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 beamwidth (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 aeriodic 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.

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 antennas 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 algo-
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, and 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 the newly updated cats 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:
  \[
P_{m}^{n} = [P_{mn}] \quad \text{where} \quad n = 1, \ldots, D , \quad (1)
  \]
  \[
  \text{Vel}_{m}^{n} = [\text{Vel}_{mn}] \quad \text{where} \quad n = 1, \ldots, D , \quad (2)
  \]
  • for each dimension the position and velocity of \( m \)-th cat is updated as:
  \[
  \text{Vel}_{m}^{n+1} = \text{Vel}_{m,n} + \omega \cdot \text{Vel}_{m,n} + C \cdot r \cdot (P_{\text{gbest}} - P_{m,n}) , \quad (3)
  \]
  \[
  P_{m}^{n+1} = P_{m,n} + \text{Vel}_{m}^{n+1} , \quad (4)
  \]
  where \( g \) represents the generation number, \( m \) is the cat’s index in a swarm, \( n \) represents the cat’s position index, \( \text{Vel}_{m,n} \) 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 \( P_{\text{gbest}} \) 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 \( \sigma \) and the mean \( \mu \) of Gaussian distribution density function is:

\[
 f_{\text{normal}}(p; \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(p-\mu)^2}{2\sigma^2}} . \quad (5)
\]

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

\[
 G(\mu, \sigma^2) = \mu + \sigma G(0,1) . \quad (6)
\]

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 \( x_{r}^{m} \) is generated by Gaussian mutation which is:

\[
 x_{r}^{m} = x_{i} + G(0, \sigma^2) = x_{i} + \sigma \cdot G(0,1) , \quad (7)
\]

where \( x_{i} \) is the unmutated individual, and \( \sigma \) is conveyed as the selected dimension’s mutated value. Therefore, the position of each dimension of \( i \)-th cat is:

\[
 x_{r}^{m} = x_{i} + \left[ \text{SRD} \cdot \left( 1 - \frac{g}{\text{gen}} \right) \cdot x_{i} \cdot G(0,1) \right] . \quad (8)
\]

To enhance the local convergence properties, we have adopted an accelerating component in the \( x_{r}^{m} \).

3.1. Time Complexity of AGMCSO

The SM time complexity is considered as \( O[\text{N}_{n}, \log(\text{N}_{n})] \), where \( \text{N}_{n} \) 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 \( O[\text{N}_{n}, \log(\text{N}_{n})] \). 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.
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 phases 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>
<thead>
<tr>
<th>Name</th>
<th>Symbol</th>
<th>Description</th>
<th>(x^*)</th>
<th>(f(x^*))</th>
<th>Search range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sphere</td>
<td>(f_1(x))</td>
<td>(\sum_{i=1}^{D} x_i^2)</td>
<td>0, \ldots, 0</td>
<td>0</td>
<td>([-5.12, 5.12]^D)</td>
</tr>
<tr>
<td>Zakharov</td>
<td>(f_2(x))</td>
<td>(\sum_{i=1}^{D} (0.5x_i + 1) \left(100(x_i - 1)^2 + x_i^2\right))</td>
<td>1, \ldots, 1</td>
<td>0</td>
<td>([-10, 10]^D)</td>
</tr>
<tr>
<td>Rosenbrock</td>
<td>(f_3(x))</td>
<td>(\prod_{i=1}^{D} -\cos(\pi x_i) + 1)</td>
<td>1, \ldots, 1</td>
<td>0</td>
<td>([-10, 10]^D)</td>
</tr>
<tr>
<td>Levy</td>
<td>(f_4(x))</td>
<td>(\sum_{i=1}^{D} \left(\frac{100}{d} - 1\right)\left(1 + 10\sin^2(\pi x_i)\right)) + (\frac{100}{d} - 1\left(1 + \sin^2(2\pi x_i)\right))</td>
<td>1, \ldots, 1</td>
<td>0</td>
<td>([-10, 10]^D)</td>
</tr>
<tr>
<td>Ackley</td>
<td>(f_5(x))</td>
<td>(-\left(\sum_{i=1}^{D} \frac{x_i^2}{x_i^2 + a} - \frac{1}{a^2}\right)) + (a + e^1)</td>
<td>0, \ldots, 0</td>
<td>0</td>
<td>([-32, 32]^D)</td>
</tr>
<tr>
<td>Rastrigin</td>
<td>(f_6(x))</td>
<td>(10d + \sum_{i=1}^{D} \left(x_i^2 - 10\cos(2\pi x_i)\right))</td>
<td>0, \ldots, 0</td>
<td>0</td>
<td>([-5, 10]^D)</td>
</tr>
<tr>
<td>Griewank</td>
<td>(f_7(x))</td>
<td>(\prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1)</td>
<td>0, \ldots, 0</td>
<td>0</td>
<td>([-600, 600]^D)</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>AGMCSO</th>
<th>CSO</th>
<th>PSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Factor</td>
<td>Amount</td>
<td>Factor</td>
</tr>
<tr>
<td>Primary cats</td>
<td>50</td>
<td>Primary cats</td>
</tr>
<tr>
<td>SRD</td>
<td>0.8 (80%)</td>
<td>SRD</td>
</tr>
<tr>
<td>CDC</td>
<td>80%</td>
<td>CDC</td>
</tr>
<tr>
<td>SMP</td>
<td>5</td>
<td>SMP</td>
</tr>
<tr>
<td>MR</td>
<td>0.8</td>
<td>MR</td>
</tr>
<tr>
<td>$r$</td>
<td>[0.1]</td>
<td>$r$</td>
</tr>
<tr>
<td>$\omega$</td>
<td>0.2–0.9</td>
<td>$\omega$</td>
</tr>
</tbody>
</table>

### Table 3

Comparison of average CPU time, convergence speed and SR

<table>
<thead>
<tr>
<th>Function</th>
<th>AGMCSO</th>
<th>CSO</th>
<th>PSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_1$</td>
<td>FEA</td>
<td>1360</td>
<td>–</td>
</tr>
<tr>
<td>Time</td>
<td>0.0877 s</td>
<td>–</td>
<td>3.45 s</td>
</tr>
<tr>
<td>SR</td>
<td>100%</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td>$f_2$</td>
<td>FEA</td>
<td>2108</td>
<td>–</td>
</tr>
<tr>
<td>Time</td>
<td>0.104 s</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>SR</td>
<td>100%</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$f_3$</td>
<td>FEA</td>
<td>340</td>
<td>–</td>
</tr>
<tr>
<td>Time</td>
<td>0.058 s</td>
<td>–</td>
<td>7.24 s</td>
</tr>
<tr>
<td>SR</td>
<td>100%</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td>$f_4$</td>
<td>FEA</td>
<td>2006</td>
<td>–</td>
</tr>
<tr>
<td>Time</td>
<td>0.0757 s</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>SR</td>
<td>100%</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$f_5$</td>
<td>FEA</td>
<td>2380</td>
<td>145800</td>
</tr>
<tr>
<td>Time</td>
<td>0.1254 s</td>
<td>10.12</td>
<td>4.53 s</td>
</tr>
<tr>
<td>SR</td>
<td>100%</td>
<td>34</td>
<td>100%</td>
</tr>
<tr>
<td>$f_6$</td>
<td>FEA</td>
<td>2312</td>
<td>–</td>
</tr>
<tr>
<td>Time</td>
<td>0.0726 s</td>
<td>–</td>
<td>4.64 s</td>
</tr>
<tr>
<td>SR</td>
<td>100%</td>
<td>9</td>
<td>199%</td>
</tr>
<tr>
<td>$f_7$</td>
<td>FEA</td>
<td>1394</td>
<td>–</td>
</tr>
<tr>
<td>Time</td>
<td>0.1143 s</td>
<td>–</td>
<td>82.42 s</td>
</tr>
<tr>
<td>SR</td>
<td>100%</td>
<td>0</td>
<td>100%</td>
</tr>
</tbody>
</table>

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 and PSO (bold figures mean the best result)

<table>
<thead>
<tr>
<th>Function</th>
<th>AGMCSO</th>
<th>CSO</th>
<th>PSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_1$</td>
<td>$7.77 \cdot 10^{-291} \pm 0$</td>
<td>$3.41 \cdot 10^{-40} \pm 1.99 \cdot 10^{-41}$</td>
<td>30.6 ± 19</td>
</tr>
<tr>
<td>$f_2$</td>
<td>$7.8 \cdot 10^{-268} \pm 0$</td>
<td>$2.72 \cdot 10^{-32} \pm 0.24 \cdot 10^{-23}$</td>
<td>5 \cdot 10^{20} ± 15</td>
</tr>
<tr>
<td>$f_3$</td>
<td>$50 \pm 0$</td>
<td>29.6 ± 2.1</td>
<td>168 ± 6.7</td>
</tr>
<tr>
<td>$f_4$</td>
<td>$0 \pm 0$</td>
<td>$3.1 \cdot 10^{-24} \pm 0.15 \cdot 10^{-40}$</td>
<td>18.1 ± 23</td>
</tr>
<tr>
<td>$f_5$</td>
<td>$8.88 \cdot 10^{-16} \pm 0$</td>
<td>$4.68 \cdot 10^{-7} \pm 3.05 \cdot 10^{-5}$</td>
<td>1 ± 0.6</td>
</tr>
<tr>
<td>$f_6$</td>
<td>$0 \pm 0$</td>
<td>$3.18 \cdot 10^{-14} \pm 0.169$</td>
<td>3.5 ± 8.9</td>
</tr>
<tr>
<td>$f_7$</td>
<td>$0 \pm 0$</td>
<td>$1.23 \cdot 10^{-7} \pm 2.5 \cdot 10^{-4}$</td>
<td>5.1 ± 1.8</td>
</tr>
</tbody>
</table>

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.
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:

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

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

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)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_1$</td>
<td>$5.7 \cdot 10^{-4}$ $\pm 1.3 \cdot 10^{-4}$</td>
<td>$1.3 \cdot 10^{-54}$ $\pm 9.2 \cdot 10^{-54}$</td>
<td>$4.4 \cdot 10^{-14}$ $\pm 1.71 \cdot 10^{-14}$</td>
<td>$1.1 \cdot 10^{-38}$ $\pm 1.21 \cdot 10^{-38}$</td>
<td>$1.3 \cdot 10^{-49}$ $\pm 7.31 \cdot 10^{-49}$</td>
<td>$0 \pm 0$</td>
<td>$1.68 \cdot 10^{-21}$</td>
<td>$7.77 \cdot 10^{-291}$ $\pm 0$</td>
</tr>
<tr>
<td>$f_2$</td>
<td>$5.0 \pm 5.8$</td>
<td>$0.3 \pm 1.1$</td>
<td>$21 \pm 2.9$</td>
<td>$1.2 \pm 1.4$</td>
<td>$51 \pm 42$</td>
<td>N/A</td>
<td>23.5</td>
<td>$7.8 \cdot 10^{-291}$ $\pm 0$</td>
</tr>
<tr>
<td>$f_3$</td>
<td>$180 \pm 2100$</td>
<td>$4.4 \cdot 10^{-15}$ $\pm 0$</td>
<td>$0 \pm 0$</td>
<td>$4.14 \cdot 10^{-14}$ $\pm 0$</td>
<td>$1.2 \cdot 10^{-14}$ $\pm 4.6 \cdot 10^{-15}$</td>
<td>$6.4 \cdot 10^{-15}$ $\pm 3.7 \cdot 10^{-15}$</td>
<td>$6.54 \cdot 10^{-12}$</td>
<td>$8.88 \cdot 10^{-16}$ $\pm 0$</td>
</tr>
<tr>
<td>$f_4$</td>
<td>$460 \pm 1200$</td>
<td>$0 \pm 0$</td>
<td>$4.85 \cdot 10^{-10}$ $\pm 0.361$</td>
<td>$0 \pm 0$</td>
<td>$4.7 \cdot 10^{-16}$ $\pm 9.71 \cdot 10^{-17}$</td>
<td>$86 \pm 10$</td>
<td>76.6</td>
<td>$0 \pm 0$</td>
</tr>
<tr>
<td>$f_5$</td>
<td>$1600 \pm 2200$</td>
<td>$2.0 \cdot 10^{-4}$ $\pm 1.4 \cdot 10^{-3}$</td>
<td>$31 \pm 0.46$</td>
<td>$0 \pm 0$</td>
<td>$170 \pm 2.21 \cdot 10^{-3}$</td>
<td>$3.50 \cdot 10^{-3}$ $\pm 7.10 \cdot 10^{-3}$</td>
<td>$0 \pm 0$</td>
<td>$0 \pm 0$</td>
</tr>
</tbody>
</table>
Unequally Spaced Antenna Array Synthesis Using Accelerating Gaussian Mutated Cat Swarm Optimization

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

$$\text{Obj}(X) = \max \left( \frac{|\text{AF}(X, \theta_0)|}{|\text{AF}_{\text{max}}|} \right),$$

where $X = (X_1, X_2, \ldots, X_N)$ is the element position vector, $\theta_0$ is defined as the angular region excluding the main lobe, the main peak of the pattern is $\text{AF}_{\text{max}}$.

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

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

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.
AGMCSO generate the value of $-23.64 \text{ dB}$, whereas CSO, DE, FIPSO and PPSO $-21.29 \text{ dB}, -18.42 \text{ dB}, -20.58 \text{ dB},$ and $-16.84 \text{ dB}$, respectively. AGMCSO shows a low PSLL level that is by $-2 \text{ dB}$ 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 $-21.29 \text{ dB}$. 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 elements 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 $-20.69 \text{ dB}$ for CSO, $-22.65 \text{ dB}$ for DE [7], $-22.75 \text{ dB}$ for CLPSO [15] and $-23.4076 \text{ dB}$ for AGMCSO. The radiation pattern achieved using AGMCSO, along with UIPA and CSO, is shown in Fig. 10. The convergence character-
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.

![Convergence plots of AGMCSO and CSO](image)

**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


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, electromagnetics fields and propagation, communication theory, soft computing, artificial neural network, radar and navigational aids, wireless and communication theory.
Unequally Spaced Antenna Array Synthesis Using Accelerating Gaussian Mutated Cat Swarm Optimization

E-mail: lakshman.pappula@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

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