Progress In Electromagnetics Research B, Vol. 101, 119–135, 2023

Element Thinning Using Discrete Cat Swarm Optimization for 5G/6G Applications

Sahiti Vankayalapati[1, *], Lakshman Pappula[1], and Debalina Ghosh[2]

Abstract—An efficient method for designing narrow beams having minimum peak side lobe level (PSLL) and maintaining power efficiency (reducing active elements) for 5G/6G base stations with large antenna arrays is proposed. To ensure high efficiency in a multi-dimensional complex nonlinear optimization problem with several constraints, thinning of antenna arrays is considered. For performing exhaustive search on the large number of feasible solutions a novel algorithm named discrete cat swarm optimization (DCSO) is used and is a binary adaptation of real-valued cat swarm optimization (CSO). To testify the efficiency of DCSO a set of standard benchmarked multimodal functions are used. The proposed algorithms exhibit heuristic nature, so the stability of the proposed method has been authenticated by using statistical test. Later the algorithm is applied to the optimization of a large planar antenna array (PAA) of size 10×20 (200 elements) to suppress the PSLL. Furthermore, the results of the synthesis are compared with literature marking low PSLL and convergence speed as pointers. The comparative results delineate the superiority of the DCSO over the existing discrete versioned traditional algorithms with respect to solution accuracy and speed of convergence. DCSO introduces a higher degree of flexibility to the field of binary-valued thinned antenna array synthesis problems.

1. INTRODUCTION

Large antenna arrays (LAAs) are commonplace entities of all key candidate technologies that are being researched for sixth generation networks (6G). Succeeding 5G, the 6G must enhance service accuracy, and base stations must cover large geographical areas using wide aperture antenna arrays. There is a need to exploit electromagnetic fields efficiently to meet the fast-growing challenges of succeeding technology. In 5G systems, the presence of sidelobes in the radiation pattern generated by the antenna array is a major source of problem as they dissipate energy in undesired directions in the transmission end, and they allow energy into the system from unwanted directions at the receiver end. This creates a necessity to design an antenna system that can practically control the sidelobe energy while maintaining a narrow beam pattern. Using antenna array pattern synthesis techniques, a design engineer can choose either the proportions of the array weights or positions of the antenna elements to achieve the desired pattern. Each element in the array receives input through a uniform feed network, and it must be disturbed for controlling the shape of the desired pattern. In practice, where power consumption is a major factor, the wide apertures with large number of elements are considered a drawback due to the mechanical complexity of feed and cost. This disadvantage can be mastered by reducing the array elements and making it energy efficient without losing its radiation properties. Wider apertures can be realized by generating a periodic antenna array. There are several methods to achieve aperiodicity, and one such approach is to modify the element positions. Practical change of positions is a complex

Received 17 May 2023, Accepted 22 June 2023, Scheduled 17 July 2023
* Corresponding author: Sahiti Vankayalapati (sahithi.vankayalapati@gmail.com).
[1] Department of ECE, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India. [2] School of Electrical Sciences, IIT Bhubaneshwar, Odisha, India.

procedure when the number of elements is large, and thinning is one technique for optimizing the element positions [5–17]. Using thinning the physical positions of the elements are untouched, but antenna element can represent itself in two states in either ON or OFF state. The ON condition is represented while feeding the elements with uniform amplitude excitations (energized), and the others are placed in the OFF (de-energized) state by terminating them at matched loads. Due to the practical advantages like limiting the cost, weight, consumption of power (energy efficiency), and reduction in complexity of the feed network, the thinning technique has been exploited for the last five decades to synthesize aperiodic antenna arrays.

Several deterministic optimization methods [1] have been used for thinning. Numerical methods like Downhill simplex, Newtons methods, Least squares, Gaussian Newtons, and Powell's methods [2, 3] optimize only continuous variables and are probable of being stuck at local minima. A few gradient-based techniques [4–6] though efficient have complex numerical calculation as most of the antenna optimizations involve multimodal nonlinear synthesis. Moreover, these procedures require a good initial guess for the algorithm to take off, and they perform one dimensional search in the search space and have large functional evaluations. Selecting optimal array aperture from largely possible combinations becomes difficult as the array gets more populated. This difficulty led the research to mold towards more natural and metaheuristic algorithms. Designing thinned arrays has seen significant improvement due to the use of metaheuristic optimization techniques as they perform global search and can handle large variables in highly nonlinear environment. Nature-inspired techniques come under the category of meta-heuristic algorithms. A few good features like self-repair, self-guidance, and evolution can be used in artificial computing for better performance. Thus, having a good number of advantages nature-inspired optimization is being encouraged in optimization of the antenna arrays.

A brief literature review involving different nature inspired techniques is discussed below. Genetic algorithm (GA) imitates the process of natural evolution [7–11]. A discrete version GA is used in 1994 [7] for large array synthesis with minimum PSLL requirement to drive the algorithm in finding the optimized element positions. Chen et al. 2007 [8] developed a modified real GA (MGA) for attaining 5.26 dB lower PSLL for a planar array of 108 elements than traditional methods. Zhang et al. 2011 [9] used an orthogonal GA for the thinning of a planar array having 20×10 dimension to achieve low PSLL -26.09 dB and -25.09 dB in the 0° and 90° planes. Oliveri and Massa in 2010 [10] used almost difference set method (ADS) and genetic algorithms (GA) called the ADSGA method of optimization for producing narrow band signals, and a PSLL of -20.64 dB was achieved. In 1996, the concept of simulated annealing (SA) [11–13] was introduced to reduce the PSLL. Meijer in 1998 [12] used discrete SA in the synthesis of a planar array of size 20×10 to achieve a PSLL of -24.4 dB from 8484 array combinations. Trucco in 1999 [13] used SA for reducing the number of elements to be energized, and a thinned planar array of 359 elements was simulated to obtain a PSLL of -21.2 dB.

Particle swarm optimization (PSO) [14, 15] which imitates the swarm of fish or particles was used to exploit the variants of algorithm and solve single and multi-objective problems to achieve low PSLL by thinning. Donelli et al. in 2009 [14] combined the deterministic method of Hadamard difference sets and PSO and applied on an array containing 576 elements to attain the PSLL of -18.97 dB. Wang et al. in 2012 [15] introduced Chaotic Binary PSO (CBPSO) where the inertia weight was obtained by chaotic mutation that prevented the early convergence. A planar array having 20×10 dimension was simulated, and the PSLL was -26.39 dB and -26.33 dB for 0° and 90° directions. Another algorithm called Ant colony optimization (ACO) [16] which imitates the colonies created by ants has seen light. In 2006 [16] Quevedo-Teruel and Rajo-Iglesias used ACO to synthesize a planar array with 20×10 dimension and achieved PSLL of -25.76 dB and -25.67 dB in 0° and 90°, respectively. Another variant of ACO is touring ant colony optimization. A few antenna optimization techniques were introduced by using that algorithm. Differential evolution (DE) [17, 18] is another evolutionary algorithm that imitates the mechanism of natural selection. Zhang et al. in 2010 [17] introduced Boolean DE (BDE) algorithm for the thinning of a planar of size 20×10 to attain PSLL of -26.09 dB and -25.09 dB in 0° and 90° directions. Rocca et al. in 2011 [18] expanded the application of DE to problems of electromagnetics. The authors pointed out the novelties of the algorithm. Next in evolutionary computing is Invasive weed optimization (IWO) which imitates the colonization of weeds [19]. Wu et al. in 2015 [20] introduced the concept of improved binary IWO (IBIWO) for thinning of array with circular geometry to reduce the SLL while maintaining specific HPBW. Roy et al. in 2011 [21] presented a mechanism of synthesizing planar and circular arrays using IWO.

A few other algorithms whose goal is analogous to the work in paper are summarized as follows. Singh and Salgotra in 2018 [22] stated that flower pollination algorithm (FPA) was better in reducing the PSLL and placing NULL in undesired directions.Dahi et al. in 2016 [23] investigated the FPA and introduced four variants of binary FPA (BFPA) to solve antenna positioning problems. Binary firefly algorithm (BFA) [24] and Grey wolf optimization (GWO) [25] were used for the synthesis of arrays with low SLL. Cuckoo search algorithm [26] and binary spider monkey algorithm [27] were quite helpful in circular and linear array synthesis. In 2018 [28] Darvish and Ebrahimzadeh utilized an improved fruit fly optimization algorithm to optimize a planar array with 68 elements, and an improvement of 9.36 dB in SLL was observed. Ravipudi and Neebha in 2018 [29] did comparative research by picking up Jaya algorithm and its variants with the main objective of achieving low SLL, while preserving antenna vital parameters.

After having considered the work in literature, a discrete variable optimization method is introduced hereto performing optimization with increased solution accuracy and great speed of convergence. This is done by keeping synthesis on large arrays which are prone to have large number of possible solutions at best interest. In the present work, the thinning of a planar antenna array (PAA) is done by a novel discrete cat swarm optimization algorithm (DCSO). The DCSO is inspired from the cat swarm optimization algorithm (CSO) which duplicates the natural behavior of the cat and was introduced by Chu et al. [30]. CSO has been quite popular for dealing with continuous real valued spaces and 31, 32].

The major contributions of this work are,

1) To propose a novel discrete cat swarm optimization algorithm for single objective binary valued problems.
2) To evaluate the potentiality of DCSO by applying benchmark functions.
3) To apply the proposed DCSO to thinned antenna array synthesis, to shape the sidelobe power.
4) To perform qualitative analysis on DCSO and other algorithms using statistical test.

2. DISCRETE CAT SWARM OPTIMIZATION

The DCSO is a unique algorithm, that imitates the natural behavior of the cat like continuous version of the CSO. The major difference being the position vector is constructed using binary digits instead of real values. All the nature inspired algorithms must be delineated to the behavioral characteristics of one or the other species in order to represent a solution set. For example, the ants' path is used in ACO; PSO solution set is built on the movement of school of fish or flock of birds; in the same way, CSO is modelled on the behavior of cats. The main objective of DCSO is to explain the notion of the adaptive nature of the cats to trace their prey and link their behavior to discrete nature.

The feature of the cat is segregated into two modes namely seeking mode (SM) and tracing mode (TM). Cat invests most of its time in rest and observes the surroundings with high alert. A mathematical field is set in DCSO to represent the functionality of cat along the two modes in a D-dimensional space and resolve various discrete optimization problems. The position vector $X_{i,d} = (X_{i1}, X_{i2}, ..., X_{iD})$ and velocity vector $V_{i,d} = (V_{i1}, V_{i2}, ..., V_{iD})$ represent the position and velocity of the ith cat in solution space which is represented by dimension d, and it extends form 1 to D. In tracing mode, cats quickly spend a lot of energy to trace the prey with quick movements. A parameter called mixed ratio (MR) is set to distribute the cats into SM and TM modes.

2.1. Seeking Mode

This mode is portrayed to represent the cat's resting position, and the cat is alert in observing the surrounding environment. The movement made here towards the next position is calculated and slow. The mode is set to flow in the below steps.

1) K copies of the ith cat are created in SM mode based on the value specified by the Seeking memory pool parameter (SMP). SMP is used to indicate the cat copies being created in SM mode.

2) $(K-1)$ copies among the K copies of the ith cat undergo mutation following the values specified by Counts of dimension to change (CDC) and Mutation probability (MP). The CDC reflects the count

of dimensions that are to be mutated. MP is specific to binary synthesis. Following the CDC, the dimensions (binary value) of the ith cat are mutated. The probability with which each parent cat in each dimension will be mutated is governed by the equation below and is represented by $MP(X_{i,d}^{k})$

$$MP(X_{i,d}^{k}) = 1 + \frac{1}{2}\tanh\left(\frac{-D_{i,d}^{k} - Z}{2}\right) + \frac{1}{2}\tanh\left(\frac{D_{i,d}^{k} - Z}{2}\right) \tag{1}$$

where $K = 1, ..., K-1$ and $d = 1, ..., D$, $X_{i,d}^{k}$ represents the position of the Kth copy of the ith cat in dimension d.

The spread distance generated by the standard gaussian number is indicated by $D_{i,d}^{k}$, and parameter Z is a constant. A Gaussian mutation number initializes the movement of the cat in the positive and negative directions. This implementation allows the cat to make smaller mutations in the parent cat neighborhood. This leads to more organized search in the vicinity of the cat, and the searching capabilities of DCSO are made more promising. The MP is defined in the range [0, 1] by tangent function and is shown in Fig. 1. The mutation probability of each cat is calculated to generate the random number 'rn' within [0, 1] range. Following the calculation of 'rn' the mutation bit is governed by the equations below. In available $K-1$ copies, the kth bit in the ith copy is updated using the equation below.

$$Y_{i,d}^{k} = \begin{cases} 1, & \text{if } \left(rn < \mathrm{MP}(X_{i,d}^{k})\right) \\ 0, & \text{else} \end{cases} \tag{2}$$

$$X_{i,d}^{k+1} = \bmod\left(x_{i,d}^{k} + y_{i,d}^{k}, 2\right) \tag{3}$$

The natural demeanor of the cat is mathematically modeled using the above equations, and the justification and flow of the mechanism are depicted as follows. In the seeking mode, the cat displacement in position is very little as it observes the environment around it. The mathematical equation should create a scope where the position updation is very much little. It can be done when the mutation rate is less. To facilitate this, the mathematical equation is modelled as in Equation (1). The rate of mutation is directly related to the mutation probability. Equation (1) is formulated by using tangent function with Gaussian mutation number. The Gaussian mutation number with a proper value of 'Z' in the tangent function allows lower mutation probability values over larger values. The mutation is said to happen if at least one bit among the string is updated. The displacement of the cat would be large if

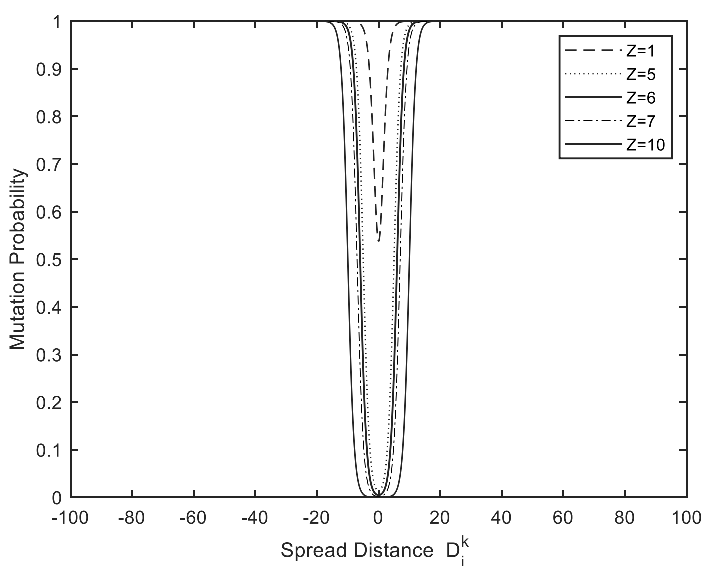


Figure 1. Mutation probability vs spread distance $D_{i,d}^{k}$.

two or more bits in the string is updated/mutated. To limit the larger displacements and the mutation in line with cats behavior, Equations (2) and (3) are developed to make the cat explore the search space systematically around the cats position. The X-OR operation in Equation (3) makes most of the cases limit to 0, while $Y_{i,d}^k$ is made in most of the cases to obey the second condition as the $MP(X_{i,d}^k)$ value span is very small compared with the span of the random number generated between [0, 1].

3) Estimate the fitness value of all cats.

4) Replace the position of the ith cat with the best one from K cats.

2.2. Tracing Mode

This mode is portrayed to present the cats nature while it is tracing the targets rather than resting. The change in positions of the cat is achieved by producing the change in their velocities.

$$V_{j,d}^{g+1} = \omega \cdot V_{j,d}^{g} + C \cdot rn \cdot \left(X_{gbest} - X_{j,d}^{g} \right) \tag{4}$$

where j is the index of a cat in the swarm, d the index of dimension in the cat, g the iteration number, $V_{j,d}^g$ the velocity of the jth cat, C the coefficient for acceleration, ω the weight of inertia, and $rn \in [0, 1]$ a random number. The global X_{gbest} is selected without domination from the external archive that contains cat positions. Using sigmoid limiting transformation S the velocity of the dth bit in the jth cat is updated and represented as below.

$$S\left(V_{j,d}^{g+1}\right) = \begin{cases} \dfrac{1}{1+e^{V_{\max}}} \to 0, & V_{j,d}^{g+1} = -V_{\max} \to \infty, \\ \dfrac{1}{2}, & V_{j,d}^{g+1} = 0, \\ \dfrac{1}{1+e^{-V_{\max}}} \to 1, & V_{j,d}^{g+1} = V_{\max} \to \infty, \end{cases} \tag{5}$$

The sigmoid limiting transformation is used to map the range $[-V_{\max}]$ to the range $\left[\frac{1}{1+e^{V_{\max}}}\right]$ which is a subset of [0, 1]. The updation of the dth bit in the jth cat is updated by the equation

$$X_{j,d}^{g+1} = \begin{cases} 1, & rn < S(V_{j,d}^{g+1}), \\ 0, & rn \geq S(V_{j,d}^{g+1}). \end{cases} \tag{6}$$

The flow of DCSO algorithm is as shown in Algorithm 1.

3. SIMULATION RESULTS FOR BENCHMARK PROBLEMS

Evaluation and simulation of the DCSO are performed using seven standard benchmark functions as listed in Table 1. Benchmarking is widely adopted for any new algorithm to test its performance using certain functions [24]. The functions (f_1, f_2, f_3) [25] are unimodal and contain only one global minimum or no minimum. The functions (f_4, f_5, f_6, f_7) are complex test functions and test the possibility of optimization getting trapped in the local minima. Depending on the problem dimension the count of local minima increases. The results obtained from this synthesis indicate a better performance of the algorithms to solve the above stated problems. The results obtained for DCSO are compared with binary GA (BGA) [33], binary PSO (BPSO) [34], and binary CSO (BCSO) [35]. 20 dimensions are considered for all the functions. The real numbers are represented in binary form by 20 bits in all the experiments. All the algorithms under this synthesis have undergone 50 independent runs, and each independent run having 200000 functional evaluations to maintain a fair comparison. Computational complexity is one metric for comparing the efficiency of stochastic algorithms for solving optimization problems. It can be measured as the number of functions evaluations. The number of function evaluations per generation for a given objective function using DCSO is given as (SM cats $*$ (SMP $-$ 1)) + (TM cats). The computation is done using MATLAB on a PC having 4 GB RAM and an operating frequency of 3 GHz. The control parameters setup of DCSO and other competing algorithms is given in Table 2 which were set following the rules in [7, 36]. Different experiments are carried out to investigate the

Algorithm 1 Discrete Cat Swarm Optimization (DCSO)

procedure
  Set all parameters N, SMP, MP, CDC, SPC, $MaxGen$, initialize a population of cats and assign velocity for every cat(solution)
  Estimate the fitness of each cat
  while (Termination condition is not met) **do**
    Separate the population of cats for seeking mode and tracing mode
    for $i = 1 \leftarrow S_s$ **do** ⊳ S_s is the count of cats in the seeking mode
      for $j = 1 \leftarrow SMP - 1$ **do**
        Select the CDC number of dimensions to vary
        Mutate the selected dimensions according to SRD
        Check the variables' range
        Compute fitness for each mutated solution
        Choose the best one from the mutated solutions and old solution
      end for
    end for
    for $k = 1 \leftarrow S_t$ **do** ⊳ S_t is the number of cats in the tracing mode
      Update the k^{th} cat solution using Equations [4] and [5]
      Check the variables' range
      Compute fitness of the updated cat solution
      Choose the best one from the updated solution and old one
    end for
    Find the best cat solution from the total population
  end while
  return The best cat solution
end procedure

effect of the selection of swarm size, CDC, and MR parameter values on the performance of the DCSO algorithm. The values of the swarm size (15, 25, 50, 75, and 100), CDC (20%, 40%, 60%, 80%, and 100%) and MR (0.2, 0.4, 0.6, 0.8, and 1) are varied to select the best values of the parameters. The values of swarm size 50, CDC 80% and MR 0.8 yield the best results in terms of solution accuracy and convergence speed. The efficiency of these algorithms is recorded by considering mean and standard deviation for all the benchmarked functions.

The results of DCSO alongside the competing algorithms are shown in Table 3. DCSO outperforms other algorithms for f_1 and f_2. f_3 exhibits multimodal property at higher dimensions, and DCSO exhibits better performance over others in terms of the rate of convergence and search space exploitation ability. The graphical representation of the unimodal evaluation is depicted in Fig. 2. The DCSO exhibits improved performance over competing algorithms while running the f_4 to f_7 functions and the output of these functions is represented in Fig. 3. The highlighted values in Table 3 represent the best values, and it can be depicted that the standard deviation of all the functions of DCSO is less than BCSO, BPSO, BGA. Standard deviation indicates the ability of the entity or value to move away from the obtained solution. So, a small value of standard deviation is desired and indicates the stability of the solution. As the purpose of our synthesis is minimization, the lowest possible mean value is desired. It is observed from Table 3 that the mean value for DCSO is better than the other three algorithms. The mean value obtained for f_1 (Sphere) is 5.01e−10 for DCSO, 8.42e−08, 4.19e+02, 3.15e+02 for BCSO, BPSO, BGA, respectively. The mean value obtained for f_6 (Griewank) is 2.26e−10 for DCSO,

Table 1. Unimodal and multimodal benchmark functions.

Name of Function	f	Mathematical representation of function under test	$f(x^*)$	Range of Search
Sphere	$f_1(x)$	$\sum_{i=1}^{D} x_i^2$	0	$[-100, 100\ ]^D$
Axis parallel hyper ellipsoid	$f_2(x)$	$\sum_{i=1}^{D}\left(\sum_{j=1}^{i} x_j^2\right)^2$	0	$[-100, 100\ ]^D$
Rosenbrock's	$f_3(x)$	$\sum_{i=1}^{D-1}(100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2)$	0	$[-10, 10\ ]^D$
Ackley's	$f_4(x)$	$-20\, exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - exp\left(\frac{1}{D}\sum_{i=1}^{D}\sqrt{\cos(2\pi x_i)}\right) + 20 + e$	0	$[-32, 32]^D$
Rastrigin's	$f_5(x)$	$\sum_{i=1}^{D}(x_i^2 - 10\cos(2\pi x_i) + 10)$	0	$[-5.12, 5.12]^D$
Griewank's	$f_6(x)$	$\sum_{i=1}^{D}\frac{x_i^2}{4000} - \prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	0	$[-600, 600\ ]^D$
Weierstrass's	$f_7(x)$	$\sum_{i=1}^{D}\left(\sum_{k=0}^{k\,max}[a^k\cos(2\pi b^k(x_i + 0.5))]\right) - D\sum_{k=0}^{k\,max}[a^k\cos(2\pi b^k \cdot 0.5)]$ $a = 0.5, b = 3, kmax = 20$	0	$[-0.5, 0.5]^D$

Table 2. Parameter composition for the DCSO, BCSO, BPSO and BGA.

DCSO		**BCSO**		**BPSO**		**BGA**	
Parameter	**Tuned value**	**Parameter**	**Tuned value**	**Parameter**	**Tuned value**	**Parameter**	**Tuned value**
Cats at initial stage	50	Cats at initial stage	50	Size of swarm	135	Population	50
SMP	3	SMP	3	c_1	2	CR	0.8
CDC	80%	CDC	20%	ω	Linearly decreases from 0.9 to 0.2	MR	0.2
MR	0.8	PMO	0.2	r_1, r_2	[0, 1]	-	-
ω	Linearly decreases from 0.9 to 0.2	MR	0.8				
c_1	2	ω	Linearly decreases from 0.9 to 0.2				
r	[0, 1]	c_1	2				
-	-	r	[0, 1]				

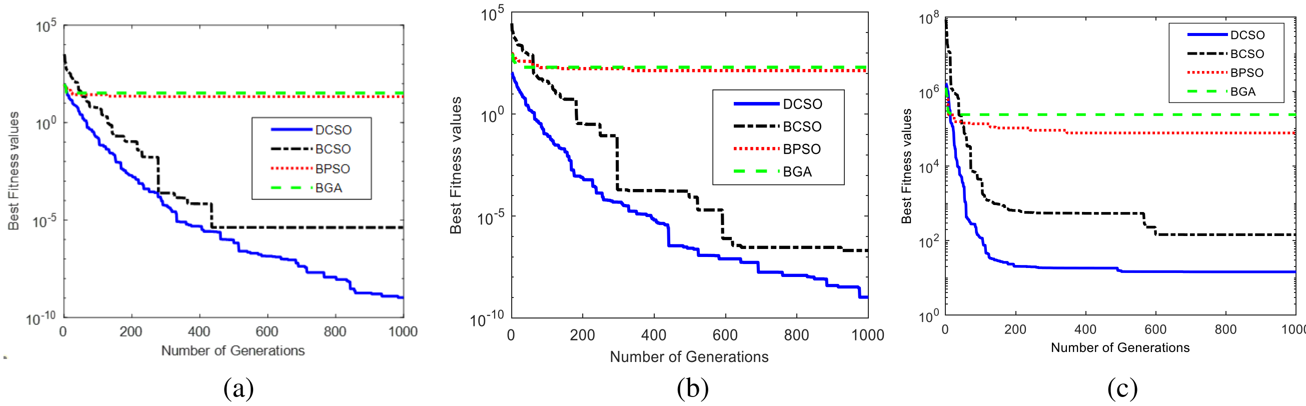


Figure 2. Convergence curve of the (a) sphere test function (f_1), (b) axis parallel ellipsoid test function (f_2), (c) Rosen Brock test function (f_3) and comparison among DCSO, BCSO, BPSO and BGA.

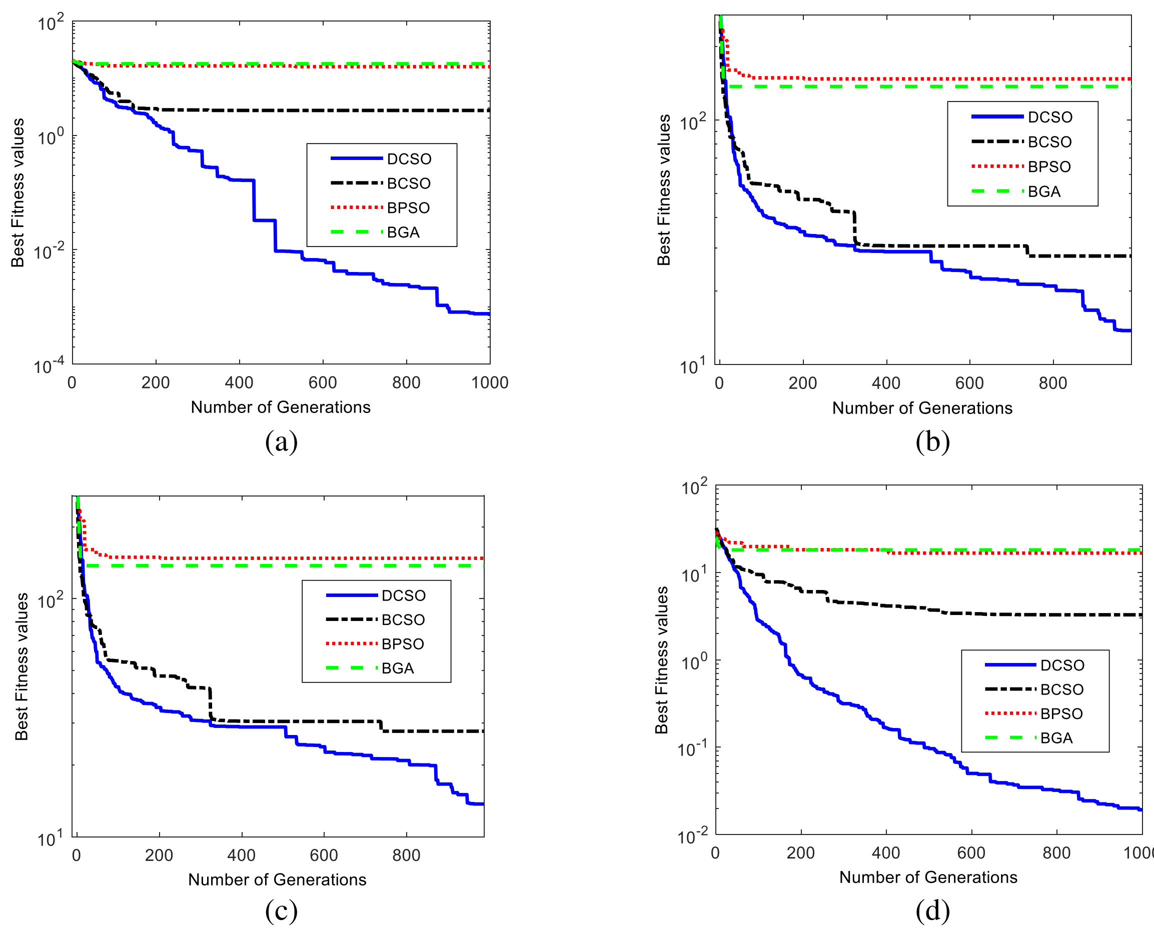


Figure 3. Convergence curve of the (a) Ackely's test function (f_4), (b) Rastrigin test function (f_5), (c) Griewank's test function (f_6), (d) Weierstrass test function (f_7) and comparison.

3.87e+00, 4.46e+04, 7.14e+04 for BCSO, BPSO, BGA, respectively. Clearly, DCSO outperformed the competing algorithms.

Table 3. Numerical analogy on DCSO, BCSO, BPSO and BGA for 20 dimensions (Mean and standard deviation representation).

Standard Function	DCSO	BCSO	BPSO	BGA
f_1	**5.01e−10±3.01e−08**	8.42e−05±2.15e−04	4.19e+02±4.11e+02	3.15e+03±0.41e+02
f_2	**7.12e−09±4.22e−09**	3.58e−04±2.58e−05	5.34e+01±8.42e+02	5.45e+02±1.02e+03
f_3	**1.51e+01±0.99e+00**	8.58e+02±7.47e+03	0.12e+04±9.23e+04	6.58e+04±1.34e+04
f_4	**3.08e−03±1.56e−03**	1.87e+00±1.1e+01	3.12e+01±7.12e−02	2.57e+01±1.27e+01
f_5	**1.16e+01±1.02e+00**	8.15e+01±7.2e+01	2.03e+03±1.32e+01	3.69e+03±0.87e+01
f_6	**2.26e−10±1.21e−09**	3.87e+00±5.46e−01	4.46e+04±4.25e+01	7.14e+04±0.68e+02
f_7	**2.31e−04±1.05e−02**	8.71e−01±4.24e+00	8.24e+04±3.13e+04	9.18e+04±7.21e+04

Bold represents the best results among all the competing ones.

4. STATISTICAL TESTS

Statistical tests generally provide the quantitative analysis about a process. In the present context these tests evaluate whether the superiority of the algorithm is significant or not over the competing algorithms. One such test is Wilcoxon signed ranked test. This test is usually used to compare the matched samples. It is also called the matched pairs test. It gives results of whether there is any pairwise statistical significance between algorithms. Here the test is carried out between DCSO and BCSO, BPSO, BGA algorithms pairwise individually. This work considers the Wilcoxon signed rank test with significance level of $\propto$. The values of $\propto$ are considered to be 0.05 and 0.1. Two hypotheses are considered for analysis $H0$ also called Null hypothesis (where no difference is observed in the performance of algorithms) and $H1$ also called Alternative hypothesis (where difference is observed in the performance of algorithms). The results of the tests are depicted Table 4 using $R+$, $R-$, and p-values. For all the considered problems, $R+$ indicates the sum of signed ranks where the first algorithm outperforms the second algorithm, and $R-$ indicates the sum of signed ranks where the second algorithm outperforms the first. From the results it can be said that the DCSO outperforms BCSO, BPSO, and BGA.

Table 4. Results that represent Wilcoxon signed rank test.

Dimension	Algorithm	$R+$	$R-$	P-value	Significance level			
					$\alpha = 0.1$		$\alpha = 0.05$	
					$p < \alpha$	H_0 Rejection	$p < \alpha$	H_0 Rejection
20	DCSO vs BCSO	28	0	0.0781	Yes	Yes	No	No
	DCSO vs BPSO	28	0	0.0156	Yes	Yes	Yes	Yes
	DCSO vs BGA	28	0	0.0156	Yes	Yes	Yes	Yes

5. APPLICATION OF DCSO TO THINNED ANTENNA ARRAY SYNTHESIS

5.1. Planar Antenna Array Geometry

Let's consider a PAA made of omnidirectional radiating elements having size $2N \times 2M$, which exhibits symmetricity over the x, y-axes and is depicted in Fig. 4. The array factor (AF) for this geometry is derived as

$$\mathrm{AF}(I, \theta, \phi) = 4 \sum_{n=1}^{N} \sum_{m=1}^{M} I_{nm} \cdot \cos\left[\pi \cdot (n - 0.5) \cdot U\right] \cdot \cos\left[\pi \cdot (m - 0.5) \cdot V\right] \tag{7}$$

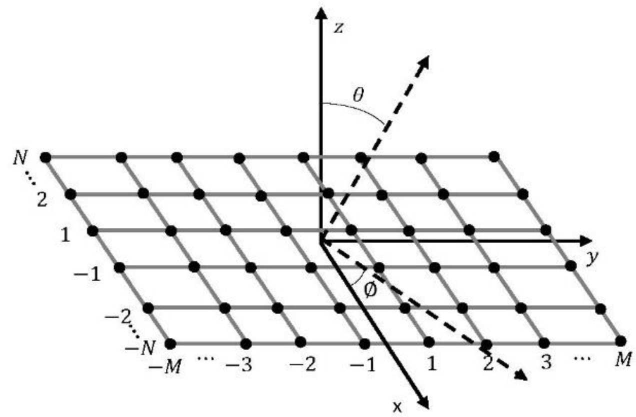


Figure 4. Geometric presentation of PAA [37] having uniformly spaced elements.

The interelement spacing is considered as 0.5λ. I_{nm} is the excitation amplitude of (n, m)th element $U = \sin(\theta)\cos(\phi)$, $V = \sin(\theta)\sin(\phi)$. In thinning process, $I_{nm} = 1$ if the (n, m)th element is switched ON and $I_{nm} = 0$ if the (n, m)th element is switched OFF.

5.2. Problem Formulation

The main objective is to achieve the suppression of PSLL while maintaining energy efficiency (a smaller number of active elements) by discovering suitable combination of 1's and 0's. PSLLs exist in principal planes ($\phi = 0°$ & $90°$) of uniformly illuminated planar antenna array radiation pattern. Quite often, researchers exercise their algorithms on reducing the SLLs in the principal places to project the effectiveness of the proposed algorithm. However, if the PSLL in $\phi = 0°$ & $90°$ planes is reduced, then proportionately the sidelobe levels in other planes increase. Also, it is necessary to control PSLL in all ϕ planes. Hence, we have considered these two scenarios of addressing the suppression of PSLL in two principal planes for $\phi = 0°$ & $90°$ and in the entire ϕ plane.

5.2.1. Case I: Defined to Reduce PSLL in $\phi = 0°$ & $90°$ Planes

The objective function is formulated to find the sum of maximum PSLL in two ϕ planes ($\phi = 0°$ & $90°$) as below

$$F(I) = \max_{(U,V\in S)}\left(\frac{\mathrm{AF}^{I}_{\mathrm{dB}}(\theta, 0°)}{\mathrm{AF}_{\max}}\right) + \max_{(U,V\in S)}\left(\frac{\mathrm{AF}^{I}_{\mathrm{dB}}(\theta, 90°)}{\mathrm{AF}_{\max}}\right) \tag{8}$$

5.2.2. Case II: Defined to Reduce PSLL in ϕ Plane

The objective function is derived to find the maximum PSLL in the entire ϕ-plane.

$$F(I) = \max_{(U,V\in S)}\left(\frac{\mathrm{AF}^{I}_{\mathrm{dB}}(\theta, \phi)}{\mathrm{AF}_{\max}}\right) \tag{9}$$

where $\mathrm{AF}_{\max}$ is the peak value generated in main beam, and F is the fitness which is considered only in the sidelobe region $S \in (\theta\phi)$ (omitting the main beam).

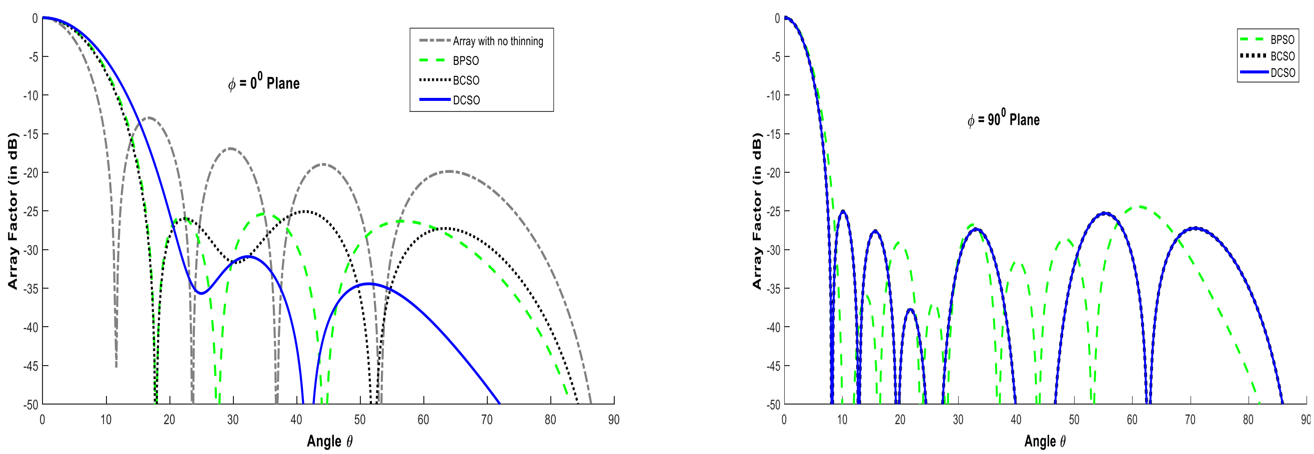
5.3. Numerical Illustrations

The proposed DCSO algorithm is applied to the synthesis of a PAA of geometric expansion 20×10 based on its popularity from previously published work [7–9, 16, 17]. The results of DCSO are compared on par with the existing BCSO [35], BPSO [34], and work in literature to test the its efficiency. The azimuth angular regions (90° to 180°) are considered for evaluating the objective functions. To generate the radiation pattern with good angular resolution 451 angular samples are considered for simulation.

10000 functional evaluations are performed for considered algorithms. For every execution the best solution is chosen after evaluating the algorithm 20 times. The parameter setups for DCSO, BCSO, and BPSO are same as listed in Table 2 except the BPSO swarm size is 100, CDC of BCSO 20%, and the MR of DCSO considered as 0.4. A series of parameter tuning experiments have been conducted for setting the parameters of DCSO.

5.3.1. Case 1: Suppression of PSLL in Two Principal Planes $\phi = 0°$ & $90°$

A uniformly illuminated periodic PAA exhibits a maximum PSLL of -13.23 dB in each of the principal planes. Next, the average PSLL is documented for each algorithm. With applying DCSO optimization a fitness value of -56 dB is attained, and BCSO and BPSO optimized PAAs produce a PSLL of -50.41 dB and -49.2 dB, respectively, while the arrays are 54 percent filled (108 elements). DCSO optimization produces a PSLL of -30.91 dB at $\phi = 0°$ and -25.08 dB at $\phi = 90°$. The patterns of the DCSO, BCSO, and BPSO optimized arrays are shown in Fig. 5, and the status of the array elements is represented in Table 5. The element status of a quadrant (x: $+ve$ in horizontal plane and y: $+ve$ in vertical plane) of the PAA is represented due to symmetricity. Fig. 6 shows the PSLL variation and filling percentage attained using DCSO, BCSO, and BPSO respectively for 20 runs. The mean PSLLs obtained by using DCSO, BCSO, and BPSO are -53.34 dB, -50.14 dB, and -49.28 dB, respectively. The convergence characteristics of all the algorithms in terms of average values are shown in Fig. 7. It can be observed that DCSO requires only 2480 and 3840 function evaluations to reach the final values obtained by BPSO and BCSO, respectively. The optimum PAA metrics obtained are represented in Table 6. The best values of the competing algorithms are marked in bold. DCSO outperforms BCSO and BPSO for producing best and average PSLL values in both the ϕ planes.



(a) (b)

Figure 5. Optimized radiation pattern (AF) for the best performing trail of 20×10 PAA at (a) $\phi = 0°$ plane and (b) $\emptyset = 90°$ plane.

Table 5. ON and OFF status indication of elements for 20×10 PAA in $\phi = 0°$ and $\phi = 90°$ planes.

Best Array Element Status for one quadrant	**Optimization Method**		
	DCSO	**BCSO**	**BPSO**
(1: ON state 0: OFF state)	1111111111	1110111110	1111111010
	1111111100	0111111101	1111011100
	1110110000	1111110000	1110100001
	1101000000	1110000000	1011100000
	0010000000	1001000000	0100000000

Table 6. Pattern metrics of optimum array design for a 20 × 10 PAA synthesis in $\phi = 0°$ and $\phi = 90°$ planes.

Optimization method	PSLL (in dB)				Array filling % (at best PSLL)
	Both planes		$\phi = 0°$	$\phi = 90°$	
	Best	Avg.			
DCSO	**−56.00**	**−53.34**	**−30.91**	−25.08	54
BCSO	−50.14	−49.27	−25.06	−25.07	54
BPSO	−49.28	−48.92	−24.45	**−25.37**	54
Array with no thinning	−26.23	−26.13	−12.97	−13.26	100

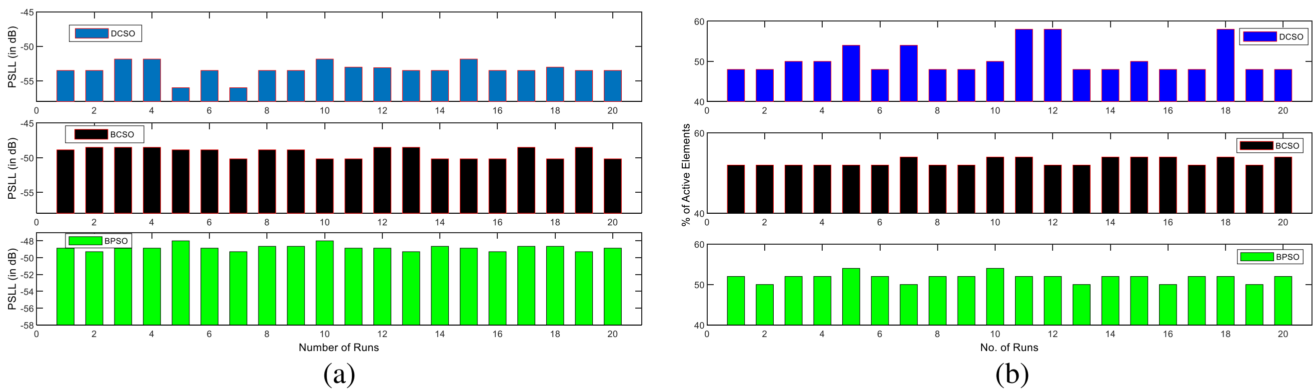


Figure 6. Variation of (a) PSLL, (b) filling percentage over 20 runs using DCSO, BCSO and BPSO for a 20 × 10 PAA in both the planes.

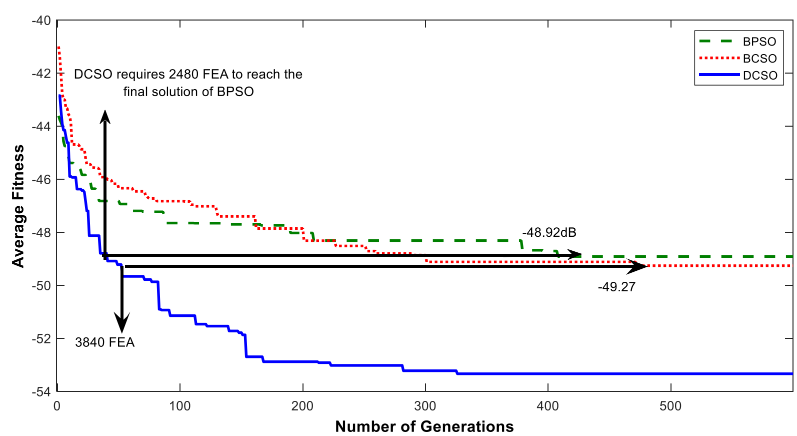


Figure 7. The convergence characteristics measured with respect to average fitness value for a 20 × 10 PAA in $\phi = 0°$ and $\phi = 90°$.

5.3.1.1 Comparison with Published Work for Case I

A 20 × 10 PAA has been synthesized by many researchers [7, 33, 36] in past, and the array obtained by DCSO approach is compared with theirs. The comparisons are listed in Table 7. From Table 7 the DCSO algorithm suppresses the PSLL around 16.17 dB over GA, 10.55 dB over MGA, 4.82 dB over OGA, 4.82 dB over BDE, and 4.57 dB over ACO. The filling percentage for all these arrays except ACO is 54%.

Table 7. Comparison with published results for 20 × 10 PAA in $\phi = (0°, 90°)$ planes.

Optimization method	PSLL in dB			Array filling % (at best PSLL)
	Both planes	0°	90°	
DCSO	**−56.00**	−30.91	−25.08	54
GA [33]	−39.83	−20.07	−19.76	54
MGA [8]	−45.45	−29.59	−15.85	54
OGA [9]	−51.18	−26.09	−25.09	54
BDE [17]	−51.18	−26.09	−25.09	54
ACO [16]	−51.43	−25.76	−25.67	68

5.3.2. Case 2: Suppression of PSLL in Entire ϕ-Plane

Figure 8 represents the 3-D pattern produced after DCSO is performed, and Fig. 9 represents the 2-D far-field patterns at $\phi = 0°$, 45°, and 90°. The optimized array produces the optimum PSLL of

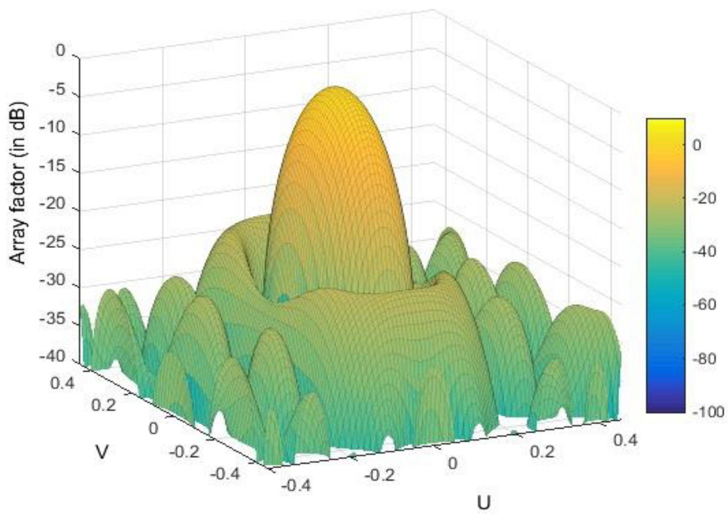


Figure 8. 3-D radiation pattern at far field region for an optimized PAA using DCSO after reduction of PSLL in all ϕ planes.

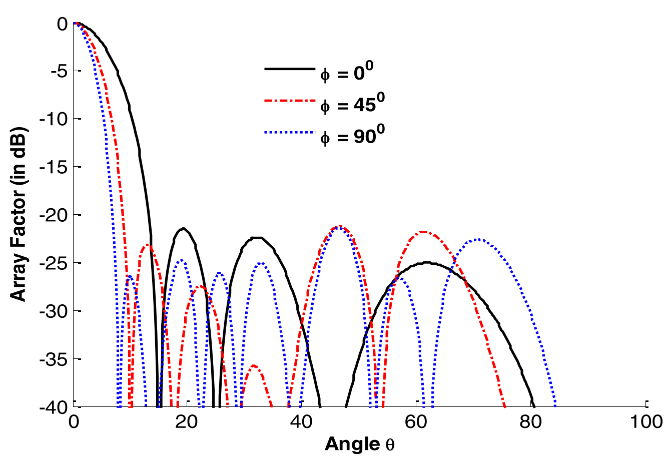


Figure 9. Optimized radiation pattern in $\phi = (0°, 45°, 90°)$ planes for a 20 × 10 PAA using DCSO.

-20.87 dB with a filling percentage of 62 in the entire ϕ-plane. The optimum PSLLs found in BCSO and BPSO are -20.16 dB and -19.56 dB, respectively. The corresponding filling percentages of these arrays are 60 and 62, respectively. The mean PSLLs obtained by using DCSO, BCSO, and BPSO are -20.47 dB, -19.84 dB, and -18.46 dB, respectively. The input status of elements for DCSO synthesis is shown in Table 8. The PSLL variation and filling percentage for 20 runs is shown in Fig. 10. The pattern metrics obtained after optimization is depicted in Table 9.

Table 8. ON and OFF status indication of elements for 20×10 PAA in entire ϕ-plane.

Best Array Element Status for one quadrant	**DCSO**
(1: ON state 0: OFF state)	1111111101 1111101010 1111110100 1101010001 0110100000

Table 9. Pattern metrics of optimum array design for a 20×10 PAA synthesis in entire ϕ-plane.

Optimization method	**PSLL (in dB) entire ϕ plane**		**% Filling**
	Best	**Avg.**	**For best PSLL**
DCSO	**−20.87**	**−20.47**	62
BCSO	−20.16	−19.84	62
BPSO	−19.56	−18.46	60

It can be seen from Table 9 that DCSO has shown superiority over BCSO and BPSO in terms of producing low PSLL in the entire ϕ-plane. The evolutionary performance in terms of average fitness value by using DCSO, BCSO, and BPSO can be seen in Fig. 11, and it can be concluded that DCSO converges faster thanBCSO and BPSO. Also, DCSO requires 2240 and 20160 function evaluations to reach final values obtained by BPSO and BCSO, respectively.

5.3.2.1 Comparison with Published Work for Case-II

It can be depicted in Table 10 that for a 20×10 thinned PAA synthesis (in entire ϕ plane), the SLL obtained by using DCSO is -20.87 dB, and 18.84 dB, -19.44 dB, -20.40 dB are the respective SLLs of MGA, OGA, BDE. Hence, the DCSO algorithm offers the suppression of PSLL by around 2.03 dB, 1.43 dB, and 0.47 dB in comparison with optimization offered by MGA [8], OGA [9], and BDE [17],

Table 10. PSLL and filling factor comparison of DCSO with MGA, OGA, BDE for 20×10 PAA synthesized in entire ϕ-plane.

Optimization method	**PSLL (in dB) entire ϕ plane**	**% Filling**
DCSO	**−20.87**	62
MGA [8]	−18.84	54
OGA [9]	−19.44	54
BDE [17]	−20.40	54

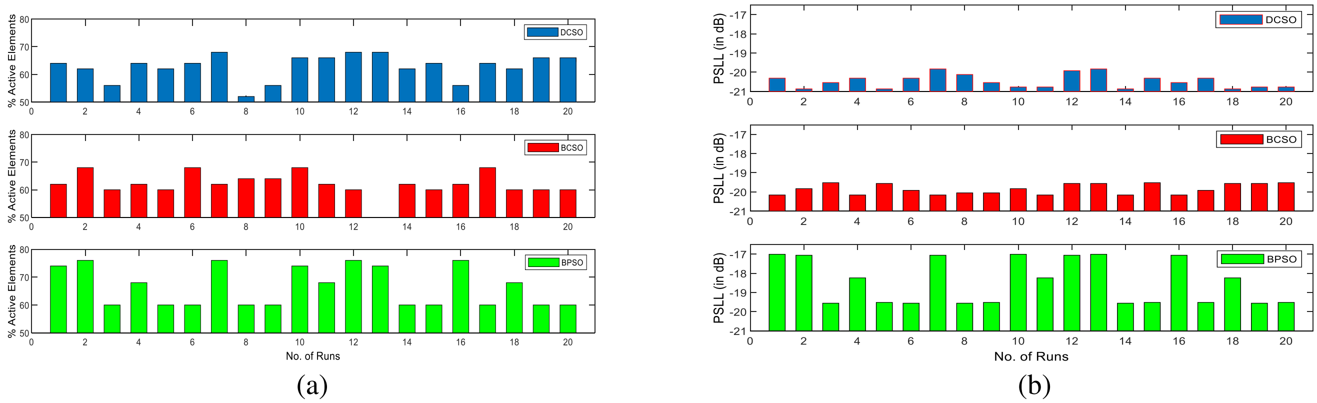


Figure 10. Variation of (a) PSLL, (b) filling percentage over 20 runs using DCSO, BCSO and BPSO for the 20×10 PAA in entire ϕ-plane.

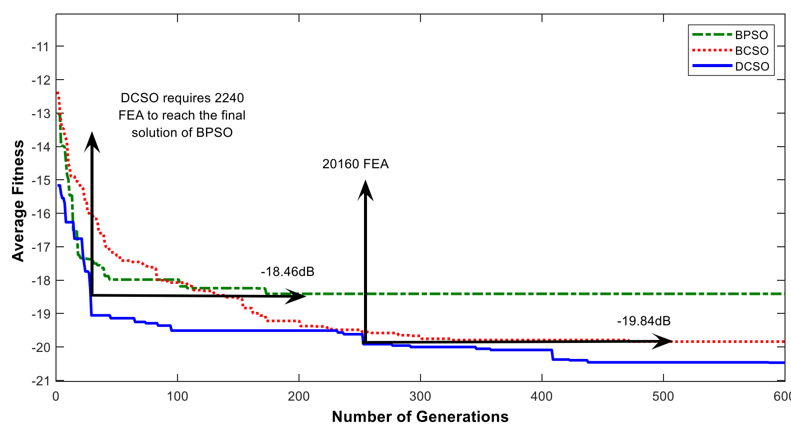


Figure 11. The convergence characteristics measured with respect to average fitness value for a 20×10 PAA in entire ϕ-plane.

respectively. The percentage of filling of the array using DCSO is 62 percent. The conclusion that can be drawn after all the functional evaluations is the DCSO outperforms the other proposed algorithms with the benchmark as solution accuracy as it produces optimal designs of the arrays.

6. CONCLUSION

In present work, a novel algorithm called DCSO that mimics the natural demeanor of the cat is designed to solve binary valued problems. A novel mutation probability equation is mathematically formulated using the tangent function with Gaussian mutation characteristics. The algorithm is applied to seven multimodal benchmark functions; the convergence curves are compared, and DCSO outperformed the BCSO, BPSO, and BGA algorithms. Statistical analysis is done by using Wilcoxon signed ranked test to show the stability of the obtained DCSO solutions. This optimization is used to perform the thinning of large antenna arrays to produce optimal switch ON and OFF status of elements. Synthesis results of 20×10 planar array show that -56 dB of PSLL is obtained for DCSO, and -50.14 dB and -49.28 dB was observed for BCSO and BPSO, respectively. There is a significant reduction in PSLL compared to existing state-of-art optimization algorithms. Other observations from above research are that the DCSO is an innovation in field of array synthesis and can be comfortably applied in the synthesis of large antenna arrays with wide apertures. Though the present paper is focused on the synthesis of PAAs, the developed algorithm is compatible for applying to other complex electromagnetic problems.

REFERENCES

1. Cheng, D. K., "Optimization techniques for antenna arrays," *Proc. IEEE*, Vol. 59, No. 12, 1664–1674, 1971, doi: 10.1109/PROC.1971.8523.
2. Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes*, 1st Edition, 1986.
3. Smith, S. K., J. C. Brggains, K. L. Melde, and F. Ares, "Analytical and optimization methods for linear arrays with high efficiency and low sidelobes," *IEEE Antennas and Propagation Society International Symposium*, 547–550, 2004.
4. Westcott, B. S. and A. A. Zaporozhets, "Beam shaping techniques based on analytical gradient iteration procedures," *IEE Colloquium on Novel Techniques for Antenna Beam Control*, No. 3, 5, 1995, doi: 10.1049/ic:19950001.
5. Peters, T. J., "A conjugate gradient-based algorithm to minimize the sidelobe level of planar arrays with element failures," *IEEE Trans. Antennas Propag.*, Vol. 39, No. 10, 1497–1504, 1991, doi: 10.1109/8.97381.
6. Sarkar, T. K., "On the application of the generalized biconjugate gradient method," *Journal of Electromagnetic Waves and Applications*, Vol. 1, No. 3, 223–242, 1987, doi: 10.1163/156939387X00036.
7. Haupt, R. L., "Thinned arrays using genetic algorithms," *IEEE Trans. Antennas Propag.*, Vol. 42, No. 7, 993–999, 1994, doi: 10.1109/8.299602.
8. Chen, K., X. Yun, Z. He, and C. Han, "Synthesis of sparse planar arrays using modified real genetic algorithm," *IEEE Trans. Antennas Propag.*, Vol. 55, No. 4, 1067–1073, 2007.
9. Zhang, L., Y. C. Jiao, B. Chen, and H. Li, "Orthogonal genetic algorithm for planar thinned array designs," *Int. J. Antennas Propag.*, Vol. 2012, 1–7, 2012, doi: 10.1155/2012/319037.
10. Oliveri, G. and A. Massa, "Genetic algorithm (GA)-enhanced almost difference set (ADS)-based approach for array thinning," *IET Microwaves, Antennas Propag.*, Vol. 5, No. 3, 305–315, 2011, doi: 10.1049/iet-map.2010.0114.
11. Murino, V., A. Trucco, and C. S. Regazzoni, "Synthesis of unequally spaced arrays by simulated annealing," *IEEE Trans. Signal Process.*, Vol. 44, No. 1, 119–123, 1996, doi: 10.1109/78.482017.
12. Meijer, C. A., "Simulated annealing in the design of thinned arrays having low sidelobe levels," *Proc. South African Symp. Commun. Signal Process., COMSIG*, 361–366, 1998, doi: 10.1109/comsig.1998.736983.
13. Trucco, A., "Thinning and weighting of large planar arrays by simulated annealing," *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, Vol. 46, No. 2, 347–355, 1999.
14. Donelli, M., A. Martini, and A. Massa, "A hybrid approach based on PSO and Hadamard difference sets for the synthesis of square thinned arrays," *IEEE Trans. Antennas Propag.*, Vol. 57, No. 8, 2491–2495, 2009, doi: 10.1109/TAP.2009.2024570.
15. Wang, W.-B., Q. Feng, and D. Liu, "Synthesis of thinned linear and planar antenna arrays using binary PSO algorithm," *Progress In Electromagnetics Research*, Vol. 127, 371–387, 2012.
16. Quevedo-Teruel, Ó. and E. Rajo-Iglesias, "Ant colony optimization in thinned array synthesis with minimum sidelobe level," *IEEE Antennas Wirel. Propag. Lett.*, Vol. 5, No. 2, 349–352, 2006.
17. Zhang, L., Y. C. Jiao, Z. B. Weng, and F. S. Zhang, "Design of planar thinned arrays using a Boolean differential evolution algorithm," *IET Microwaves, Antennas Propag.*, Vol. 4, No. 12, 2172–2178, 2010, doi: 10.1049/iet-map.2009.0630.
18. Rocca, P., G. Oliveri, and A. Massa, "Differential evolution as applied to electromagnetics," *IEEE Antennas Propag. Mag.*, Vol. 53, No. 1, 38–49, 2011, doi: 10.1109/MAP.2011.5773566.
19. Liu, C. and H. Wu, "Synthesis of thinned array with side lobe levels reduction using improved binary invasive weed optimization," *Progress In Electromagnetics Research M*, Vol. 37, 21–30, 2014.
20. Wu, H., C. Liu, and X. Xie, "Thinning of concentric circular antenna arrays using improved binary invasive weed optimization algorithm," *Hindawi Publ. Corp. Math. Probl. Eng.*, Vol. 2015, 1–8, 2015, doi: 10.1155/2015/365280.

21. Roy, G. G., S. Das, P. Chakraborty, and P. N. Suganthan, "Design of non-uniform circular antenna arrays using a modified invasive weed optimization algorithm," *IEEE Trans. Antennas Propag.*, Vol. 59, No. 1, 110–118, 2011, doi: 10.1109/TAP.2010.2090477.
22. Singh, U. and R. Salgotra, "Synthesis of linear antenna array using flower pollination algorithm," *Neural Comput. Appl.*, Vol. 29, No. 2, 1–11, 2018, doi: 10.1007/s00521-016-2457-7.
23. Dahi, Z. A. E. M., C. Mezioud, and A. Draa, "On the efficiency of the binary flower pollination algorithm: Application on the antenna positioning problem," *Appl. Soft Comput. J.*, Vol. 47, 395–414, 2016, doi: 10.1016/j.asoc.2016.05.051.
24. Chatterjee, A., G. K. Mahanti, and G. Ghatak, "Synthesis of satellite footprint patterns from rectangular planar array antenna by using swarm-based optimization algorithms," *Int. J. Satell. Commun. Netw.*, Vol. 32, 25–47, 2014, doi: 10.1002/sat.
25. X. L. and K. M. Luk, "The grey Wolf optimizer and its applications in electromagnetics," *IEEE Trans. Antennas Propag.*, Vol. 68, No. 3, 2186–2197, 2020, doi: 10.1109/TAP.2019.2938703.
26. Singh, U. and M. Rattan, "Design of linear and circular antenna arrays using Cuckoo optimization algorithm," *Progress In Electromagnetics Research C*, Vol. 46, 1–11, 2013.
27. Singh, U., R. Salgotra, and M. Rattan, "A novel binary spider monkey optimization algorithm for thinning of concentric circular antenna arrays," *IETE J. Res.*, Vol. 62, No. 6, 1–10, 2016, doi: 10.1080/03772063.2015.1135086.
28. Darvish, A. and A. Ebrahimzadeh, "Improved fruit-fly optimization algorithm and its applications in antenna arrays synthesis," *IEEE Trans. Antennas Propag.*, Vol. 66, No. 4, 1–11, 2018, doi: 10.1109/TAP.2018.2800695.
29. Ravipudi, J. L. and M. Neebha, "Synthesis of linear antenna arrays using Jaya, self-adaptive Jaya and chaotic Jaya algorithms," *AEU — Int. J. Electron. Commun.*, Vol. 18, 1–27, 2018, doi: 10.1016/j.aeue.2018.05.022.
30. Chu, S.-C., P. Tsai, and J.-S. Pan, *Cat Swarm Optimization*, LANI, Vol. 4099, 854–858, Springer-Verlag Berlin Heidelberg, 2006, doi: 10.1007/978-3-540-36668-3_94
31. Pappula, L. and D. Ghosh, "Linear antenna array synthesis using cat swarm optimization," *AEU - Int. J. Electron. Commun.*, Vol. 68, No. 6, 540–549, 2014, doi: 10.1016/j.aeue.2013.12.012.
32. Pappula, L. and D. Ghosh, "Synthesis of linear aperiodic array using Cauchy mutated cat swarm optimization," *AEU - Int. J. Electron. Commun.*, Vol. 72, 52–64, 2017, doi: 10.1016/j.aeue.2016.11.016.
33. Haupt, R. L., J. J. Menozzi, and C. J. McCormack, "Thinned arrays using genetic algorithms," *AP-S Int. Symp. (IEEE Antennas Propag. Soc.*, Vol. 2, 712–715, 1993, doi: 10.1109/aps.1993.385248.
34. Kennedy, J. and R. C. Eberhart, "Discrete binary version of the particle swarm algorithm," *Proc. IEEE Int. Conf. Syst. Man Cybern.*, Vol. 5, 4104–4108, 1997, doi: 10.1109/icsmc.1997.637339.
35. Sharafi, Y., M. A. Khanesar, and M. Teshnehlab, "Discrete binary cat swarm optimization algorithm," *2013 3rd IEEE Int. Conf. Comput. Control Commun., IC4 2013*, 1–6, 2013, doi: 10.1109/IC4.2013.6653754.
36. Marcano, D. and F. Duran, "Synthesis of antenna arrays using genetic algorithms," *IEEE Antennas Propag. Mag.*, Vol. 42, No. 3, 12–20, 2000.
37. Pappula, L. and D. Ghosh, "Synthesis of thinned planar antenna array using multiobjective normal mutated binary cat swarm optimization," *Applied Computational Intelligence & Soft Computing*, Vol. 2016, 2016.