



M.Sc. Thesis

A Modified Genetic Algorithm for Sparse Optical Phased Array Design

Kaishen Lin B.Sc

Abstract

With the continuous advancement of autonomous driving technology, the precision and efficiency of perception systems have become increasingly critical. Among various sensors, LiDAR plays a central role, and solid-state optical phased arrays (OPAs) are widely regarded as a promising future direction. However, traditional uniform OPAs often face challenges such as high power consumption and limited scalability.

This thesis addresses the design and optimization of sparse non-uniform OPAs, aiming to balance trade-offs among the number of antennas, element spacing, beamwidth, and side lobe level. We propose a novel formulation that simultaneously considers array sparsity and performance while enforcing distance constraints, which is solved using a modified genetic algorithm. The simulation results reveal a clear trade-off between sparsity and array performance, while also offering practical solutions to the constraints faced by current LiDAR systems. Furthermore, we investigate the impact of array configuration on beam steering and introduce a mathematical transformation that reformulates the steering design problem to be compatible with our model. The comparison results demonstrate that the proposed approach significantly improves array performance across the entire steering range.

A Modified Genetic Algorithm for Sparse Optical Phased Array Design

THESIS

submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

in

ELECTRICAL ENGINEERING

by

Kaishen Lin B.Sc
born in Shenzhen, China

This work was performed in:

Signal Processing Systems Group
Department of Microelectronics
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology



Delft University of Technology

Copyright © 2025 Signal Processing Systems Group
All rights reserved.

DELFT UNIVERSITY OF TECHNOLOGY
DEPARTMENT OF
MICROELECTRONICS

The undersigned hereby certify that they have read and recommend to the Faculty of Electrical Engineering, Mathematics and Computer Science for acceptance a thesis entitled “**A Modified Genetic Algorithm for Sparse Optical Phased Array Design**” by **Kaishen Lin B.Sc** in partial fulfillment of the requirements for the degree of **Master of Science**.

Dated: 29 August 2025

Chairman:

dr. Geethu Joseph

Advisors:

dr. Geethu Joseph

dr.ir. Jac Romme

Committee Members:

prof.dr.ir. G.J.T. Leus

dr.ir. J.N. Driessen

Abstract

With the continuous advancement of autonomous driving technology, the precision and efficiency of perception systems have become increasingly critical. Among various sensors, LiDAR plays a central role, and solid-state optical phased arrays (OPAs) are widely regarded as a promising future direction. However, traditional uniform OPAs often face challenges such as high power consumption and limited scalability.

This thesis addresses the design and optimization of sparse non-uniform OPAs, aiming to balance trade-offs among the number of antennas, element spacing, beamwidth, and side lobe level. We propose a novel formulation that simultaneously considers array sparsity and performance while enforcing distance constraints, which is solved using a modified genetic algorithm. The simulation results reveal a clear trade-off between sparsity and array performance, while also offering practical solutions to the constraints faced by current LiDAR systems. Furthermore, we investigate the impact of array configuration on beam steering and introduce a mathematical transformation that reformulates the steering design problem to be compatible with our model. The comparison results demonstrate that the proposed approach significantly improves array performance across the entire steering range.

Acknowledgments

First and foremost, I would like to express my deepest appreciation to my supervisors, dr. Geethu Joseph and dr.ir. Jac Romme. Their support and guidance have been invaluable throughout this project. Without their help, I would have easily lost direction at an early stage. Their critical questions constantly challenged me to think deeper, broadened my perspectives, and pushed me to validate my models from different angles.

I am also sincerely grateful to my dear friends in the Jingniang group. Thank you for standing by me through the gloomy Dutch weather, endless exams, and deadlines, and for sharing both the struggles and the joys of this journey. Beyond the study, your companionship in exploring the world with me—especially the “Dezhou” world—has left me with memories I will always treasure.

Finally, my heartfelt gratitude goes to my parents. Thank you for your unconditional love, for the freedom you gave me in pursuing my studies abroad, and for being my greatest source of strength and support throughout this journey.

Kaishen Lin B.Sc
Delft, The Netherlands
29 August 2025

Contents

Abstract	v
Acknowledgments	vii
1 Introduction	1
1.1 LiDAR system	2
1.2 The components of OPA	3
1.3 Radiation pattern and performance metrics	4
1.4 Motivation for sparse non-uniform design	7
1.5 Outline	7
2 Literature Review	9
2.1 Optimization Problem Formulations for OPA Design	9
2.2 Review of different sparse non-uniform OPA methods	11
2.2.1 Evolutionary algorithms	11
2.2.2 Algorithms based on matrix decomposition	12
2.2.3 Convex optimization and compressed sensing algorithms	12
2.3 Chapter summary	13
3 Problem Formulation and Design of Non-steering OPA	15
3.1 Proposed Method and Problem Formulation	15
3.2 Classical GA Introduction	16
3.3 Modified GA	17
3.3.1 Fitness function	17
3.3.2 Operators Selection	17
3.3.3 Encoding operator	18
3.4 Tuning Crossover and Mutation Rates	21
3.4.1 Impact of Crossover rate and mutation rate	21
3.5 Model Verification	23
3.5.1 Construction of test solution space	23
3.5.2 Validation I: Finding optimal PSLL without considering sparsity ($\alpha'=0$)	23
3.5.3 Validation II: Evaluation of sparsity controller	24
3.6 Results and Discussion	25
3.6.1 Simulation setups	26
3.6.2 Model Speed-Up	27
3.6.3 Results	27
3.6.4 Experiment on distance control	28
3.6.5 Comparison and discussion	30
3.7 Conclusion	32

4	Design of steering OPA	33
4.1	Impact on array performance during steering process	33
4.2	Design of steering OPA	34
4.3	Results and discussion	36
5	Conclusion and Future Work	41
5.1	Conclusion	41
5.2	Future work	42

List of Figures

1.1	Various sensors in autonomous vehicles [1]	1
1.2	Different steering types of LiDAR System	2
1.3	OPA components	3
1.4	Linear phased array diagram	4
1.5	Beam pattern of a uniform linear array with $N = 11$ elements and element spacing of 1.5 wavelength.	5
2.1	An example of the desired beam pattern	10
3.1	An example of a chromosome under grid architecture design	17
3.2	Variety of operators used in GA [2]	18
3.3	Simplified illustration of the tournament selection process with a tournament size of two	20
3.4	Grid search results of <i>PSLL</i> under different crossover rate and mutation rate combination	22
3.5	(a) is the best result in each iteration ($\alpha' = 0$), and (b) is the array pattern of the best solution using 14 antennas	24
3.6	(a) is the best result in each iteration ($\alpha = 44.5$), and (b) is the array pattern of the best solution using 14 antennas	25
3.7	(a) is the best result in each iteration ($\alpha = 45$), and (b) is the array pattern of the best solution using 10 antennas	26
3.8	(a) is the best result in each iteration ($\alpha = 59$), and (b) is the array pattern of the best solution using 8 antennas	26
3.9	Array pattern of the optimal 641-element configuration obtained when $\alpha = 0$	28
3.10	Array pattern of the optimal 425-element configuration obtained when $\alpha = 15$	29
3.11	Array pattern of the optimal 139-element configuration obtained when $\alpha = 24$	30
3.12	The relationship between α and number of antenna and <i>PSLL</i>	31
3.13	Different grid spaces searching results	32
4.1	<i>PSLL</i> of the optimized array across the entire steering angle range: (a) with 641 elements and (b) with 425 elements	33
4.2	Array patterns of the optimized 641-element array at steering angles of (a) 2° and (b) 10°	34
4.3	Array patterns of the optimized 425-element array at steering angles of (a) 2° and (b) 10°	34
4.4	Array pattern of the optimal 643-element configuration obtained by the new pattern expression.	37
4.5	Performance Comparison Without and With Steering Process Consideration(≈ 640 Antennas)	38

4.6	Performance Comparison Without and With Steering Process Consideration(≈ 427 Antennas)	38
4.7	Performance Comparison Without and With Steering Process Consideration(≈ 139 Antennas)	39

List of Tables

3.1	The chosen operators	21
3.2	Detailed grid results of average PSLI under different crossover rate and mutation rate combinations	22
3.3	Maximum <i>PSLI</i> for chromosomes with the same number of antennas .	24
3.4	The jump point during the exhaustive test of α	25
3.5	Array pattern characteristics of the best solution under different values of α	28
3.6	The results of using reweighted l_1 -norm method in paper [3]	31
4.1	Performance Comparison under Varying α' and Chromosome Size . . .	37

Introduction

In recent years, with the rapid advancement of sensor technologies and artificial intelligence, autonomous driving has achieved significant breakthroughs and is gradually moving toward large-scale applications. Autonomous driving systems are typically composed of three core components: perception of the environment, behavior planning, and motion execution [4]. Among them, the perception system serves as the “eyes” of the vehicle, responsible for capturing and interpreting data from the environment surrounding the vehicle. It provides crucial input to the planning module and forms the foundation for ensuring the safety and stability of the entire autonomous driving system. Several sensors are commonly used in the perception system, including LiDAR (Light Detection and Ranging), millimeter-wave radar, cameras, and ultrasonic sensors. Figure 1.1 shows various sensors equipped in autonomous vehicles.

In all these sensors, LiDAR stands out as the only sensor capable of delivering precise three-dimensional spatial information about surrounding objects with both high spatial and temporal resolution. In this thesis, we focus on LiDAR as it has been rapidly

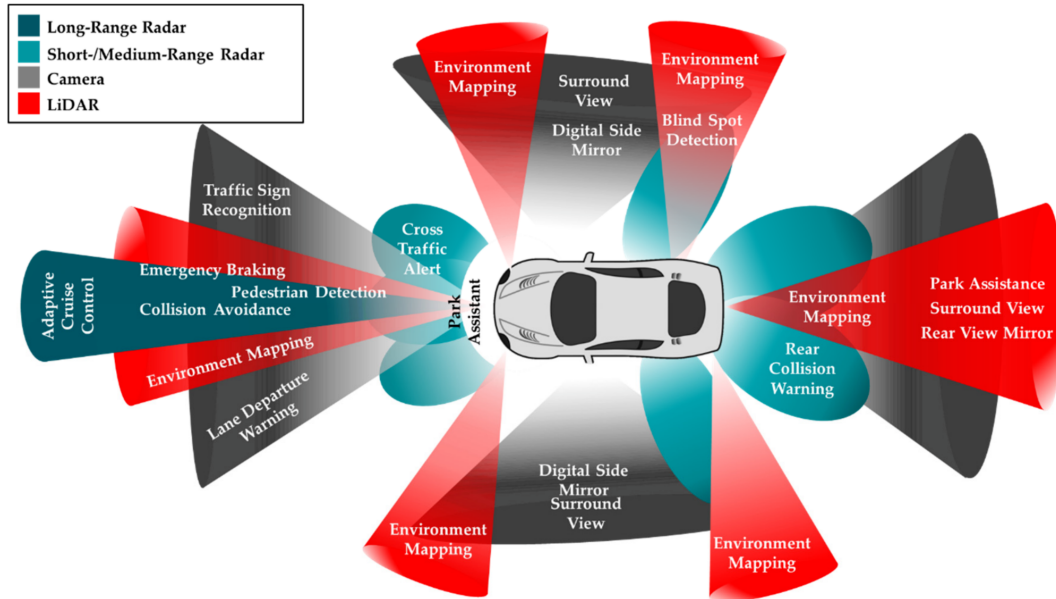


Figure 1.1: Various sensors in autonomous vehicles [1]

advancing with ongoing developments in industrial technology. Compared to radar, LiDAR operates at a shorter wavelength, which provides higher angular resolution. The working principle of LiDAR is similar to that of radar: it emits pulsed light waves and measures the time it takes for the light to reflect off a surface and return to the sensor. This information is then used to generate spatial data, including an object’s

position, velocity, shape, and distance. In contrast, camera sensors lack the ability to accurately capture distance information.

1.1 LiDAR system

In order to get information from the surrounding environment, LiDAR systems need to steer their beams. The steering principle can be broadly classified into three types based on their scanning mechanisms: mechanical rotation LiDAR, flash LiDAR, and solid-state LiDAR [5].

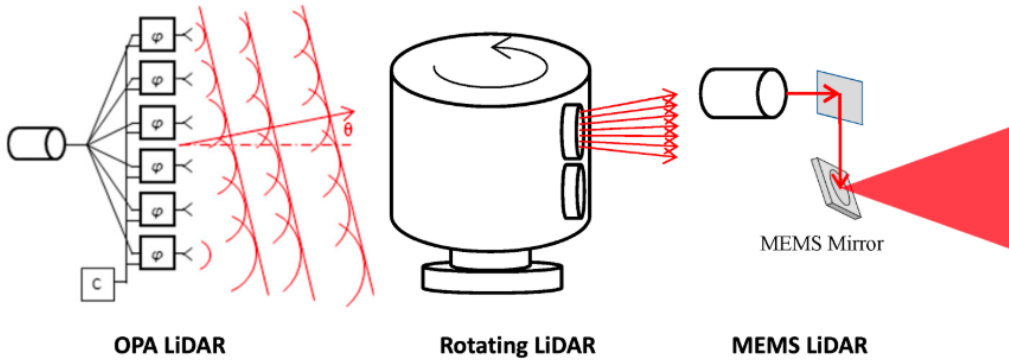


Figure 1.2: Different steering types of LiDAR System

Mechanical rotation LiDAR uses rotating mechanical components to direct the laser emitter, enabling panoramic scanning. This enables comprehensive environmental coverage but comes at the cost of increased size, high manufacturing expense, limited integrability, and susceptibility to mechanical wear. A well-known example of this type is Velodyne’s multi-beam rotating LiDAR system.

Flash LiDAR adopts a camera-like approach, emitting a broad laser pulse to illuminate the entire scene at once and capturing the reflected light to form a point cloud. This design eliminates moving parts and simplifies system architecture, improving robustness and integration potential. However, it suffers from lower spatial resolution due to the lack of point-by-point scanning, and the power spread across the wide beam limits its effective detection range.

In contrast, solid-state LiDAR eliminates the need for large mechanical components, making it more compact and robust. Among solid-state solutions, the micro-electromechanical system approach is the most widely used. It employs micro-mirrors to reflect the light to control the direction and scanning range of laser beams via high-speed mechanical motion. It is not entirely solid-state since it relies on moving micro-mirrors for beam steering. The limited oscillation amplitude of these mirrors restricts the field of view, and their intricate micro-mechanical structures demand high manufacturing precision, resulting in higher costs.

Another solid-state approach is the Optical Phased Array (OPA). OPA represents one of the most promising scanning approaches and is also the focus of this thesis. OPA utilizes light wave interference, controlling the phase of each emitter to steer the

beam without the need for mechanical components. This principle is similar to that of phased array radar.

Compared to traditional LiDAR systems, OPAs are smaller and lighter because they eliminate bulky mechanical parts. They are also more robust, as the lack of moving components makes them resistant to shocks and vibrations. In addition, OPAs can be implemented with Silicon Photonics. This makes it possible to create photonic integrated circuits using standard semiconductor fabrication and avoiding costly optical components like lenses, significantly lowering manufacturing costs, making large-scale production more practical and affordable.

Motivated by the advantages of OPA, this thesis investigates the design of OPA for LiDAR applications, with the goal of achieving compact solid-state beam steering. The following section provides a detailed discussion of OPAs.

1.2 The components of OPA

An OPA transmitter typically contains several key components, including a laser input, power splitters, waveguides, grating couplers, and phase shifters [6]. Figure 1.3 shows the components in OPA.

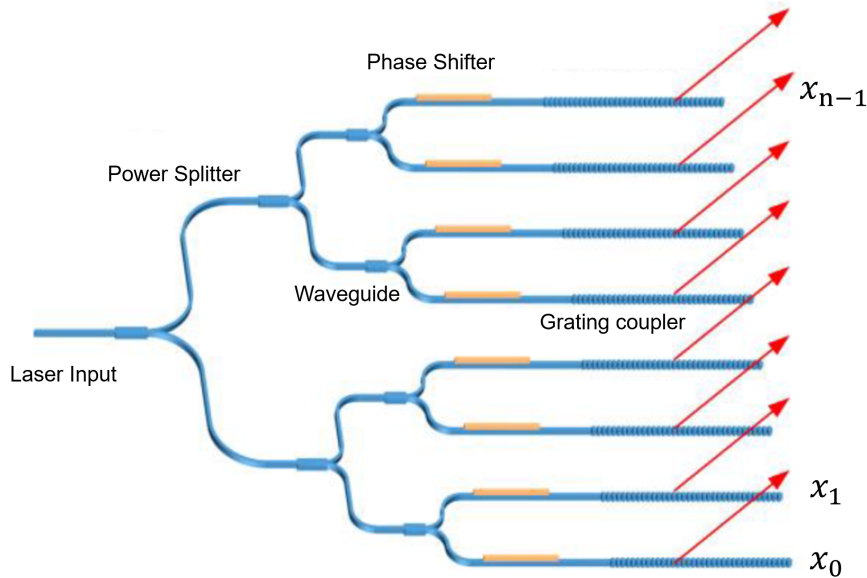


Figure 1.3: OPA components

The laser input provides the optical signal for the whole transmitter, and the power splitters distribute the power into two paths. The split power travels through the waveguide, which is very similar to the function of a wire in an electrical circuit. The grating coupler is a device for converting the light in the waveguide to free-propagating optical radiation [7]. In the design, grating couplers are optical antennas.

The phase shifter is responsible for controlling the phase of each element. There are two main types of phase shifters: thermo-optic and electro-optic phase shifters. Both

operate by changing the refractive index of the waveguides. Among them, thermo-optic phase shifters are the most widely used due to their lower optical loss and compatibility with CMOS manufacturing processes. However, thermo-optic shifters have the drawback of high power consumption, requiring up to tens or hundreds of milliwatts to achieve a π phase shift [3].

1.3 Radiation pattern and performance metrics

The concept of OPA still adheres to the fundamental principle of phased arrays: using wave interference to shape and steer the beam solely through phase control of individual antennas. The overall radiation pattern produced is the beam pattern.

The radiation from a single optical antenna can be described by a complex exponential representing its amplitude and phase:

$$p(t) = ae^{j\omega t + \phi},$$

where a denotes the amplitude, ϕ is the initial phase shift, ω is the angular frequency, and t is the time at which the signal is evaluated. Consider a 1D array with N optical elements. In the far field, the total electric field is the superposition of the contributions from all array elements. In a phased array, all elements emit simultaneously and use phase delays to steer the beam. Therefore, the beam pattern can be described as:

$$P(\theta) = \sum_{n=1}^N a_n e^{j(kx_n \sin \theta + \phi_n)}, \quad (1.1)$$

where $k = \frac{2\pi}{\lambda}$ denotes the wave number in free space, a_n denotes the amplitude of the n -th element, x_n is the position of each antenna, θ is the azimuth angle of the observation, λ is the working wavelength and ϕ_n is the phase shift of the n -th element.

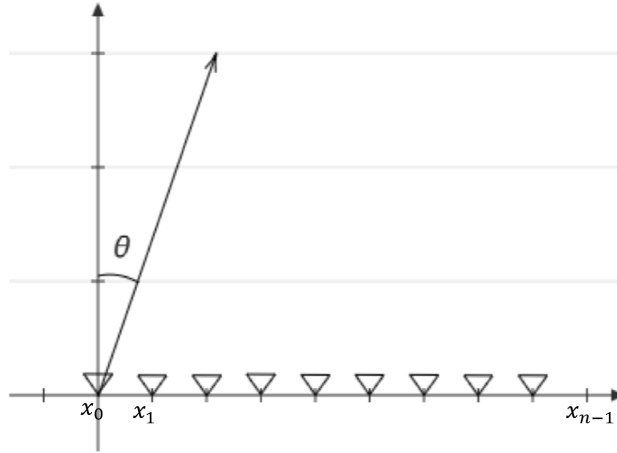


Figure 1.4: Linear phased array diagram

This beam pattern formulation is derived under several key assumptions: the array operates in the far-field region, and the antenna elements are assumed to exhibit

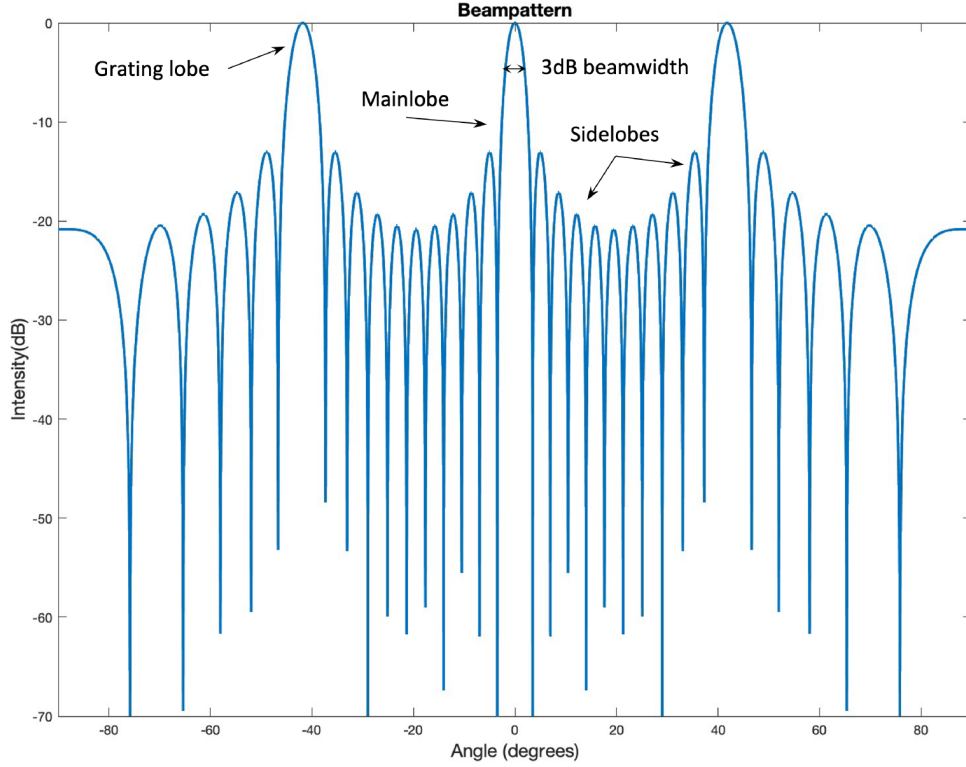


Figure 1.5: Beam pattern of a uniform linear array with $N = 11$ elements and element spacing of 1.5 wavelengths.

isotropic radiation and operate independently without mutual coupling. In (1.1), the direction of the main lobe of the beam can be controlled by adjusting $\phi_n = -kx_n \sin \theta_0$; the main beam aligns with the desired direction θ_0 . This illustrates the principle of a phased control array.

The beam pattern is typically represented as a three-dimensional pattern in all spatial directions. To facilitate analysis and optimization, this 3D pattern is commonly cut into an azimuth plane. Furthermore, due to the inherent symmetry configurations and the fact that real-world applications often focus only on the forward-facing region, it is common practice to retain only the azimuth range $\theta \in [-90^\circ, 90^\circ]$. An example of a uniform linear array beam pattern is shown in Figure 1.5, where the beam pattern is normalized so that its maximum value equals 0 dB.

Based on the above 1D beam pattern formulation and simplification, we will introduce several important concepts critical for the array design:

- **Main lobe:** The main lobe is the primary lobe in the beam pattern, representing the direction of maximum radiated power for transmitting or receiving signals.
- **Beamwidth:** Beamwidth refers to the angular width of the main lobe, typically defined as the angle between the points where the radiation pattern drops to half of its peak value.

- **Sidelobes:** Sidelobes are weaker lobes that occur in directions other than the main lobe, representing unwanted signals. Their strength is typically much lower than that of the main lobe. Sidelobes can cause interference by receiving unwanted signals, making their minimization crucial in autonomous applications.
- **Grating lobes:** Grating lobes are a specific type of sidelobe that occur in a phased array when the spacing between adjacent elements exceeds half the wavelength. Unlike typical sidelobes, the intensity of grating lobes can be as high as the main lobe, making them particularly problematic. In applications, they can lead to inaccurate information about the surroundings, like strong false reflections. Grating lobes arise from the physical structure of the array and can be mitigated by optimizing the spacing between elements.
- **Field of View (FOV):** FOV is expressed as an angular range in this thesis, representing the angular extent of the observable area. This range indicates the angular region within which the sensor can detect objects. In Figure 1.5, the FOV is $[-90^\circ, 90^\circ]$.

In the industry, there are some requirements for beamwidth to meet the demands of autonomous driving. A main lobe beamwidth of approximately 0.1° is necessary to distinguish potential hazards even at a distance of 200 meters [8]. In a uniform linear array, according to the Rayleigh criterion, the beamwidth is determined only by the aperture size D and the wavelength λ [3],

$$\Delta\theta \approx \frac{1.22\lambda}{D}. \quad (1.2)$$

Therefore, to achieve a narrower beamwidth, the aperture size needs to be increased, as the operating wavelength of the OPA is fixed in both design and application. FOV is another important performance metric in the design. Although grating lobes appear when the spacing between adjacent elements exceeds half the wavelength, a practical approach is to utilize only the grating lobe-free angular range and also to steer the beam within this region. The FOV is defined as the grating lobe-free region. Moreover, sidelobes can degrade detection performance by introducing ambiguity and reducing signal-to-noise ratio. Therefore, in OPA design, it is desirable to suppress sidelobes and ensure a high ratio between the main lobe and the sidelobes. To quantitatively evaluate this, the peak sidelobe level (PSLL) is adopted as a key performance metric, and it is computed as:

$$\text{PSLL} = 20 \cdot \log_{10} \left(\frac{\text{main lobe}}{\max(\text{sidelobe})} \right), \quad (1.3)$$

While increasing the number of antennas can be a method to enlarge the aperture size and achieve a narrower main lobe beamwidth, it also leads to increased system power consumption—primarily due to the phase shifters required in an OPA. Therefore, for a given set of design specifications, such as achieving a specific PSLL, a smaller number of antennas is better.

In conclusion, the following four performance metrics will be employed in the design analysis:

- The beamwidth of the main lobe
- PSLL: In this thesis, the higher value of PSLL means a lower peak sidelobe to main lobe ratio.
- FOV: The FOV is defined by the grating lobe free region.
- The number of antennas required in the OPA

In this work, we aim to find solutions that are sparse while maintaining acceptable array performance.

1.4 Motivation for sparse non-uniform design

In most uniform space designs, the space between adjacent elements is set to less than half the wavelength to avoid grating lobes occurring in the beam pattern. However, this naive design method has several major drawbacks.

First, it is a great challenge for energy storage and cooling of the system [3]. In the application of autonomous driving vehicles, the objective is to obtain a high angular resolution of around 0.1° [9] while simultaneously avoiding the grating lobe effect. According to the Rayleigh criterion (1.2), this requires at least 2000 elements when the adjacent space is set to half wavelength. Using 2000 elements leads to a power consumption of around 10 Watts for phase shifting only [10]. Also, when the working wavelength is set to 1550 nm, the physical length of the array is around 1.55 mm. There is a significant amount of heat generated in such a small area, which also presents a challenge for cooling the system [3]. Second, designs with spacing less than half wavelength are also difficult to implement in practice. Third, mutual coupling occurs between elements in an array, which alters the array's beam pattern. As the spacing between elements decreases, the mutual coupling effect increases, which significantly impacts the beam pattern.

In the design of OPAs for autonomous vehicles, several goals must be satisfied: a narrow beamwidth on the order of 0.1° , a wide FOV of approximately 30° to enable reliable detection at distances up to 200 meters, and an acceptable PSLL. Achieving all these objectives concurrently—while maintaining a half-wavelength spacing between elements and minimizing the total number of antennas—is fundamentally infeasible under conventional uniform array configurations due to physical and fabrication constraints in the optical domain. However, this challenge can be effectively mitigated through the adoption of sparse non-uniform array designs [11].

In this thesis, we aim to design a sparse non-uniform array configuration that meets all the requirements of the array pattern while satisfying the distance constraints.

1.5 Outline

This thesis is organized as follows:

- **Chapter 2 – Literature Review:** This chapter investigates the two architectures of sparse OPA design. We also introduce two general problem formulations

corresponding to the two main design objectives and review three widely used methods for the design.

- **Chapter 3 – Problem Formulation and Design of Non-Steering OPA:** This chapter first formulates the objective function that incorporates all objectives and constraints. Then, it presents a modified genetic algorithm (GA) based on a grid architecture, designed to satisfy all the desired objectives. We detail the construction of the modified GA model, including the choice of operators, parameter tuning, and performance validation. We then perform exhaustive testing to demonstrate the two main purposes served by the model. Finally, we show the trade-off between sparsity and PSLL and also optimal solutions for various numbers of antenna elements.
- **Chapter 4 – Design of Steering OPA:** This chapter first explores the impact of beam steering on array performance. We then propose an extended optimization strategy that incorporates the steering process into the model, enabling the use of the modified GA framework developed in Chapter 3. Several design results are presented for different numbers of antenna elements, considering the effect of steering.
- **Chapter 5 – Conclusion and Future Work:** This chapter summarizes the key findings of the thesis and outlines potential directions for future research.

The design of a non-uniform OPA can draw inspiration from the design of sparse phased arrays, since the operating principle of the OPA is similar to that of the phased array radar. The current design of sparse arrays is categorized into three types based on their architecture: thinned arrays, non-uniformly spaced arrays, and clustered arrays [12]. A thinned array architecture is based on a uniform array achieved by deactivating some elements in the corresponding uniformly spaced array. This process is similar to selecting elements in a grid architecture. A non-uniformly spaced array allows the elements to be placed without the constraint of a predefined grid, meaning they can be positioned arbitrarily within the antenna aperture. The clustered array is similar to the thinned array, which is also achieved by deactivating elements within a grid architecture. However, a clustered array operates at the level of subsets or clusters rather than individual elements. Each grid cell in a clustered array contains multiple elements, which form a small array within the cluster. In OPA design, clustered arrays are rarely used. The high accuracy required for beamforming in optical systems necessitates that each element be independently controlled to ensure optimal performance, which is difficult to achieve with the shared resources in a clustered array design. According to the spatial structure of the array, it can also be divided into 1D and 2D designs.

2.1 Optimization Problem Formulations for OPA Design

As discussed previously, in the 1D case, sparse OPA designs can be generally classified into two categories: thinned arrays and non-uniformly spaced arrays. A thinned array is based on a grid architecture, where the optimization focuses on selecting the optimal subset of elements from a predefined uniform grid. In such arrays, without considering amplitude weighting, we can assume $a_n \in \{0, 1\}$, and the element distance can be written as $x_n = nd$, where d is the grid space. A non-uniformly spaced array follows an off-grid architecture; we do not fix x_n and the optimization is performed directly over the element positions $\{x_1, x_2, \dots, x_N\}$ [12].

There are two primary types of optimization objectives in the design of sparse OPA [13]. The first type focuses on improving the array performance, like the PSL, by optimizing sensor parameters such as element positions and amplitude weights. In this approach, the number of antenna elements N is typically fixed. It is noticed that the given value of N is smaller than that required in a uniformly spaced array, making this design a form of a sparse (non-uniform) array. The following presents a general problem formulation for this kind of objective. Here, d_i is the distance between the i and $i + 1$ antennas, and $d_{\min}$ is the minimum distance of the adjacent elements, which is constrained by the physical limitations of the fabrication process. Also, $d_{\max}$ is the largest distance of the adjacent elements, which is constrained to prevent large empty

regions (voids) in the array aperture.

$$\begin{aligned} \max \quad & \text{PSLL} \\ \text{s.t.} \quad & \min(d_1, d_2, \dots, d_{N-1}) \geq d_{\min}, \\ & \max(d_1, d_2, \dots, d_{N-1}) \leq d_{\max}. \end{aligned} \quad (2.1)$$

The second type focuses on minimizing the number of antennas N while ensuring that the designed beam pattern matches the desired beam pattern within an acceptable matching error. In this design objective, to match the process between the designed beam pattern and the desired beam pattern, the FOV is first sampled into L direction angles, and then the designed beam pattern and the desired beam pattern are compared at each of these L sampled direction angles.

$$\begin{aligned} \min \quad & \|\mathbf{a}\|_0 \\ \text{s.t.} \quad & \|P(\theta) - P_d(\theta)\|_2 \leq \epsilon \end{aligned} \quad (2.2)$$

where $P(\theta)$ and $P_d(\theta)$ are the designed beam pattern and the desired beam pattern, $\mathbf{a}$ is an excitation vector containing the amplitude of each antenna. To be more specific, it matches the $L \times 1$ vectors and $P(\theta)$ with minimal elements. An example of the desired beampattern is shown in Figure 2.1, which has an ideal narrow beamwidth and a sidelobe of 0. In both cases, since the array pattern is an exponential function of

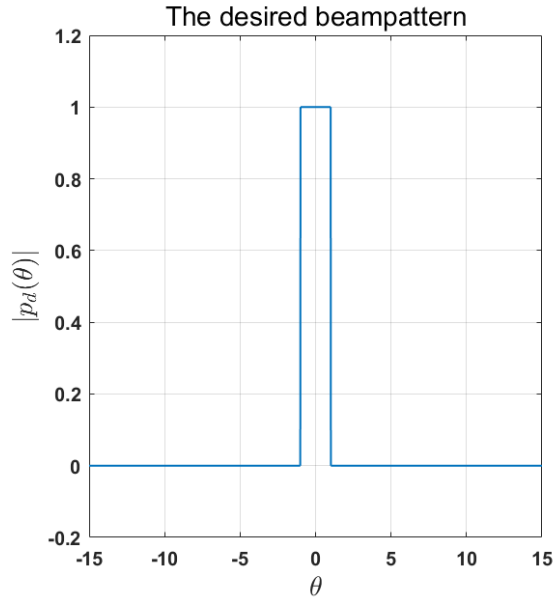


Figure 2.1: An example of the desired beampattern

the element positions, the resulting optimization problem is inherently nonlinear and non-convex.

The primary objective of this thesis is to achieve a high-performance beam pattern with a minimal number of antenna elements. This involves a fundamental trade-off: reducing the number of antennas typically results in degraded beamforming performance.

Balancing these competing objectives poses a significant challenge. However, the previous general formulations (2.1) and (2.2) are not the only way to define the problem. According to the idea presented in [13], after appropriate mathematical transformation, it is possible to jointly optimize the array performance while minimizing the number of elements as (2.3). In (2.3), $\mathbf{a}$ denotes the weight vector of the array elements. The ℓ_0 -norm, defined as the number of non-zero entries in $\mathbf{a}$, serves as a measure of the array sparsity. The parameters α_1 and α_2 are weighting coefficients used to balance the trade-off between the PSL and the sparsity of the array.

$$\begin{aligned}
& \max \quad \alpha_1 \text{PSL} - \alpha_2 \|\mathbf{a}\|_0 \\
& \text{s.t.} \quad \min(d_1, d_2, \dots, d_{N-1}) \geq d_{\min}; \\
& \quad \max(d_1, d_2, \dots, d_{N-1}) \leq d_{\max} \\
& \quad \text{Beamwidth} \leq \epsilon_2.
\end{aligned} \tag{2.3}$$

2.2 Review of different sparse non-uniform OPA methods

Many of the methods are adaptable to both design goals with only minor modifications in implementation. The following will introduce three main approaches to the design.

2.2.1 Evolutionary algorithms

The main algorithms employed in sparse OPA design are evolutionary algorithms, such as the GA and particle swarm optimization (PSO). These algorithms are both inspired by biological evolution: GA mimics natural selection by encoding parameters into individuals (candidate solutions) and exploring the solution space through crossover and mutation. PSO, on the other hand, represents parameters as particles within a solution space, with the particles moving to find the optimal solution. Essentially, evolutionary algorithms use multiple entities (either individuals or particles) to explore the solution space randomly, with the goal of identifying the global optimum. GA was first used in the 1990s to optimize thinned arrays, and since then, other methods like stochastic approximation, PSO, and their modified versions have been introduced [14].

The main challenge of evolutionary algorithms is the potential for premature convergence, leading to suboptimal solutions. In [15], this issue is addressed by employing a two-step optimization process in each iteration: a cursory optimization stage and a precise optimization stage. In the previous chapter, the naive approach to mitigating the grating lobe effect was to restrict the FOV. Another method to reduce the impact of grating lobes is to suppress their level. In [16], a modified GA is used to suppress the grating lobe. In (2.1), the goal is to optimize a single steering angle. However, when the direction angle changes, performance deteriorates. To address this issue, some papers consider not only one direction angle but also the performance at other steering angles. In [17], the authors use a double-weighted fitness function to optimize two steering angles simultaneously. However, determining the optimal weights for each angle can be challenging. To resolve this, [18] employs the non-dominated sorting genetic algorithm II, a classical multi-objective optimization algorithm, to solve the problem of different beam steering angles by using multiple objectives (multi-fitness functions) at once. In

some OPA chip designs, each antenna does not have a uniform amplitude, so optimizing the amplitude is also important. For example, [19] first uses PSO to find the optimal positions of a uniform amplitude array, followed by a convex optimization algorithm to determine the optimal amplitude by optimizing the performance in other steering angles. Further, [20] follows a similar approach, but the second step uses PSO to optimize the amplitude. All the methods discussed above set the minimum spacing to a low value, mostly around three wavelengths. However, to meet the angular resolution requirements with such a small minimum spacing, a large number of antennas may still be needed, or the maximum distance between adjacent antennas will be quite large. To address this, [21] introduces a new algorithm that uses weighted antenna spacing and a Hamming window based on GA. This approach can achieve good performance, for example, a 512-channel aperiodic OPA with antenna spacing between 10λ and 15λ , achieving more than 15 dB side lobe suppression.

2.2.2 Algorithms based on matrix decomposition

One approach to improving performance is to directly match the desired beam pattern. Matrix decomposition algorithms can be applied to change the form of the desired pattern matrix, enabling the designed pattern to more effectively approximate the desired pattern. As described in the previous section, in (2.2), the direction angles are first sampled into L direction angles to make it convenient for the comparison of the designed pattern and the desired pattern. After sampling, these two patterns become $L \times 1$ vectors, $P(\theta)$ and $P_d(\theta)$. These vectors can be used to construct a Hankel matrix, enabling the extraction of key information from the matrix through matrix decomposition. By applying matrix decomposition, the number of elements can be effectively reduced. A technique called matrix pencil method (MPM) is a kind of matrix decomposition method to design a sparse array. The core idea of this method is to perform SVD on the Hankel matrix constructed from the desired pattern. From the results of the SVD, it can be seen that some singular values are relatively small, indicating that their contribution to the radiation pattern is minimal. These small singular values can be discarded to obtain a low-rank approximation of the Hankel matrix. The noniterative MPM effectively estimates synthesis parameters for sums of complex exponentials, offering faster convergence and better results. However, it struggles with asymmetric beam patterns because it only considers the real parts of element positions, leading to unquantifiable negative effects on array performance. This problem can be partially solved by utilizing the forward-back MPM [22], changing the Hankel matrix to a Hankel-Toeplitz matrix. In [23], a unitary matrix pencil method is presented to reduce the computational complexity by converting the decomposition procedures into real ones through a unitary transformation. However, these matrix decomposition methods can not directly control the elements' spacing.

2.2.3 Convex optimization and compressed sensing algorithms

There are numerous powerful tools available for solving convex optimization problems, which can be used to address the second general problem formulation. The formulation (2.2) is not a convex problem. Therefore, the first step is to reformulate the problem

into a convex one. In [24], the authors transform the minimization of the ℓ_0 -norm into the minimization of the PSLL also with the ℓ_1 -norm of the weights associated with each element in the array. Another classical approach [3] is to relax the ℓ_0 -norm into a reweighted ℓ_1 -norm. By employing this minimization, the authors achieved effective pattern performance while maintaining low sparsity.

In [25], the authors introduce the concept of atomic norm minimization (ANM), and the ℓ_0 -norm is replaced with the atomic norm to make the problem convex. Convex optimization tools are then used to solve the ANM problem, resulting in a low-rank Toeplitz matrix that encodes the array’s parameters. Next, Vandermonde decomposition and the least squares method are applied to determine the element layouts and excitations. This ANM method differs from the previous two approaches by operating in a continuous space, while the first two methods select antenna positions from a predefined grid. The continuous space framework provides greater design freedom so that the desired patterns can be reconstructed with fewer elements and lower matching errors.

Other Compressed sensing (CS) techniques can also be applied to solve the second type of general problem formulation. The approach in [26] uses a CS-based method for non-uniform planar arrays that achieve low computational complexity while considering the minimum inter-element spacing. It focuses on designing a 2D array, which extends slightly beyond the current scope. Nevertheless, the mathematical modeling of 2D arrays is a natural extension of the linear array model. Unlike traditional CS strategies that rely on predefined candidate grid positions for element selection, this paper [26] introduces an off-grid compressed sensing approach using an iterative mechanism. This method enables better matching performance and lower computational complexity. This off-grid orthogonal matching pursuit method significantly reduces computational complexity while ensuring satisfactory performance.

2.3 Chapter summary

In this chapter, we discuss two fundamental architectures and two general formulations for OPA design. We then review three major design methodologies widely adopted in the literature. However, most of the methods discussed above focus solely on improving array performance, without exploring the relationship between sparsity and performance. To address this limitation, the problem needs to be reformulated to simultaneously consider both sparsity and final performance, while satisfying the distance constraints during the optimization process. In the next chapter, we reformulate the problem and present our proposed method.

Problem Formulation and Design of Non-steering OPA

3

3.1 Proposed Method and Problem Formulation

In a sparse non-uniform OPA design, direct optimization of element positions is a nonlinear, non-convex problem. Evolutionary algorithms are particularly well-suited for addressing such nonlinear non-convex challenges, making them the primary methods employed in sparse OPA design. Moreover, evolutionary algorithms can be easily adapted to optimize various performance objectives by modifying their objective functions. Some research indicates that evolutionary algorithms are often time-consuming when optimizing a large number of antennas [12]. However, real-time positional adjustment of the array elements is not required in OPA chip design, rendering the array design time less critical. Furthermore, since the transition from theoretical design to physical production typically spans several months, the duration of the design optimization process becomes relatively insignificant in the overall development timeline.

Most existing evolutionary algorithms focus on a single optimization target, such as minimizing PSLL, without incorporating sparsity or beamwidth as explicit optimization objectives. Consequently, these approaches fall short of the present study's goal: to identify and analyze the trade-off between array sparsity and performance (PSLL and beamwidth).

This thesis adopts a modified genetic algorithm that jointly considers sparsity and array performance while satisfying distance constraints. The original constrained optimization problem in (2.3), which enforced the distance as a hard constraint, is reformulated into (3.1) by incorporating a penalty term into the objective function. By assigning a sufficiently large penalty value, infeasible configurations can be effectively avoided. In this way, sparsity, performance, and constraint violations are explicitly balanced within a single optimization problem. In GAs, the objective function is referred to as the *fitness function*, which assigns a fitness value to each candidate solution. Here, $F(\mathbf{c})$ denotes the fitness function, where $\mathbf{c}$ represents an antenna configuration.

$$F(\mathbf{c}) = \text{PSLL} - \alpha \cdot Q - \beta \cdot I, \quad (3.1)$$

where

- Q : the sparsity level of the array.
- I : represents the number of illegal antenna pairs that violate the inter-element spacing constraint.
- α : a parameter that controls the sparsity level by presetting a different value
- β : a penalty factor for illegal antenna pairs.

The optimization problem thus becomes identifying the candidate $\mathbf{c}$ that maximizes the fitness function $F(\mathbf{c})$ across all potential solutions.

3.2 Classical GA Introduction

In recent years, metaheuristic algorithms have gained significant attention for addressing complex real-world problems across various domains, including economics, engineering, politics, and management [2]. GA is one of the metaheuristic algorithms that aims to identify the solutions with the highest fitness value, as evaluated by a predefined fitness function. The core principles of GA are derived from biological processes such as chromosome crossover and mutation. These mechanisms have inspired the development of various encoding schemes that allow feasible solutions to be effectively represented and manipulated. Combined with selection strategies, these genetic operators form the basis of genetic algorithms. Rooted in the principle of survival of the fittest, GA iteratively evolves a population of candidate solutions by evaluating their fitness and applying selection, crossover, and mutation operations. Over successive generations, chromosomes with higher fitness are more likely to propagate, guiding the search toward optimal or near-optimal solutions. GAs are particularly effective in addressing the nonlinear optimization challenges inherent in OPA design.

The procedure of classical GA is as follows: First, randomly generate M solutions, which are called chromosomes in GA, and the whole set of chromosomes is an initial population. After calculating the fitness value of each chromosome, different selection operators can be applied based on the fitness values to select a chromosome pair from the population. A crossover and mutation operators are then applied to each selected pair with certain probabilities (crossover rate and mutation rate), producing new offspring. In the final step, a new population is formed, replacing the old one. This iterative process continues until reaching the maximum iteration number. Algorithm 1 shows the pseudo code of the classical genetic algorithm.

Algorithm 1 Classical Genetic Algorithm

```

Generate initial population of  $M$  chromosomes  $\mathbf{c}_i$  ( $i = 1, 2, \dots, N$ )
Set iteration counter  $t = 0$ 
Compute the fitness value of each chromosome through fitness function
while  $t < \text{maxiter}$  do
    Select a pair of chromosomes from the population based on fitness
    Apply crossover operation on selected pair with crossover rate
    Apply mutation on the offspring with mutation rate
    Replace old population with newly generated population
    Increment the current iteration  $t$  by 1
end while
return The chromosome with the highest fitness value:  $\mathbf{c}_{best}$ 

```

3.3 Modified GA

In our modified GA model, we discuss two key components: the first is the design of the fitness function, and the second focuses on the design of genetic operators, including encoding, crossover, mutation, and selection mechanisms.

3.3.1 Fitness function

In most of the GA model I have surveyed, the number of array elements is fixed, without explicitly considering the trade-off between sparsity level and array performance [15–18] [21]. To address this, the grid-based architecture is adopted in this design. As a result, each chromosome will be encoded as a binary vector $\mathbf{c} \in \{0, 1\}^N$. In this vector, a ‘1’ represents an active position (with an antenna), and a ‘0’ represents an inactive position (without an antenna). Figure 3.1 gives an example of a chromosome under grid architecture design. Based on this grid architecture, we can also take beamwidth

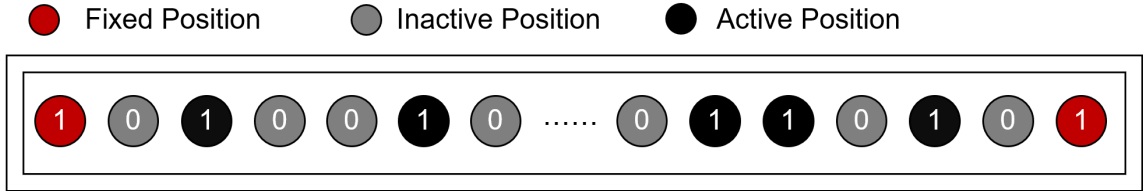


Figure 3.1: An example of a chromosome under grid architecture design

into account. According to Rayleigh’s criterion (1.2), beamwidth is determined only by the wavelength and the aperture size. If we maintain a sufficiently large and dense aperture, we can achieve a narrow beamwidth. Therefore, during the searching process, we ensure that the first and last positions of $\mathbf{c}$ in the binary vector are set to 1. We also interpret this binary set as the weight assigned to each antenna element, thereby avoiding the need for additional thresholding methods to achieve sparsity. The fitness function (3.1) can now be changed into (3.2).

$$\begin{aligned} \max_{\mathbf{c}} \quad & F(\mathbf{c}) = \text{PSLL} - \alpha \cdot Q - \beta \cdot I, \quad (\mathbf{c} \in \{0, 1\}^N) \\ \text{s.t.} \quad & \mathbf{c}(1) = 1, \quad \mathbf{c}(N) = 1, \end{aligned} \quad (3.2)$$

In the grid-based architecture, the Q can be defined by the ratio as:

$$Q = \frac{\|\mathbf{c}\|_0}{N}. \quad (3.3)$$

Also, with fixed potential positions for element placement, we can ignore the constraint on the maximum distance between adjacent elements, because the grid itself limits the spacing, so only the minimum distance constraint needs to be enforced.

3.3.2 Operators Selection

GA employs various operators during the search process, including encoding, selection, crossover, and mutation. The choice and combination of these operators can

significantly affect the convergence speed and will lead to worse results under limited iterations. Figure 3.2 illustrates different operator choices in GAs.

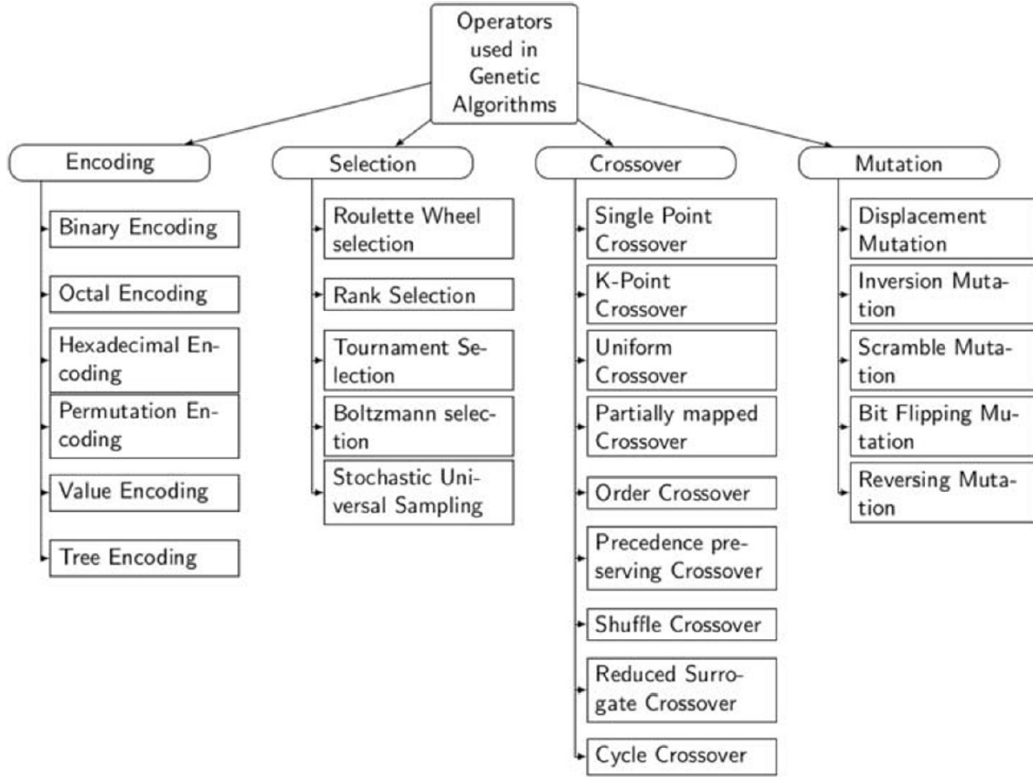


Figure 3.2: Variety of operators used in GA [2]

3.3.3 Encoding operator

Grid-based architecture is used to build the OPA, making binary encoding a straightforward choice for chromosome representation. In this case, the binary output directly represents the solution, eliminating the need for further conversion. Additionally, binary encoding facilitates easier and faster implementation of crossover and mutation operators.

3.3.3.1 Selection operator

The selection operator plays a crucial role in guiding the search for optimal solutions by picking out appropriate chromosomes to take the next step. The selection pressure also matters. Selection pressure refers to how strongly the selection process favors the best chromosomes in a population when choosing parents for reproduction. On one hand, we aim to impose some selection pressure to ensure that chromosomes with a higher fitness value survive. However, excessive pressure can lead to premature convergence by consistently selecting only the fittest chromosomes in a given generation, thereby reducing population diversity. Thus, maintaining a suitable selection pressure is also

essential. We will compare two common selection operators: Roulette wheel selection and Tournament selection.

Roulette wheel selection, also known as fitness-proportionate selection, selects chromosomes based on their fitness values; the higher the fitness value, the greater the probability of selection. The fitness values are converted into selection probabilities using the following equation:

$$p_i = \frac{F(\mathbf{c}_i)}{F} \quad (3.4)$$

where p_i represents the probability of the chromosome $\mathbf{c}_i$, $F(\mathbf{c}_i)$ denotes its fitness value, and F is the sum of all fitness values in one population. This mechanism resembles a roulette wheel in a casino, where each chromosome is assigned a proportion of the wheel based on its probability. A random selection is performed by simulating spins of the wheel. Each spin selects one chromosome, and the process continues until the number of selected chromosomes matches the initial population size. Since selection is based on probability, even weak chromosomes have a chance of being chosen, which helps preserve genetic diversity by retaining a potentially useful schema.

However, a principal limitation of roulette wheel selection is its dependence on the variance of fitness values [2]. When most individuals in a large population exhibit similar fitness and the fitness variance is low, the selection pressure is significantly reduced. In such cases, small differences in fitness lead to nearly uniform selection probabilities, thereby diminishing the algorithm's ability to guide the search toward superior solutions. As a result, the selection process may resemble a random search. In our implementation, each iteration uses at least 1000 chromosomes, with fitness values typically ranging from 15 to 35. Under these conditions, the drawback becomes very obvious. Instead, we adopt tournament selection as an alternative operator.

Tournament selection operates by randomly selecting a given set size of chromosomes to compete. The chromosome with the highest fitness value within the set will be selected. This process continues until the required number of chromosomes is chosen. The selection pressure in tournament selection is the set size, which is called the tournament size. As the tournament size increases, competition intensifies, making weaker chromosomes less likely to be selected. A simplified selection process with a tournament size of two is shown in Figure 3.3. In Figure 3.3, among the ten potential chromosomes, two are first selected randomly, the red and green. Then their fitness values are compared. The chromosome with the higher fitness value, in this case the green one, is then chosen. In this project, we use a tournament size of 2 to mitigate premature convergence and preserve solution diversity.

Due to the stochastic crossover and mutation, the optimal solution in the current generation may be disrupted or lost in the next generation. To mitigate this, we apply the elite selection method before tournament selection. The elite selection method preserves the best chromosomes by directly passing them to the next generation without undergoing further genetic operations, thereby preventing potential degradation. In this project, we save one chromosome pair in each iteration.

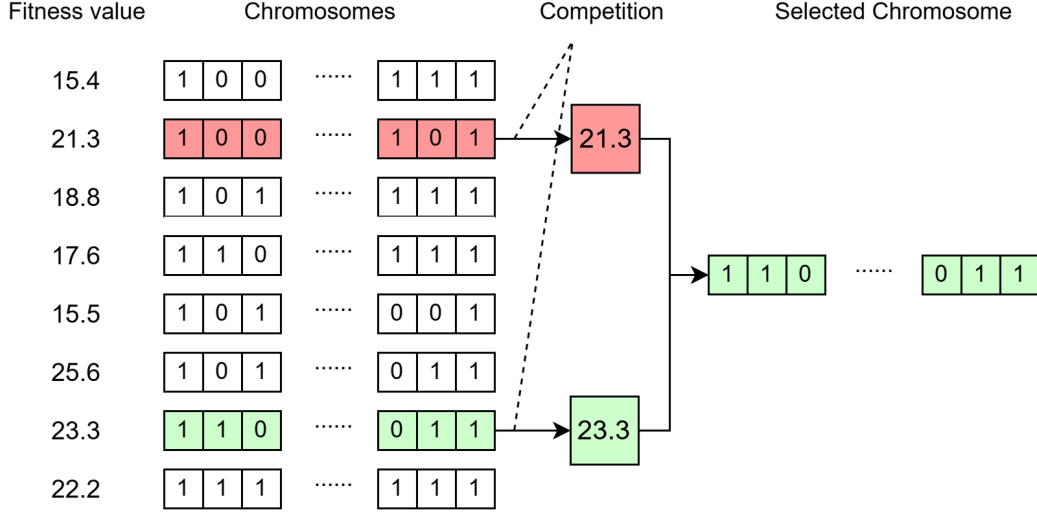


Figure 3.3: Simplified illustration of the tournament selection process with a tournament size of two

3.3.3.2 Crossover and mutation operator

The crossover process simulates genetic recombination in biological inheritance, allowing offspring to inherit advantageous genes from their parents, thereby increasing population diversity and accelerating convergence. The purpose of crossover is to combine information from two parent chromosomes to generate new chromosomes, enabling the algorithm to explore the search space more effectively. In this project, we talk about two different crossover operators: the one-point crossover and the two-point crossover.

In one-point crossover, a random crossover point $c(i)$ is selected, and the genetic information of the two parent chromosomes beyond this point is exchanged. For example, the segments $c_a(i + 1)$ to $c_a(N)$ and $c_b(i + 1)$ to $c_b(N)$ are swapped:

$$c_a(i + 1:N) \longleftrightarrow c_b(i + 1:N).$$

In two-point crossover, two crossover points, $c(i)$ and $c(j)$ with $i < j$, are randomly selected. The genetic information between these two points is exchanged between the two parent chromosomes. For example, the segments from $c_a(i)$ to $c_a(j)$ and from $c_b(i)$ to $c_b(j)$ are swapped:

$$c_a(i:j) \longleftrightarrow c_b(i:j).$$

In the case of binary encoding, one-point crossover tends to have a faster convergence speed because information exchange begins at a single point. However, it may also disrupt important schema. In general, the differences between one-point, two-point crossover in binary encoding are not significant [2]. In order to have a faster convergence speed, we will use one-point crossover in this thesis.

The mutation operator allows chromosomes to generate new genetic combinations during inheritance, thereby increasing population diversity and preventing the algorithm from getting trapped in local optima. In the GAs, crossover primarily facilitates local search, whereas mutation enhances global exploration, improving the algorithm’s search capability. The simplest way to achieve mutation is through simple inverse mutation. In this mutation operation, two indices i and j ($i < j$) are randomly selected. The segment $c(i:j)$ is first reversed to form $c(j:i)$, and then this reversed segment is inserted at a randomly selected position within the chromosome. Mutation also occurs with a probability, the mutation rate. In nature, the mutation rate is generally much lower than the crossover rate because the “damage” it causes to the chromosome can be significant. Table 3.1 lists all the operators used in this model.

Encoding	Binary Encoding
Selection	Elite selection + Tournament selection
Crossover	One-point crossover
Mutation	Simple inverse mutation

Table 3.1: The chosen operators

3.4 Tuning Crossover and Mutation Rates

For ease of comparison, we adopted the preset conditions established by Kunlei [3]. As described in his work, the FOV was set from -18° to 18° . The minimum distance constraint between each antenna pair was set to one wavelength. This model is applied to a 1000 grid architecture, where the grid distance is one wavelength, thereby negating the need to consider minimum distance constraints. The population size is set to 1000×1000 , and the algorithm runs for 2000 iterations. The entire algorithm and simulation framework are implemented in MATLAB.

Under the established presets, a rough grid search will be employed to find out the optimal values for these two parameters.

3.4.1 Impact of Crossover rate and mutation rate

In the GA, a low crossover and mutation rate helps preserve high-quality chromosomes already found, preventing their disruption. However, excessively low rates may cause the algorithm to become trapped in local optima due to insufficient exploratory capabilities, leading to premature convergence. Conversely, higher crossover and mutation rates increase population diversity, which helps the algorithm escape local optima and enhances its global search capability, but high rates can introduce too much randomness into the population, disrupting beneficial patterns within the chromosomes. This may hinder convergence and make it difficult to find optimal chromosomes. As a result, selecting appropriate values for both rates is important.

This preliminary grid search is conducted to investigate the impact of mutation rate and crossover rate on the PSLI in the model, with the goal of identifying the optimal combination of these parameters. The parameter ranges are defined as follows:

- **Mutation rate:** {0.01, 0.05, 0.1, 0.3, 0.6}
- **Crossover rate:** {0.6, 0.7, 0.8, 0.9, 0.95, 0.97}

This resulted in a total of 30 unique parameter combinations. For each combination, the model was independently run four times. The final PSL values were averaged to evaluate performance. Figure 3.4 shows the average results of different combinations. More detailed results are shown in Table 3.2.

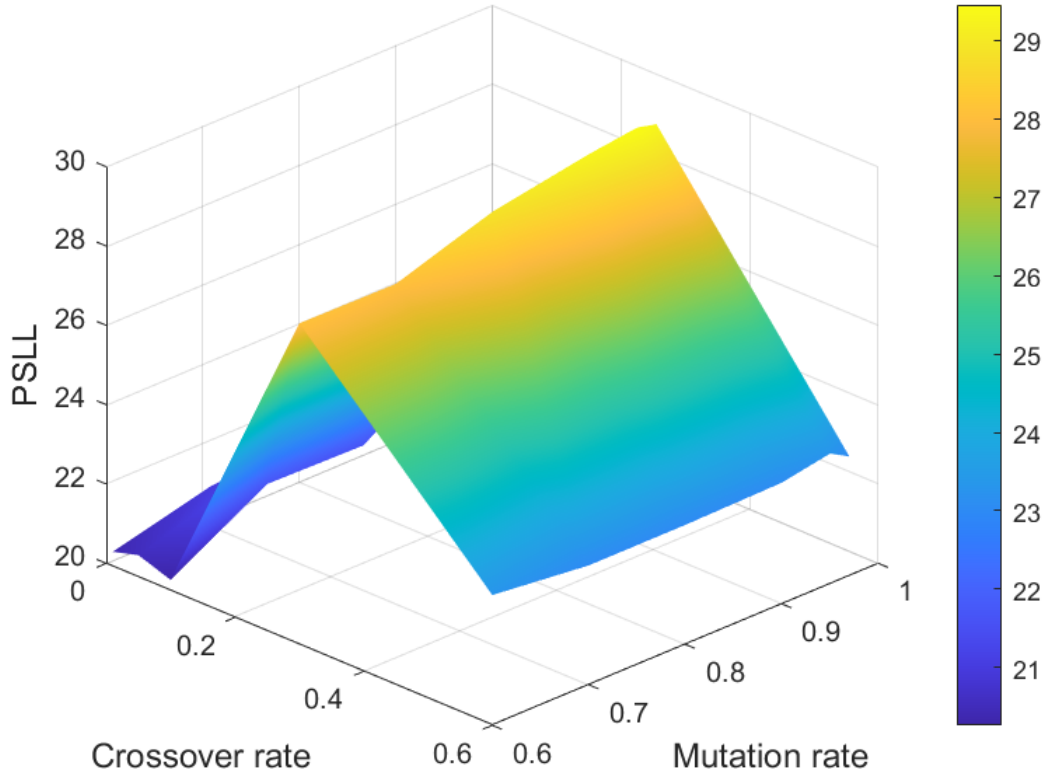


Figure 3.4: Grid search results of *PSL* under different crossover rate and mutation rate combination

Mutation/Crossover	0.6	0.7	0.8	0.9	0.95	0.97
0.01	20.360	20.970	21.240	21.968	22.218	22.145
0.05	20.547	21.307	21.720	22.036	22.146	22.106
0.1	20.263	21.658	21.617	23.662	23.802	23.749
0.3	28.094	28.069	28.864	29.296	29.456	29.339
0.6	23.283	23.011	23.012	23.056	23.280	23.020

Table 3.2: Detailed grid results of average PSL under different crossover rate and mutation rate combinations

The results reveal a clear trend along both parameter axes. As the mutation rate increases, the average PSLI initially increases, reaches a peak, and then decreases, with the highest performance observed at a mutation rate of 0.3. A similar trend is observed with the crossover rate, where the best performance is achieved at a value of 0.95. Based on these findings, the optimal values for these two rates are

- **Mutation rate:** 0.3
- **Crossover rate:** 0.95

3.5 Model Verification

Before using the model for actual optimization, we conducted an exhaustive search to validate its correctness. This verification serves two main purposes:

1. To determine whether the modified GA can successfully find the true best solution.
2. To test whether the sparsity level controller (α) can accurately find the optimal solution while effectively controlling the sparsity.

3.5.1 Construction of test solution space

Under the previous configuration, where 998 potential positions were available and no minimum distance constraint was applied, the total number of possible chromosomes is 2^{997} (note: the symmetric configuration gets the same results). The search space is too large to test all the chromosomes. Therefore, we performed this verification in a smaller solution space: the chromosome length was set to 30, with a minimum distance constraint of 2, and an additional constraint limiting the maximum distance to 5. Under these settings, the number of candidate chromosomes was reduced to 82114. Then calculate the PSLI of these 82114 chromosomes and identify the best results for different numbers of antennas, as shown in the Table 3.3.

3.5.2 Validation I: Finding optimal PSLI without considering sparsity ($\alpha'=0$)

In this experiment, the objective is to verify the effectiveness of the first goal of the modified GA. The algorithm is tested with a population size of 50, chromosome size of 30, and a maximum of 500 iterations. The grid distance is set to 0.4 wavelength.

Figure 3.5a illustrates the optimization progress over iterations, where the X represents the iteration number and the Y indicates the best fitness value found in each iteration. The results show that the algorithm converges after approximately 203 iterations, beyond which no further improvement is observed. The final solution employs 14 antennas and achieves a *PSLI* of 16.1266 dB. These results demonstrate that the modified GA is capable of effectively finding the best solution. Figure 3.5b shows the array pattern of the best solution.

Solution index	Number of antenna	PSLL
80977	7	0.5715
82112	7	0.5715
53713	8	6.2087
81972	8	6.2087
52797	9	8.0825
81996	9	8.0825
13392	10	10.1395
63855	10	10.1395
1681	11	11.3142
81138	11	11.3142
38513	12	12.3684
72503	12	12.3684
13081	13	13.5687
57644	13	13.5687
3	14	16.1266
72453	14	16.1266
1	15	13.8787
34889	15	13.8787

Table 3.3: Maximum $PSLL$ for chromosomes with the same number of antennas

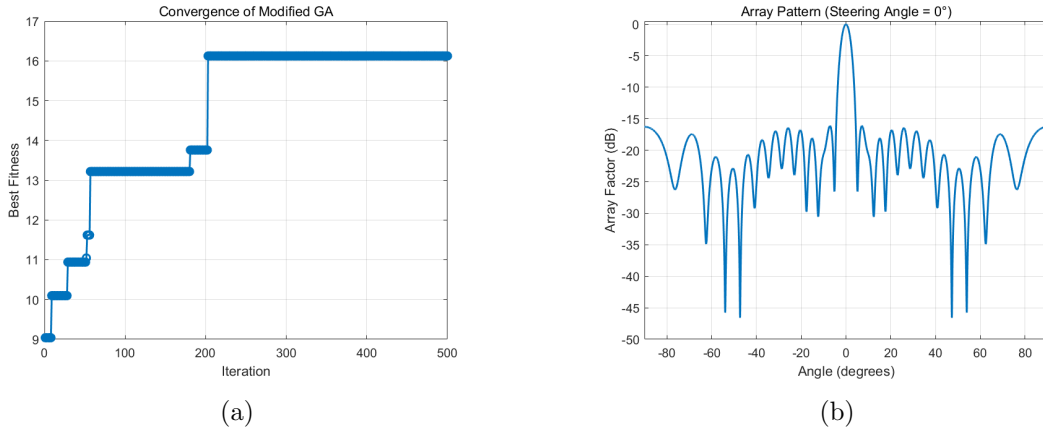


Figure 3.5: (a) is the best result in each iteration ($\alpha' = 0$), and (b) is the array pattern of the best solution using 14 antennas

3.5.3 Validation II: Evaluation of sparsity controller

To further evaluate the effectiveness and reliability of the sparsity control mechanism, a second experiment is conducted. In this experiment, the parameter α is varied from 0 to 100 with a fixed step of 0.5. For each value of α , we compute the optimal fitness value using equation (3.1), which incorporates sparsity, distance constraints, and the $PSLL$.

By analyzing the fitness value of all chromosomes, several distinct transition

points—where the trade-off behavior changes—can be identified. Comparing the generated results with the reference data in Table 3.4 enables us to verify whether the α -based sparsity controller can accurately identify the optimal solution while effectively controlling the sparsity of the array.

Number of Antenna	PSLL	α	Fitness Values
14	16.1266	0	16.1266
14	16.1266	44.5	-4.6401
10	10.1395	45	-4.8605
10	10.1395	58.5	-9.3605
8	6.2087	59	-9.5247
8	6.2087	100	-20.4581

Table 3.4: The jump point during the exhaustive test of α

We select three jump points $\alpha = 44.5$, $\alpha = 45$ and $\alpha = 59$ for testing, the other preset values are the same as those above discussion. The results are shown in Figures 3.6, 3.7 and 3.8. These results clearly demonstrate that the parameter α can effectively control the sparsity level and enable the identification of optimal solutions under different numbers of antennas. It is worth noting that, in all conducted tests, no illegal antenna positions were observed, and all chromosomes strictly satisfied the distance constraint.

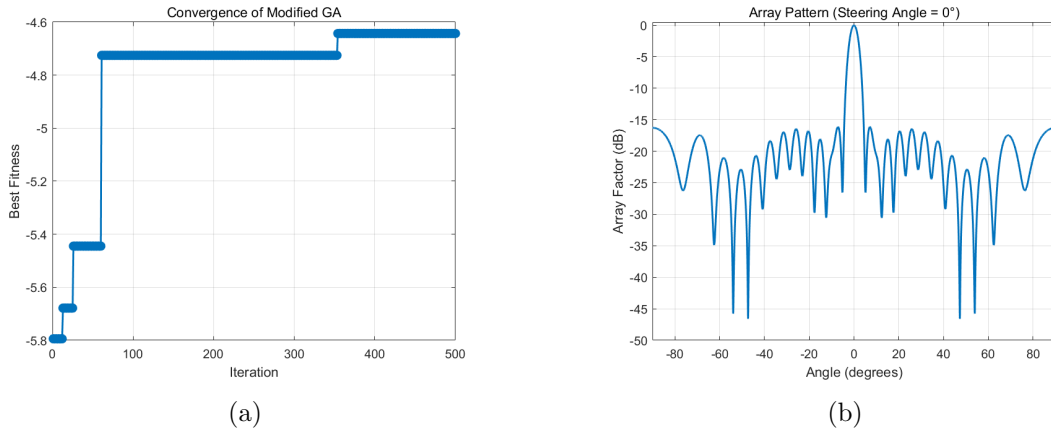
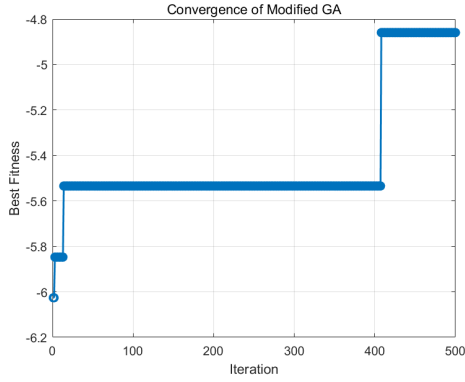


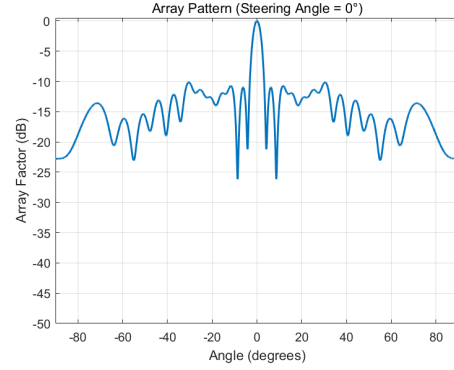
Figure 3.6: (a) is the best result in each iteration ($\alpha = 44.5$), and (b) is the array pattern of the best solution using 14 antennas

3.6 Results and Discussion

In the previous section, we discussed the details of the proposed model and evaluated its effectiveness using an exhaustive search method. In this section, we apply the model to generate solutions that meet the specific requirements of automotive vehicle applications.

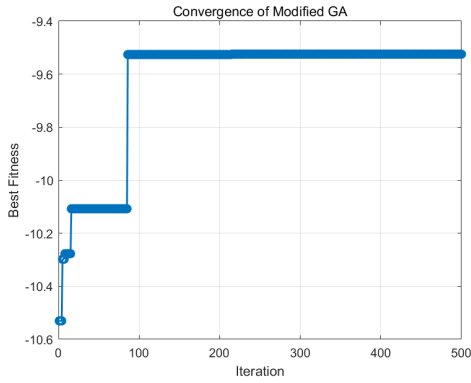


(a)

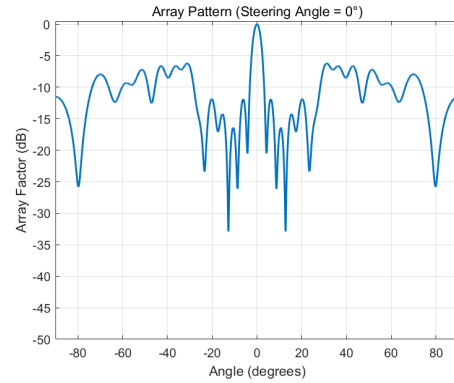


(b)

Figure 3.7: (a) is the best result in each iteration ($\alpha = 45$), and (b) is the array pattern of the best solution using 10 antennas



(a)



(b)

Figure 3.8: (a) is the best result in each iteration ($\alpha = 59$), and (b) is the array pattern of the best solution using 8 antennas

3.6.1 Simulation setups

To simplify future comparisons, the following parameter settings follow the setups in [3]:

- The operating wavelength λ is set to 1550nm;
- The detect fov is $[-18^\circ, 18^\circ]$;
- The number of sampling angle L is set to 10001;
- Beam steering is still not considered in this case; the steering angle is fixed at 0° ;
- The grid space is set to 1λ ;
- Chromosome size = 1000, population size = 1000; the maximum number of iterations = 2000;

- The crossover rate = 0.95, mutation rate = 0.3;
- Tournamentsize = 2, with 2 elite chromosomes retained in each generation;

3.6.2 Model Speed-Up

As previously discussed, one major drawback of using GAs is their substantial computational cost. Although a fast response to different settings is not required—since only the final optimal solution is of interest—the process still consumes a significant amount of time during testing and tuning. In this work, we propose two methods to accelerate the computation.

By employing MATLAB’s profiling tools, we observe that approximately 95% of the total runtime is consumed by the function responsible for calculating the PSL of each solution. The direct computation of PSL can be performed using matrix multiplications. However, due to the grid-based structure of our model, the element positions x_n in (1.1) can be simplified to $x_n = nd$. Thus, the beam pattern expression can be reformulated as:

$$P(u) = \sum_{n=1}^N a_n e^{jkn du}, \quad (3.5)$$

where $u = \sin(\theta) - \sin(\phi)$.

Equation (3.5) represents a finite Fourier series that relates the excitation coefficients a_n of the linear array to its beam pattern through a discrete Fourier transform. As a result, we can utilize MATLAB’s built-in `fft` function to efficiently compute the beam pattern, significantly reducing the overall computation time.

Furthermore, since the PSL of all candidate solutions must be evaluated in each generation, we implement parallel computing using the `parfor` loop in MATLAB. It is important to note that `parfor` cannot be nested within another loop, which prevents its use in the outer GA iteration loop.

3.6.3 Results

We first give a zero value to α to see the results without considering the sparsity constraints. Figure 3.9 shows the array pattern of the best results found by the modified GA. From the array pattern, it can be observed that no grating lobes are present, and the beamwidth satisfies the requirements for autonomous applications.

In the experiment, α can also take a wide range of values, and two representative 15 and 27 are selected for testing. Figures 3.10 and 3.11 show the best solutions obtained using the modified GA when $\alpha = 15$ and $\alpha = 24$, respectively. Table 3.5 gives the details data of the array pattern under different values of α .

Figure 3.12 illustrates the relationship between α , the number of active antennas, and the resulting PSL. As shown in the figure, α effectively controls the sparsity level of the solution. When the value of α increases, the weight of the sparsity term in the fitness function (3.1) becomes more significant, leading to a reduction in the number of antennas used in the final solution. Correspondingly, as the number of active antennas decreases, the PSL also tends to decrease. Notably, a rapid decline in both the number

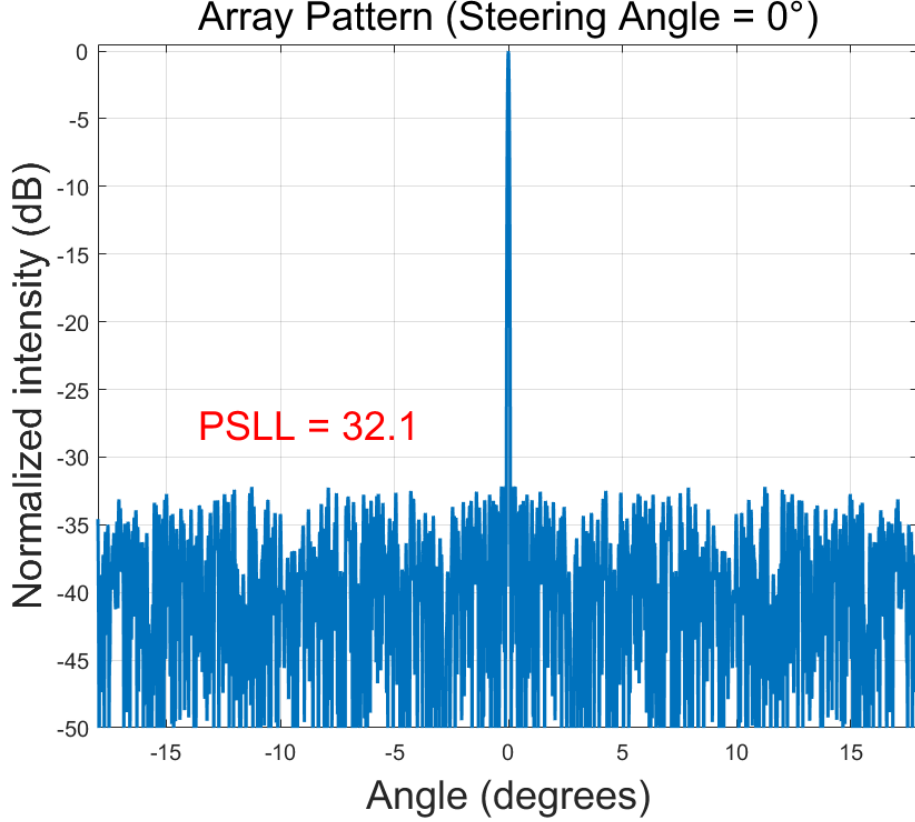


Figure 3.9: Array pattern of the optimal 641-element configuration obtained when $\alpha = 0$

α	Number of antenna	PSLL	Beamwidth
0	641	32.1	0.05
15	425	28.9	0.05
24	139	23.4	0.07

Table 3.5: Array pattern characteristics of the best solution under different values of α

of antennas and PSLL is observed when α increases from 5 to 25, indicating a sensitive trade-off region.

3.6.4 Experiment on distance control

In the work presented in [3], the author cleverly sets the grid size equals to the minimum distance constraint, allowing the optimization process to bypass this constraint. However, in our model, the spacing between elements is explicitly controllable, enabling us to relax the grid size to values even smaller than the minimum spacing if desired.

In the following experiments, we refine the spatial resolution by reducing the grid spacing and use the parameter β to enforce a minimum spacing between adjacent elements. Specifically, we consider grid spacings of half-wavelength and quarter-wavelength. To maintain the same aperture size under different grid spaces, the chro-

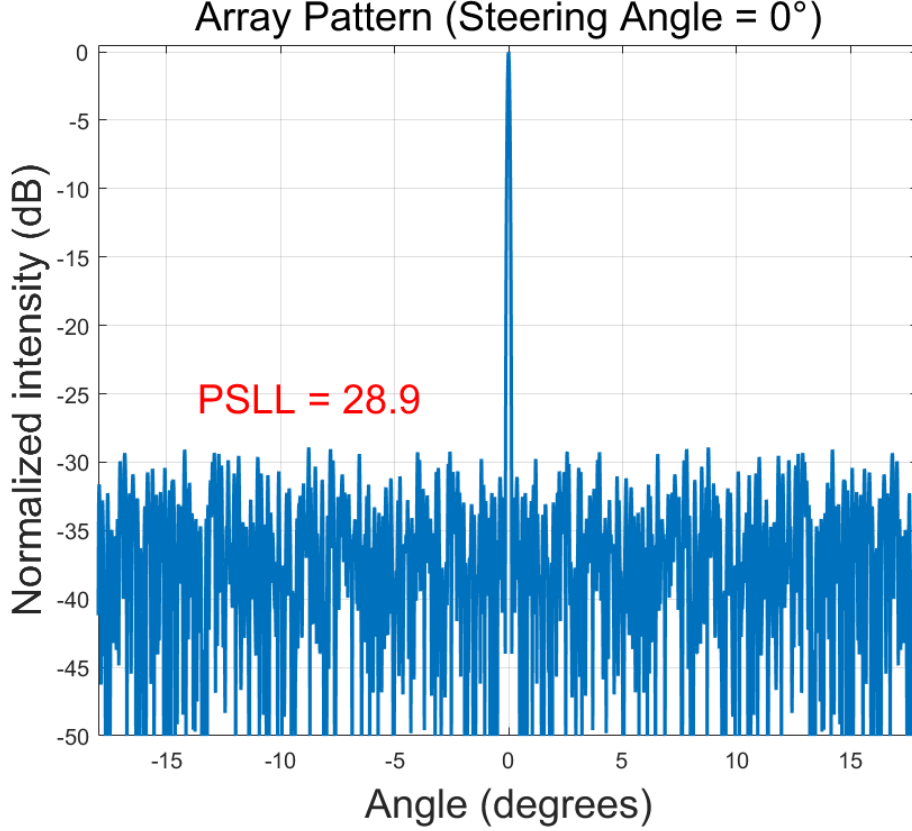


Figure 3.10: Array pattern of the optimal 425-element configuration obtained when $\alpha = 15$

mosome size is increased to 2000 and 4000, respectively. In these settings, the minimum allowable spacing between two active elements corresponds to 2 and 4 grid units, respectively.

The parameter β is introduced to penalize “illegal” configurations where the spacing constraint is violated. However, during the early stages of the optimization process, we allow a greater diversity of solutions, including those with promising beam patterns that may temporarily violate spacing constraints. Therefore, β is initially set to a small value. As the search progresses, β is gradually increased to enforce stricter adherence to the spacing constraints, guiding the algorithm toward physically realizable solutions. As a result, we adopt an iterative update strategy for the parameter β , using an exponential function to gradually increase its value over the course of the optimization.

$$\beta_{iter} = 10 \cdot \frac{e^{\frac{5 \cdot iter}{\max iter}} - 1}{e^5 - 1} \quad (3.6)$$

The Figure 3.13 presents the results for different grid spacings evaluated under various values of α' . To ensure the same sparsity control comparison across different chromosome lengths, we normalize the regularization coefficient as $\alpha' = \frac{\alpha}{N}$, where N denotes the chromosome size. The fitness function in this case will be changed into:

$$F(\mathbf{c}) = \text{PSLL} - \alpha' \cdot \|\mathbf{c}\|_0 - \beta_{\text{adap}} \cdot I, \quad (3.7)$$

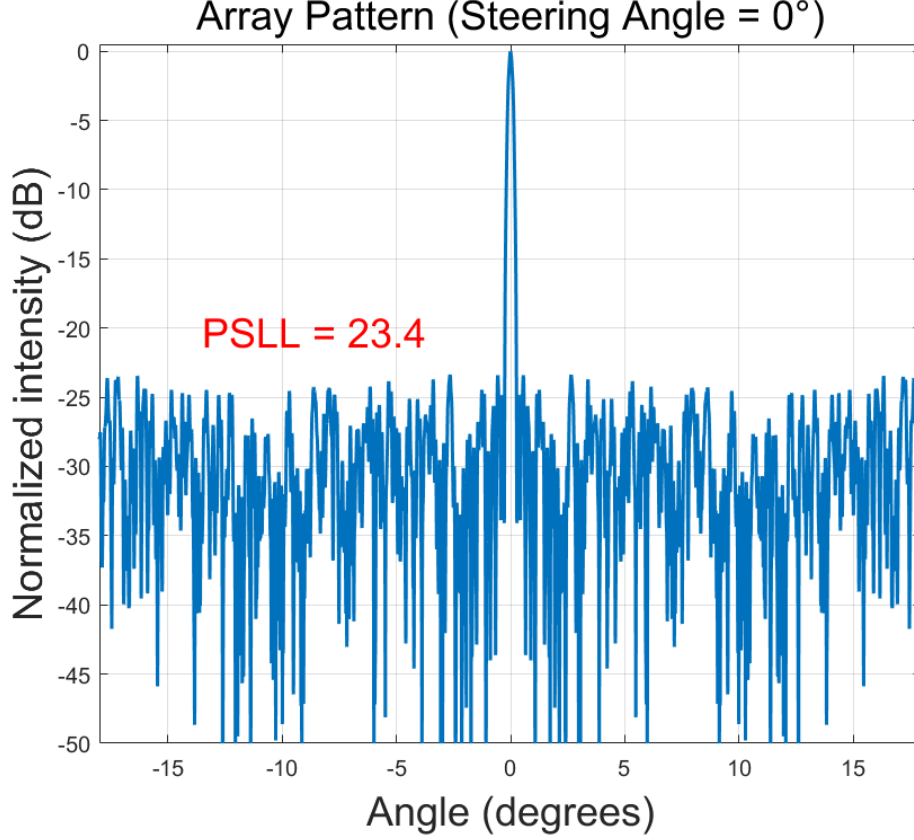


Figure 3.11: Array pattern of the optimal 139-element configuration obtained when $\alpha = 24$

where β_{adap} is the adaptive penalty for illegal position situations.

3.6.5 Comparison and discussion

In this project, the simulation setup follows the methodology presented in [3]. In that work, the author formulates the general problem using the general problem formulation (2.2) and applies a reweighted ℓ_1 -norm to relax it into a convex optimization problem. Additionally, an extra component, the amplitude controller, is introduced to control the amplitude of each antenna element.

To ensure a fair comparison, the main lobe magnitude is normalized to 1. The resulting array patterns' characteristics are shown in Table 3.6. Comparing Tables 3.6 and 3.5, it can be observed that the optimal solutions in both cases utilize approximately 660 antenna elements. However, the PSLL achieved by the proposed modified GA is 8.7 dB higher than that reported in [3]. Furthermore, to achieve a PSLL of 23.4 dB, the solution generated by the modified GA requires only 139 antenna elements—significantly fewer than the 676 antennas used in [3].

There are several possible reasons to explain this phenomenon. First, the reweighted ℓ_1 -norm relaxation method does not directly solve the original problem; it relaxes the nonconvex problem into a convex optimization problem. Also, it is primarily intended

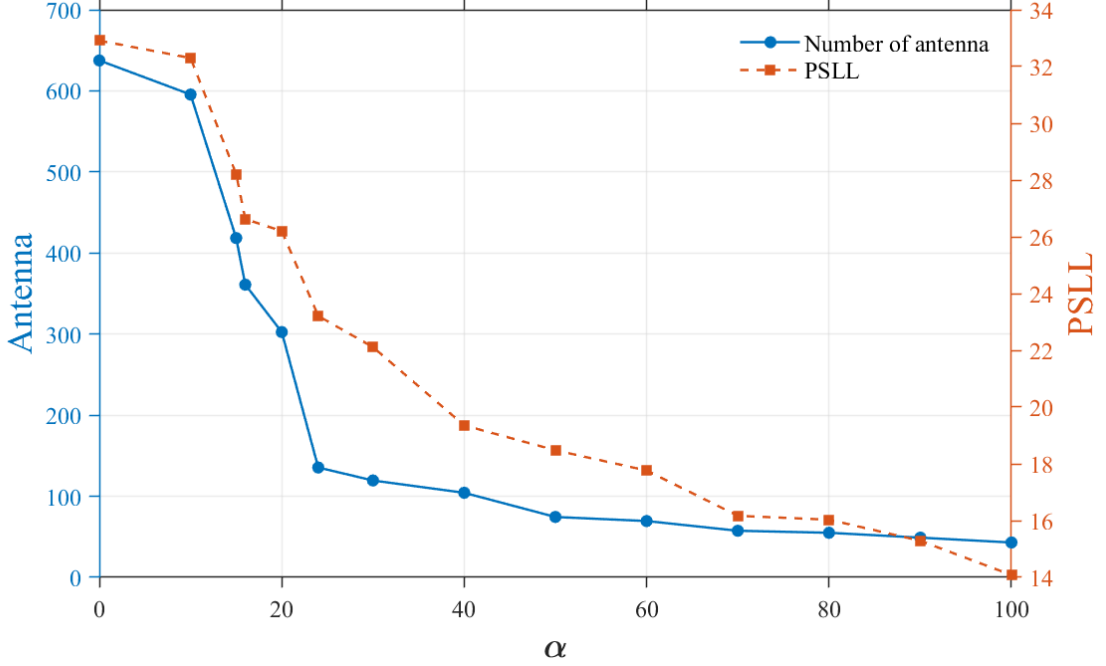


Figure 3.12: The relationship between α and number of antenna and PSLL

to enhance the sparsity of an initial solution. If the initial iteration yields a poor result, the algorithm may become trapped in an undesirable local minimum [27]. Second, the reweighted approach is more suited for generating continuous-valued solutions. However, in this work, we require a discrete solution. Therefore, even if the amplitude of certain antenna elements is very close to zero, a thresholding process must still be applied to force them to zero. This thresholding step significantly influences the final result and may degrade overall performance.

Number of antenna	PSLL	Beamwidth
676	23.4	0.05
418	21.4	0.05
190	13.8	0.09
34	5.4	0.39

Table 3.6: The results of using reweighted l_1 -norm method in paper [3]

After modifying the grid spacing, we evaluated the performance of the penalty parameter β . First, none of the resulting configurations violated the minimum distance constraint, demonstrating that the β is effective in enforcing spacing requirements between elements. Moreover, it can be observed that across different grid spacings, the parameter α and α' continues to effectively regulate the sparsity of the solution. As the number of potential element positions increases—thereby enhancing the design’s degrees of freedom—the overall array performance improves accordingly.

However, although computation time is not an evaluation metric in this thesis, it

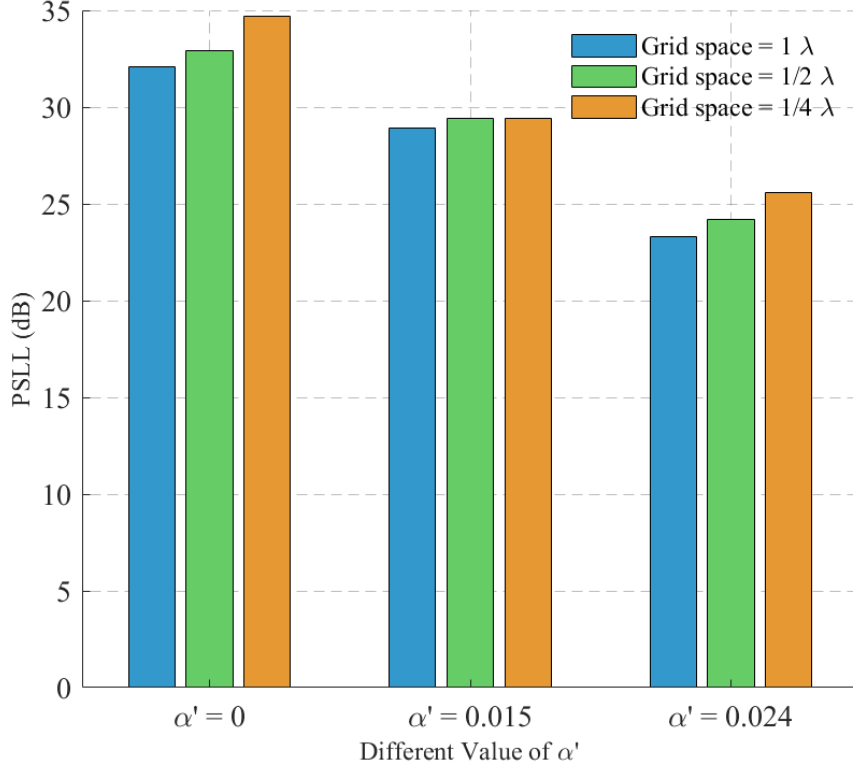


Figure 3.13: Different grid spaces searching results

is worth noting that the test case with a chromosome size of 4000 required approximately 2.5 hours to complete. This duration is prohibitively long and underscores the substantial computational burden associated with such a high-resolution optimization.

3.7 Conclusion

This chapter focuses on developing the modified GA and evaluating the effectiveness of the proposed model. Based on the results from the exhaustive search, we conclude that when the solution space is relatively small, the optimal solution can be accurately found. Subsequently, the modified GA is employed to generate optimal solutions under predefined scenarios, satisfying the requirements of autonomous applications. Furthermore, the model demonstrates the ability to control the sparsity level of the resulting array by adjusting the parameter α . We further investigate the effectiveness of the penalty parameter β , demonstrating that it successfully enforces the minimum spacing constraint. Increasing the number of potential antenna positions enhances design flexibility and leads to improved performance.

All of the above discussions are based on a steering angle of zero; the beam steering process has not yet been considered. Beam steering is a key capability of OPA. The next chapter will discuss the steering mechanism in detail.

Design of steering OPA

The previous chapter focused on array design without considering the beam steering process. In this chapter, we first investigate the impact of the beam steering process on the performance of several optimized OPA designs. Next, we propose a new optimization problem that incorporates the steering process and present a corresponding solution method. Finally, we compare the performance of several solutions across the entire steering range.

4.1 Impact on array performance during steering process

In the previous discussion, all optimization was performed under a single steering angle. Such an optimal solution may not retain its performance across the entire steering range, potentially resulting in degraded array performance at other angles. To investigate the impact of the steering process on array performance, we evaluate two previously obtained optimal configurations, using 641 and 425 antenna elements, respectively. We then get their performance under different steering angles. Figure 4.1 shows the PSL values across the whole steering angle range using 641 and 425 antenna elements. It can be observed that as the steering angle increases away from 0° , the PSL consistently degrades. This degradation may be attributed to stronger sidelobes located outside the FOV, which are shifted into the FOV as the main beam is steered, thereby increasing the measured PSL. It indicates that the steering process has an impact on array performance and therefore must be taken into account during the optimization process.

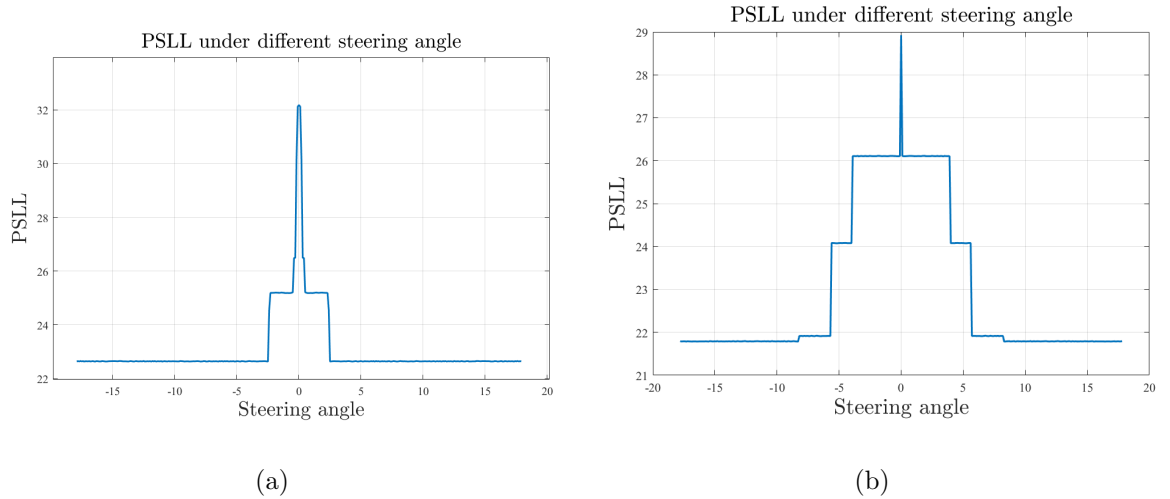


Figure 4.1: PSL of the optimized array across the entire steering angle range: (a) with 641 elements and (b) with 425 elements

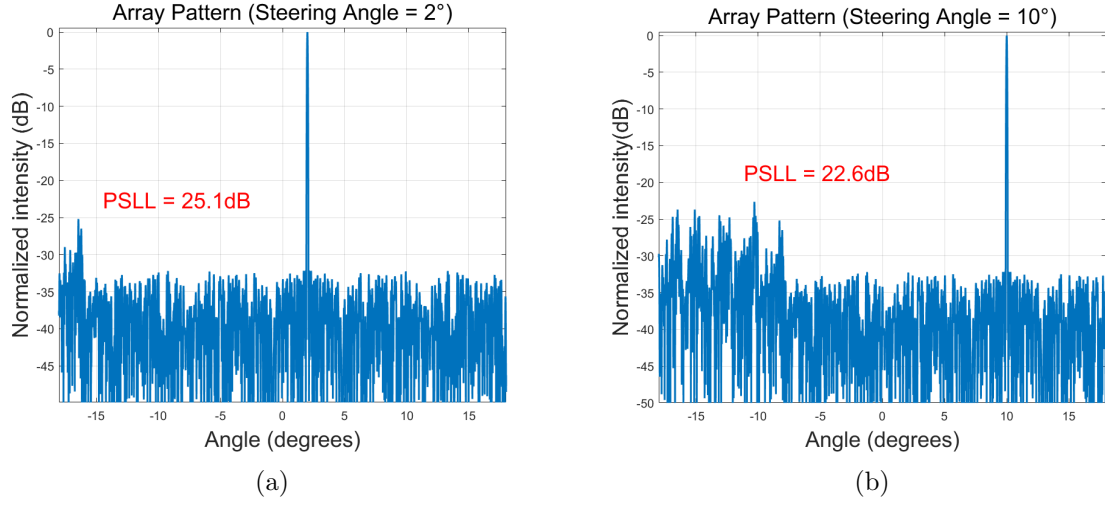


Figure 4.2: Array patterns of the optimized 641-element array at steering angles of (a) 2° and (b) 10°

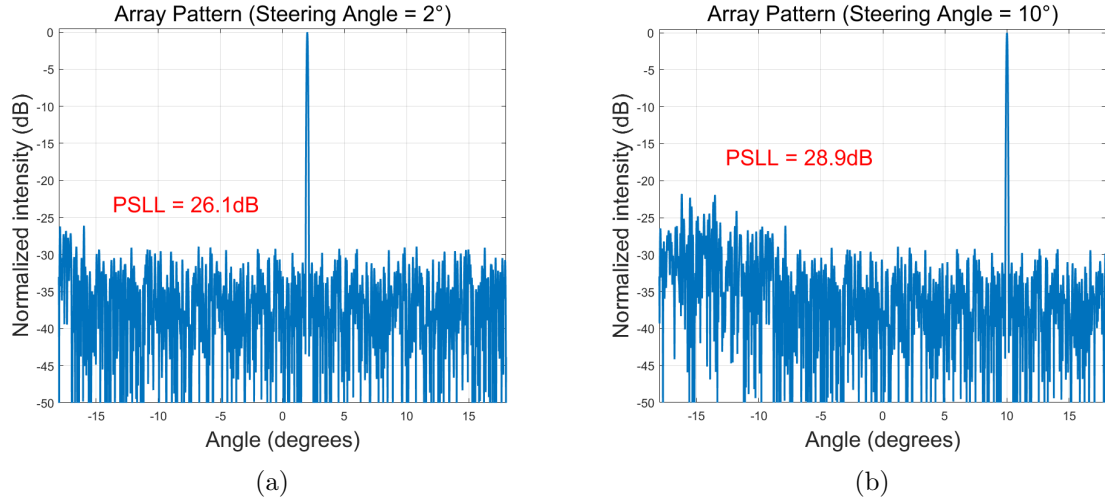


Figure 4.3: Array patterns of the optimized 425-element array at steering angles of (a) 2° and (b) 10°

4.2 Design of steering OPA

In the formulation (1.1), θ_n denotes the phase shift of the n th element, which is given by $\theta_n = -kx_n(\sin \phi)$, ϕ is the steering angle, also the direction of the main beam. By combining the exponential terms, we obtain the intensity of the far-field radiation pattern of an OPA with N elements,

$$I(\theta, \phi) = \left| \sum_{n=1}^N a_n e^{jkx_n(\sin(\theta) - \sin(\phi))} \right|^2, \quad (4.1)$$

where k is the wavenumber, x_n the position of the n th antenna. In our case, $a_n \in \{0, 1\}$ and denote $\sum a_n = N_a$ and $x_n = nd$, d is the grid space. For the 1-dimensional case, the general definition of the PSL function S given the steering angle ϕ is defined as follows:

$$S(\phi) = N_a^2 - \max_{\theta \in [\theta_{min}, \theta_{max}]} (I(\theta, \phi) \setminus N_a^2), \quad (4.2)$$

where:

- N_a^2 is the maximal intensity of the beam.
- $[\theta_{min}, \theta_{max}]$ denotes the FoV, which is assumed to be symmetric around zero for mathematical convenience and is used in practice as such.
- $(I(\theta, \phi) \setminus N_a^2)$ represents the second largest local maximum of the function $I(\theta, \phi)$ within the same range, excluding the global maximum.

To optimize performance across all detection angles θ and steering angles ϕ of the design, we are facing a two-dimensional optimization problem. Intuitively, the steering process can be viewed as a translation of the array pattern. Consequently, it is natural to investigate the exact connection of this translation. The primary objective of optimization is to minimize the peak side lobe level $P(\phi)$ over the steering range $[\phi_{min}, \phi_{max}]$. The Minimum Peak Sidelobe Level (MPSLL) M is defined as follows:

$$\begin{aligned} M &= \min_{\phi \in [\phi_{min}, \phi_{max}]} (S(\phi)) \\ &= \min_{\phi \in [\phi_{min}, \phi_{max}]} \left(N_a^2 - \max_{\theta \in [\theta_{min}, \theta_{max}]} (I(\theta, \phi) \setminus \{N_a^2\}) \right) \\ &= N_a^2 - \max_{\phi \in [\phi_{min}, \phi_{max}]} \left(\max_{\theta \in [\theta_{min}, \theta_{max}]} (I(\theta, \phi) \setminus \{N_a^2\}) \right), \end{aligned} \quad (4.3)$$

Hence, the computation of the MPSLL is fully characterized by the second-highest PSL identified within a two-dimensional angular region. A two-dimensional discretization of the $I(\theta, \phi)$ to calculate M quickly becomes memory and computational intensive. Since the GA must evaluate the PSL of numerous OPA configurations repeatedly, this approach becomes impractical for large values of N . However, the function $I(\theta, \phi)$ can be substituted by a shift-invariant function $f(kd(\sin(\theta) - \sin(\phi)))$, where

$$f(z) = \left| \sum_{n=1}^N a_n e^{jnz} \right|^2 = N_a^2 + \sum_{n=1}^N \sum_{m=n+1}^N 2a_n a_m \cos((n-m)z) \quad (4.4)$$

The function $f(z)$ satisfies the properties $f(z) = f(-z)$ and $f(z) = f(z+2\pi)$, indicating that it is both even and periodic. Owing to this shift-invariance, the MPSLL, can be evaluated by computing the PSL over a single one-dimensional segment, like

$$M = N_a^2 - \max_{z \in [z_{min}, z_{max}]} (f(z) \setminus \{N_a^2\}), \quad (4.5)$$

where

- $z_{max} = kd(\sin(\theta_{max}) - \sin(\phi_{min}))$
- $z_{min} = kd(\sin(\theta_{min}) - \sin(\phi_{max}))$

In the application, we want to detect every angle in the FOV, so the steering range will be the same as FOV, which makes $z_{min} = -z_{max} = -2kd(\sin(\theta_{max}))$. As the function f is even and the segment is symmetric around zero, z_{min} can be replaced by zero without affecting the result and all the peak side lobe occur within the finite segment $[0, \pi]$. In other words, the MPSLL is fully characterized by

$$M = N_a^2 - \max_{z \in [0, \min(\pi, z_{max})]} (f(z) \setminus \{N_a^2\}) \quad (4.6)$$

Hence, the GA can calculate the MPSLL M using a one-dimensional discretization of a bounded segment.

In this setting, the problem formulation stays the same as (3.2), but PSL is calculated by the alternative pattern:

$$f(z) = \sum_{n=0}^{N-1} a_n e^{jnz}, \quad z \in [0, \min(\pi, z_{max})]$$

4.3 Results and discussion

By giving different values of α , we can get some optimal solutions using different numbers of antennas. Figure 4.4 shows the array pattern of the configuration with around 643 elements, considering the steering process. It can be observed that at a steering angle of 0° , the PSL is reduced by approximately 3 dB compared to the previous design using 641 elements. Figure 4.5 compares the PSL of these two solutions over the full steering range. It is evident that the red curve only exceeds the blue curve near the steering angle of 0° . Across the entire steering range, the red line consistently maintains higher and more stable values. In autonomous applications, achieving strong and clear signals across all detection angles throughout the steering range is essential. Therefore, the new solution better satisfies our needs. After further experiment, we obtained other solutions that utilize a similar number of antennas as those shown in Table 3.5. Figures 4.6 and 4.7 show a comparison of PSL over the full steering range.

In this model, which considers the steering process, the grid spacing can be modified to provide greater design flexibility. Table 4.1 presents the results obtained using the same α' . With further searching, the fitness value can be improved or, at the very least, be maintained like row one and two. Comparing the results in row three and four, although the PSL decreases, the number of antennas used in the solution is also reduced, and the fitness value still increases. Comparing row five and six, the PSL increases slightly, yet the number of antennas is reduced, which is still favorable, and the fitness value increases as well.

In conclusion, the steering process indeed affects the PSL, and therefore must be taken into account during optimization. By leveraging the relationship between the original two-dimensional problem and a newly formulated one-dimensional problem, a new optimal solution can be obtained. This solution improves the performance across

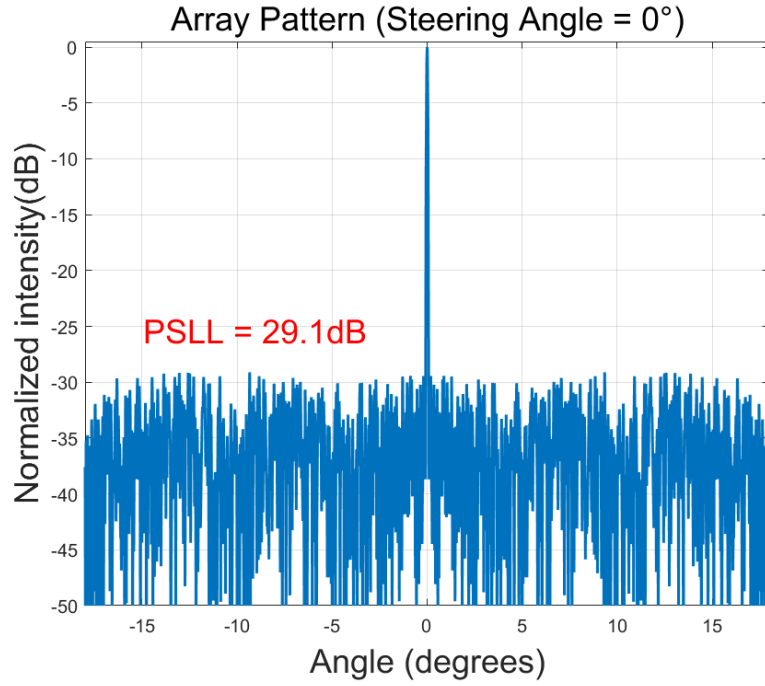


Figure 4.4: Array pattern of the optimal 643-element configuration obtained by the new pattern expression.

Chromosome size	α'	Number of antenna	PSLL(dB)	Fitness value
1×1000	0.007	643	29.11	24.61
1×1999	0.007	643	29.11	24.61
1×1000	0.013	430	26.30	20.71
1×1999	0.013	325	24.95	20.73
1×1000	0.03	142	19.66	15.44
1×1999	0.03	135	19.94	15.89

Table 4.1: Performance Comparison under Varying α' and Chromosome Size

the whole FOV. Increasing the number of grid points may not necessarily improve array performance; however, the fitness value will always increase, which aligns with the intended design philosophy of the model.

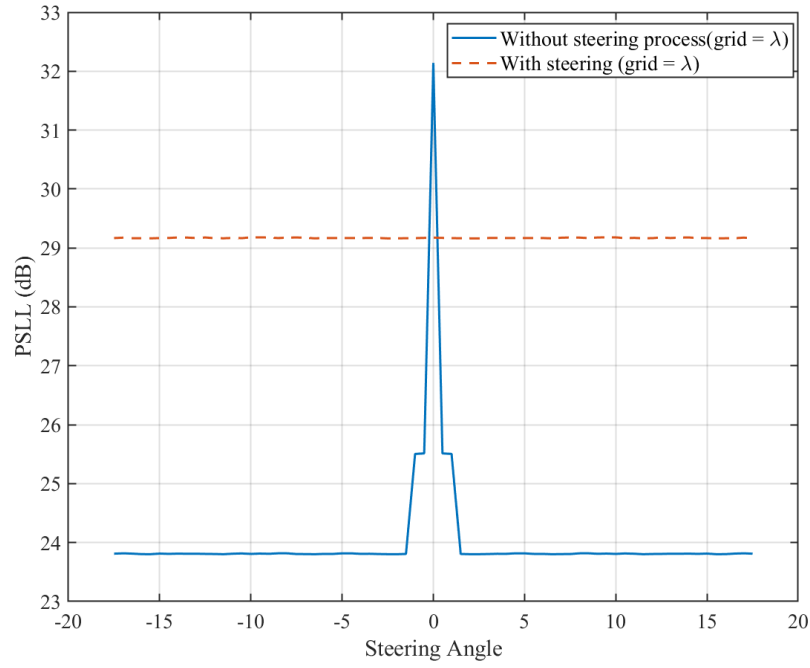


Figure 4.5: Performance Comparison Without and With Steering Process Consideration(≈ 640 Antennas)

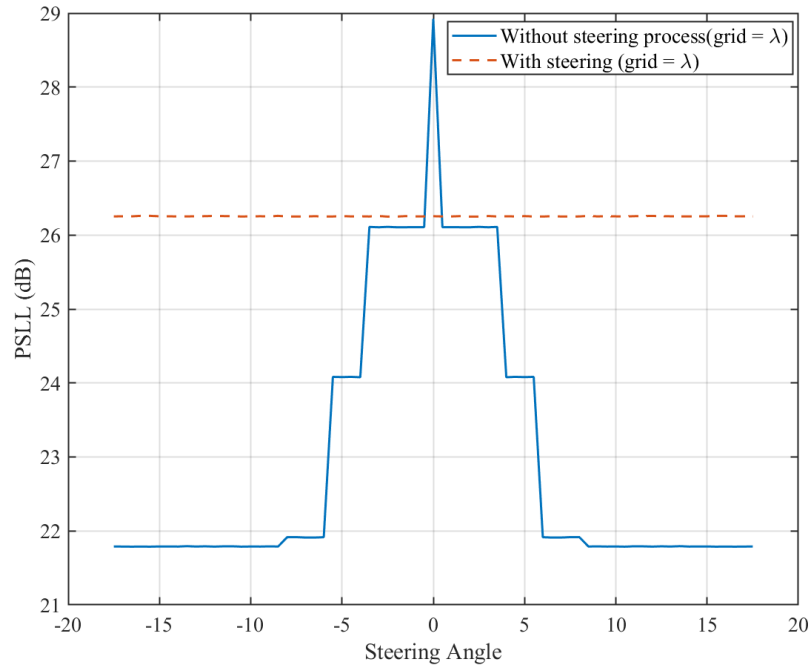


Figure 4.6: Performance Comparison Without and With Steering Process Consideration(≈ 427 Antennas)

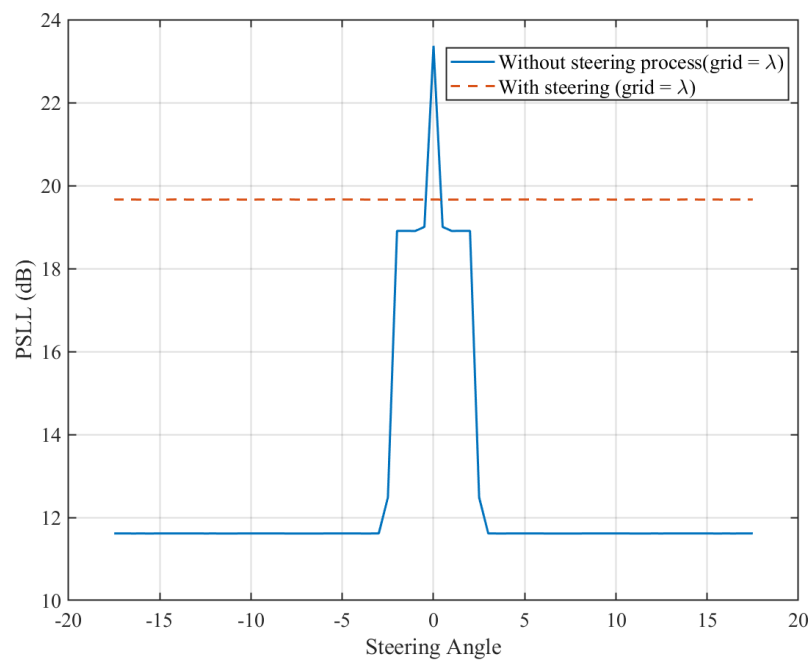


Figure 4.7: Performance Comparison Without and With Steering Process Consideration(≈ 139 Antennas)

Conclusion and Future Work

5.1 Conclusion

With the continuous advancement of autonomous driving technology, the precision and efficiency of perception systems have become increasingly critical. Among various sensors, LiDAR plays a central role, and solid-state OPA systems are being considered the future of LiDAR systems. However, designing OPAs remains challenging, particularly in balancing the number of elements and array performance. To address these challenges, this thesis adopts a sparse array design based on grid architecture and proposes a modified GA that accounts for both element density and array performance. Using this enhanced algorithm, we identify optimal array configurations across various element counts. The key contributions of this thesis are as follows:

1. **OPA Background and Motivation:** We introduce the background of OPA and outline industrial design requirements relevant to autonomous driving applications. A particular focus is given to the motivation behind using sparse OPA arrays. Due to the short wavelength of light and fabrication constraints in practical implementations, it is difficult to maintain half wavelength element spacing, which leads to the grating lobe effect. To mitigate side lobes effect and expand the FOV, a sparse array design approach is adopted. After analyzing several sparse array design methodologies, the widely used GA is selected for our OPA design.
2. **Modified GA for non-steering sparse OPA Design:** To accommodate the dual optimization goals—element density and array performance—we modify the GA’s fitness function and propose a modified GA based on a grid architecture. Through extensive tests and comparative analyses, we identify optimal GA operators and parameter configurations tailored to the design goals. Exhaustive evaluations show that our model effectively controls array sparsity while achieving relatively good PSL. Compared with the method in [3], under the same pre-set situation, both approaches utilize approximately 650 elements—our method achieves an improvement in PSL of nearly 8 dB. Moreover, when targeting the same optimal PSL, our design requires only 139 elements, 537 fewer than using method in [3], marking a significant improvement in efficiency.
3. **Modified GA for steering sparse OPA Design:** During the beam steering process, we observe PSL degradation, known as scan loss, especially when the steering angle deviates from boresight. To alleviate this issue, we reformulate the original 2D optimization problem, which considers both steering and detection angles, into a 1D optimization problem. This transformation enables us to identify array configurations that maintain robust performance across the full steering range.

5.2 Future work

OPAs represent a highly significant and board area of research. This thesis builds upon existing sparse OPA design concepts and explores sparse non-uniform array configurations. Future research directions include the following:

1. **2D Array Design:** This study focuses solely on 1D array configurations. Extending the design methodology to 2D arrays is essential for broader beamforming applications and more realistic system implementations.
2. **Off-grid Architectures:** This study focuses on a grid architecture; however, adopting off-grid configurations can offer greater design flexibility and potentially lead to improved array performance.
3. **Integration of Alternative and AI-Based Optimization Methods:** Future work could explore the application of alternative optimization techniques, including those based on artificial intelligence, to better handle complex multi-objective design problems.

The source code is available on [GitHub](#)

Bibliography

- [1] D. J. Yeong, G. Velasco-Hernandez, J. Barry, and J. Walsh, “Sensor and sensor fusion technology in autonomous vehicles: A review,” *Sensors*, vol. 21, no. 6, p. 2140, 2021.
- [2] S. Katoch, S. S. Chauhan, and V. Kumar, “A review on genetic algorithm: past, present, and future,” *Multimedia tools and applications*, vol. 80, pp. 8091–8126, 2021.
- [3] K. Yu, “Sparse non-uniform optical phased array design,” 2023, master’s thesis.
- [4] E. Marti, M. A. De Miguel, F. Garcia, and J. Perez, “A review of sensor technologies for perception in automated driving,” *IEEE Intelligent Transportation Systems Magazine*, vol. 11, no. 4, pp. 94–108, 2019.
- [5] Y. Li and J. Ibanez-Guzman, “Lidar for autonomous driving: The principles, challenges, and trends for automotive lidar and perception systems,” *IEEE Signal Process. Mag.*, vol. 37, no. 4, pp. 50–61, 2020.
- [6] T. Kim, P. Bhargava, C. V. Poulton, J. Notaros, A. Yaacobi, E. Timurdogan, C. Baiocco, N. Fahrenkopf, S. Kruger, T. Ngai *et al.*, “A single-chip optical phased array in a wafer-scale silicon photonics/cmos 3d-integration platform,” *IEEE J. Solid-State Circuits*, vol. 54, no. 11, pp. 3061–3074, 2019.
- [7] J. He, T. Dong, and Y. Xu, “Review of photonic integrated optical phased arrays for space optical communication,” *IEEE Access*, vol. 8, pp. 188 284–188 298, 2020.
- [8] C.-P. Hsu, B. Li, B. Solano-Rivas, A. R. Gohil, P. H. Chan, A. D. Moore, and V. Donzella, “A review and perspective on optical phased array for automotive lidar,” *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 27, no. 1, pp. 1–16, 2020.
- [9] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, “Pointpillars: Fast encoders for object detection from point clouds,” in *Proceedings of the IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 12 697–12 705.
- [10] W. Tong, E. Yang, Y. Pang, H. Yang, X. Qian, R. Yang, B. Hu, J. Dong, and X. Zhang, “An efficient, fast-responding, low-loss thermo-optic phase shifter based on a hydrogen-doped indium oxide microheater,” *Laser Photonics Rev.*, vol. 17, no. 9, p. 2201032, 2023.
- [11] T. Komljenovic, R. Helkey, L. Coldren, and J. E. Bowers, “Sparse aperiodic arrays for optical beam forming and lidar,” *Optics express*, vol. 25, no. 3, pp. 2511–2528, 2017.
- [12] P. Wu, Y. Liu, Z. Zhao, and Q. H. Liu, “Sparse antenna array design methodologies: a review,” *J. Electron. Sci. Technol.*, vol. 100, p. 100276, 2024.

- [13] L. Cen, W. Ser, W. Cen, and Z. L. Yu, "Linear sparse array synthesis via convex optimization," in *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*. IEEE, 2010, pp. 4233–4236.
- [14] S. Vankayalapati, L. P, K. Kumar, N. V. Batch, A. Dalali, N. D. Inti, and N. R. Netinti, "Application of binary particle swarm optimization algorithm for thinned planar antenna array synthesis," in *2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS)*, vol. 1, 2022, pp. 1414–1418.
- [15] C. Liu, W. Xu, L. Zhou, L. Lu, and J. Chen, "Multi-agent genetic algorithm for sparse optical phased array optimization," in *Proc. Asia Commun. Photonics Conf.* Optica Publishing Group, 2019, pp. M4A–286.
- [16] D. Zhang, F. Zhang, and S. Pan, "Grating-lobe-suppressed optical phased array with optimized element distribution," *Opt. Commun.*, vol. 419, pp. 47–52, 2018.
- [17] K. Du, R. Wang, J. Guo, R. Jiang, D. Kan, and Y. Zhang, "Design of a sparse array for a one-dimensional non-uniform optical phased array," *JOSA B*, vol. 39, no. 4, pp. 1141–1146, 2022.
- [18] Z. Wang, P. Ma, Y. Yang, L. Cui, L. Yu, G. Luo, P. Wang, J. Pan, Y. Su, and Y. Zhang, "Non-uniform optical phased array with a large steering angle and high side mode suppression ratio," *Applied Optics*, vol. 61, no. 36, pp. 10 788–10 793, 2022.
- [19] T. Dong, J. He, X. He, Y. Xu, and J. Zhou, "Hybrid design approach of optical phased array with wide beam steering range and low side-lobe level," *Opt. Lett.*, vol. 47, no. 4, pp. 806–809, 2022.
- [20] X. He, T. Dong, J. He, and Y. Xu, "A design approach of optical phased array with low side lobe level and wide angle steering range," in *Photonics*, vol. 8, no. 3. MDPI, 2021, p. 63.
- [21] B. Yang, H. Chen, S. Yang, and M. Chen, "An improved aperiodic opa design based on large antenna spacing," *Opt. Commun.*, vol. 475, p. 125852, 2020.
- [22] Y. Liu, Q. H. Liu, and Z. Nie, "Reducing the number of elements in the synthesis of shaped-beam patterns by the forward-backward matrix pencil method," *IEEE Trans. Antennas Propag.*, vol. 58, no. 2, pp. 604–608, 2009.
- [23] H. Shen, B. Wang, and X. Li, "Shaped-beam pattern synthesis of sparse linear arrays using the unitary matrix pencil method," *IEEE Antennas Wirel. Propag. Lett.*, vol. 16, pp. 1098–1101, 2016.
- [24] L. Cen, W. Ser, W. Cen, and Z. L. Yu, "Linear sparse array synthesis via convex optimization," in *Proc. IEEE Int. Symp. Circuits Syst.* IEEE, 2010, pp. 4233–4236.
- [25] Z. Wang, G. Sun, J. Tong, and Y. Ji, "Pattern synthesis for sparse linear arrays via atomic norm minimization," *IEEE Antennas Wirel. Propag. Lett.*, vol. 20, no. 12, pp. 2215–2219, 2021.

- [26] S. Yang, B. Liu, Z. Hong, and Z. Zhang, “Low-complexity sparse array synthesis based on off-grid compressive sensing,” *IEEE Antennas Wireless Propag. Lett.*, vol. 21, no. 12, pp. 2322–2326, 2022.
- [27] E. J. Candes, M. B. Wakin, and S. P. Boyd, “Enhancing sparsity by reweighted ℓ_1 minimization,” *Journal of Fourier analysis and applications*, vol. 14, pp. 877–905, 2008.