is called swarm. During the search, each particle adapts its position according to its own experience and the experience of neighbouring particles, making use of the best position experienced by itself ("personal best'') and its neighbours ("global best''). The basic PSO technique updates the velocity and position of each particle according Eqs. (a.21) and (a.22) respectively [Wang *et al.* (2012)]

$$v_i(t+1) = v_i\ (t) + c_1\ r_1 \left[p_i(t) - x_i(t) +\ c_2\ r_2\left[p_g(t) - x_i(t)\right]\right] \qquad (a.21)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \qquad (a.22)$$

where i = 1…….N, N represents the number of particles, c_1 and c_2 are the acceleration constants, r_1 and r_2 are two random numbers within the range [0, 1], v_i (t) and x_i (t) are the velocity and position of the ith particle at tth iteration respectively, $p_i(t)$ is the best position of ith particle at tth iteration, also called ("personal best''), and "global best'' $p_g(t)$ is the best position found by the swarm at tth iteration.

A Flow chart depicting the general PSO Algorithm is shown in Fig. a-4. In the present study, PSO algorithm is implemented using MATLAB ver. 8.0. The numerical factors chosen for Eq. (a.21) are: c_1 = 0.4, c_2 = 0.6, and r_1 and r_2 are two random numbers between 0 and 1.

A2.2 Inertia Weighted Particle Swarm Optimization (IW-PSO)

The IW-PSO algorithm [Khodier and Christodoulou (2005)] is proposed to solve the problem at hand. IW-PSO is a variant of standard PSO algorithm in which the velocity of the current iteration is multiplied with an inertia weighted factor to ensure convergence controls of momentum of the particle. The algorithm is a population based stochastic optimization technique in which each swarm number is called a particle that represents a potential solution. The swarm initially consists of a population of random particles and the Algorithm depends on the social interaction among independent particles during their search for the optimum solution. A swarm of particles moves through the problem space. During its movement, each particle aligns its search direction based on its own best position, which is assorted with best solution (Pbest) it has attained so far, and the global best position, which is assorted with best solution (Gbest) obtained up to now by any particle in a swarm. For a swarm of N particles traversing a D-dimensional space, the velocity and position of each particle are updated as [Chen *et al.* (2006)]

$$v_i^d(t+1) = \omega \times v_i^d(t) + c_1 \times r_{1,i}(t) \times \left[p_i^d - x_i^d(t)\right] + c_2 \times r_{2,i}(t) \times \left[p_g^d - x_i^d(t)\right] \qquad (a.23)$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \qquad (a.24)$$

where d = 1, 2, 3⋯, D represents the particle's dimension index, i = 1, 2, 3 ⋯, and N_p is the particle index. The constants c_1 and c_2 are called the cognitive and social parameters. The variables v_i^d and x_i^d are the velocity and position of the ith particle corresponding to its dth dimension while p_i^d and p_g^d are the particle's local best and the swarm's global best positions for the dth dimension respectively. The variables $r_{1,i}$ and $r_{2,i}$ are drawn from a uniform ergodic distribution [0, 1] and are the sources of stochasticity in the search conduct of the swarm while ω is an inertia weight pertaining to the distribution [0, 1].

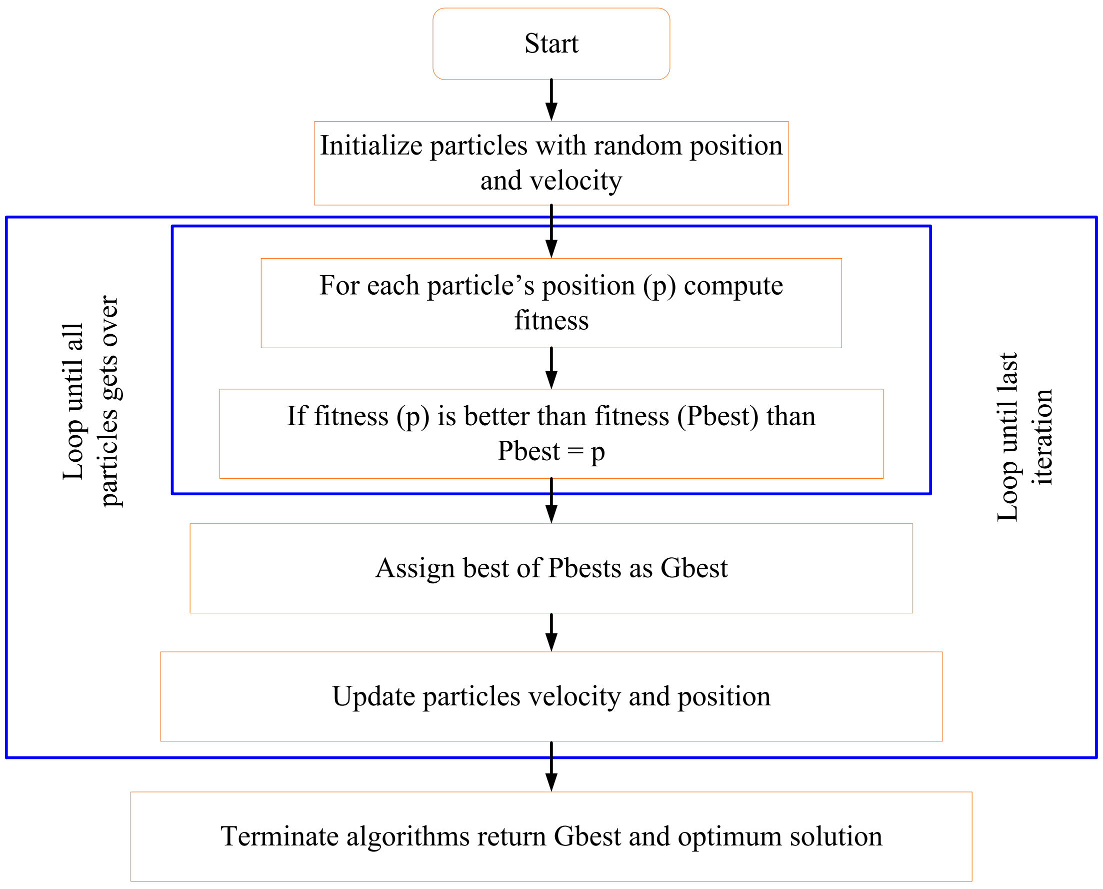

Start
Initialize particles with random position and velocity
For each particle's position (p) compute fitness
If fitness (p) is better than fitness (Pbest) than Pbest = p
Assign best of Pbests as Gbest
Update particles velocity and position
Terminate algorithms return Gbest and optimum solution
Loop until all particles gets over
Loop until last iteration

Fig. a-4: Flow chart showing the basic steps involved in PSO.

APPENDIX-‘B’: GUI TOOL FOR SPARSE ANTENNA ARRAY OPTIMIZATION

B1. Graphical User Interface (GUI) Development

The concept of GUI is came into existence in order to provide easiness to the user who may or may not require to know the architecture of optimization programs using GA or PSO algorithms and their implementation in MATLAB. The user should be able to directly get into the problem and evaluate/optimize same as per the required application. Towards this end, it is aimed to develop visual formulation of the problem in MATLAB ver. 8.0. The optimization programs perform search and computation and graphical presentation of the obtained results is viewed as plots.

B2. Realization of Synthesis and Optimization Tool for Sparse Antenna Arrays

The GA and PSO based optimization tool as shown in Fig. b-1 perform search and computation of mainly three types of sparse antenna array problems as mentioned below and displays the radiation patterns of optimized antenna arrays along with the obtained parameters.

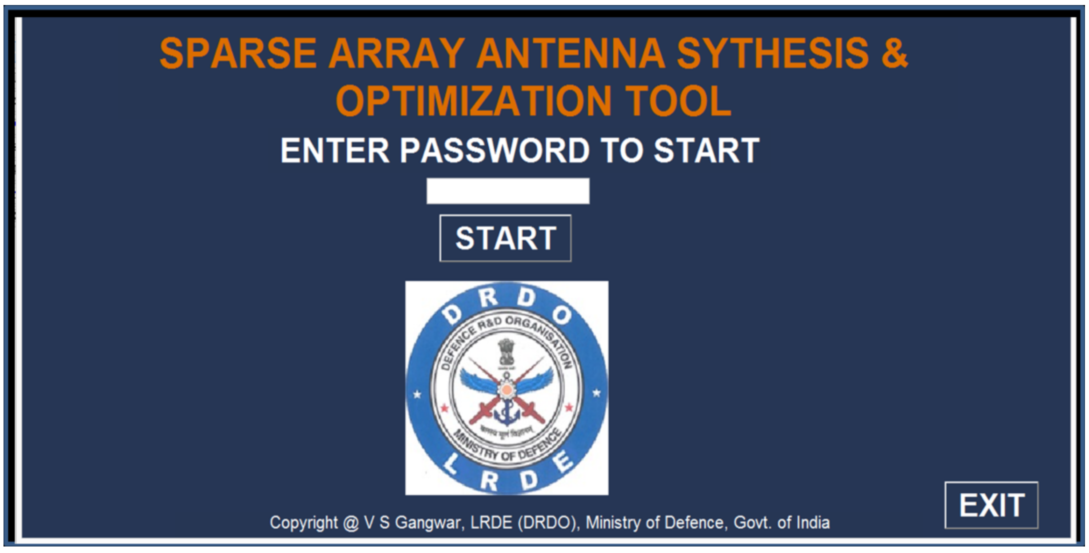
- o Optimization of thinned antenna arrays
- o Optimization of non-uniformly spaced antenna arrays
- o Optimization of shared aperture antenna arrays

B2.1 Thinned Antenna Array Optimization

The Thinned antenna array optimization tool is shown in Fig. b-2. The tool is having two options for selecting the array configurations as listed below. Similarly, the non-uniformly spaced and shared aperture arrays are also providing options for selecting the array configurations.

- Thinned linear antenna array
- Thinned planar antenna array

The input and output parameters window along with optimized radiation patterns for thinned linear and planar antenna arrays are depicted in Figs. b-3 – b-5 respectively.



Copyright @ V S Gangwar, LRDE (DRDO), Ministry of Defence, Govt. of India

Fig. b-1: Main window of sparse antenna array optimization tool.

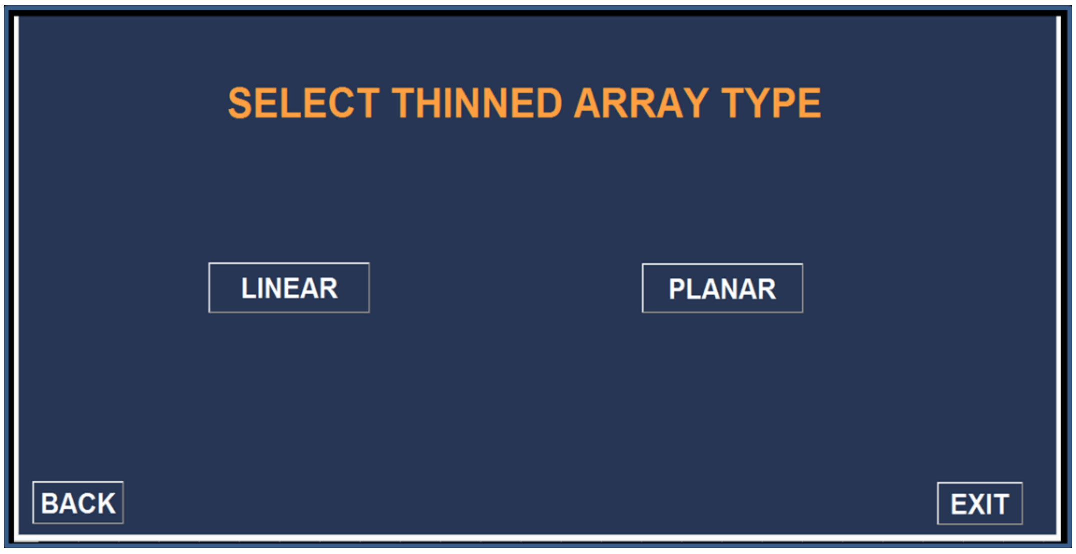


Fig. b-2: Window of thinned antenna array optimization tool.

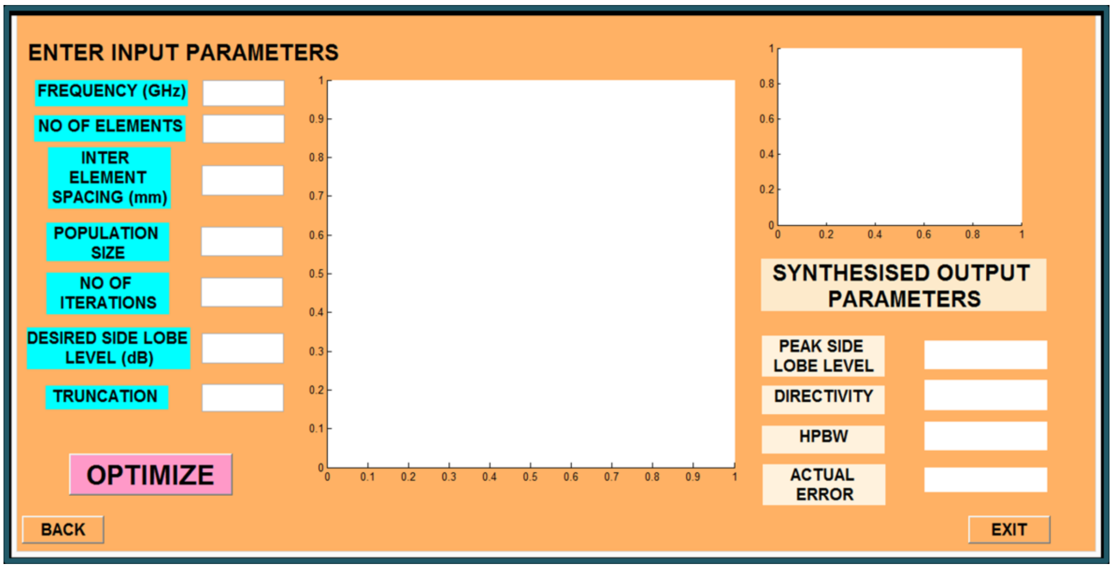
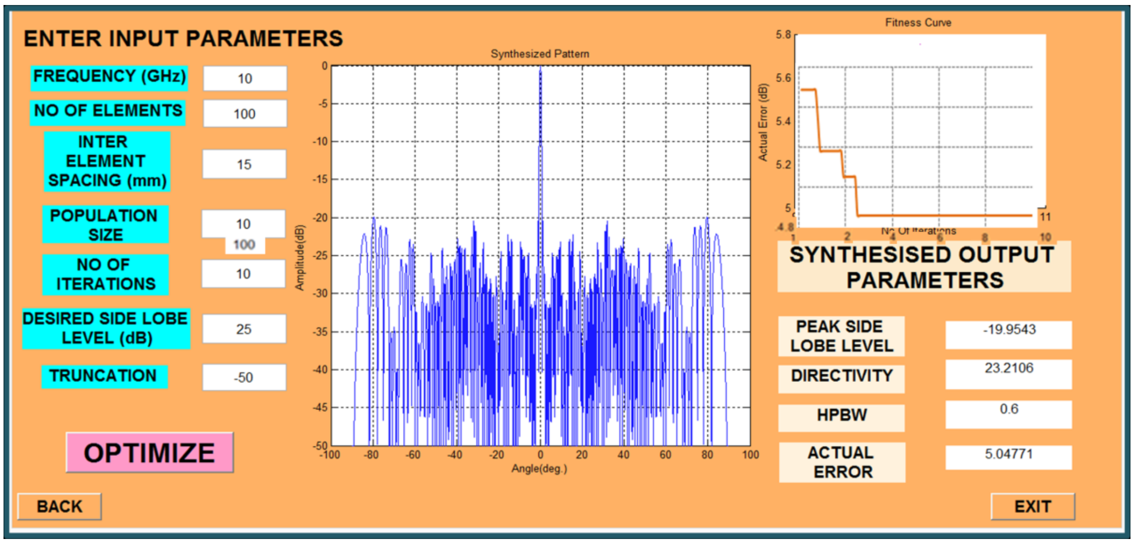


Fig. b-3: Input parameter window for thinned linear antenna array optimization.



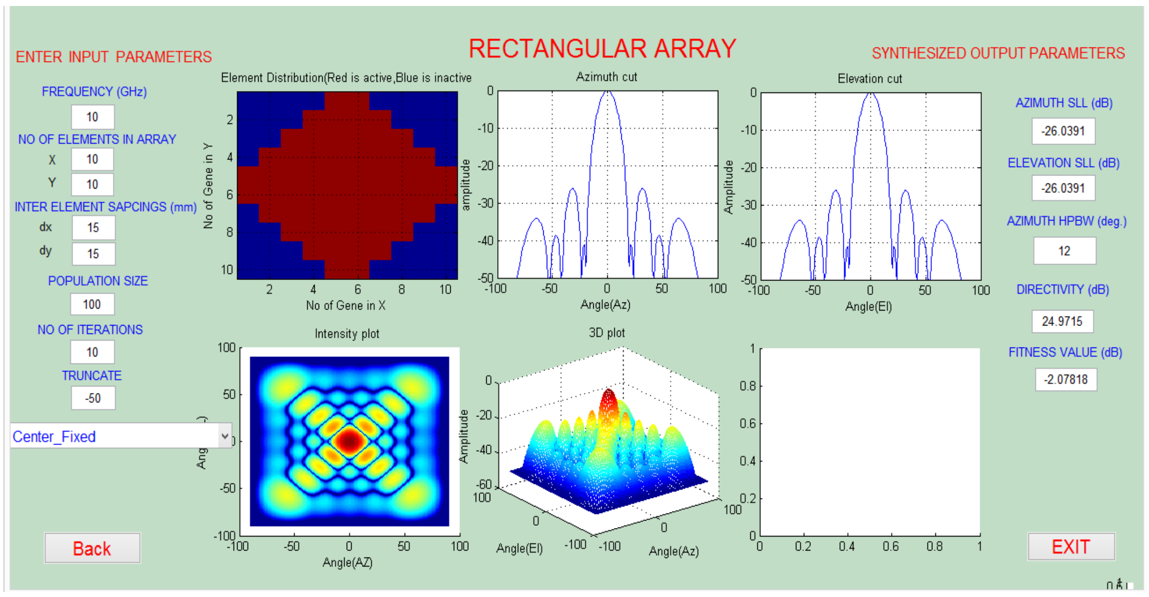
ENTER INPUT PARAMETERS

FREQUENCY (GHz)	10
NO OF ELEMENTS	100
INTER ELEMENT SPACING (mm)	15
POPULATION SIZE	10
NO OF ITERATIONS	10
DESIRED SIDE LOBE LEVEL (dB)	25
TRUNCATION	-50

SYNTHESISED OUTPUT PARAMETERS

PEAK SIDE LOBE LEVEL	-19.9543
DIRECTIVITY	23.2106
HPBW	0.6
ACTUAL ERROR	5.04771

Fig. b-4: Output parameter window for thinned linear antenna array optimization.



RECTANGULAR ARRAY

ENTER INPUT PARAMETERS

FREQUENCY (GHz)	10
NO OF ELEMENTS IN ARRAY X	10
NO OF ELEMENTS IN ARRAY Y	10
INTER ELEMENT SAPCINGS (mm) dx	15
INTER ELEMENT SAPCINGS (mm) dy	15
POPULATION SIZE	100
NO OF ITERATIONS	10
TRUNCATE	-50
	Center_Fixed

SYNTHESIZED OUTPUT PARAMETERS

AZIMUTH SLL (dB)	-26.0391
ELEVATION SLL (dB)	-26.0391
AZIMUTH HPBW (deg.)	12
DIRECTIVITY (dB)	24.9715
FITNESS VALUE (dB)	-2.07818

Fig. b-5: Input and output parameter window for thinned planar antenna array optimization.