Paper

Unequally Spaced Antenna Array Synthesis Using Accelerating Gaussian Mutated Cat Swarm Optimization

Prasanna K. Kumar, Lakshman Pappula, B. T. P. Madhav, and V. S. V. Prabhakar

Department of ECE, Koneru Lakshmaiah Education Foundation, Guntur, India

https://doi.org/10.26636/jtit.2022.154821

Abstract—Low peak sidelobe level (PSLL) and antenna arrays with high directivity are needed nowadays for reliable wireless communication systems. Controlling the PSLL is a major issue in designing effective antenna array systems. In this paper, a nature inspired technique, namely accelerating Gaussian mutated cat swarm optimization (AGMCSO) that attributes global search abilities, is proposed to control PSLL in the radiation pattern. In AGM-SCO, Gaussian mutation with an acceleration parameter is used in the position-updated equation, which allows the algorithm to search in a systematic way to prevent premature convergence and to enhance the speed of convergence. Experiments concerning several benchmark multimodal problems have been conducted and the obtained results illustrate that AGMCSO shows excellent performance concerning evolutionary speed and accuracy. To validate the overall efficacy of the algorithm, a sensitivity analysis was performed for different AGMCSO parameters. AGMCSO was researched on numerous linear, unequally spaced antenna arrays and the results show that in terms of generating low PSLL with a narrow first null beamwidth (FNBW), AGMCSO outperforms conventional algorithms.

Keywords—Gaussian mutation, cat swarm optimization, linear antenna array, PSLL.

1. Introduction

Numerous antenna arrays are used in mobile, satellite, radar, and other wireless communication systems as they offer good signal quality, enhanced directivity, extended spectrum efficiency and wide coverage. To avoid interference with other communication systems operating nearby, there is a need to maintain a low peak sidelobe level (PSLL). Because of the increasing electromagnetic deterioration, nulls need to be kept at the desired directions with low sidelobes and fixed first null beam width (FNBW).

There are many approaches to shaping side lobe power in the radiation pattern. The designer may either alter the antenna's position or may use complex weights to obtain the desired radiation pattern (low PSLL). The weights apply to amplitude and phase inputs of each radiation element in the antenna array. Implementation of non-uniform amplitude and phase weights in uniformly placed antenna arrays is a complex problem. Instead, unequally spaced antenna arrays with uniform feeding provide greater flexibility in shaping side lobe power in the radiation pattern [1].

In this paper, we rely on aperiodic antenna array synthesis due to its simple feed network. An illustration of an unequally spaced array is shown in Fig. 1. It may be designed by altering the distance between the antenna's elements. The problem of unequally spaced antenna array synthesis involves non-linear and non-convex optimization using a set of classical gradient-based algorithms deployed in nature-inspired optimization techniques. Several nature-inspired optimization techniques, namely genetic algorithm (GA) [2]–[6], differential evolution (DE) [7]–[12], particle swarm optimization (PSO) [13]–[22], ant colony optimization (ACO) [23]–[25], cat swarm optimization [26]–[32], grey wolf optimization (GWO) [33]–[34], and bee colony optimization (BCO) [35]–[36] have been implemented while synthetizing unequally spaced antenna arrays.

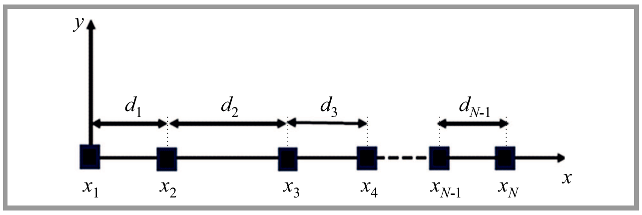


Fig. 1. A linear antenna array with non-uniform spacing.

Some of the important algorithms relied upon in antenna array synthesis have been discussed in detail in the literature. Yan *et al*. [2] proposed a simple and versatile GA for antenna array pattern synthesis and the approach involves array excitation weighting vectors used as complex number chromosomes. GA has been applied to synthesize linear and circular arrays to achieve –20 dB PSLL levels. Chen KS *et al*. proposed, in paper [4], a modified real genetic algorithm (MGA) for the synthesis of sparse linear arrays by optimizing the elements' positions to reduce peak sidelobe level. The result of the synthesis showed that MGA achieved a PSLL of −20.562 dB with 37 elements. Numerical simulations demonstrated superb efficiency and robustness of this algorithm. The improved genetic algorithm (IGA) was proposed by Cen *et al*. [5]. It simultaneously changes the weight coefficients and inter-sensor spacings of a linear aperiodic array. The authors extended the research to include the impact of steering angles on the sidelobe level with fixed main beamwidth. The results have shown superiority of PSLL compared to other GA variants. A computationally efficient global optimization method of differential evolution (DE) was proposed for the synthesis of uniform amplitude arrays by Kurup *et al*. [7]. Phase-only synthesis and position-only syntheses were discussed to achieve low sidelobe levels by Lin *et al*. [9].

Goudos *et al*. [11] proposed a design technique based on a self-adaptive DE (SADE) algorithm which has been applied to real-valued microwave design problems including linear-array synthesis, patch-antenna design, and microstrip filter design. The authors compared the SADE strategy with popular PSO and DE variants and proved its effectiveness. Zhang *et al*. [17] enhanced search diversity by integrating a parameter selection strategy into classical differential evolution. In this research, a modified DE based on a harmony search algorithm known as HSDEA was developed to optimize linear aperiodic arrays by ensuring a minimum peak sidelobe level. Simulations showed that a PSLL of -22.631 dB had been achieved for a 32-elements array. HSDEA converges faster and requires fewer calculations for synthetizing a linear aperiodic array, compared with other methods. PSO [13] is a recently developed high-performance optimizer. It is similar to GA or evolutionary algorithms, but requires fewer computational resources. Boeringer *et al*. [13] proposed PSO to synthetize antenna arrays using amplitude-only, phase-only, and complex tapering. Several comparative studies were conducted by comparing PSO with GA and ACO. Khodier *et al*. [14] proposed PSO for a synthesis of linear antenna arrays and formulated the objective function for PSLL and null placement. Rajo-Iglesias *et al*. [23] proposed ACO using real numbers for synthesizing linear antenna arrays. They synthesized 10- and 32-element array systems. For a 32-element linear array, ACO gives a PSLL of -17.5 dB and a 7.7° beamwidth.

To increase the efficiency of antenna arrays by ensuring high directivity and low sidelobe levels, CSO was proposed for synthesizing linear antenna arrays by Pappula *et al*. [31]. CSO is used to optimize the distance between the antenna's parts in order to generate a radiation pattern with low PSLL and deep nulls in the desired directions. CSO has shown superiority over GA and PSO. Li *et al*. [34] proposed the GWO algorithm, which mimics the social behavior of grey wolves, to synthesize linear arrays. The objective is to suppress peak sidelobe level under various constraints. Performance is further verified while optimizing the design of a dual-band E-shaped patch antenna and a wideband magnetoelectric dipole antenna.

All these techniques have demonstrated at alternatives to traditional gradient-based algorithms exist and may be relied upon while searching for the global solution. But in the synthesis of antenna arrays, the feasible range of solutions is extremely wide and the search for an optimal solution with a fast convergence rate poses a major challenge. Algorithms that incorporate an exhaustive search function are needed to seek the optimal solution with a fast convergence rate.

Cat swarm optimization (CSO) [26] is a newly developed technique that mimics the original behavior of cats. Chu and Tsai introduced this technique in [37]. It has been implemented while dealing with numerous engineering problems in the real world [37] and has shown improved performance over GA and PSO.

However, while solving complex non-linear problems, conventional CSO suffers from premature convergence and locks in local minima. In a position-updated equation of CSO, due to the random mutation process, this leads to the aforementioned problems. This issue is restricted to a wide range of applications of the traditional CSO.

In this paper, we have introduced Gaussian mutation with an accelerating parameter in the position-updated equation – a solution offering fast convergence that may be accurately compared with CSO. The proposed AGMCSO is applied while synthesizing unequally spaced antenna arrays to suppress PSLL, while simultaneously maintaining narrow FNBW. Then, a detailed comparison of AGMCSO with state-of-the-art algorithms is presented.

This article is organized as follows: details of the traditional CSO approach are discussed in Section 2 and are followed by the introduction of AGMCSO in Section 3. Section 4 presents the test functions on which AGMCSO is being implemented and a comparison of numerical outcomes for 30-dimensional problems between AGMCSO, CSO, PSO is obtained. Section 5 addresses the application of AGMCSO in complex EM design problems.

2. Modified Cat Swarm Optimization

CSO is modeled by observing the hunting skills of a cat. The algorithm is classified into two modes of operation: seeking mode and tracing mode. Cats are assigned to the mode depending on the mixture ratio (MR).

In the seeking mode (SM), by observing the surroundings, a cat being in its rest position will always be on alert. The cat's movements are very slow. The relevant model uses the following [32]:

- seeking range of the selected dimension (SRD), which determines the amount of available range,
- counts of dimensions to change (CDC) – this parameter determines the number of dimensions to be mutated.,
- seeking memory pool (SMP) – determining the number of copies of cats to be created for mutation.

The following are the phases observed in the SM:

- build K copies of i-th cat based on SMP;
- $(K-1)$ copies are subject to the mutation mechanism. All dimensions are randomly mutated according to CDC and SRD, either by adding SRD to or subtracting it from the parent location;
- the fitness values of newly updated copies are analyzed;
- choose the best value from the K copies is chosen and replaced with the cat's position.

In the tracing mode (TM), cats attempt to change their locations rapidly by tracking the targets. The shift in the location is statistically inferred by following the cat's tracing actions. In this mode, the algorithm's steps are:

- in the D-dimensional solution space the position and velocity of the m-th cat is:

$$P_m^g = [P_{mn}] \quad \text{where} \quad n = 1, \ldots, D \;, \tag{1}$$

$$Vel_m^g = [Vel_{mn}] \quad \text{where} \quad n = 1, \ldots, D \;, \tag{2}$$

- for each dimension the position and velocity of m-th cat is updated as:

$$Vel_m^{g+1} = [Vel_{m,n}] = \omega . Vel_{m,n}^g + C.r.(P_{gbest} - P_{m,n}^g) \;, \tag{3}$$

$$P_{m,n}^{g+1} = P_{m,n}^g + Vel_{m,n}^{g+1} \;, \tag{4}$$

where g represents the generation number, m is the cat's index in a swarm, n represents the cat's position index, $Vel_{m,n}^g$ is the velocity of the m-th particle, C represents the acceleration coefficient, r is the arbitrary number between 0 and 1, P is the weight of the inertia, and P_{gbest} is the best cat's position.

The fitness values are assessed after the tracing mode. If the required solution is not attained based on the mixture ratio, the adjusted cats are dispersed to SM and TM. This is repeated until the desired solution is acquired. However, it has been observed that in SM mode, the random mutation process leads to a poor and premature convergence rate.

3. Accelerating Gaussian Mutation Based CSO

The probability of sensing range is steadily decreased as the cat is in a resting position. It resembles a Gaussian distribution curve with a zero mean. The sensing range is focused around the cat's rest position and gradually becomes low as it moves far from the cat's position, i.e. compared to large mutations, the probability of developing lower mutations is higher. The Gaussian distribution curve resembles the cat's behavior in the seeking mode, as illustrated in Fig. 2.

It may be observed that there is a higher likelihood of minor mutations which lead to a more rigorous local search along with a global search. The standard deviation σ and the mean μ of Gaussian distribution density function is:

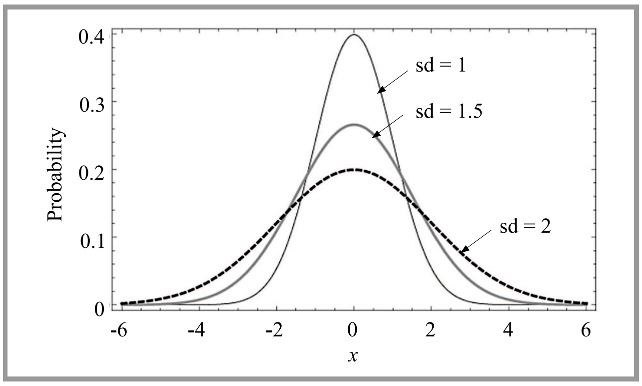


Fig. 2. Gaussian density function with three standard deviations.

$$f_{normal}(p; \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(p-\mu)^2}{2\sigma^2}} \;. \tag{5}$$

According Eq. (5), the Gaussian random number G is:

$$G(\mu, \sigma^2) = \mu + \sigma G(0,1) \;. \tag{6}$$

Here $G(0,1)$ is the Gaussian random number normally distributed with a standard deviation of 1 and a zero mean. From Fig. 2 it is evident that both large and small mutation values can be produced from the standard deviation value of 1. The method is modified with the help of Gaussian mutation to improve solution accuracy and convergence.

A mutant individual x_i^m is generated by Gaussian mutation which is:

$$x_i^m = x_i + G(0, \sigma^2) = x_i + \sigma \cdot G(0,1) \;, \tag{7}$$

where x_i is the unmutated individual, and σ is conveyed as the selected dimension's mutated value. Therefore, the position of each dimension of i-th cat is:

$$x_i^m = x_i + \left[SRD \cdot \left(1 - \frac{g}{gen}\right) \cdot x_i \cdot G(0,1) \right] \;. \tag{8}$$

To enhance the local convergence properties, we have adopted an accelerating component in the x_i^m.

3.1. Time Complexity of AGMCSO

The SM time complexity is considered as $O[N_n \cdot \log(N_n)]$, where N_n is $S_n \cdot \text{Dim} \cdot (\text{SMP} - 1)$ [32], and S_n is the number of cats in SM, Dim is the number of dimensions, the SMP pool searching for a memory. The process time complexity of TM may be mentioned as $O(N_n)$, where N_n is $T_n \cdot \text{Dim}$, and T_n is tracing the number of cats. The number of cats in TM is to be smaller than the number of cats in SM. The worst-case time complexity of the tracing mode process is dominated by SM worst-case time complexity. The suggested method's total worst-case time complexity is considered as $O[N_n \cdot \log(N_n)]$. Time complexity of the proposed AGMCSO approach and of the conventional CSO algorithm is the same, as we did not introduce any complicated variants in the proposed AGMCSO.

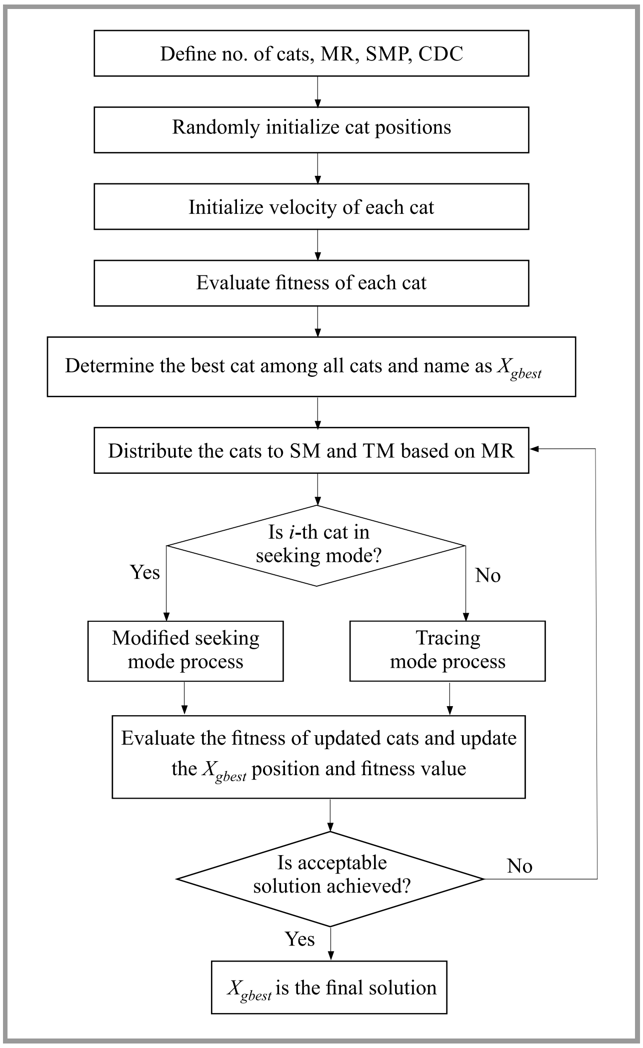


Fig. 3. Steps of the modified CSO algorithm.

3.2. Description of AGMCSO Algorithm

Figure 3 demonstrates the 7 phases of the modified CSO algorithm, which are:

1. in the D-dimensional solution space, a finite amount of cats is initialized randomly;
2. the velocity of the cats is initialized;
3. the fitness value of each cat is calculated, the cat with the highest fitness value is picked and the appropriate position of the cat is stored in the memory as X_{gbest};
4. the cats are shifted to the SM and TM depending on their flags, according to MR. In turn, if the cat's flag is set to SM, the cat will be directed to SM. Otherwise, the cat will move to the TM process;
5. the fitness of each altered cat is calculated after two modes have been completed and the cat's best position is stored as $X_{i,j}$;
6. X_{gbest} and $X_{i,j}$ fitness values are compared and the best position is updated as X_{gbest};
7. the program ends, if the required solution is obtained or else steps from 4–7 are repeated.

4. Benchmark Functions

In order to estimate the efficiency of techniques influenced by nature, common benchmark issues are used. They are classified into a few different categories and are considered to be multimodal or unimodal. Table 1 presents the characteristics of such benchmark problems. The global optimum x^*, global solution $f(x^*)$, acceptable solution and

Table 1
Benchmark test functions

Name	Symbol	Description	x^*	$f(x^*)$	Search range
Sphere	$f_1(x)$	$\sum_{i=1}^{D} x_i^2$	$0,\ldots,0$	0	$[-5.12, 5.12]^D$
Zakharov	$f_2(x)$	$\sum_{i=1}^{E} x_i^2 + \left[\sum_{i=1}^{D} 0.5ix_i\right]^2 + \left[\sum_{i=1}^{D} 0.5ix_i\right]^4$	$0,\ldots,0$	0	$[-5, 10]^D$
Rosenbrock	$f_3(x)$	$\sum_{i=1}^{D-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right]$	$1,\ldots,1$	0	$[-5, 10]^D$
Levy	$f_e(x)$	$\sin^2(\pi\omega_1) + \sum_{i=1}^{D-1} (\omega_i - 1)[1 + 10\sin^2(\pi\omega_i + 1)] +$ $(\omega_d - 1)^2[1 + \sin^2(2\pi\omega_d)]$	$1,\ldots,1$	0	$[-10, 10]^D$
Ackley	$f_5(x)$	$-20\mathrm{e}^{-b\sqrt{\frac{1}{d}\sum_{i=1}^{D} x_i^2}} - \mathrm{e}^{\frac{1}{d}\sum_{i=1}^{D} \sqrt{\cos(cx_i)}} + a + \mathrm{e}^1$	$0,\ldots,0$	0	$[-32, 32]^D$
Rastrigin	$f_6(x)$	$10d + \sum_{i=1}^{D} \left[x_i^2 - 10\cos(2\pi x_i)\right]$	$0,\ldots,0$	0	$[-5, 10]^D$
Griewank	$f_7(x)$	$\sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$0,\ldots,0$	0	$[-600, 600]^D$

search range of the benchmark problems have been listed. Problems $f_1 - f_4$ are unimodal and $f_5 - f_7$ are multimodal. For all cases the acceptable solution is set at 10^{-6}.

Numerous trials have been performed using seven benchmark functions to compare the proposed AGMCSO using a 30-dimensional problem with the classic PSO and CSO approaches. The simulation parameters are listed in Table 2. For all the experiments the average value, the average standard deviation and the number of function evaluations needed to achieve the acceptable solution (FEA) are listed in Table 3. The accuracy of solutions obtained using the proposed AGMCSO, CSO and PSO approaches for a 30-dimensional problem is presented in Table 4.

Table 2
AGMCSO, CSO and PSO parameters

AGMCSO		CSO		PSO	
Factor	Amount	Factor	Amount	Factor	Amount
Primary cats	50	Primary cats	50	Swarm size	135
SRD	0.8 (80%)	SRD	0.3 (30%)	r_1, r_2	[0, 1]
CDC	80%	CDC	80%	c_1	2
SMP	5	SMP	5	ω	0.2–0.9
MR	0.8	MR	0.8	–	–
r	[0, 1]	r	[0, 1]	–	–
ω	0.2–09	ω	0.2–0.9	–	–
c_1	2	c_1	2	–	–

Table 3
Comparison of average CPU time, convergence speed and SR

Function		AGMCSO	PSO	CSO
f_1	FEA	**1360**	–	47450
	Time	**0.0877** s	–	3.45 s
	SR	**100%**	0	100%
f_2	FEA	**2108**	–	–
	Time	**0.104** s	–	–
	SR	**100%**	0	0
f_3	FEA	**340**	–	38090
	Time	**0.058** s	–	7.24 s
	SR	**100%**	0	100%
f_4	FEA	**2006**	–	–
	Time	**0.0757** s	–	–
	SR	**100%**	0	0
f_5	FEA	**2380**	145800	38480
	Time	**0.1254** s	10.12	4.53 s
	SR	100%	34	100%
f_6	FEA	**2312**	–	69550
	Time	**0.0726** s	–	4.64 s
	SR	**100%**	9	199%
f_7	FEA	**1394**	–	194740
	Time	**0.1143** s	–	82.42 s
	SR	**100%**	0	100%

Notes:
Bold figures indicate the best results obtained (with all algorithms taken into consideration).
"–" means no runs performed by the algorithm have achieved the acceptable solution.
FEA is calculated to achieve the adequate solution over the number of successful runs.
SR shows the percentage of independent runs that have efficiently found the adequate solution.

Table 4
Comparison of solution accuracy for 30-D problems between AGMCSO, CSO, PSO (bold figures mean the best result)

	AGMCSO	CSO	PSO
f_1	$\mathbf{7.77 \cdot 10^{-291} \pm 0}$	$3.41 \cdot 10^{-40} \pm 1.99 \cdot 10^{-41}$	30.6 ± 19
f_2	$\mathbf{7.8 \cdot 10^{-268} \pm 0}$	$2.72 \cdot 10^{-52} \pm 0.24 \cdot 10^{-23}$	$5 \cdot 10^{20} \pm 15$
f_3	$\mathbf{50 \pm 0}$	29.6 ± 2.1	168 ± 6.7
f_4	$\mathbf{0 \pm 0}$	$3.1 \cdot 10^{-24} \pm 0.15 \cdot 10^{-40}$	18.1 ± 23
f_5	$\mathbf{8.88 \cdot 10^{-16} \pm 0}$	$4.68 \cdot 10^{-5} \pm 3.05 \cdot 10^{-5}$	1 ± 0.6
f_6	$\mathbf{0 \pm 0}$	$3.18 \cdot 10^{-14} \pm 0.169$	3.5 ± 8.9
f_7	$\mathbf{0 \pm 0}$	$1.23 \cdot 10^{-4} \pm 2.5 \cdot 10^{-4}$	5.1 ± 1.8

Computational complexity is the main performance metric for evaluating performance of an algorithm. It may be measured by the average CPU time or by the average FEA required to reach an acceptable solution. The success rate (SR) indicator is specified as the percentage of independent runs that have effectively achieved the desired solution.

The results obtained illustrate that AGMCSO outperforms CSO and PSO in terms of convergence rate and solution

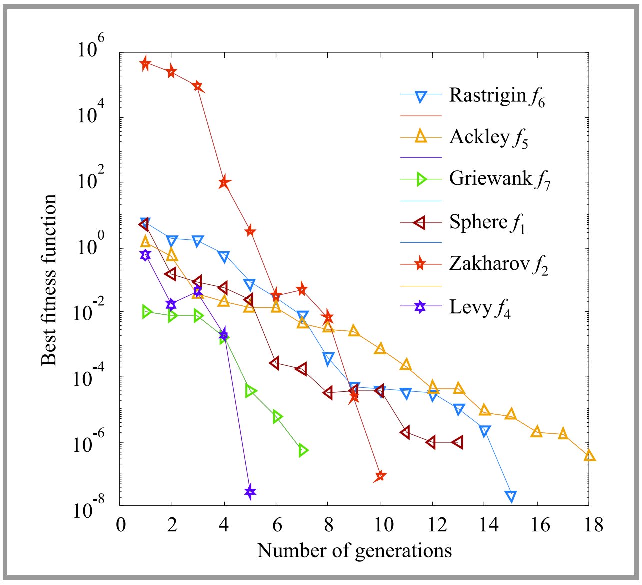


Fig. 4. Evolutionary process of fitness functions f_1, f_2, and f_4–f_6 for 30 dimensions.

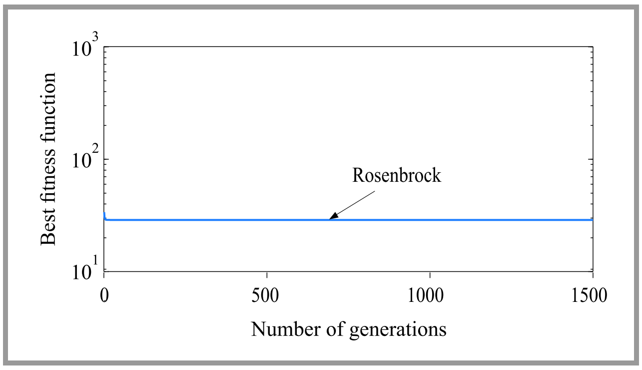


Fig. 5. Evolutionary process of fitness function f_3.

accuracy. AGMCSO achieved global solutions for all 30-dimensional multi-modal problems with minimum function evaluations. AGMCSO achieves acceptable solutions faster than CSO and PSO. For example, if we consider the 30-D multimodal Griewank function f_7 AGMCSO requires, on average, 1394 function evaluations vs. a 194740 needed by CSI in order to reach adequate solution accuracy of 10^{-6}. Table 3 shows that for all the benchmark functions, the SR for AGMCSO is 100% for all the dimensions whereas the PSO and CSO fails to reach 100% of SR.

Figure 4 shows the convergence plots between the number of generations and the fitness function for f_1, f_2, f_4–f_6. A similar convergence plot for f_3 is shown in Fig. 5.

4.1. Comparisons with Other Evolutionary Algorithms

Table 5 presents a comparison of the proposed AGMCSO solution with a few existing 30-D algorithms. The algorithms include fast evolutionary algorithm (FEP), adaptive differential evaluation with optional archive (JADE), comprehensive learning PSO (CLPSO), orthogonal learning particle swarm optimization (OLPSO-L), detecting, shrinking and local learning strategy PSO (DSLPSO), enhanced parallel cat swarm optimization (EPCSO), and cat swarm optimization with adaptive parameter control (NCSO). Except for the Rosenbrock function f_3, AGMCSO offers the best accuracy and also outperforms the current updated CSO algorithms. The average number of FEAs needed to achieve the suitable solution is also smaller.

5. AGMCSO Applications

Here, two examples of the AGMCSO algorithm are presented based on linear array designs selected from the literature.

5.1. Linear Antenna Array

Consider an M-element, uniformly illuminated linear antenna array positioned on the x axis (Fig. 6). The antenna array factor in the azimuth plane of $M = 2N$ is:

$$AF(X,\theta) = 2\sum_{n=1}^{N}\cos[kX_n\cos(\theta)] \quad M = 2N\ , \tag{9}$$

where the azimuthal angle is given as θ, the n-th element position is given as X_n, the wavenumber is given by $k = \frac{2\pi}{\lambda}$ and the wavelength by λ.

Selection of the distance between the antenna's elements is crucial. The positioning of adjacent elements too far apart leads to grating lobes, and positioning the too close to each other leads to mutual coupling. Thus, the constraint of adjacent element spacing has to be considered during the optimization process. The distance between antenna elements within the array is constrained as $|x_i - x_j| \geq 0.25\lambda$. With the parameter configuration for AGMCSO, CSO and PSO algorithms retrieved from Table 2, the algorithm is executed 10 times to show the efficacy of the suggested method.

Table 5
Comparison of AGMCSO with evolutionary algorithms (bold print = best result)

Function	FEP [38]	JADE [8]	CLPSO [15]	OLPSO-L [18]	DSL-PSO [20]	EPCSO [27]	NCSO [29]	AGMCSO
f_1	$5.7\cdot10^{-4}$ $\pm1.3\cdot10^{-4}$	$1.3\cdot10^{-54}$ $\pm9.2\cdot10^{-54}$	$4.4\cdot10^{-14}$ $\pm1.71\cdot10^{-14}$	$1.1\cdot10^{-38}$ $\pm1.21\cdot10^{-38}$	$1.3\cdot10^{-49}$ $\pm7.31\cdot10^{-49}$	0 ± 0	$1.68\cdot10^{-21}$	$\mathbf{7.77\cdot10^{-291}}$ $\mathbf{\pm0}$
f_2	–	–	–	–	–	–	–	$\mathbf{7.8\cdot10^{-291}}$ $\mathbf{\pm0}$
f_3	5.0 ± 5.8	0.3 ± 1.1	21 ± 2.9	1.2 ± 1.4	51 ± 42	N/A	23.5	$\mathbf{-1.019}$
f_4	–	–	–	–	–	–	–	$\mathbf{0\pm0}$
f_5	180 ± 2100	$4.4\cdot10^{-15}$ ±0	0 ± 0	$4.14\cdot10^{-14}$ ±0	$1.2\cdot10^{-14}$ $\pm4.6\cdot10^{-15}$	$6.40\cdot10^{-15}$ $\pm3.7\cdot10^{-15}$	$6.54\cdot10^{-12}$	$\mathbf{8.88\cdot10^{-16}}$ $\mathbf{\pm0}$
f_6	460 ± 1200	0 ± 0	$4.85\cdot10^{-10}$ ±0.361	0 ± 0	$4.7\cdot10^{-16}$ $\pm9.71\cdot10^{-17}$	86 ± 10	76.6	$\mathbf{0\pm0}$
f_7	1600 ± 2200	$2.0\cdot10^{-4}$ $\pm1.4\cdot10^{-3}$	31 ± 0.46	0 ± 0	170 $\pm2.21\cdot10^{-3}$	$3.50\cdot10^{-3}$ $\pm7.10\cdot10^{-3}$	0	$\mathbf{0\pm0}$

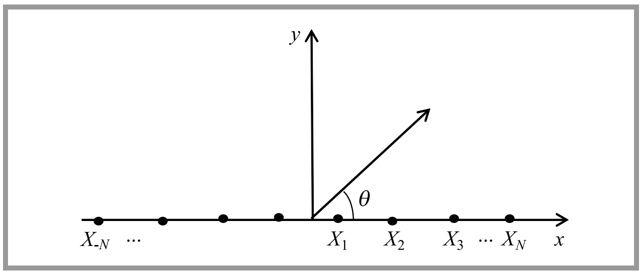


Fig. 6. Illustration of linear antenna array with non-uniform spacing.

In the design process, the aim is to minimize the peak sidelobe level in the sidelobe region by optimizing the spacings between the antenna's elements using the proposed AGMCSO method. The objective function can be modeled as:

$$Obj(X) = \max\left(\frac{|AF(X,\theta_0)|}{|AF_{\max}|}\right), \tag{10}$$

where $X = (X_1, X_2, \ldots, X_N)$ is the element position vector, θ_0 is defined as the angular region excluding the main lobe, the main peak of the pattern is $AF_{\max}$.

Table 6
Positions of a 20-element array optimized using AGMCSO and CSO

Element (n)	Position $\frac{x_n}{\lambda}$	
	AGMCSO	CSO
1	0.2642	0.2504
2	0.5288	0.7625
3	0.7920	1.2441
4	1.0920	1.7153
5	1.3470	2.3773
6	1.7122	2.8824
7	2.0428	3.5296
8	2.4654	4.3169
9	2.9505	5.2229
10	3.5994	6.0549

In the first example, a 20-element array is synthesized using the proposed AGMCSO and CSO approaches to minimize PSLL in the sidelobe region. Table 6 shows AGMCSO- and CSO-optimized element positions relative to Z. The array patterns obtained using the AGMCSO algorithm along with the CSO-optimized array and uniformly illuminated periodic array (UIPA) are shown in Fig. 7. Convergence characteristics for 10 independent runs are shown in Fig. 8. The comparison of convergence characteristics of AGMCSO and CSO is shown in Fig. 9. The comparison of PSLL obtained using CSO, fully informed particle swarm optimization (FIPSO) [17], perturbation particle swarm optimization (PPSO) [17] and AGMCSO is presented in Table 7.

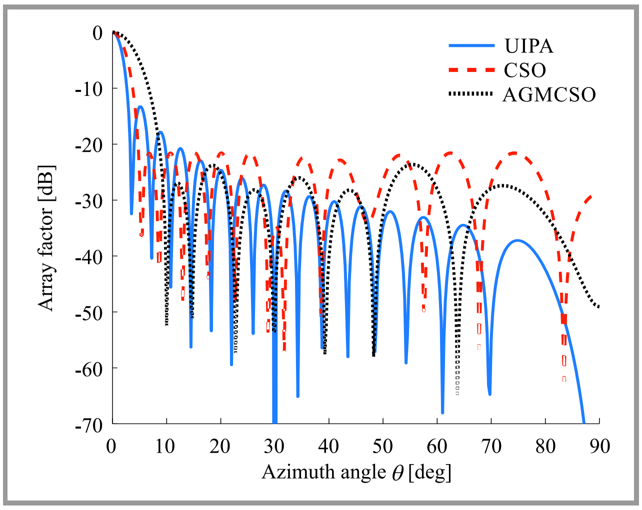


Fig. 7. Radiation pattern of the 20-element array using AGMCSO, CSO and UIPA.

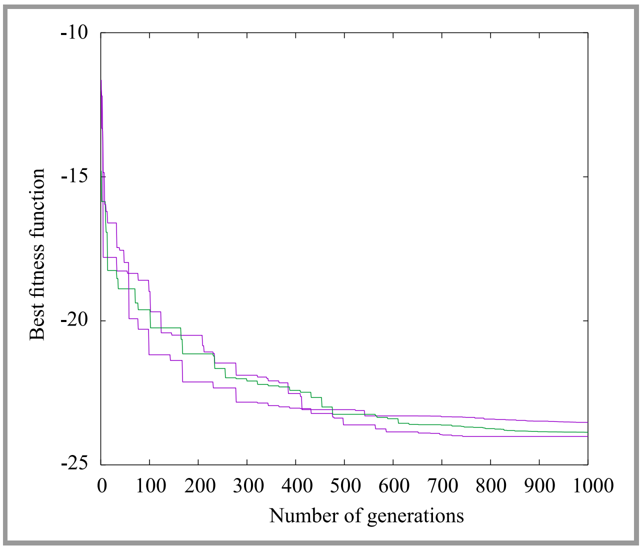


Fig. 8. Evolutionary process of the fitness value of a 20-element array using AGMCSO.

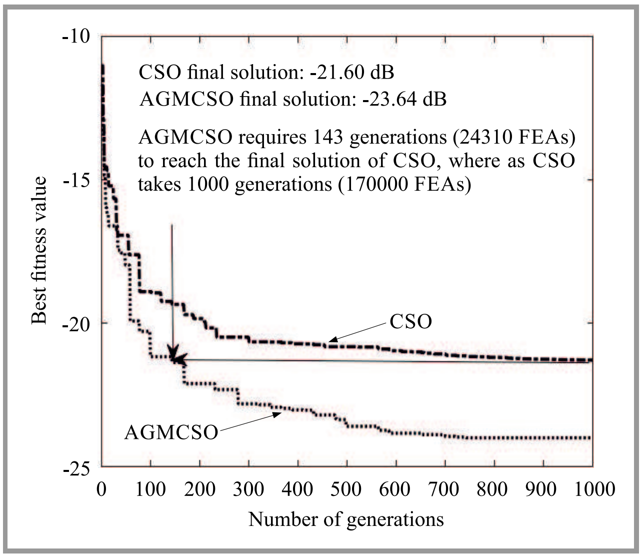


Fig. 9. Evolutionary process of the fitness value of a 20-element linear array using CSO and AGMCSO.

Table 7
Comparative results for the synthesis of a 20-element linear unequally spaced array

Number of elements	Algorithm	PSLL [dB]	Aperiodic FNBW	UIPA FNBW
20	CSO	−21.60	10	3.5
	FIPSO	−20.58		
	PPSO	−16.85		
	AGMCSO	−23.64		

AGMCSO generate the value of −23.64 dB, whereas CSO, DE, FIPSO and PPSO −21.29 dB, −18.42 dB, −20.58 dB, and −16.84 dB, respectively. AGMCSO shows a low PSLL level that is by −2 dB lower compared with CSO. It can be seen from Fig. 9 that AGMCSO outperforms CSO in terms of the convergence rate. AGMCSO takes 24310 FEAs to reach the final solution with CSO of −21.29 dB. The success rate of achieving a similar final value of AGMCSO is evident from Fig. 8.

5.2. 32-element Linear Array

In the second example, a 32-element array is synthesized to achieve the minimum PSLL. Table 8 shows the positions of ele-ments optimized by using AGMCSO and CSO. A comparison of the PSLL obtained using CSO, DE, CLPSO and AGMCSO is shown in Table 9. The best PSLL for a 32-element linear array in 10 runs was found to be −20.69 dB for CSO, −22.65 dB for DE [7], −22.75 dB for CLPSO [15] and −23.4076 dB for AGMCSO. The radiation pattern achieved using AGMCSO, along with UIPA and CSO, is shown in Fig. 10. The convergence character-

Table 8
Positions of a 32-element array optimized using AGMCSO and CSO

Element (n)	Position $\frac{x_n}{\lambda}$	
	AGMCSO	CSO
1	0.3934	0.2952
2	0.6988	0.9181
3	1.0263	1.5182
4	1.3399	2.1504
5	1.7294	2.8029
6	2.1038	3.4503
7	2.3565	4.1665
8	2.8756	4.8664
9	3.2624	5.5887
10	3.7256	6.3456
11	4.1031	7.0395
12	4.7333	7.8415
13	5.2702	8.6343
14	5.8879	9.7310
15	6.7224	11.4967
16	7.4994	12.2483

Table 9
Comparative results for the synthesis of a 32-element linear unequally spaced array

Number of elements	Algorithm	PSLL [dB]	Aperiodic FNBW	UIPA FNBW
32	CSO	−20.69	5	3.5
	DE	−22.65		
	CLPSO	−22.75		
	AGMCSO	−23.40		

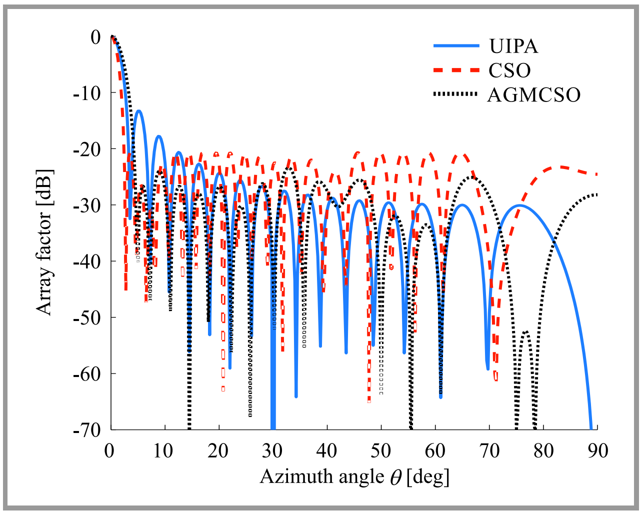


Fig. 10. Radiation pattern of the 32 elements array using AGMCSO, CSO, and UIPA.

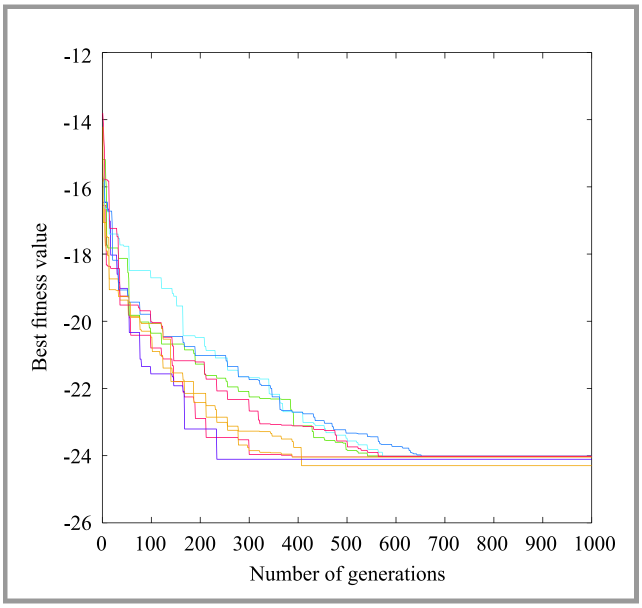


Fig. 11. Evolutionary process of the fitness value of the 32 elements array using AGMCSO.

istics for a 32-element linear array using AGMCSO for 10 independent runs are shown in Fig. 11. The convergence plots of AGMCSO and CSO are shown in Fig. 12.

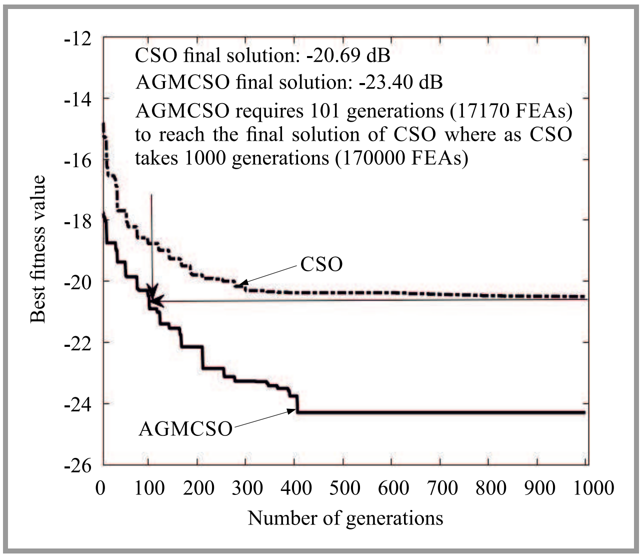


Fig. 12. Evolutionary process of the fitness value of a 32-element linear array using CSO and AGMCSO.

AGMCSO produces a PSLL that is by -2.71 dB lower compared with CSO (Table 9). CSO takes 170000 FEAs to reach its final solution. AGMCSO requires 24310 FEAs to reach the final solution of CSO (Fig. 12) and the deviation in achieving the final solution is low for several independent runs and shows the reliability of the proposed AGMCSO method (Fig. 11).

Overall, AGMCSO outperforms the traditional CSO approach in terms of low PSLL and convergence speed. Accelerated Gaussian mutation leads to defining positions located at better locations, by preventing premature convergence. AGMCSO had shown superior results compared to the classic CSO approach, in terms of solution accuracy and offers a PSLL value that is by -2 dB lower compared to CSO. Computational speed is greatly enhanced by the proposed AGMCSO methods. AGMCSO outperforms CSO in terms of convergence speed and requires 15% of the CSO's FEAs to reach the final solution of CSO.

6. Conclusion

In this paper, unequally spaced arrays with low PSLL have been designed using a modified AGMCSO optimization algorithm. The Gaussian mutation with an acceleration parameter has been introduced in the position-updated equation of the traditional CSO approach to enhance solution accuracy and convergence rate. The effectiveness of AGMCSO has been benchmarked using multimodal, 30-, 100- and 1000-dimensional problems. The simulations show that the proposed AGMCSO algorithm outperforms popular optimization techniques in terms of solution accuracy and convergence rate. A detailed analysis of the impact of all AGMCSO parameters on its overall performance has been carried out. We have applied AGMCSO in the synthesis of unequally spaced antenna arrays to suppress PSLL. 20- and 32-element linear arrays have been synthesized and the numerical results illustrate that AGMCSO outperforms the conventional and modified algorithm in terms of low PSLL with narrow FNBW.

References

[1] D. Khzmalyan and A. S. Kondrat'yev, "Phase-only synthesis of antenna array amplitude pattern", *Int. J. Electron.*, vol. 81, no. 5, pp. 585–589, 1996 (DOI: 10.1080/002072196136490).

[2] K. K. Yan and Y. Lu, "Sidelobe reduction in array pattern synthesis using genetic algorithm", *IEEE Trans. Anten. Propag.*, vol. 45, no. 7, pp. 1117–1122, 1997 (DOI: 10.1109/8.596902).

[3] F. J. Ares-Pena, J. A. Gonzalez, E. Lopez, and S. R. Rengarajan, "Genetic algorithms in the design and optimization of antenna array patterns", *IEEE Trans. Anten. Propag.*, vol. 47, no. 3, pp. 506–510, 1999 (DOI: 10.1109/8.768786).

[4] K. Chen, Z. He, and C. Han, "A modified real GA for the sparse linear array synthesis with multiple constraints", *IEEE Trans. Anten. Propag.*, vol. 54, no. 7, pp. 2169–2173, 2006 (DOI: 10.1109/TAP.2006.877211).

[5] L. Cen, Z. L. Yu, and W. Ser, "Linear aperiodic array synthesis using an improved genetic algorithm", *IEEE Trans. Anten. Propag.*, vol. 60, no. 2, pp. 895–902, 2012 (DOI: 10.1109/TAP.2011.2173111).

[6] T. Reshma, K. V. Reddy, D. Pratap, and V. Agilan, "Parameters optimization using fuzzy rule-based multi-objective genetic algorithm for an event-based Rainfall-Runoff model", *Water Resour. Manag.*, vol. 32, no. 3, pp. 1501–1516, 2018 (DOI: 10.1007/s11269-017-1884-2).

[7] D. G. Kurup, M. Himidi, and A. Rydberg, "Synthesis of uniform amplitude unequally spaced antenna arrays using the differential evolution algorithm", *IEEE Trans. Anten. Propag.*, vol. 51, no. 9, pp. 2210–2217, 2003 (DOI: 10.1109/TAP.2003.816361).

[8] J. Q. Zhang and A. C. Sanderson, "JADE: Adaptive differential evolution with optional external archive", *IEEE Trans. Evol. Comput.*, vol. 13, no. 5, pp. 945–958, 2009 (DOI: 10.1109/TEVC.2009.2014613).

[9] C. Lin, A.-Y. Qing, and Q.-Y. Feng, "Synthesis of unequally spaced antenna arrays by using differential evolution", *IEEE Trans. Anten. Propag.*, vol. 58, no. 8, pp. 2553–2561, 2010 (DOI: 10.1109/TAP.2010.2048864).

[10] S. K. Goudos, K. Siakavara, and T. Samaras, "Sparse linear array synthesis with multiple constraints using differential evolution with strategy adaptation", *IEEE Anten. Wirel. Propag. Lett.*, vol. 10, pp. 670–673, 2011 (DOI: 10.1109/LAWP.2011.2161256).

[11] S. K. Goudos *et al.*, "Self-adaptive differential evolution applied to real-valued antenna and microwave design problems", *IEEE Trans. Anten. Propag.*, vol. 59, no. 4, pp. 1286–98, 2011 (DOI: 10.1109/TAP.2011.2109678).

[12] F. Zhang, W. Jia, and M. Yao, "Linear aperiodic array synthesis using differential evolution algorithm", *IEEE Anten. Wirel. Propag. Lett.*, vol. 12, pp. 797–800, 2013 (DOI: 10.1109/LAWP.2013.2270930).

[13] D. W. Boeringer and D. H. Werner, "Particle swarm optimization versus genetic algorithms for phased array synthesis", *IEEE Trans. Anten. Propag.*, vol. 52, no. 3, pp. 771–779, 2004 (DOI: 10.1109/TAP.2004.825102).

[14] M. M. Khodier and C. G. Christodoulou, "Linear array geometry synthesis with minimum sidelobe level and null controlling using particle swarm optimization", *IEEE Trans. Anten. Propag.*, vol. 53, no. 8, pp. 2674–2679, 2005 (DOI: 10.1109/TAP.2005.851762).

[15] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions", *IEEE Trans. Evol. Comput.*, vol. 10, no. 3, pp. 281–295, 2006 (DOI: 10.1109/TEVC.2005.857610).

[16] S. K. Goudos *et al*., "Application of comprehensive learning particle swarm optimizer to unequally spaced linear array synthesis with side lobe level suppression and null control", *IEEE Anten. Wirel. Propag. Lett*., vol. 9, pp. 125–129, 2010 (DOI: 10.1109/LAWP.2010.2044552).

[17] W.-B. Wang, Q.-Y. Feng, and D. Liu, "Application of chaotic particle swarm optimization algorithm to pattern synthesis of antenna arrays", *Progr. in Electromag. Res*., vol. 115, pp. 173–189, 2011 (DOI:10.2528/PIER11012305).

[18] Z. H. Zhan, J. Zhang, and Y. Li, "Orthogonal learning particle swarm optimization", *IEEE Trans, Evol. Comput*., vol. 15, no. 6, pp. 281–295, 2011 (DOI: 10.1109/TEVC.2010.2052054).

[19] R. Bhattacharya, T. K. Bhattacharyya, and R. Garg, "Position mutated hierarchical particle swarm optimization and its application in synthesis of unequally spaced antenna arrays", *IEEE Trans. Anten. Propag*., vol. 60, no. 7, pp. 3174–2181, 2012 (DOI: 10.1109/TAP.2012.2196917).

[20] X. Xia, J. Liu, and Z. Hu, "An improved particle swarm optimizer based on tabu detecting and local learning strategy in a shrunk search space", *Appl. Soft Comput*., vol. 23, pp. 76–90, 2014 (DOI:10.1016/j.asoc.2014.06.012).

[21] L. A. Bewoor, V. C. Prakash, and S. U. Sapkal, "Evolutionary hybrid particle swarm optimization algorithm for solving np-hard no-wait flow shop scheduling problems", *Algorithms*, vol. 10, no. 4, pp. 1–17, 2017 (DOI: 10.3390/a10040121).

[22] A. Ahilan *et al*., "Segmentation by fractional order Darwinian particle swarm optimization based multilevel thresholding and improved lossless prediction based compression algorithm for medical images", *IEEE Access*, vol. 7, pp. 89570–89580, 2019 (DOI: 10.1109/ACCESS.2019.2891632).

[23] R. R. Eva and Q. T. Oscar, "Linear array synthesis using an ant colony optimization based algorithm", *IEEE Trans. Anten. Propag*., vol. 49, no. 2, pp. 70–79, 2007 (DOI: 10.1109/MAP.2007.376644).

[24] M. S. Kumar and M. Venkatesan, "Multi-objective task scheduling using hybrid genetic ant colony optimization algorithm in cloud environment", *Wirel. Pers. Commun*., vol 107, pp. 1835–1848, 2019 (DOI: 10.1007/s11277-019-06360-8).

[25] C. J. Malar *et al*., "Multi constraints applied energy efficient routing technique based on ant colony optimization used for disaster resilient location detection in mobile ad-hoc network", *J. of Ambient Intell. and Human. Comput*., vol. 12, pp. 4007–4017, 2021 (DOI: 10.1007/s12652-020-01767-9).

[26] G. Panda, P. M. Pradhan, and B. Majhi, "IIR system identification using cat swarm optimization", *Expert Syst. Appl.: an Int. J*., vol. 38, no. 10, pp. 12671–12683, 2011 (DOI: 10.1016/j.eswa.2011.04.054).

[27] P.-W. Tsai, J.-S. Pan, S. M. Chen, and B. Y. Liao, "Enhanced parallel cat swarm optimization based on the Taguchi method", *Expert Syst. with Appl*., vol. 39, no. 7, pp. 6309–6319, 2012 (DOI:10.1016/j.eswa.2011.11.117).

[28] L. Pappula and D. Ghosh, "Linear antenna array synthesis using cat swarm optimization", *AEU – Int. J. Electron. Commun*., vol. 68, no. 6, pp. 540–549, 2014 (DOI: 10.1016/j.aeue.2013.12.012).

[29] J. Wang, "A new cat swarm optimization with adaptive parameter control", in *Genetic and Evolutionary Computing: Proceeding of the Eighth International Conference on Genetic and Evolutionary Computing, October 18-20, 2014, Nanchang, China*, H. Sun *et al*., Eds. *AISC*, vol. 329, pp. 69–78. Springer, 2015 (DOI: 10.1007/978-3-319-12286-1_8).

[30] T. Nireekshana, G. Kesava Rao, and S. Sivanaga Raju, "Available transfer capability enhancement with FACTS using cat swarm optimization", *Ain Shams Engin. J*., vol. 7, no. 1, pp. 159–167, 2016 (DOI: 10.1016/j.asej.2015.11.011).

[31] L. Pappula and D. Ghosh, "Synthesis of linear aperiodic array using Cauchy mutated cat swarm optimization", *AEU – Int. J. of Electron. and Commun*., vol. 72, pp. 52–64, 2017 (DOI: 10.1016/j.aeue.2016.11.016).

[32] L. Pappula and D. Ghosh, "Cat swarm optimization with normal mutation for fast convergence of multimodal functions", *Appl. Soft Comput*., vol. 66, pp. 473–491, 2018 (DOI: 10.1016/j.asoc.2018.02.012).

[33] R. Rajakumar, J. Amudhavel, P. Dhavachelvan, and T. Vengattaraman, "GWO-LPWSN: Grey Wolf optimization algorithm for node localization problem in wireless sensor networks", *J. of Comp. Netw. and Commun*., vol. 2, pp. 1–10, 2017 (DOI:10.1155/2017/7348141).

[34] X. Li and K. M. Luk, "The grey wolf optimizer and its applications in electromagnetics", *IEEE Trans. on Antenn. and Propag*., vol. 68, no. 3, 2020 (DOI: 10.1109/TAP.2019.2938703).

[35] M. Rajeswari, J. Amudhavel, S. Pothula, and P. Dhavachelvan, "Directed bee colony optimization algorithm to solve the nurse rostering problem", *Computat. Intell. and Neurosci*., vol. 2017, Article ID 6563498, 2017 [Online]. Available: https://downloads.hindawi.com/journals/cin/2017/6563498.pdf

[36] R. S. Raghav, J. Amudhavel, and P. Dhavachelvan, "Enhanced artificial bee colony optimization for solving economic load dispatch, *IIOAB J*., vol. 8, no. 2, pp. 208–216, 2017 [Online]. Available: https://www.iioab.org/IIOABJ_8.2_208-216.pdf

[37] S.-C. Chu and P.-W. Tsai, "Computational intelligence based on the behavior of cats", *Int. J. Innov. Comput., Inform. and Contr.* (*IJICIC*), vol. 3, no. 1, pp. 163–173, 2007 [Online]. Available: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.103.3698&rep=rep1&type=pdf

[38] X. Yao, Y. Liu, and G. M. Lin, "Evolutionary programming made faster", *IEEE Trans. Evol. Comput*., vol. 3, no. 2, pp. 82–102, 1999 (DOI: 10.1109/4235.771163).

K. Prasanna Kumar completed his Ph.D. from Koneru Lakshmaiah Education Foundation in 2021, focusing on antenna arrays and optimization techniques. He is currently working as the Head of R&D at St. Peter's Engineering College, Hyderabad. He had obtained his M.Tech. from Pydah College of Engineering and Technology and B.Tech. in ECE from MVGR College of Engineering. He has 13 years of teaching experience and handled various academic and administrative roles.

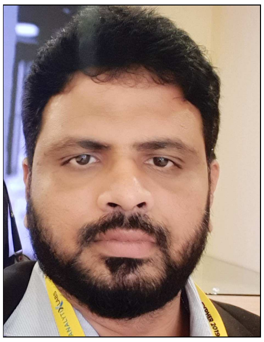
E-mail: prasannakorada@gmail.com
Department of ECE
St. Peter's Engineering College
Hyderabad, India

Lakshman Pappula completed his Ph.D. from Indian Institute of Technology Bhubaneswar in 2017, specializing in antenna arrays and optimization techniques. He also completed his M.Sc. in RF and Microwave Engineering. Currently, he is working as the Deputy HOD and an Associate Professor at the Department of Electronics and Communications Engineering, KLEF, Guntur, Andhra Pradesh, India. His interests include antenna theory, electromagnetic fields and propagation, communication theory, soft computing, artificial neural network, radar and navigational aids, wireless and communication theory.

E-mail: lakshman.pappula@kluniversity.in
Department of ECE
Koneru Lakshmaiah Education Foundation
Guntur, India

B. T. P. Madhav works as Professor and Associate Dean at KLEF. His research interests include antennas, liquid crystals applications and wireless communications.

E-mail: btpmadhav@kluniversity.in
Department of ECE
Koneru Lakshmaiah Education Foundation
Guntur, India



V. S. V. Prabhakar completed his Ph.D. from JNTU, Kakinada in the field of VLSI design. He completed his M.Sc. in chip systems in Sweden, and B.Tech. from Osmania University. Currently, he is working as Professor and Director, Industry Connect, at the Koneru Lakshmaiah Education Foundation. He also worked as Cisco Chair in IIDT, Tirupathi, as Head of Department, ECE in KLEF, and as Principal at the Avanthi Institute of Engineering and Technology.

E-mail: prabhakarvsv@kluniversity.in
Department of ECE
Koneru Lakshmaiah Education Foundation
Guntur, India