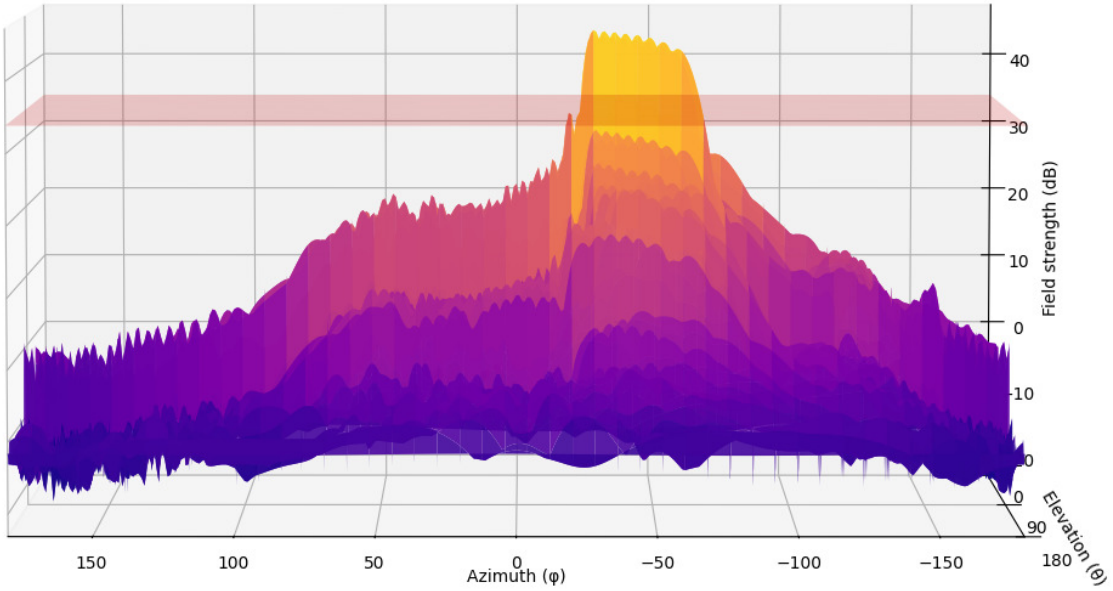


Stochastic Methods for Beam Weight Optimization for Large Antenna Arrays

A Time Efficient and Robust Method for Beam Weight Optimization Using Non-ideal Data

Master's thesis in Data Science and AI
Master's thesis in Engineering Mathematics and Computational Science

MATILDA SELVARAJ TIVESTEN
JOHAN ÖDESJÖ

DEPARTMENT OF MATHEMATICAL SCIENCES
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2025
www.chalmers.se

Master's thesis 2025

Stochastic Methods for Beam Weight Optimization for Large Antenna Arrays

A Time Efficient and Robust Method for Beam Weight Optimization Using Non-ideal Data

MATILDA SELVARAJ TIVESTEN
JOHAN ÖDESJÖ

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Mathematical Sciences
Division of Applied Mathematics and Statistics
Chalmers University of Technology
Gothenburg, Sweden 2025

Stochastic Methods for Beam Weight Optimization for Large Antenna Arrays
A Time Efficient and Robust Method for Beam Weight Optimization Using Non-ideal Data
MATILDA SELVARAJ TIVESTEN
JOHAN ÖDESJÖ

© MATILDA SELVARAJ TIVESTEN & JOHAN ÖDESJÖ, 2025.

Supervisor: Dr. Artem Roev, Ericsson AB
Examiner: Prof. Rebecka Jörnsten, Department of Mathematical Sciences

Master's Thesis 2025
Department of Mathematical Sciences
Division of Applied Mathematics and Statistics
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Radiation pattern of a reference field used during beam weight optimization.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2025

Stochastic Methods for Beam Weight Optimization for Large Antenna Arrays
A Time Efficient and Robust Method for Beam Weight Optimization Using Non-ideal Data
Matilda Selvaraj Tivesten, Johan Ödesjö
Department of Mathematical Sciences
Chalmers University of Technology

Abstract

Traditional methods for beam weight optimization suffer from poor convergence and long run times for large antenna arrays and are limited to ideal antenna element models. To overcome these limitations, more efficient optimization algorithms are needed. Previous research has demonstrated that stochastic approaches such as genetic algorithms and particle swarm optimization are promising alternatives for solving this problem. This thesis investigates the tailoring of these stochastic optimization algorithms for beam weight optimization using high-dimensional, realistic, simulated data.

We propose the use of an absolute radiation pattern scale as opposed to a relative one, which allows us to simplify the objective function and promote a more efficient energy usage. We decrease runtimes by restricting the optimization to a low-rank search space, implementing warm starts, and leveraging GPU acceleration.

We find that these tailored methods show robust performance and converge significantly faster compared to current methods. The resulting beams also consistently outperform those currently in use. We demonstrate the effectiveness of our approach on antenna arrays of up to 1,152 subarrays, where antennas of 288 subarrays or more are considered large.

Keywords: antenna arrays, beamforming, advanced antenna systems, stochastic optimization, genetic algorithms, particle swarm optimization.

Acknowledgements

This thesis would not have been possible without the guidance, support, and encouragement from many different people.

Thank you to our academic supervisor, Prof. Rebecka Jörnsten, for being a leading light in the worlds of statistics and optimization.

Thank you to our Ericsson supervisor, Dr. Artem Roev, for sharing your expertise on antenna arrays.

Thank you to Ericsson for hosting us, to our line manager Dr. Bhushan Billade for having us, and to Håkan Karlsson for answering all of our questions, even the bad ones.

Thank you to our proofreaders Alexander Mayer, Dr. Helena Ödesjö, Ulf Ödesjö, and Alma Cavallin for your time and feedback.

Thank you to Bella for the endless coffee and enthusiasm.

Matilda Selvaraj Tivesten & Johan Ödesjö, Gothenburg, June 2025

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

BW	Beam Weights
CAR	Coverage Angular Range
CM	Current Method
EIRP	Effective Isotropic Radiated Power
ESAP	Embedded Subarray Patterns
EI	Expected Improvement
GA	Genetic Algorithms
GoB	Grid of Beams
MSE	Mean Square Error
PSO	Particle Swarm Optimization

Nomenclature

Below is the nomenclature of indices, sets, parameters, and variables that have been used throughout this thesis.

Indeces

m	Number of rows of antenna array
n	Number of columns of antenna array

Coordinates

$\{ x, y, z\}$	Cartesian coordinate system
$\{ \theta, \varphi \}$	Far field coordinate system, $\theta = \frac{\pi}{2}$, $\varphi = 0$ corresponds to propagation at normal incidence
$\{ r, \theta, \varphi \}$	Spherical coordinates

Antennas

$\mathbf{E}(r, \theta, \varphi)$	Electrical field
$\mathbf{G}(\theta, \varphi)$	Far field function
$\mathbf{G}_A(\theta, \varphi)$	Resulting far field function of an antenna array
$U(\theta, \varphi)$	Radiation intensity or Radiation pattern
$D(\theta, \varphi)$	Directivity
$\mathcal{G}(\theta, \varphi)$	Gain
$\mathcal{G}_R(\theta, \varphi)$	Realized gain
$\mathrm{EIRP}(\theta, \varphi)$	Effective Isotropic Radiated Power
P_R	Total radiated power
P_O	Total accepted power

P_{M_2}	Power available from the matched transmission line feeding the antenna
$ESAP_j(\theta, \varphi)$	Embedded subarray pattern j
bw_n	Beam weight n
N	Number of beam weights

Optimization

$\mathcal{F}_{main}$	Set of angles (θ, φ) that defines the main lobe area
$\mathcal{F}_{side}$	Set of angles (θ, φ) that defines the side lobe area
$\mathcal{L}(bw)$	Objective function
$\mathcal{F}(bw)$	Fitness function

Contents

List of Acronyms **ix**

Nomenclature **xi**

List of Figures **xv**

List of Tables **xix**

List of Algorithms **xxi**

1 Introduction **1**
1.1 Problem Statement 1
1.2 Constraints 2

2 Electromagnetic Field Theory **3**
2.1 Azimuth and Elevation Coordinates 3
2.2 The Far Field 3
2.3 Directivity and Gain 4
2.4 EIRP 6
2.5 Units 6
2.6 Polarisation of Plane Waves 7
2.7 Aperture Size 7

3 Antenna Arrays **9**
3.1 Beam Types and Uses 9
3.2 Antenna Arrays 12
3.3 The Superposition Principle 14
3.4 Beamforming with Antenna Arrays 15
3.5 Single- and Dual-Polarized Beamforming 16
3.6 Metrics 16

4 Optimization **21**
4.1 Classical Optimization 21
4.2 Stochastic Optimization 21
4.3 Bayesian Optimization 26

5 Method **31**
5.1 Specification of the Issue Being Investigated 31

5.2 Algorithm Overview . . . 32
5.3 Design of Objective Function . . . 32
5.4 Optimization Algorithms . . . 34
5.5 Search Space . . . 34
5.6 t-Tests . . . 36
5.7 Hyperparameter Tuning . . . 40
5.8 Extension to Dual-Polarized Beamforming . . . 41

6 Results **43**
6.1 Performance . . . 44
6.2 Comparison to Current Method . . . 46
6.3 Dimensionality Tests . . . 50
6.4 Stability Analysis . . . 54
6.5 Runtime in Real Application . . . 56

7 Discussion **59**
7.1 Fulfilment of Desiderata . . . 59
7.2 Search Space Dimension . . . 61
7.3 Limitations . . . 62

8 Conclusion **65**
8.1 Fulfilment of the Desiderata . . . 65
8.2 Further Studies . . . 66

Bibliography **69**

A Method Appendix **I**
A.1 Paired t-Tests . . . I

B Result Appendix **XI**
B.1 Performance . . . XI
B.2 Comparison to Current Method . . . XII
B.3 p-Values . . . XIV

List of Figures

2.1 Coordinate system used for antennas, with a Cartesian coordinate system for reference. The antenna array is situated in the yz-plane and faces the positive x-direction. 3
2.2 In the far field region, the electrical field can approximated as a plane wave. 4

3.2 Wide beams are used to identify the optimal traffic beam with which to serve a user. 10
3.3 Wide beams are used to identify the optimal traffic beam with which to serve a user. The beams are arranged in a GoB that cover the CAR. 11
3.4 Creation of an envelope over a set of traffic beams. 11
3.5 One antenna element consisting of two antennas rotated 45° in opposite directions. 13
3.6 A 3×1-subarray consists of three antenna elements arranged into three rows and one column. 13
3.7 A 3×8 antenna array consists of 24 subarrays arranged into 3 rows and 8 columns. 14
3.8 Constructive and destructive interference between two vectors (green and red arrows respectively). The black arrow represents the resulting vector. 15
3.9 Representation of how the electrical field is calculated from the ESAPs and BWs. 16
3.10 Cut of an example beam. The pink line is the reference field in the main lobe area and the blue line is the optimized field. The black and grey dotted lines are the side lobe and main lobe thresholds, respectively. The red area represents the main lobe loss. 17
3.11 An example of the radiation pattern of a wide beam and its main lobe. 18
3.12 Cut of an example beam. The pink line is the reference field in the main lobe area and the blue line is the optimized field. The black and grey dotted lines are the side lobe and main lobe thresholds, respectively. The red area represents the side lobe loss. 19

4.1 Visualization of a generation of GA with its main operations. The blue ovals are populations, and the smaller circles are individuals. . . 22
4.2 Visualization of a swarm of particles and their velocities. The blue star represents the current best position of the blue particle and the pink star represents the current best position of the entire swarm. . . 25

4.3 The mean of the surrogate function is shown as a dashed green line, and the area within 1.96 standard deviations from the mean of the surrogate model is shown shaded in green. The mean of the true function is shown as a dashed red line, and all sampled points are shown as red dots. The acquisition function is shown in blue, with a blue dot denoting the next query point. 28

5.1 A flowchart of the method from data and user inputs to optimized beam weights. The dotted line only applies when GA is the algorithm used in the optimization step and represents the use of traffic beam BWs as warm starts. 32
5.2 Comparison of two ways of initializing beam weights. Each point represents one beam weight. 36
5.3 Result of paired t-test for GA for initial positions that are half random and half traffic beam weights. α is the level of noise added unto the traffic beams. 40

6.1 Comparisons of main lobes of beams optimized by GA and PSO and the reference beam. 44
6.2 Comparisons between cuts of beams (blue lines) optimized by GA and PSO, and the reference beam (pink line). Main and side lobe thresholds are shown as well (grey and black line respectively). . . . 45
6.3 Comparison of best loss of final iteration for different metrics for 20 runs of GA and PSO. 46
6.4 A wide beam optimized by CM. In b) and c) the pattern optimized by CM is blue and the radiation pattern of the reference field is pink. The grey and black dashed lines represent the side and main lobe thresholds respectively. 47
6.5 GoB of wide beams optimized by different methods and a GoB of the corresponding traffic beams. 48
6.6 Total loss for all beams in the GoB shown in Figure 6.5 and difference in total loss between CM and both GA and PSO. Each loss corresponds to the wide beam in the same position in the GoB. . . . 49
6.7 Distributions of final loss of 20 runs of GA and PSO for search spaces of rank 1/2, 1, and 2. 50
6.8 Comparison between time and loss for different algorithms for different ranks. Results for GA are shown in blue, and results for PSO are shown in pink. 51
6.9 Loss distributions GA and PSO for different ranks. Results for GA are shown in blue, and results for PSO are shown in pink. 52
6.10 Comparison between time and loss for different algorithms for different array sizes. Results for GA are shown in blue, and results for PSO are shown in pink. 53
6.11 Total loss for 20 runs of GA and PSO until convergence using single- and dual-polarized beamforming. 54

6.12 Variation of total loss for 20 beam weights with Gaussian noise added to either phase or amplitude. Red arrow indicates maximum allowed error according to internal Ericsson standards. 55
6.13 Comparisons of beams with noise with $3\sigma_P = 22°$ added to phases and $3\sigma_A = 2$dB added to amplitudes. Also a beam with no noise added for reference. 56
6.14 Beams marked 1, 2, and 3 can be approximately calculated using beam weights of the beam marked X by mirroring its amplitudes vertically, horizontally, or both. 57

A.1 Results from paired t-test for optimization using genetic algorithm and side lobe thresholds relative to optimized field vs reference field. Both methods were ran 20 times. II
A.2 Results from paired t-test for particle swarm optimization and side lobe thresholds relative to optimized field vs reference field. Both methods were ran 20 times. III
A.3 Results from paired t-test for optimization using constrained vs unconstrained GA. IV
A.4 Results from paired t-test for constrained vs unconstrained PSO. . . . V
A.5 Results from paired t-test for PSO with initial positions completely random vs half random and half traffic beam beam weights. VI
A.6 Results from paired t-test for GA with initial positions completely random vs half random and half traffic beam beam weights. VII
A.7 Result of paired t-test for GA for initial positions that are half random and half traffic beam weights. α is the level of noise added unto the traffic beams. VIII

B.1 Comparison of best taper loss of final iteration for 20 runs of GA and PSO. XI
B.2 Taper loss for all beams in the GoB shown in Figure 6.5, and difference in total loss between CM and both GA and PSO. XII
B.3 Main lobe loss for all beams in the GoB shown in Figure 6.5, and difference in main lobe loss between CM and both GA and PSO. . . . XIII
B.4 Side lobe loss for all beams in the GoB shown in Figure 6.5, and difference in main lobe loss between CM and both GA and PSO. . . . XIV

List of Tables

6.1 Mean loss and runtime over 20 runs of GA and PSO. 45
6.2 Standard deviation of final loss for all loss metrics for GA and PSO. . 46
6.3 Mean loss and runtime over 20 runs of GA and PSO when terminated using convergence criterion. 47
6.4 Mean difference in total loss and taper loss between GoBs optimized by CM compared to GA and PSO. For reference, taper loss is usually in the range two to three. 49
6.5 Mean runtime over 20 runs for different search space ranks. 51
6.6 Mean total loss and runtime for 20 runs of GA and PSO for an antenna array with 32 subarrays. 53
6.7 Mean total loss and runtime for 20 runs of GA and PSO for an antenna array with 1152 subarrays. 53
6.8 Mean runtime over 20 runs for single- and dual-polarisation. 54
6.9 Mean total loss per beam over GoB for different kinds of warm starts. 57
6.10 Mean runtime per beam over GoB for different kinds of warm starts. 58
6.11 Estimated times needed to optimize the entire GoB of an antenna with 288 subarrays for five frequencies using different warm start methods. 58

A.1 p-values for all paired t-tests, rounded to two significant figures. . . . IX

B.1 Mean taper loss over 20 runs of GA and PSO. XI
B.2 Standard deviation of final taper loss for GA and PSO. XI
B.3 Mean difference in total loss and taper loss between all 24 wide beam optimized by current method compared to GA and PSO. XIV
B.4 p-values for all two-sample t-tests, rounded to two significant figures. XV

List of Algorithms

1 Genetic Algorithm for High Dimensional Problems 23
2 Crossover 23
3 Mutation 24
4 Particle Swarm Optimization with Inertia 26
5 Bayesian Optimization 27

1
Introduction

Ericsson is one of the leading providers of information and communication technologies to service providers, while also investing in technical development. One area under research at Ericsson is *antenna arrays*. These can conduct traffic with user equipment by creating peaks in the intensity of the electrical field, known as *beams*, in the direction of the user by manipulating signals called *beam weights*. Beam weights that result in simple beams can be calculated analytically, while more complicated beams require more complex tools, such as optimization methods.

A larger antenna array can create stronger beams and better specify direction, and therefore, the current trend appears to be toward larger antenna arrays for 6G. However, current methods for beam weight optimization used at Ericsson suffer from long run times and poor convergence for larger products. Producing beam weights for large antenna arrays in an efficient and precise manner is a challenge that Ericsson needs to tackle to unlock further development opportunities within the area of large antenna arrays.

The performed initial studies show that the use of tailored algorithms is a promising alternative to the generic optimizers used thus far, for example through the use of CNNs to predict optimal beam weight values, such as by Frilund and Olofsson [1], although this method was only explored for small antenna arrays. The aim of this thesis is to create an optimization algorithm tailored to the problem of beam weight optimization for large antenna arrays. The algorithm needs to be able to consistently converge in a reasonable time while keeping the desired spatial performance of Ericsson's Active Antenna System products.

1.1 Problem Statement

The objective of this thesis is to design a time-efficient optimization algorithm that can be used to optimize beam weights for large antenna arrays such that the resulting electrical field is fitted to a given reference electrical field.

Previously used methods are unable to use real embedded subarray pattern data in the optimization process and can only handle ideal data. This is a drawback, as the physical characteristics of the antenna array affect the resulting electrical field.

Ericsson uses a number of different metrics to evaluate the quality of a beam weight

choice for a given reference field. For the purposes of this master thesis, a subset of these metrics will be selected to be used as components of the objective function, but all of them will be used when evaluating the quality of the solutions. These metrics are introduced in Section 3.6.

The total list of desiderata for the final algorithm is stated here. The algorithm should:

- Consistently produce beam weights that correspond to an electrical field with desired metric values with respect to a given reference field.
- Run and converge in a shorter time than the current methods.
- Produce results that are robust against small variations resulting from the manufacturing of antenna and measurement components.
- Optimize beam weights using measured and simulated realistic ESAPs.

1.2 Constraints

In this thesis, there are a number of constraints set from the beginning to limit the scope of the project. One such limitation is that we do not take into account emissions outside the frequency bandwidth of the antenna, i.e., all antennas are assumed to have no out-of-band emissions. We further assume that all electrical fields propagate through free space and do not consider other mediums. Finally, we use realistic simulated data, and no data measured from real antennas since no such measurements exist for the specific antenna size we mainly investigate. Due to the computational cost of producing realistic simulated embedded subarray patterns, we only have one such set available. This data is generated using the first two assumptions. However, data could be simulated without these two assumptions, though they have a negligible effect and are therefore generally included as it simplifies the process.

2

Electromagnetic Field Theory

In this chapter, we introduce the majority of the electromagnetic field theory that is necessary to understand the method and results of this thesis. We explain concepts such as relevant coordinate systems, scales, units, and polarisations.

2.1 Azimuth and Elevation Coordinates

When working with antennas, it is common to use a coordinate system that is similar to spherical coordinates. We use the *angle in azimuth* φ to refer to the angle between the x-axis and the projection of a point onto the xy-plane. The *angle in elevation* θ refers to the angle between the z-axis and the point. This means that φ is in the range $[-180, 180]$ and θ is in the range $[0, 180]$. The coordinate system is illustrated in Figure 2.1.

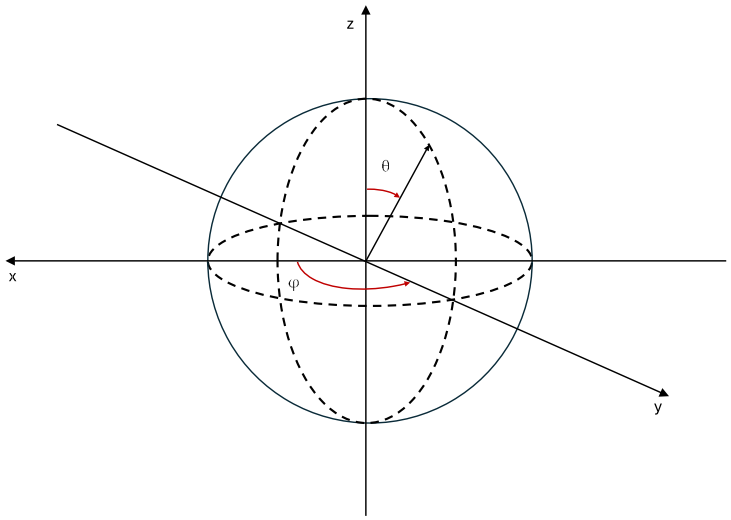


Figure 2.1: Coordinate system used for antennas, with a Cartesian coordinate system for reference. The antenna array is situated in the yz-plane and faces the positive x-direction.

2.2 The Far Field

When working with antennas, the focus is often on the *far field*, which is defined as the region where

$$r \geq \frac{2D^2}{\lambda},$$

where λ is the wavelength, r is the distance from the source to the measuring point, and D is the largest diameter of the antenna [2, p. 33].

Figure 2.2: In the far field region, the electrical field can approximated as a plane wave.

The far field is of interest because the electrical field can be approximated as a plane wave [3], as shown in Figure 2.2. There, the electrical field's dependency on r becomes nearly perfectly separable from the θ- and φ-dependency and the field can be approximately expressed as

$$\mathbf{E}(r, \theta, \varphi) = \frac{1}{r} e^{-j\frac{2\pi}{\lambda}r} \mathbf{G}(\theta, \varphi), \tag{2.1}$$

where $\mathbf{G}(\theta, \varphi)$ is the directional dependency of the field, known as the *far field function*. This is the reason that the azimuth/elevation coordinate system omits a radial coordinate. The far field function is complex valued and therefore also affects the phase of $\mathbf{E}$ [2, p. 35].

Given the far field function $\mathbf{G}(\theta, \varphi)$ the radiated power per unit solid angle can be calculated as

$$U(\theta, \varphi) = \frac{1}{2\zeta} |\mathbf{G}(\theta, \varphi)|^2,$$

where $U(\theta, \varphi)$ is known as the *radiation intensity* and ζ is the wave impedance [2, p. 36]. When we refer to the radiation intensity for a set of angles, we call it the *radiation pattern*.

2.3 Directivity and Gain

The *directivity* $D(\theta, \varphi)$ of an antenna in a given direction is defined as the ratio between the radiation intensity in that direction and the radiation intensity averaged

over all directions [4, p. 12]. This is written as,

$$D(\theta, \varphi) = 4\pi \frac{U(\theta, \varphi)}{P_{\mathrm{R}}},$$

where P_{R} is the total radiated power of the antenna and is defined as,

$$P_{\mathrm{R}} = \int_0^{2\pi} \int_0^{\pi} U(\theta, \varphi) \sin(\theta) d\theta d\varphi.$$

When directivity is mentioned without a directional dependency, it is usually assumed that it is in the direction that maximizes the radiation intensity [5, p. 39]. However, in this thesis, we always refer to directivity with a directional dependence.

Directivity does not measure the antennas' efficiency, only its directional capabilities, and therefore *gain*, $\mathcal{G}(\theta, \varphi)$ is often used as accounts for both. The *gain* of an antenna in a given direction is defined by IEEE as the ratio between the radiation intensity in that direction and the radiation intensity produced if the power accepted by the antenna was radiated isotropically, i.e., equally in all directions [4, p. 17]. Thus, gain is defined as

$$\mathcal{G}(\theta, \varphi) = 4\pi \frac{U(\theta, \varphi)}{P_{\mathrm{O}}}, \tag{2.2}$$

where P_{O} is the total accepted power of the antenna. According to [4, p. 30] $P_{\mathrm{O}} \cdot \eta = P_{\mathrm{R}}$, where $\eta \in [0, 1]$ is the *radiation efficiency*, which is a measure of how large part of the accepted power is radiated by the antenna. Thus, equation (2.2) can be rewritten as,

$$\mathcal{G}(\theta, \varphi) = 4\pi\eta \frac{U(\theta, \varphi)}{P_{\mathrm{R}}} = \eta D(\theta, \varphi).$$

In the same way as for directivity, when gain is mentioned without a direction, it is usually assumed that it is the direction that maximizes the radiation intensity [5, p. 59]. Likewise, in this thesis, we always refer to gain with a directional dependence.

Gain is not enough when we want to know the performance of our antenna in relation to the supplied power. It accounts for the losses inside the antenna, but to account for the energy loss between the energy source and the antenna, we use *realized gain*. It is defined as the gain of an antenna multiplied with an *impedance mismatch factor*, $M_2 \in [0, 1]$, which is the ratio between the total accepted power of the antenna, P_O and the power available from the matched transmission line feeding the antenna, P_M [4, pp. 19, 31]. Thus, realized gain is

$$\mathcal{G}_R(\theta, \varphi) = M_2 \mathcal{G}(\theta, \varphi).$$

In the same way as for gain and directivity, when realized gain is mentioned without a direction, it is usually assumed that it is the direction that maximizes the radiation intensity. As with directivity, we always refer to realized gain with a directional dependence in this thesis.

2.4 EIRP

Effective Isotropic Radiated Power (EIRP) is a measure of how much energy an antenna is transmitting in a given direction and is often used when studying the performance of an active antenna. According to "IEEE Standard for Definitions of Terms for Antennas" [4, p. 15] it can be calculated in the following way,

$$\mathrm{EIRP}(\theta,\varphi) = P_{\mathrm{O}}\mathcal{G}(\theta,\varphi). \tag{2.3}$$

Since gain can be rewritten as $\mathcal{G}(\theta,\varphi) = \frac{1}{M_2}\mathcal{G}_R(\theta,\varphi)$ and $P_O = M_2 P_M$, the relation can also be written as

$$\mathrm{EIRP}(\theta,\varphi) = P_{\mathrm{M}}\mathcal{G}_R(\theta,\varphi). \tag{2.4}$$

Furthermore, since gain can also be rewritten as $\mathcal{G}(\theta,\varphi) = \eta D(\theta,\varphi)$ and $P_{\mathrm{O}} = \frac{1}{\eta}P_{\mathrm{R}}$, we can also write this relation as

$$\mathrm{EIRP}(\theta,\varphi) = P_{\mathrm{R}}D(\theta,\varphi). \tag{2.5}$$

Equation (2.5) is used more often than (2.3) as P_{R} can be measured directly, whereas P_{O} is very difficult to measure. However, Equation (2.4) is the easiest to use, as the power in the matched transmission line, P_M, is the easiest to measure.

2.5 Units

In this thesis, the primary focus will be on the far field, where we measure directivity and gain in dBi. The i in dBi stands for isotropic because we measure relative an isotropic antenna, i.e., an antenna that radiates equally in all directions. This is according to convention defined as

$$\begin{aligned} D_{\mathrm{dBi}} &= 10 \cdot \log_{10}(D)\ [\mathrm{dBi}] \\ \mathcal{G}_{\mathrm{dBi}} &= 10 \cdot \log_{10}(\mathcal{G}/\mathcal{G}_{\mathrm{iso}})\ [\mathrm{dBi}], \end{aligned}$$

where $\mathcal{G}_{\mathrm{iso}}$ is the gain of an isotropic antenna.

The unit dBm is used to express power, for example $U(\theta,\varphi)$, P_O, or P_R, in dB relative to 1 mW. That is, it is used to express a power P, measured in mW, in logarithmic scale [6, p. 30]. It is converted according to

$$P_{\mathrm{dBm}} = 10 \cdot \log_{10}(P)\,[\mathrm{dBm}].$$

If the power, P, is in W we convert it to dBm according to

$$P_{\mathrm{dBm}} = 10 \cdot \log_{10}(P) + 30\,[\mathrm{dBm}].$$

EIRP is also often expressed in dBm, and if it is measured in W it is converted according to

$$EIRP_{\text{dBm}} = 10 \cdot \log_{10}(EIRP) + 30\,[\text{dBm}].$$

Every time directivity, gain, or EIRP is mentioned in the context of this thesis, it is in units of dBi and dBm, respectively.

2.6 Polarisation of Plane Waves

As mentioned in Section 2.2, all electrical fields can be approximated as a plane wave if considered in a sufficiently small area in the far field. The electrical field of a plane wave propagating in the $\hat{\mathbf{z}}$-direction can according to [2, p. 23] be described as

$$\mathbf{E} = \mathbf{E}_{\text{t}} e^{-j\frac{2\pi}{\lambda}z} = [E_{\text{x}}\hat{\mathbf{x}} + E_{\text{y}}\hat{\mathbf{y}}]e^{-j\frac{2\pi}{\lambda}z}$$

where $\mathbf{E}_{\text{t}}$ is the $\mathbf{E}$-field with the propagation vector $e^{-j\frac{2\pi}{\lambda}z}$ omitted. It is possible to decompose $\mathbf{E}_{\text{t}}$ into its two *polarisations*, $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$. The components E_{x} and E_{y} can be expressed as the projection of $\mathbf{E}$ onto the orthogonal unit vectors $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ respectively. We use $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ as an example here, but this is applicable to any set of orthogonal unit vectors.

If the polarisations can be individually controlled, then it is possible to send twice the amount of information by sending different signals along the different polarisations. This is true for any pair of orthogonal polarisations, however, for this to work well the polarisations must be well separated, i.e. one must be able to detect one of them without the other interfering.

Each antenna aims to send only one polarisation, which we refer to as the *co-polar* polarisation for that antenna. In the non-ideal case it is almost certain that some signals will be sent in the other polarisation, which is referred to as *cross-polar* [2, p. 23].

Thus, the electrical field from an antenna can be decomposed into the co-polar and cross-polar components, E_{co} and E_{xp}, where E_{co} is parallel to the co-polar unit vector $\hat{\mathbf{co}}$ and E_{xp} is parallel to the cross-polar unit vector $\hat{\mathbf{xp}}$. Therefore, the electrical field can be written as

$$\mathbf{E} = [E_{\text{co}}\hat{\mathbf{co}} + E_{\text{xp}}\hat{\mathbf{xp}}]e^{-j\frac{2\pi}{\lambda}z}. \tag{2.6}$$

The propagation direction $\hat{\mathbf{z}}$ is orthogonal to both $\hat{\mathbf{co}}$ and $\hat{\mathbf{xp}}$ which are generally orthonormal vectors [2, p. 24].

2.7 Aperture Size

The *aperture* of an antenna is the area where most of the radiated power passes through [2, p. 57]. In general it is difficult to exactly define the aperture of an antenna and some antenna types lack the concept completely.

The *effective aperture area* of an antenna in a given direction is

$$A_{\mathrm{e}} = \frac{P_{\mathrm{O}}}{W_t},$$

where P_{O} is the total accepted power by the antenna and W_t is the power density of a plane wave in the given direction [4, p. 13].

The max directivity of an antenna is directly proportional to the effective aperture area when the field of the antenna has a uniform distribution over the aperture [2, pp. 57, 240]. Max directivity can therefore be calculated as

$$D_{\max} = A_{\mathrm{e}} \frac{4\pi}{\lambda^2}.$$

This means that larger antennas allow us to create narrower, more focused beams, concentrating more of the power in one chosen direction.

3

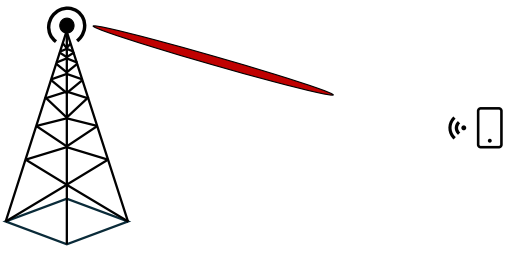
Antenna Arrays

This chapter aims to describe the fundamentals of antenna arrays and their use in beamforming. We begin with a description of the different beam types and their uses, before outlining the architecture of an antenna array. We then detail the use of antenna arrays in beamforming. Following that, we also introduce the metrics that are used to evaluate the quality of an optimized set of beam weights.

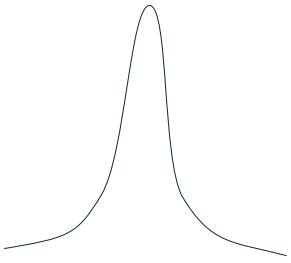
3.1 Beam Types and Uses

The main utility of antenna arrays lies in the ability to manipulate their electrical fields as desired through adjustment of the phases and amplitudes of the beam weights. In this way, antenna arrays allow for steering which directions they are radiating energy in. When this is used to radiate energy in a chosen direction, we refer to this peak in the radiation pattern as a *beam*. We refer to the manipulation of beams through, for example, the choice of beam weights, as *beamforming* [7, p. 2]. In 5G, antenna arrays are mainly used to create two types of beams, *traffic beams* and *wide beams* [7, pp. 2–3].

Traffic beams, or pencil beams, are beams with low side lobe levels and a single main lobe with small half-power beamwidth in both azimuth and elevation [4]. They have high max directivity, which is used to send and receive information from user equipment [7, pp. 2–3]. Figure 3.1a illustrates how a traffic beam communicates with a user device, and Figure 3.1b shows a representation of a cross-section of the radiation pattern of an ideal traffic beam.



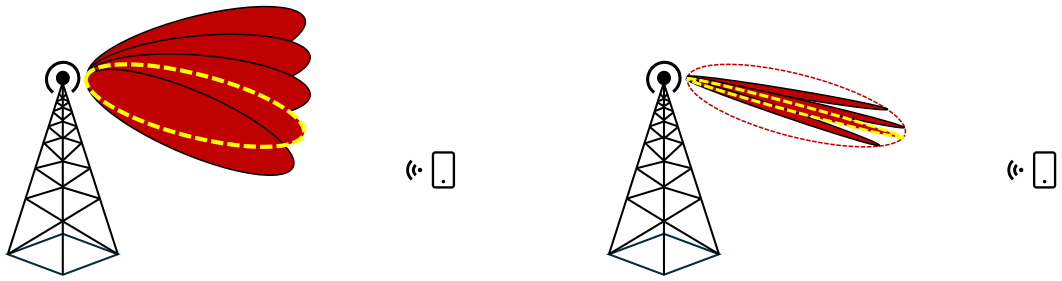
(a) Representation of a traffic beam conducting traffic to and from a user device.



(b) Representation of a cross section of the radiation pattern of an ideal traffic beam.

Each antenna array has a set of traffic beams that cover the entire angular range that the antenna serves, also known as the coverage angular range (CAR). The traffic beam with the strongest connection to the user equipment is used to provide service to the user. However, to choose which traffic beam to use, we need to know which creates the strongest signal. This can be done by testing each traffic beam one at a time and then choosing the traffic beam that created the strongest signal for the user. The problem with this approach is that there are many traffic beams available to an antenna, making it inefficient to test each one. To aid in this process, a second type of beam called a *wide beam* [7, p. 3] is used.

Wide beams are wider than traffic beams, so fewer are needed to cover the CAR. Each wide beam corresponds to a subset of the traffic beams. When searching for the best traffic beam to provide service to a user, each wide beam is tested to find the one that creates the strongest signal for the user. The traffic beams corresponding to the chosen wide beam are then tested to find the one with the strongest connection to the user. In this way, it is possible to find the best traffic beam to serve the user without having to check every single traffic beam. This process is illustrated in Figure 3.2.

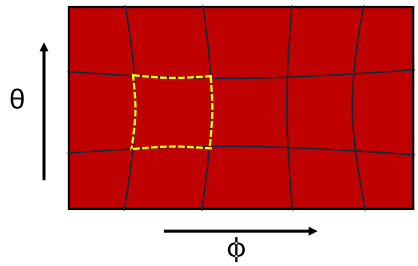


(a) The wide beams are tested to find the one with the best connection.

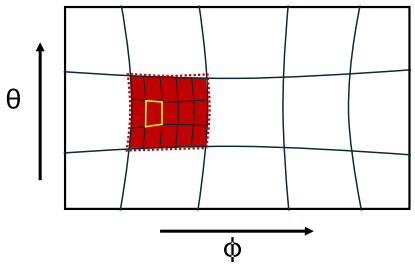
(b) The traffic beams corresponding to the best wide beam are tested to find the one with the best connection.

Figure 3.2: Wide beams are used to identify the optimal traffic beam with which to serve a user.

The wide beams are arranged in a grid that covers the CAR. This grid is known as a grid of beams (GoB), which can refer to either a grid of wide beams, or to traffic beams, which are also arranged in a grid. The process of selecting a traffic beam is also illustrated in Figure 3.3.



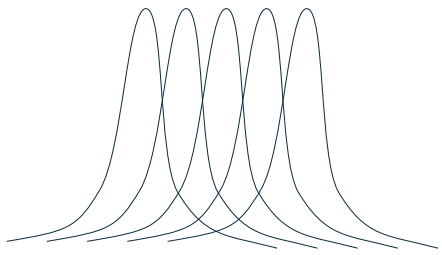
(a) The wide beam with the best connection to the user equipment is selected, and is outlined here in yellow.



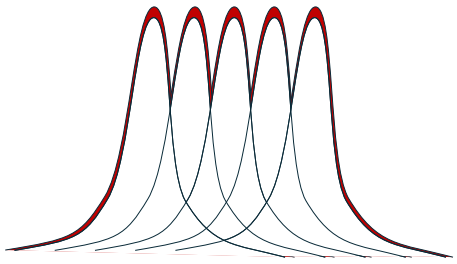
(b) The traffic beams corresponding to the best wide beam are tested to find the one with the best connection to the user equipment. The selected traffic beam is outlined here in yellow.

Figure 3.3: Wide beams are used to identify the optimal traffic beam with which to serve a user. The beams are arranged in a GoB that cover the CAR.

The desired wide beam shape is one that represents the strength of the signal for the corresponding traffic beams as accurately as possible. In the ideal case, the strength of the signal between a user and a wide beam is directly proportional to the strength of the signal between the user and the best of the traffic beams corresponding to that wide beam. For this reason, we build our desired wide beam radiation pattern, which will be referred to as the *reference field*, from the radiation patterns of the corresponding traffic beams. This is done by creating an *envelope* around these beams, which refers to taking the maximum over all the traffic beams at each point, as represented in Figure 3.4b.



(a) Representation of a cross section of several ideal traffic beams corresponding to a single wide beam.



(b) Representation of a cross section of the envelope of several ideal traffic beams corresponding to a single wide beam.

Figure 3.4: Creation of an envelope over a set of traffic beams.

The radiation pattern is a representation of the amount of energy that is sent in different directions. A traffic beam utilizes all the energy available from the antenna. For this reason, an envelope of several traffic beams creates a curve that implies that the antenna is emitting more energy than is available to it. This is rather improbable. For this reason, the envelope is divided by the square root of the number of traffic beams used to create the beam, which lowers the power to a level where the

energy output is not greater than the energy available to the antenna. This is the ideal wide beam shape.

It is possible to calculate the beam weights that create a traffic beam analytically. However, this does not work for wide beams due to their complicated shape. For this reason, optimization methods are employed to find beam weights that create a radiation pattern that is similar to that of a given reference field.

3.2 Antenna Arrays

As was shown in Section 2.7, larger antennas have higher directivity and for some uses, for example long-distance communication, much higher directivity than antennas of ordinary size can create is required. For this purpose, larger antennas or *antenna arrays* can be used. Antenna arrays consist of multiple *antenna elements* that are distributed over a surface, the aperture area, and thus antenna arrays with large apertures can be constructed. They are also useful in that they provide a large flexibility when shaping the radiation pattern [8, p. 285].

There are three different types of antenna arrays: linear, planar, and conformal. The type mainly discussed in this thesis is *planar antenna array*. The antenna elements of a planar array are spread out on a plane surface [2, p. 334]. The antenna elements can be any type of antenna or even several antennas per element, for example, two orthogonal half-wave dipole antennas.

3.2.1 Antenna Elements

As mentioned above, the structure of antenna elements can vary, but in the scope of this thesis we will work with antenna elements which consists of two orthogonal antennas. In this case, the antennas produce fields orthogonal to each other that propagate in the same direction.

In Figure 3.5 we present an example of an antenna element with two orthogonal antennas. This setup is called ± 45-polarisation, since both antennas are rotated 45° in opposite directions from the top of the antenna element. Another very common way to build such an antenna element is called H/V-polarisation and consists of one antenna rotated 90° and one rotated 0° from the top, i.e., one is horizontal and the other is vertical.

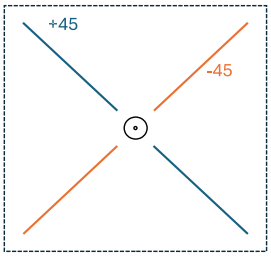


Figure 3.5: One antenna element consisting of two antennas rotated 45° in opposite directions.

In the ideal case, the electrical field of each antenna will consist of only one polarisation. We illustrate this using the antenna element in Figure 3.5. In the ideal case, the E_{+45} component of the field will be created entirely by the +45 antenna and the E_{-45} component will be created entirely by the −45 antenna. However, in reality, each antenna usually creates some low-level noise on its cross-polarisation.

3.2.2 Several Antenna Elements Make a Subarray

Several antenna elements can be arranged next to each other to create a *subarray*. Figure 3.6 shows an example of a subarray with 3 rows and 1 column. It is referred to as a 3×1-subarray. Each subarray has 2 *ports*. Each port corresponds to one polarisation, in this case either +45 or −45.

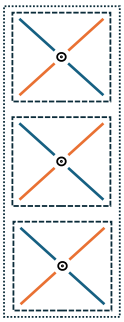


Figure 3.6: A 3×1-subarray consists of three antenna elements arranged into three rows and one column.

The excitations sent through the ports are called *beam weights* (BWs). These are the smallest steerable parameters that will be discussed in this thesis. Beam weights are complex valued, and when a beam weight is set to a non-zero value, the corresponding elements in that subarray create an electrical field. Multiplying the beam weight by some factor a will create an electrical field that is a times stronger, and adding a phase θ will create an electrical field that is shifted θ degrees in phase.

3.2.3 Several Subarrays Make an Antenna Array

An *antenna array* consists of a number of subarrays arranged in a grid. An example with 3 rows and 8 columns can be seen in Figure 3.7. An antenna array with m

rows and n columns has $2 \cdot m \cdot n$ ports, one for each polarisation for each subarray. By manipulating the beam weights, it is possible to steer the direction and modify the shape of the electrical field created by the antenna array. Each port has a corresponding beam weight, so there are $2 \cdot m \cdot n$ beam weights to choose for an $m \times n$ antenna array.

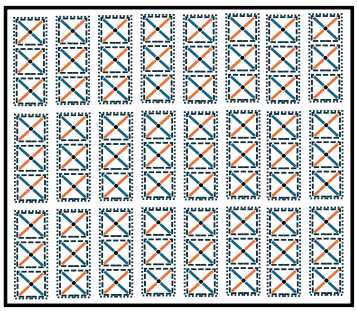


Figure 3.7: A 3×8 antenna array consists of 24 subarrays arranged into 3 rows and 8 columns.

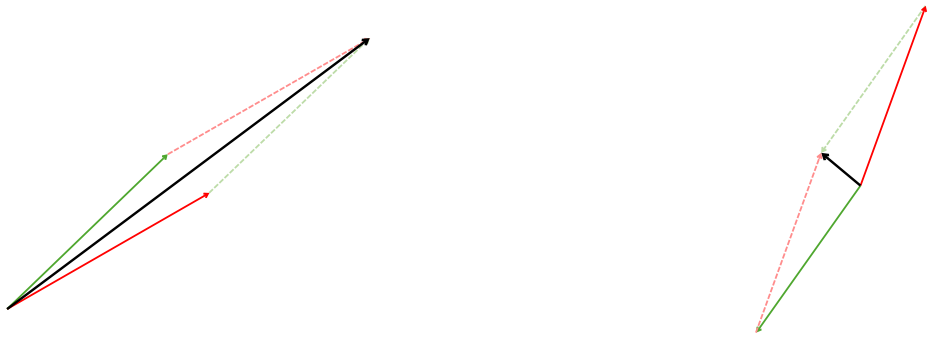
3.3 The Superposition Principle

Electric fields are vector fields and can therefore be combined according to the superposition principle. When there are several sources creating electrical fields in the same space, the total electric field at each point is the sum of the vectors of the different electrical fields in that point [2, p. 135]. The total electrical field $\mathbf{E}_{\text{tot}}$ can therefore be written as

$$\mathbf{E}_{\text{tot}}(\mathbf{x}) = \sum_{j \in S} \mathbf{E}_j(\mathbf{x}) \tag{3.1}$$

where S is the set of sources, $\mathbf{E}_j$ is the electrical field created by the source $j \in S$, and $\mathbf{x}$ is any point in the space.

If the vectors are pointing in a similar direction, the resulting vector will be longer than either, while if the phase difference is large, the resulting vector will be shorter than both. This is called *constructive* and *destructive interference* respectively, and is illustrated in Figure 3.8.



(a) Constructive interference. **(b)** Destructive interference.

Figure 3.8: Constructive and destructive interference between two vectors (green and red arrows respectively). The black arrow represents the resulting vector.

3.4 Beamforming with Antenna Arrays

For each port on an antenna array, an electrical field is created when the corresponding beam weight is set to one, and all other beam weights are set to zero. The resulting far field function is called an *embedded subarray pattern* (ESAP). As there is one ESAP for each port, an $m \times n$ antenna array has $2 \cdot m \cdot n$ ESAPs. For each ESAP we can calculate the resulting far field function when that same beam weight is instead set to $re^{i\theta}$, and all other beam weights are still set to zero, by multiplying the ESAP with $re^{i\theta}$. This is done by multiplying the amplitude in each point by r and adding θ to the phase of the vector in each point, as illustrated in Figure 3.9a. By summing over the products of each ESAP with its corresponding beam weight, we get the resulting far field function $\mathbf{G}_A(\theta, \varphi)$ of the antenna array, which we can write as

$$\mathbf{G}_A(\theta, \varphi) = \sum_{j \in S} bw_j \cdot ESAP_j(\theta, \varphi), \tag{3.2}$$

where bw_j is the beam weight corresponding to $ESAP_j(\theta, \varphi)$, and S is the set of ports. This calculation is illustrated in Figure 3.9b. By shifting the phases of different beam weights we can steer where we will see constructive and destructive interference in the far field, and we can affect the strength of the field in different points by adjusting the amplitudes of the different beam weights. In this way, we are able to create desired radiation patterns by manipulating the beam weights [7, p. 2].

(a) The field created by one polarisation of a subarray is the embedded subarray pattern multiplied with the beam weight.

(b) The field created by an antenna array is the sum of the fields created by each polarisation of each subarray.

Figure 3.9: Representation of how the electrical field is calculated from the ESAPs and BWs.

3.5 Single- and Dual-Polarized Beamforming

In Section 3.2.2 we stated that each subarray of an antenna array has two ports, one for each polarisation. Sometimes we choose to optimize the beam weights in such a way that the two ports for a subarray are assigned the same beam weight. This is called *single polarized beamforming* and gives us $m \cdot n$ complex variables to optimize for an antenna array of size $m \times n$. The alternative is called *dual polarized beamforming* which means that we optimize individual beam weights for each port. This results in $2 \cdot m \cdot n$ complex variables to optimize. These two approaches have different strengths and weaknesses. With single polarized beamforming there are half as many beam weights that need to be optimized, which can reduce the runtime of an optimization algorithm substantially depending on the time complexity of the optimization. On the other hand, dual polarized beamforming has more degrees of freedom which might allow the algorithm to find a more exact solution.

3.6 Metrics

In this section, we introduce three metrics used when evaluating the quality of a choice of beam weights. For some of these metrics, we need a reference field to compare the resulting radiation pattern against. We assume that the far field function are measured in discrete points, at every whole degree of θ and φ, as this is a standard format for measured ESAPs.

3.6.1 Taper Loss

Taper loss is a measure of how much of the maximum available power of an antenna is utilized for a particular set of beam weights. This is calculated from the amplitudes of the beam weights. Each beam weight has a maximal amplitude constrained by the amount of energy available per port. This implies that when the amplitudes of all beam weights are set to the maximum allowed value, the antenna utilizes all available power. In this case, the taper loss is zero. It is usually assumed that we

are attempting to create the strongest possible beam with the power available to the antenna array. For this reason, we normalize the beam weights and assume that the largest amplitude is set to the largest allowed value. To account for this normalization, we divide each beam weight amplitude by the largest beam weight amplitude in the taper loss equation, effectively performing the normalization as a part of the taper loss calculation in case the beam weights have not been normalized.

The taper loss is defined as

$$\mathscr{L}_T := -10 \cdot \log_{10}\Big(\sum_{n=1}^{N}\Big(\frac{|bw_n|}{\max_n |bw_n|}\Big)^2/N\Big),$$

where N is the number of beam weights and bw_n is the beam weight corresponding to the nth ESAP.

3.6.2 Main Lobe Loss

One very important loss is the *main lobe loss* which measures the difference between the field calculated from the investigated beam weights, and the reference field, by taking the mean square error (MSE) of the radiation pattern of both fields in the EIRP scale. However, the fields are not compared in all indices, but only in the *main lobe area* $\mathcal{F}_{\text{main}}$. The main lobe loss is illustrated in cross-section in Figure 3.10. The total main lobe loss is the volume between the reference beam and optimized beam. Note that the figure is in a logarithmic scale and that the side and main lobe thresholds are lower than they might appear in the figure.

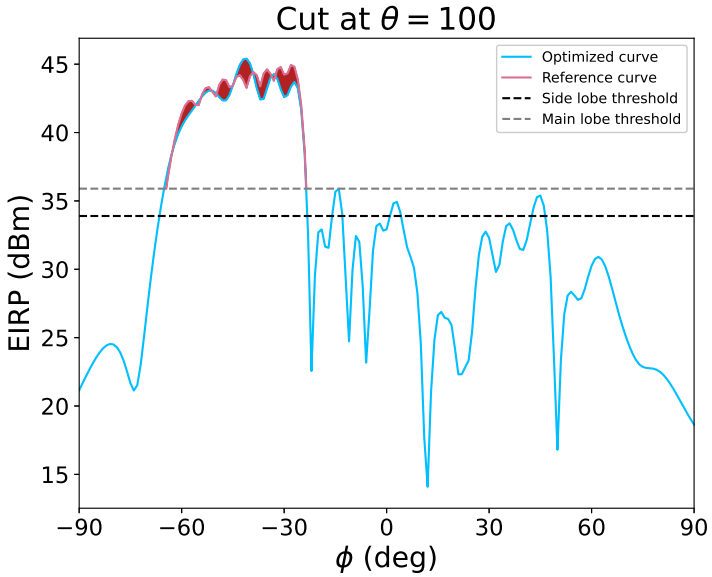
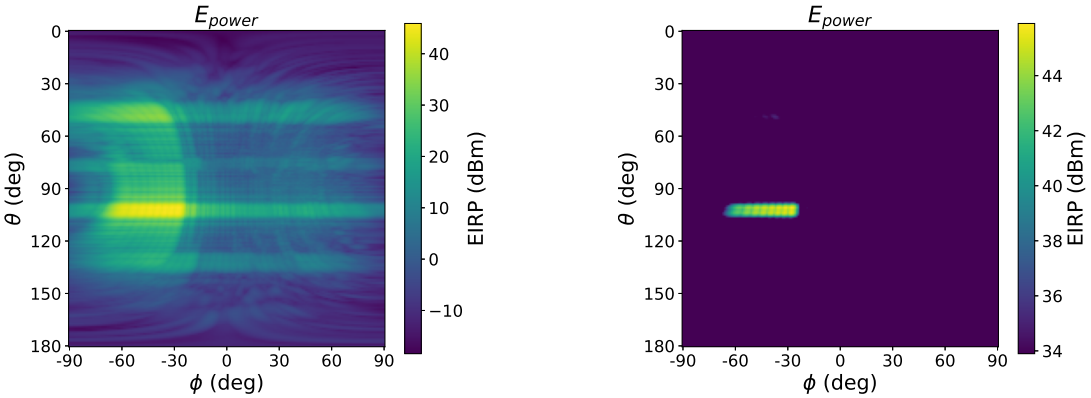


Figure 3.10: Cut of an example beam. The pink line is the reference field in the main lobe area and the blue line is the optimized field. The black and grey dotted lines are the side lobe and main lobe thresholds, respectively. The red area represents the main lobe loss.

The first step of calculating the main lobe area is finding the highest value of the radiation pattern of the reference field, $U_{\max}$, and its corresponding index, $(i_{\max}, j_{\max})$. Then let P be the set of all indices of the reference field that have a radiation intensity higher than $U_{\max} - 10$ [dBm], which is referred to as the *main lobe threshold*. We then define the main lobe area $\mathcal{F}_{\text{main}}$ to be the largest connected subset of S that contains the index $(i_{\max}, j_{\max})$. For more details, see the implementation in Section 5.3.2.



(a) An examplatory wide beam.

(b) The main lobe of the examplatory wide beam.

Figure 3.11: An example of the radiation pattern of a wide beam and its main lobe.

The main lobe loss is then defined as

$$\mathscr{L}_M := \frac{1}{|\mathcal{F}_{main}|} \sum_{\theta,\varphi \in \mathcal{F}_{main}} \left([U_{\mathbf{G}_A}(\theta,\varphi)]_{dBm} - [U_R(\theta,\varphi)]_{dBm} \right)^2,$$

where $U_{\mathbf{G}_A}(\theta,\varphi)$ is the radiation pattern of the antenna array and $U_R(\theta,\varphi)$ is the radiation pattern of the reference field.

3.6.3 Side Lobe Loss

A side lobe is an unwanted peak in the radiation pattern of an antenna array. They are unwanted because they leak energy in undesirable directions, which wastes energy, and can also interfere with other antennas. The side lobe loss is a measure of how much the side lobes exceed the *side lobe threshold*, $U_{\max} - 12$ [dBm]. If no side lobes exceed the threshold the loss is zero. The side lobe loss is illustrated in cross-section in Figure 3.12. The total side lobe loss is the volume between the optimized beam and the side lobe threshold. Note that the figure is in a logarithmic scale and that the side and main lobe thresholds are lower than they might appear in the figure.

Similar to the main lobe loss, this is not calculated for all indices, only the *side lobe area* $\mathcal{F}_{\text{side}}$. We define this set as the complement set of $\mathcal{F}'_{\text{main}}$, which is calculated in

almost the same way as $\mathcal{F}_{\text{main}}$ in Section 3.6.2. The only difference is that we instead use $U_{\max} - 12$ [dBm] as the threshold. For more details, see the implementation in Section 5.3.2.

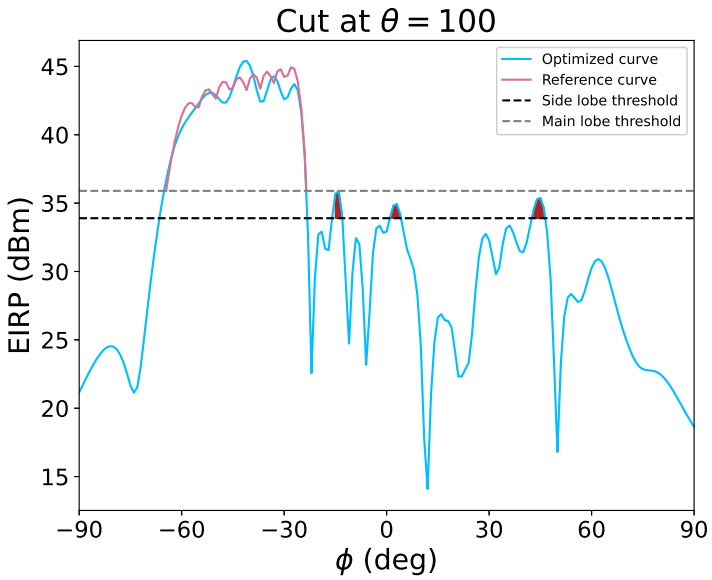


Figure 3.12: Cut of an example beam. The pink line is the reference field in the main lobe area and the blue line is the optimized field. The black and grey dotted lines are the side lobe and main lobe thresholds, respectively. The red area represents the side lobe loss.

The side lobe loss is then defined as

$$\mathscr{L}_S := \sum_{\theta,\varphi \in \mathcal{F}_{side}} \max\{0, [U_{\mathbf{G}_A}(\theta,\varphi)]_{dBm,} - [U_{\max} - 12]_{dBm}\}.$$

Note that since we are working in a logarithmic scale, the side lobe threshold, $U_{\max} - 12$ [dBm], would be approximately $\frac{1}{16} U_{\max}$ in a linear scale.

4
Optimization

In the field of optimization, one studies methods for finding the *optimum* of a function. The optimum is the best possible value, often defined as the point within a set search space where the function achieves either its maximum or minimum value. There are many different approaches to solving optimization problems, the oldest and most well-studied branch being *classical optimization* [9, pp. 1–2]. However, the main focus of this thesis will be methods from a separate branch, namely *stochastic optimization*. In this chapter, we begin with a brief introduction to these two branches, before detailing the specific algorithms that will be used in this thesis.

4.1 Classical Optimization

Classical optimization methods are deterministic algorithms for solving optimization problems where the objective function can be written as a well-defined mathematical function. Most classical optimization algorithms are poorly suited to non-convex problems, and problems where the objective function is highly non-linear or of high dimension. For problems of these types, it is often prudent to turn to other branches of optimization, such as stochastic optimization [9, pp. 1–2, 33].

4.2 Stochastic Optimization

Stochastic optimization methods are, in contrast to classical optimization methods, never deterministic. They have elements of randomness in them, though this does not mean they are completely without rhyme or reason. Many stochastic optimization methods are inspired by phenomena in nature, such as foraging behaviour of ant colonies, swarms of animals, and evolution. From these phenomena, principles such as cooperation and adaption, which can be regarded as a type of optimization, have been taken and converted to mathematical equations to be used in optimization. Two stochastic methods inspired by nature are *genetic algorithms* (GA) and *particle swarm optimization* (PSO) [9, pp. 2–4].

4.2.1 Genetic Algorithms

Genetic algorithms (GA) are a type of *evolutionary algorithms*, which are stochastic optimization methods inspired by the biological phenomena of evolution as described by Charles Darwin [9, p. 35]. Similar to how a gradual change of a population over generations can increase individuals' chances to survive and reproduce, i.e. increase

their fitness, we want to get closer and closer to an optimum over the iterations of an optimization algorithm.

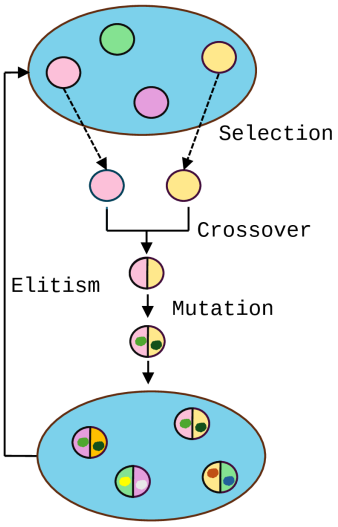


Figure 4.1: Visualization of a generation of GA with its main operations. The blue ovals are populations, and the smaller circles are individuals.

The first step of genetic algorithms is to define an encoding of solutions to *chromosomes*. Next, a *population* is created, which consists of several chromosomes. Then, a *fitness function* is defined. Its purpose is to evaluate each chromosome. The resulting value is referred to as a chromosome's fitness. A common way to define the fitness function is $F(x) = \frac{1}{\mathscr{L}(x)}$, where $\mathscr{L}(x)$ is an objective function which we want to minimize. The main part of genetic algorithms consists of manipulating chromosomes to increase their fitness over *generations*. A new generation is created from the previous generation using three operations. The first operation is called *crossover*, which creates new chromosomes using parts of current chromosomes. Secondly, there is *mutation* which makes stochastic changes to elements of the chromosomes, also known as *genes*. Lastly, there is *elitism*, which selects chromosomes for the new generation based on their fitness. Chromosomes with higher fitness are more likely to be selected.

There are many ways the three operations; crossover, mutation, and elitism, can be implemented. For this project we used the operations as defined by M L Shahab, F Azizi, B A Sanjoyo, *et al.* [10] which are modified to work well in high dimensions. Algorithm 1 is a pesudocode for this genetic algorithm, and Figure 4.1 presents a simple representation of a generation of GA. The operations are described in more detail in the following sections.

Algorithm 1 Genetic Algorithm for High Dimensional Problems

```
Define a chromosome representing the solution
Define fitness function F(x)
Generate an initial population consisting of m chromosomes
Initialize the maximum number of generations n
for t = 1,2...,n do
    for i = 1,2...,m do
        Generate child by Crossover
        Update obtained child by Mutation
    end for
    Elitism
end for
```

Crossover

The first step of crossover is selecting two chromosomes from the population. The method used is tournament selection with size n and tournament probability p_{tour}. Thus, n random chromosomes are chosen from the population, and the one with the highest fitness is chosen with probability p_{tour}. If it is not chosen it is discarded, and the same process is repeated recursively until one individual is chosen or there is only one left. In that case, the last remaining one is chosen. This process is executed twice, so that two individuals are chosen in total.

Genes from the two chosen chromosomes are then mixed as shown in Algorithm 2 to create a new chromosome. The new chromosome always has the same or higher fitness than its parent chromosomes due to the construction of the algorithm.

Algorithm 2 Crossover

$x \leftarrow$ Chromosome of length N chosen via tournament selection
$y \leftarrow$ Chromosome of length N chosen via tournament selection
$x^c \leftarrow$ Copy of x
for $i = 1, 2, ..., N$ **do**
    $x^{(1)} \leftarrow x^c$
    $x^{(2)} \leftarrow x^c$ with value at index i set to y_i
    $x^{(3)} \leftarrow x^c$ with value at index i set to $\frac{y_i + x_i}{2}$
    $j \leftarrow \text{argmax}_{j \in \{1,2,3\}} \{F(x^{(j)})\}$
    $x^c \leftarrow x^{(j)}$
end for
return x^c

Mutation

We perform mutation on the chromosomes returned by crossover with mutation probability set to one, which means each index is mutated. The process of mutation is described in Algorithm 3. The mutated chromosome is always better than the

original. Mutation is important since it contributes to chromosomes moving out of local optima of the fitness function and adds genetic diversity to the population.

Algorithm 3 Mutation

$n \leftarrow$ The number of the current generation
$x \leftarrow$ Chromosome of length N created using Crossover
$x^c \leftarrow$ Copy of x
for $i = 1, 2, ..., N$ **do**
  $r \leftarrow$ random value sampled from uniform distribution on interval $[-\frac{1}{2n}, \frac{1}{2n}]$
  $x^{(1)} \leftarrow x^c$
  $x^{(2)} \leftarrow x^c$ with value at index i set to $x_i + r$
  $x^{(3)} \leftarrow x^c$ with value at index i set to $x_i - r$
  $j \leftarrow \text{argmax}_{j \in \{1,2,3\}}\{F(x^{(j)})\}$
  $x^c \leftarrow x^{(j)}$
end for
return x^c

Elitism

The version of elitism used in this project is called "Elitism replacement with filtration" and was introduced by M L Shahab, F Azizi, B A Sanjoyo, *et al.* [10]. It chooses the next generation as the m chromosomes with the highest fitness out of both the last generation and the m newly generated chromosomes. However, we first remove duplicate chromosomes and generate random chromosomes if there are in total less than m chromosomes left. This is to counteract potential “inbreeding”, i.e., if the chromosomes become too similar, by removing duplicates and introducing new genes.

4.2.2 Particle Swarm Optimization

We also investigated particle swarm optimization (PSO). PSO is a type of stochastic optimization algorithm that mimics the behaviour of a swarm of animals or insects. Similar to the population in GA, PSO has a group of possible solutions, which we refer to as a *swarm*. Each solution in the swarm is referred to as a *particle*. The particles in a swarm are able to share information about where they have found the lowest loss values in the search space. The loss of a particle x is calculated using the objective function $f(x)$. Each particle stores information about the position with the lowest loss that it has found, as well as the position with the lowest loss that any member of the swarm has found. This is visualized in Figure 4.2.

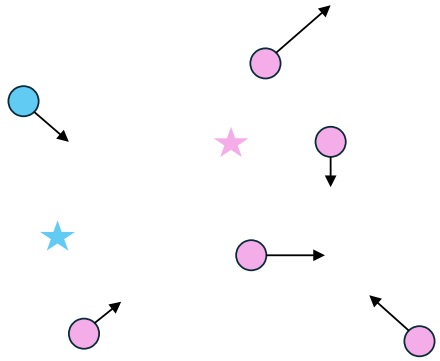


Figure 4.2: Visualization of a swarm of particles and their velocities. The blue star represents the current best position of the blue particle and the pink star represents the current best position of the entire swarm.

The first step in the algorithm is to initialize the swarm with all its particles. Each particle is initialized at a random position in the search space, with a random velocity. The search space is assumed to be a hyperrectangle defined by the maximal and minimal values $x_{j,\max}$ and $x_{j,\min}$ for each dimension j.

In each iteration, each particle i updates its position $\mathbf{x}_i$ based on the position of their personal best $\mathbf{x}_i^{pb}$, and the position of the swarm best x^{sb}. How much the particles steers toward these points is determined by two positive constants: the cognition factor c_1 which affects how much the particles move toward their personal bests, and the social factor c_2 which affects how much the particles steer toward the swarm best.

Next, the velocity $\mathbf{v}_i$ is checked to ensure that it does not exceed a set speed limit $v_{\max}$. If it does, the vector is normalized to be at the speed limit. Then the position of the particle is calculated from the current position of the particle and its updated velocity together with the time step Δt. This new position is then evaluated, and if the loss is lower than its personal best, then the particle's personal best is updated. If the loss is lower than the swarm best, then the swarm best is updated as well.

A common version of PSO is PSO with inertia, where inertia ω is a nonnegative constant that affects how much of the previous velocity is kept in the velocity update. The pseudocode for this PSO variant is presented in Algorithm 4.

Algorithm 4 Particle Swarm Optimization with Inertia

Initialize particles with random positions and velocities

$x_{ij} \leftarrow x_{j,\min} + U[0,1](x_{j,\max} - x_{j,\min}), \quad i = 1, ..., n, j = 1, ..., n$

$v_{ij} \leftarrow \frac{\alpha}{\Delta t}(-\frac{x_{j,\max} - x_{j,\min}}{2} + U[0,1](x_{j,\max} - x_{j,\min})), \quad i = 1, ..., n, j = 1, ..., n$

for $k = 1, 2, ..., K$ **do**

Evaluate each particle and update their personal best and the swarm best

$\mathbf{x}_i^{pb} \leftarrow \mathbf{x}_i$ if $f(\mathbf{x}_i) < f(\mathbf{x}_i^{pb}), \quad i = 1, ..., n$

$\mathbf{x}^{sb} \leftarrow \mathbf{x}_i$ if $f(\mathbf{x}_i) < f(\mathbf{x}^{sb}), \quad i = 1, ..., n$

Update each particle's velocity

$v_{ij} \leftarrow \omega v_{ij} + c_1 r_{1j} \left(\frac{x_{ij}^{pb} - x_{ij}}{\Delta t} \right) + c_2 r_{2j} \left(\frac{x_j^{sb} - x_{ij}}{\Delta t} \right), \quad i = 1, ..., n, j = 1, ..., m$

Restrict the velocities to the speed limit

$\mathbf{v}_i \leftarrow \frac{v_{max}}{|\mathbf{v}_i|} \mathbf{v}_i$ if $|\mathbf{v}_i| > v_{max}, \quad i = 1, ..., n$

Update each particle's position

$\mathbf{x}_i \leftarrow \mathbf{x}_i + \mathbf{v}_i \Delta t, \quad i = 1, ..., n$

end for

return $\mathbf{x}^{sb}$

Here, r_1 and r_2 are random numbers between 0 and 1.

4.3 Bayesian Optimization

Bayesian optimization is a non-parametric optimization method suitable for problems where the objective function acts as a black-box function that is very expensive to evaluate. It attempts to find the point that minimizes the objective function using as few evaluations as possible by strategically choosing the next point to evaluate in each iteration [11]. Here we present a brief overview of the optimization method and introduce pseudocode, before discussing the main elements of the method in more detail.

The method uses a Gaussian process as a surrogate model for the objective function. The surrogate model attempts to model the behaviour of the objective function as accurately as possible. The method also uses an acquisition function. For each point in the domain, the acquisition function estimates the value gained from sampling the objective function in that point. This value is usually defined in terms of the expected amount of information gained and the value of the surrogate function in that point. In each iteration, the acquisition function is maximized to find the optimal point to sample in. Next, the objective function is evaluated in the chosen point, and the result is used to update the surrogate model using Gaussian process regression [11]. Algorithm 5 summarizes Bayesian optimization in pseudocode as presented by Frazier [11].

Algorithm 5 Bayesian Optimization

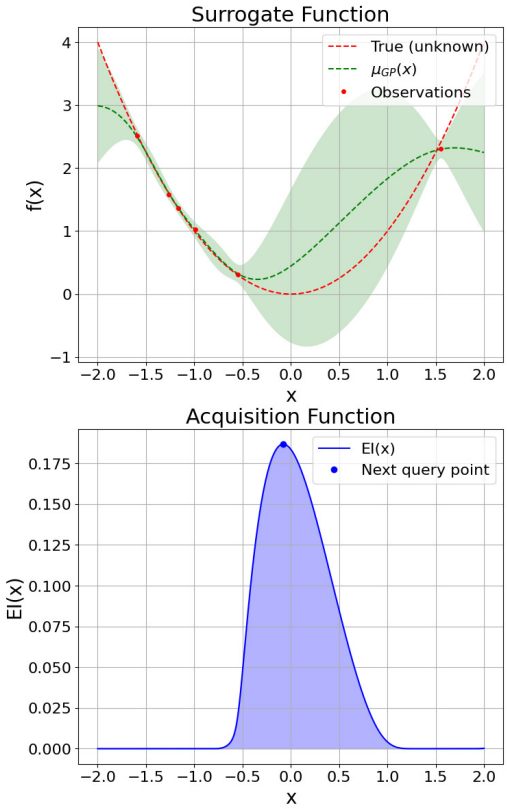
Place a Gaussian prior on f
Observe f at n_0 points according to an initial space-filling experimental design.
Set $n = n_0$
while $n \leq N$ **do**
  Update the posterior probability distribution on f using all available data.
  Let x_n be a maximizer of the acquisition function over x, where the acquisition function is computed using the current posterior distribution.
  Observe $y_n = f(x_n)$.
  Increment n
end while
Return a solution: either the point evaluated with the largest $f(x)$, or the point with the largest posterior mean.

Figure 4.3 illustrates an example of how one iteration of the algorithm affects the surrogate model and the acquisition function. Observe how an additional observation is added in the point that maximized the acquisition function and that this reduces the uncertainty in the surrogate model and improves the estimated mean so that it is now closer to the true value. Note that this is a toy example and that in most cases we would not have access to the true shape of the function or any of its derivatives when performing Bayesian optimization.

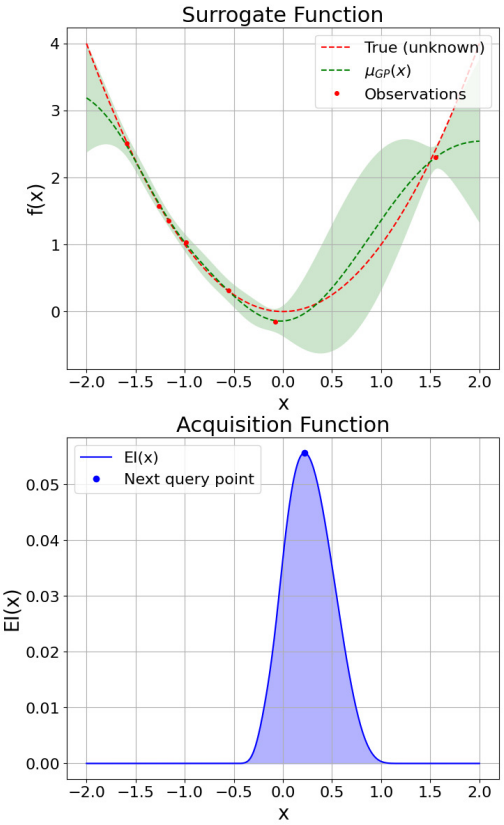
4.3.1 Gaussian Process Regression

A Gaussian process is defined as a collection of random variables, of which any finite collection has a multivariate Gaussian distribution. Each of these random variables is tied to a point in some domain D, often time or space. The Gaussian process is completely defined by its mean function $\mu(\mathbf{x})$ and its covariance function $\Sigma(\mathbf{x}, \mathbf{x}')$. We often choose $\mu_0(\mathbf{x}) = 0$ for all $\mathbf{x} \in D$ as our prior. The prior covariance matrix $\Sigma_0(\mathbf{x}, \mathbf{x}')$ is chosen to reflect our beliefs about the covariance between different points and is sometimes referred to as a *kernel function* [12].

The kernel function is chosen to encode our beliefs about the correlation between different points in the domain based on the distance between them. A common assumption is that points that are closer should have a higher correlation. This would mean that when we evaluate the objective function in a point x' and use this to update the surrogate model, points that are close to x' in the domain are affected more by the update. If we know that the function is periodic with a known period λ, then we can choose a kernel function that represents this, so that when we update the prior in a point x', the prior is most affected in the points $x' + n\lambda$ for any $n \in \mathbf{Z}$.



(a) Surrogate model with the true unknown model (top) and acquisition function (bottom) after updating the prior with 6 random points.



(b) Surrogate model with the true unknown model (top) and acquisition function (bottom) after 1 iteration of Bayesian optimization.

Figure 4.3: The mean of the surrogate function is shown as a dashed green line, and the area within 1.96 standard deviations from the mean of the surrogate model is shown shaded in green. The mean of the true function is shown as a dashed red line, and all sampled points are shown as red dots. The acquisition function is shown in blue, with a blue dot denoting the next query point.

We assume that the prior placed on the mean is $\mu_0(\mathbf{x}) = 0$ and that σ_0 is constant over D. Then, given a kernel function $\Sigma_0(\mathbf{x}_1, \mathbf{x}_2)$ and a set of new data points $(\mathbf{x}_{1:n}, y_{1:n})$, where $\mathbf{x}_{1:n}$ denotes a set of n points and $y_{1:n}$ denotes the corresponding objective function values, we update the mean and variance functions according to Rasmussen and Williams [12] in the following way:

$$\mu_n(x) = \Sigma_0(x, x_{1:n}) \left(\Sigma_0(x_{1:n}, x_{1:n}) + \sigma_0^2 I\right)^{-1} f(x_{1:n})$$
$$\sigma_n^2(x) = \Sigma_0(x, x) - \Sigma_0(x, x_{1:n}) \left(\Sigma_0(x_{1:n}, x_{1:n}) + \sigma_0^2 I\right)^{-1} \Sigma_0(x_{1:n}, x).$$

4.3.2 Acquisition Function

The acquisition function usually reflects some combination of the uncertainty in the surrogate model and the expected mean. In Figure 4.3 we use expected improvement

(EI) as our acquisition function. The improvement from sampling at a point is 0 if the objective value y' is higher than or equal to the lowest value we have found so far, $y_{\min}$, and the improvement is $y_{\min} - y'$ if $y' < y_{\min}$. Since we cannot know what the improvement will be at any point without sampling there, we instead calculate the expected improvement. Therefore, the expected improvement prioritizes points where the surrogate model has a low mean but a high variance, since points with high variance carry more uncertainty and could potentially be much better than the expectation [11].

5
Method

In this chapter, we specify the goals of this thesis in more detail. We then proceed to give an overview of the algorithms and detail the design choices made during the development of the central beam weight optimization methods of this thesis.

5.1 Specification of the Issue Being Investigated

The main objective of this thesis is to create an algorithm for optimizing beam weights such that the corresponding radiation pattern minimizes the objective function, which we defined as a linear combination of the main lobe loss and the side lobe loss. The algorithm should take as input; the embedded subarray patterns of the antenna for which the beam weights should be optimized, the corresponding traffic beam beam weights, and the indices of the traffic beams to be used in the construction of the reference field. The reference field is a theoretical beam that is constructed from traffic beams as described in Section 3.1.

The total list of desiderata for the final algorithm is stated in more detail here. The algorithm should:

- Consistently produce beam weights that correspond to a radiation pattern with desired metric values with respect to a given reference field.
  - The metrics used to evaluate beam weights are main lobe loss, side lobe loss, and taper loss, as introduced in Section 3.6.
  - The desired metric levels are based on the results of the current method.
- Run and converge in a shorter time than the current methods.
  - Current methods need approximately 1-2 hours per beam weight optimization.
  - The algorithm should also have a reasonable time complexity with regard to the dimension of the problem.
- Produce results that are robust against small variations resulting from the manufacturing of antenna and measurement components.
  - These variations may include noise in the ESAPs and rounding errors for beam weights applied in real situations.
- Optimize beam weights using measured and simulated realistic ESAPs.
  - This is in contrast to current methods which are only able to handle ideal ESAPs.

5.2 Algorithm Overview

In Figure 5.1 a flowchart illustrates the most important steps of the algorithm between the input data (ellipses) and the output (circle).

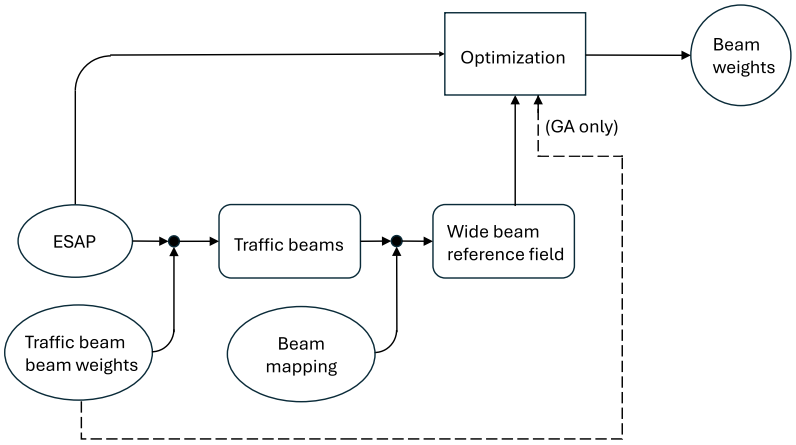


Figure 5.1: A flowchart of the method from data and user inputs to optimized beam weights. The dotted line only applies when GA is the algorithm used in the optimization step and represents the use of traffic beam BWs as warm starts.

The ESAPs and traffic beam BWs are used to calculate the radiation pattern of the traffic beams. The beam mapping is a list of the indices of the traffic beams to be used in the construction of the reference field. The traffic beam radiation patterns and the beam mapping are used to calculate the reference field, as described in Section 3.1. The ESAPs and the reference field are then passed to the optimizer, which returns a set of optimized beam weights. The optimizer minimizes the objective function, which we define in Section 5.3.

5.3 Design of Objective Function

The objective function evaluates the quality of a set of beam weights and is composed of two components, the side lobe loss and the main lobe loss, which are both calculated using a chosen reference field. Originally, the objective function consisted of the three metrics introduced in Section 3.6 which are commonly used at Ericsson for beamforming, for example in the master thesis written at Ericsson in 2024 by Frilund and Olofsson [1]. However, due to a difference in the choice of scale compared to Frilund and Olofsson [1], the taper loss component was rendered unnecessary and could be omitted, see Section 5.3.1. Thus, the final objective function $\mathscr{L}(bw)$, is a weighted sum of the main lobe loss and the side lobe loss,

$$\mathscr{L}(bw) = \alpha_1 \mathscr{L}_M(bw) + \alpha_2 \mathscr{L}_S(bw),$$

where $\mathscr{L}_M(bw)$ is the main lobe loss and $\mathscr{L}_S(bw)$ is the side lobe loss of the beam weights bw, and α_1 and α_2 are positive constants. The exact values of α_1 and α_2

are omitted from this report for confidentiality reasons. $\mathscr{L}(bw)$ is henceforth what we refer to as the total loss. In the case of GA, the fitness function is defined as $F(bw) = \frac{1}{\mathscr{L}(bw)}$.

5.3.1 Taper Loss

When working in a scale that is relative to the used energy, such as realized gain, it is necessary to have some separate measure of the amount of radiated energy. For this reason, previous algorithms, such as the one by Frilund and Olofsson [1], include taper loss as part of the objective function. However, our algorithms perform the optimization in EIRP scale which is an absolute scale that also measures the radiated energy, and thus the taper loss becomes a superfluous component of the objective function, and we can remove it.

5.3.2 Main and Side Lobe Areas

The definitions of the main and side lobe areas mentioned in Section 3.6 have the restriction that the area must be connected, as without that restriction a side lobe of the reference field with a high amplitude might be considered part of the main lobe area. Earlier implementations, such as Frilund and Olofsson [1], did not impose such restrictions, and thus we had to devise the following method to find those areas.

We assume that the radiation intensity of the antenna array, $U_{\mathbf{G}_A}(\theta, \varphi)$, is measured at the points (θ_i, φ_j), $i = 1, .., n_\theta$, $j = 1, .., n_\varphi$, where n_θ and n_φ is the size of the set of θ and φ values, respectively. Let $\mathcal{M}$ be a matrix of size $n_\theta \times n_\varphi$, where each index corresponds to a point at which $U_{\mathbf{G}_A}(\theta, \varphi)$ is measured. We define

$$\mathcal{M}_{ij} = \begin{cases} 1, & \text{if } U_{\mathbf{G}_A}(\theta_i, \varphi_j) > U_{\max} - 10, \\ 0, & U_{\mathbf{G}_A}(\theta_i, \varphi_j) \leq U_{\max} - 10, \end{cases} \tag{5.1}$$

where $U_{\max}$ is the peak of the radiation pattern of the reference field. In other words, $\mathcal{M}_{ij}$ is set to one if the corresponding point in $U_{\mathbf{G}_A}(\theta_i, \varphi_j)$ is larger than the main lobe threshold and to zero otherwise. Therefore, there is a large connected area of ones around the index corresponding to $U_{\max}$, i.e. where our main lobe is, and potentially other smaller areas of ones as well. We now want to remove all smaller connected areas such that only the one corresponding to the main lobe is left.

Convert $\mathcal{M}$ to a graph where $\mathcal{M}_{ij}$ is a neighbour to all values in $\{\mathcal{M}_{i+1,j}, \mathcal{M}_{i-1,j}, \mathcal{M}_{i,j+1}, \mathcal{M}_{i,j-1}\}$ that exist. If (i, j) is an index at the edge of $\mathcal{M}$ then some of the four neighbours won't exist. On this graph we perform a modified breadth first search starting in the index which corresponds to $U_{\max}$. The breadth first search only cares about the indices which have value one, and ignores indices with value zero, and thus we never explore outside the main lobe.

The process of finding the side lobe area is similar but not identical. Each element of $\mathcal{M}$ is set to one if the corresponding point of the radiation pattern is instead larger than the side lobe threshold, $U_{\max} - 12$. The breadth first search is performed

the same as for the main lobe area, and then the side lobe area can be found as the complement set of the chosen indices.

5.4 Optimization Algorithms

Many optimization algorithms were considered for this project, but the ones chosen were genetic algorithms and particle swarm optimization. We mainly investigated stochastic optimization methods as they generally perform well for problems with complex objective functions, and there are a plethora of methods with different strengths and weaknesses that can be utilized. One reason we chose GA and PSO specifically was due to them being successfully applied to similar problems such as by Boeringer and Werner [13] who used both PSO and GA for optimizating beam weights and Duzel, Saoud, Shayea, *et al.* [14] who made a comprehensive review of the use of GA to optimize the phase of beam weights. The version of GA we have used was designed by M L Shahab, F Azizi, B A Sanjoyo, *et al.* [10] to perform better in high dimensions, which is of import since our thesis concerns specifically large antenna arrays.

5.5 Search Space

The search space of an optimization problem is the subset of all feasible solutions that we allow the optimizer to explore. Therefore, the dimension of our search space is the number of parameters that we need to optimize, which depends on several factors. These include the size of the antenna, the rank of the beam weights, which we discuss in Section 5.5.1, and whether we are performing single- or dual polarized beamforming. For full rank dual polarized beamforming we need to optimize two $m \times n$ matrices of complex beam weights. Furthermore, we regard the phase and amplitude of each beam weight as separate variables during optimization, and therefore, have a total of $4 \cdot m \cdot n$ values to optimize. For single-polarized beamforming the dimension instead becomes $2 \cdot m \cdot n$.

Another option is to represent them as the real and imaginary part, however, that would have complicated certain calculations, as we wanted the option to constrain the amplitudes, and in certain cases keep the amplitudes fixed during optimization.

5.5.1 Low-Rank Restrictions

Let A be an $m \times n$ matrix. If A has exactly r linearly independent columns, then A is said to be of *rank* r. This means that it is possible to express A as

$$A = BC \tag{5.2}$$

where B is an $m \times r$ matrix and C is an $r \times n$ matrix. This implies that A, which has mn variables, can be expressed as the matrix product of B and C, which have a total of $r(m + n)$ variables. In the case where r is small, this can be a significant

reduction. If $r = \min\{m, n\}$, we say that A is of *full rank*.

We reduce the dimension of the problem by restricting each beam weight matrix to be of low rank. For each beam weight matrix BW this is accomplished by optimizing two matrices, BW_1 and BW_2, of sizes $m \times r$ and $r \times n$ respectively. We assume

$$BW = BW_1 BW_2, \tag{5.3}$$

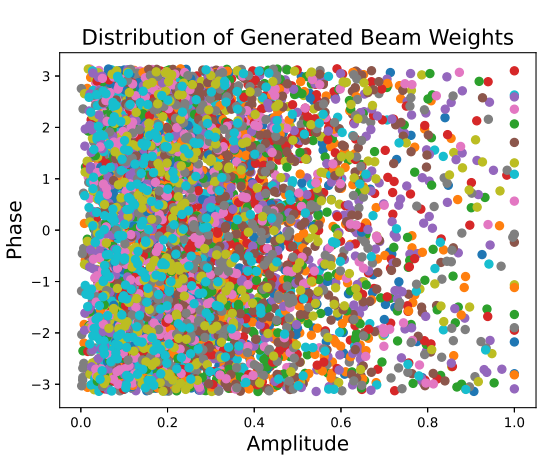
i.e., that BW is of rank r. BW_1 and BW_2 are complex valued matrices, but as explained in Section 5.5, we do not optimize the complex values, but rather the phases and amplitudes separately. Therefore, the values we optimize are $\arg BW_1$ and $\arg BW_2$, and the amplitudes $|BW_1|$ and $|BW_2|$. This means that the number of optimized parameters is $2r(m + n)$ for each beam weight matrix. Thus, the dimension of the search space is $2r(m + n)$ for single-polarized beamforming and $4r(m + n)$ for dual-polarized beamforming.

For large antenna arrays, this significantly reduces the number of variables to optimize, and thereby speeds up the optimization algorithms. Previous work done at Ericsson has used beam weights of low rank, and the initial tests of lower ranks yielded positive results. To limit the scope of the project, we restricted ourselves to only investigating rank 1 solutions in depth, and most decisions were made based on results for the rank 1 case.

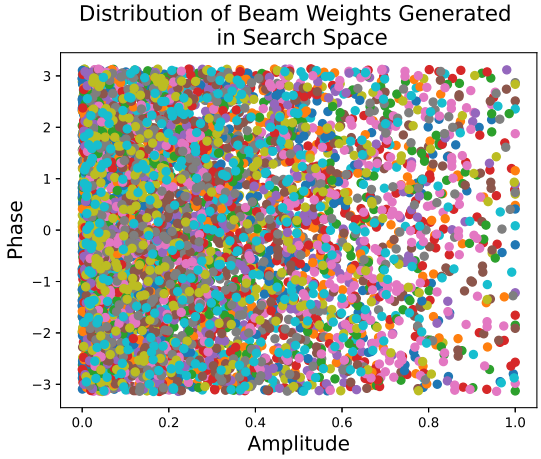
5.5.2 Initialization

When generating random starting points for GA and PSO, we decided between two different methods. The alternatives were generating beam weights and encoding them to our search space or generating directly in the search space. The difference is that the generated beam weights must be encoded to the search space, which has low rank, and thus information will be lost in the encoding. We wanted to see how this affected the distribution when they are decoded compared to the values generated directly in the search space.

We generated 20 sets of beam weights for an antenna array with 288 subarrays, i.e., 5760 beam weights. Their representations after they are encoded and decoded are shown in Figure 5.2a in a phase-amplitude space. We also generated the same number of beam weights directly in a rank 1 search space, and their representations after they are encoded are shown in Figure 5.2b. The values generated directly in the search space have a more spread out distribution compared to the generated beam weights, and thus we chose to generate directly in the search space, as the goal is to have the initial values cover as large area as possible to easier be able to find the optimum. We also want amplitudes to be close to the maximum allowed value in the final beam weights as this allows more of the available energy to be utilized.



(a) 5760 uniformly generated beam weights that have been encoded to rank 1 and then decoded.



(b) 5760 decoded beam weights generated uniformly in the rank 1 search space.

Figure 5.2: Comparison of two ways of initializing beam weights. Each point represents one beam weight.

5.6 t-Tests

Paired t-tests are used to test if there is a significant difference in mean value between two sets of observations, X_0 and X_1, which are continuous and normally distributed. There are two kinds of t-tests which are applicable in different situations depending on the dependence between observations in X_0 and X_1 [15].

5.6.1 Paired t-Tests

Paired t-tests are used to compare the mean values of two sets of observations, X_0 and X_1, when we can pair each observation in X_0 with one in X_1 in such a way that for these two observations all controllable variables are identical except for the ones being tested. By calculating the mean difference of each pair, we obtain a more accurate measure of the effect of the changed variable and reduce the impact of other environmental variables and differences between subjects [15].

More specifically, paired t-tests are used to test the null hypothesis $H_0 : \mu_1 - \mu_0 = 0$ under the assumptions that X_0 and X_1 are drawn from two continuous normal distributions, and each sample of X_0 is paired with a sample of X_1 in such a way that the pairs $X_j = \{X_{0j}, X_{1j}\}$ $j = 1, 2, ..., n$ are independent, but the data within each pair is correlated [15].

We calculate $X_{dj} = X_{1j} - X_{0j}$ and $\overline{X}_d = \frac{\sum_{j=1}^{n} X_{dj}}{n}$, and estimate the variance of $\overline{X_d}$ as $S_d^2 = \frac{1}{n(n-1)} \sum_{j=1}^{n} (X_{dj} - \overline{X_d})^2$. Thus, we can calculate

$$t = \frac{\overline{X_d}}{S_d}, \tag{5.4}$$

and the p-value is calculated via the student's t-distribution with $n-1$ degrees of freedom [15]. The exact calculation depends on the formulation of the alternative hypothesis, and the variants we use are presented in Section 5.6.3.

5.6.2 Two-Sample t-Tests

Two-sample t-tests are used when all observations in both X_0 and X_1 are independent of each other. Aside from that, all other assumptions made for paired t-tests hold for two-sample t-tests as well [15].

We therefore calculate the estimated variance for both sets of observations, S_0^2 and S_1^2, and if the true variances are equal we can estimate the variance of $\overline{X_d}$ as $S_d^2 = (\frac{1}{n_0} + \frac{1}{n_1})\frac{(n_0-1)S_0^2+(n_1-1)S_1^2}{n_0+n_1-2}$, where n_0 and n_1 are the number of observations in X_0 and X_1. Thus, we calculate the test statistic

$$t = \frac{\overline{X_d}}{S_d}, \tag{5.5}$$

and the p-value is calculated via the student's t-distribution with n_0+n_1-2 degrees of freedom [15]. The exact calculation depends on the formulation of the alternative hypothesis, and the variants we use are presented in the next section.

5.6.3 Use of t-Tests

We use t-tests to compare the distribution of the total loss for optimized beam weights for two methods of optimization. The observations in X_0 are the total loss of beam weights optimized using the standard or current method, and they are compared against the observations in x_1 from the alternative method. Using an alternative hypothesis H_1 we determine whether the alternative method results in a statistically significant change in the total loss. All p-values are also presented in Table A.1 in Appendix A or Table B.4 in Appendix B depending on which chapter they were calculated in.

We use two different alternative hypotheses H_1. $H_1 : \mu_0 > \mu_1$ is mostly used in Chapter 5 when we investigate alternatives to a standard method. This is because in those cases we are only interested in if the alternative method results in a decreased loss. This is a lower-tailed test and the p-value is calculated as $P(T < t)$, where T is the student's t-distribution with degree's of freedom as specified above, and t is the test statistic. $H_1 : \mu_0 \neq \mu_1$ is used mainly in Chapter 6 when we investigate differences between different cases of using our optimization algorithms. In those cases, it is also interesting if a case causes the loss to significantly increase. This case is two-tailed and therefore the p-value is calculated $p = 2 \cdot P(T > |t|)$. We always discard the null hypothesis in favour of the alternative hypothesis if the p-value is less than 0.05.

The results of paired t-tests are represented as boxplots of X_{dj} and above the boxes, we use asterisks to represent the p-values where the number of asterisks n signify a p-value less than 10^{-n}, with a maximum of three asterisks.

In some cases there are separate t-tests for each loss component as well as the total loss, i.e., side lobe loss plus main lobe loss, but it is the p-values for the total loss that are the deciding factor for which method we choose to use.

The default settings for the optimization used in the following tests were rank 1 single-polarized beamforming for an antenna array with 288 subarrays.

5.6.4 Side Lobe Threshold

One alternative version of calculating the side lobe loss was tested. Instead of defining the side lobe area using the side lobe threshold $U_{\max} - 12$ [dBm], it was defined using the threshold $U'_{\max} - 12$ [dBm], where $U'_{\max}$ is the maximum value of the radiation pattern corresponding to the optimized beam weights and not the reference field. The reason behind this alternative method was that even if the optimized beam did not reach the same amplitudes as the reference field, we could ensure that the side lobes were still suppressed below -12 [dBm] relative to the optimized curve. However, if the optimization managed to reach the same amplitudes levels as the reference field, this change should be no different from the current method.

We compared the two methods for defining the side lobe threshold using a paired t-test. The current method uses a side lobe threshold relative to the reference field, and the alternative method is relative to the optimized field. We correlated the pairs $\{X_{0j}, X_{1j}\}$ $j = 1, 2, ..., 20$ by using the same initial values for the optimization. The results of the paired t-test are presented in Appendix A and neither GA nor PSO showed a significant decrease in total loss when the side lobe areas were calculated relative to the optimized curve, as opposed to relative to the reference field. An important detail is that the total loss for both tests was evaluated using the same objective function, where the side lobe area is relative to the reference field. This is because a comparison using two different objective functions would not give any meaningful results. An interesting detail is that taper loss showed a decrease for both PSO and GA, with $p < 0.05$. Since the side lobe defined relative to the optimized beam did not result in a better performance, we decided that it would not be used.

5.6.5 Constrained vs Unconstrained Optimization

As previously mentioned, the amplitudes of the beam weights are constrained. The beam weights are therefore always normalized before being returned from the optimization to ensure this condition holds. However, we can choose to either constrain the amplitudes during optimization as well, never letting them exceed the maximal value, or let them be unconstrained and normalize at the end. We refer to these variants as constrained and unconstrained optimization, respectively.

For this paired t-test we used unconstrained optimization as the current method and constrained method as the alternative method for PSO as it is most commonly unconstrained. However, for GA the opposite is true, it is usually constrained, and thus we set the alternative method as unconstrained optimization for GA. We correlated the pairs $\{X_{0j}, X_{1j}\}$ $j = 1, 2, ..., 20$ by using the same initial values for the optimization. The result of the test is shown in Appendix A and the difference was not significant for either PSO or GA, and so we chose to continue using unconstrained PSO and constrained GA.

5.6.6 Warm Starts

When solving a problem, usually an optimization problem, solutions to nearby problems can sometimes be used as a *warm start* to facilitate and speed up the optimization process [16]. In our case, we can use traffic beam BWs as warm starts to find BWs for wide beams. More specifically, the approach we tested was to use the traffic beam weights, which were used to create the reference fields. We compared the distance in the rank 1 search space between a given wide beam BW and the BWs of the traffic beams used to create the reference field. The Euclidean distance between these was shorter than the distance between the wide beam BWs and randomly generated starting points.

We performed a paired t-test with random initialization as described in Section 5.5.2 as the current method and initialization where half the initial positions were random and half were traffic beams randomly selected from the traffic beams used to create the reference field as the alternative method. The pairs $\{X_{0j}, X_{1j}\}$ $j = 1, 2, ..., 20$ were correlated by using the same values for half of the initial positions for the optimization.

The results of the test are shown in Appendix A. For GA, we saw that $p < 0.05$ and the null hypothesis was discarded in favour of the alternative hypothesis. However, for PSO, the null hypothesis could not be discarded. So, warm starts give a statistically significant decrease of the total loss for GA, but no significant change for PSO.

Since the warm start method showed promise for GA, we decided to try adding noise onto the encoded traffic beam BWs. This was mostly to avoid GA getting stuck in local optima because too many of the initial values were too close to each other. For this purpose, we added Gaussian noise. For the amplitudes we subtracted the absolute value of the noise which had mean value zero and standard deviation α and for the phases we added noise with mean zero and standard deviation $\pi \cdot \alpha$ for varying values of α. The noise levels we tried were $\alpha = \{0.0, 0.1, 0.25, 0.5\}$ where $\alpha = 0.0$ means there is no noise, just the traffic beam weights. The results of the test are shown in Figure 5.3 and Appendix A and no tried value of α resulted in $p < 0.05$. We decided to let α be optimized by the hyperparameter tuning in Section 5.7 as the results of this test were inconclusive. If no noise is the best solution, then the Bayesian optimization should hopefully set α to zero.

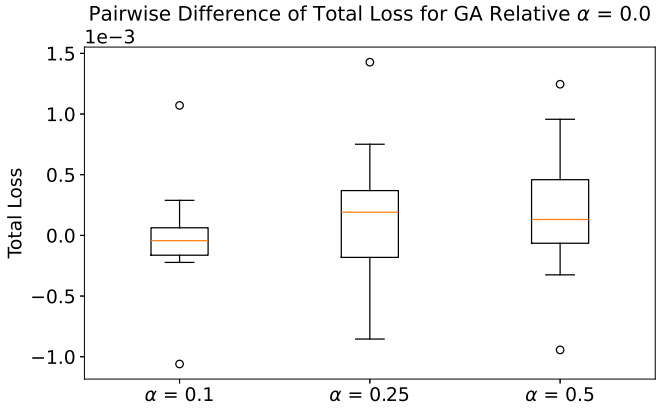


Figure 5.3: Result of paired t-test for GA for initial positions that are half random and half traffic beam weights. α is the level of noise added unto the traffic beams.

5.7 Hyperparameter Tuning

PSO has 6 hyperparameters that need to be chosen by the user. To perform a grid search with 3 different values for each hyperparameter would require 3^6 runs of the algorithm. PSO takes approximately 10 minutes per run, which would mean that the total run time would be about $10 \cdot 3^6$ minutes, which is equal to approximately 122 hours. However, testing only three values per hyperparameter is not a very thorough search, but if we want to test 10 values per hyperparameter, the runtime becomes 19 years. This is clearly an infeasible solution. Instead, we used Bayesian optimization to tune the hyperparameters for GA and PSO, which gave us preciser results and more manageable runtimes.

We used Bayesian Optimization to select all hyperparameters for each of the methods, choosing the set of hyperparameter values that minimized the objective function for each method. For GA, this included the standard deviation α of the noise added to the traffic beams in the warm starts, as described in section 5.6.6. We used $\mu_0 = 0$ and σ_0 constant for the prior.

For each hyperparameter tuned, we allowed the algorithm to test values within the feasible interval for that hyperparameter. For hyperparameters where the feasible interval is infinite, we limited the interval to a subset of the feasible interval which we deemed reasonable. In one instance, we found that one hyperparameter was chosen that was at the very edge of the interval we had chosen. We then extended that interval and ran the optimization again to prevent confining hyperparameter values to smaller intervals than necessary.

We tuned the hyperparameters specifically for rank 1 single-polarized beamforming for an antenna array with 288 subarrays. Despite using Bayesian optimization, hyperparameter tuning was a lengthy process, and we performed it only for these default settings.

The chosen hyperparameter values are omitted from this report for confidentiality reasons.

5.8 Extension to Dual-Polarized Beamforming

In all previous tests, we have handled only the single-polarized case. However, we extended our models to work for dual-polarized beamforming as well through minor changes to certain key functions. We chose to make no other changes to the algorithms, and used the same optimization algorithms and objective function as for single-polarized beamforming. The cases are similar and running all tests again for dual-polarized beamforming would have been too time-consuming.

6
Results

In this chapter, we present the results from a number of tests that were run to investigate the behaviour of our developed optimization methods. We begin with results that show the general performance of the algorithms, which we then compare to the method currently in use at Ericsson. We subsequently study the behaviour of the methods when the dimension of the problem varies, before investigating the stability of the results when exposed to varying levels of noise. Finally, we investigate the use of warm starts to expedite the optimization process, and present the time taken to optimize all necessary wide beams for an exemplary antenna array.

The results presented in this section were all produced using the same reference field unless explicitly stated otherwise. This is to reduce the number of external factors since we are mainly interested in comparing the algorithms to each other. The data used was simulated using Ansys HFSS to represent the ESAPs of an antenna array with 288 subarrays.

When we refer to GA and PSO in this chapter we mean the optimization algorithms as detailed in Chapter 5. Furthermore, when a run of either GA or PSO is referenced it will, if nothing else is stated, be single-polarized beamforming with rank 1 search space and run for 20 or 6000 iterations, respectively. These numbers were chosen to produce similar run times to enable a fair comparison between the algorithms. In some cases the results were run until convergence, meaning that we allowed the algorithm to run until the convergence criterion was fulfilled, but for a maximum of 60 or 8000 iterations for GA and PSO, respectively. These numbers were chosen to ensure that each run stopped eventually, while still allowing most runs to fulfill the convergence criterion. The convergence criterion is satisfied if the total loss improves by less than 10^{-5} for 2 consecutive iterations of GA, or for 1000 consecutive iterations for PSO. All tests were performed with an NVIDIA A40-48Q.

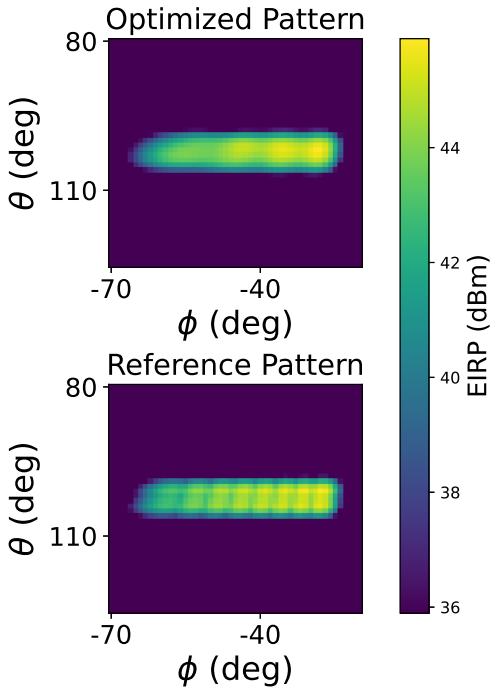
6.1 Performance

We ran GA and PSO 20 times each to investigate their performance and present the results here.

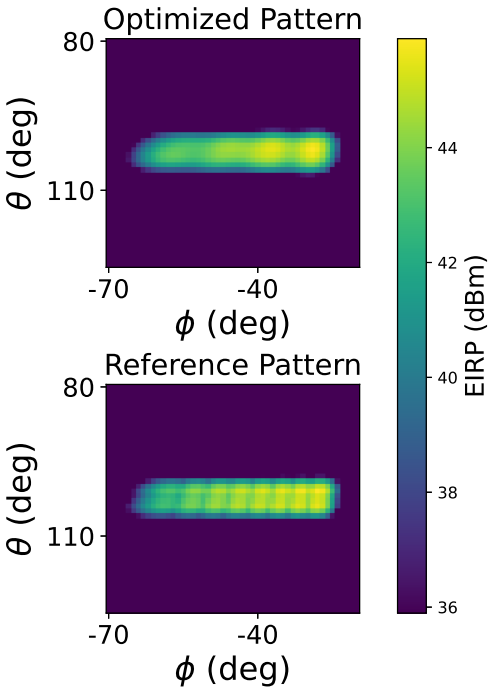
6.1.1 Example Beams

Figure 6.1 shows example beams from the results of GA and PSO. They were chosen as the beams with loss values closest to the median. The main lobes of the resulting radiation patterns are shown along with the main lobe of the used reference field. In Figure 6.2 we show cuts in elevation and azimuth of the same beam to allow for a closer comparison of the beams and the reference beam.

The results of the two algorithms are similar enough in performance that it is difficult to decide which algorithm is better from a purely visual inspection. To facilitate a comparison, we present the mean of each loss component and the total loss in Table 6.1, as well as the mean runtime.

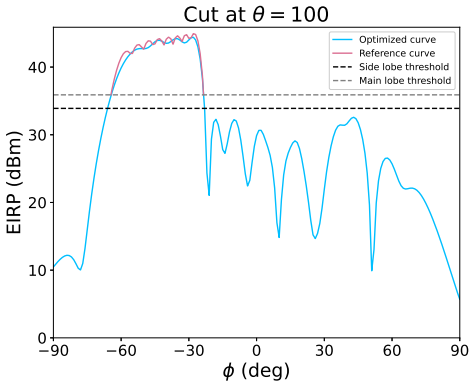


(a) The main lobe of the beam optimized by GA and the reference beam.

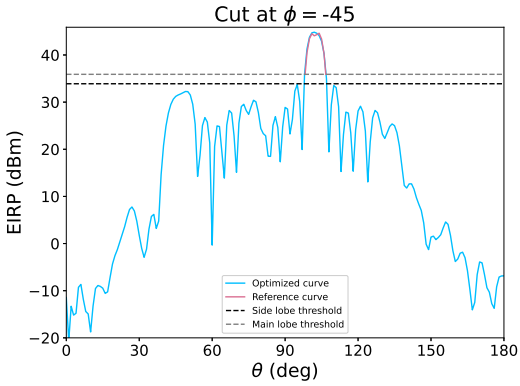


(b) The main lobe of the beam optimized by PSO and the reference beam.

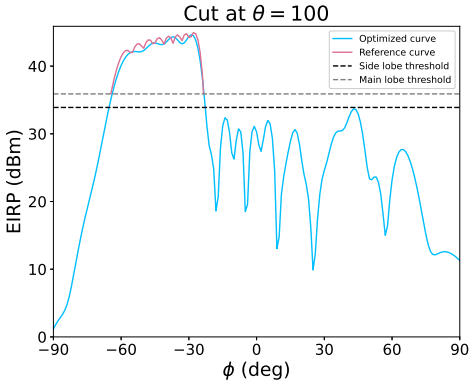
Figure 6.1: Comparisons of main lobes of beams optimized by GA and PSO and the reference beam.



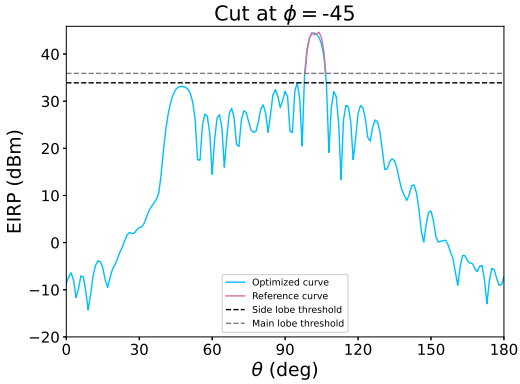
(a) Cut in azimuth of beam optimized by GA and reference beam.



(b) Cut in elevation of beam optimized by GA and reference beam.



(c) Cut in azimuth of beam optimized by PSO and reference beam.



(d) Cut in elevation of beam optimized by PSO and reference beam.

Figure 6.2: Comparisons between cuts of beams (blue lines) optimized by GA and PSO, and the reference beam (pink line). Main and side lobe thresholds are shown as well (grey and black line respectively).

6.1.2 Loss Distribution and Runtimes

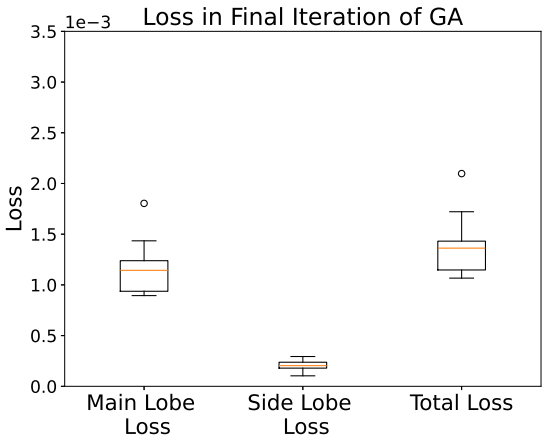
The run times were similar between GA and PSO, and the two algorithms produced results where the mean total loss is in the same order of magnitude. Interestingly, GA appears to be better at minimizing side lobes, while PSO appears to be better at optimizing the main lobe shape. Although similar, GA has a slightly lower total loss. Taper loss is omitted from this table, but can be found in Appendix B.

Table 6.1: Mean loss and runtime over 20 runs of GA and PSO.

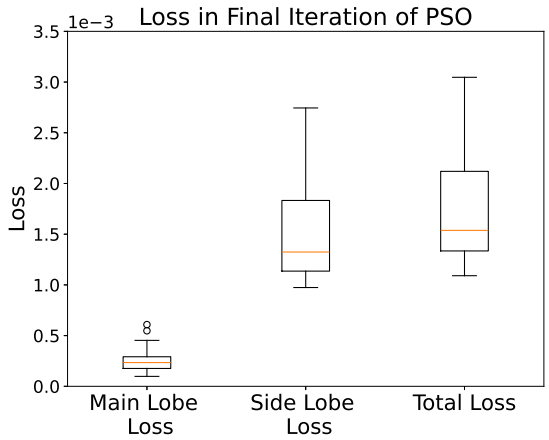
	Main lobe loss	Side lobe loss	Total loss	Runtime
GA	$1.15 \cdot 10^{-3}$	$2.08 \cdot 10^{-4}$	$1.35 \cdot 10^{-3}$	10 m 54 s
PSO	$2.63 \cdot 10^{-4}$	$1.58 \cdot 10^{-3}$	$1.85 \cdot 10^{-3}$	11 m 16 s

To further study the performance of GA and PSO we investigated the distribution of the final loss, i.e. the best loss in the final iteration, using the standard deviation

of the final loss in Table 6.2 and boxplots of the final loss in Figure 6.3. Similar to the mean values, the standard deviations of the total loss are of the same order of magnitude for GA and PSO.



(a) Best loss of final iteration from 20 runs of GA.



(b) Best loss of final iteration from 20 runs of PSO.

Figure 6.3: Comparison of best loss of final iteration for different metrics for 20 runs of GA and PSO.

Table 6.2: Standard deviation of final loss for all loss metrics for GA and PSO.

	Main lobe loss	Side lobe loss	Total loss
GA	$2.18 \cdot 10^{-4}$	$4.86 \cdot 10^{-5}$	$2.42 \cdot 10^{-4}$
PSO	$1.38 \cdot 10^{-4}$	$6.98 \cdot 10^{-4}$	$8.08 \cdot 10^{-4}$

6.1.3 Results with Convergence Criterion

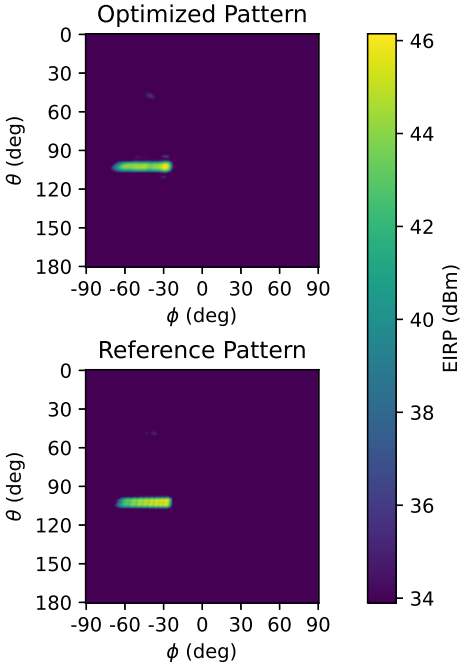
In Table 6.3 we present the mean total loss and runtime over 20 runs of GA and PSO where we used the convergence criterion to terminate the algorithms. This can be compared to Table 6.1 which contains the average losses and runtimes over 20 runs of GA and PSO when run for a set number of iterations. We find that the total loss and runtimes are similar for GA, but that PSO has a lower runtime but higher loss when stopped using the convergence criterion.

6.2 Comparison to Current Method

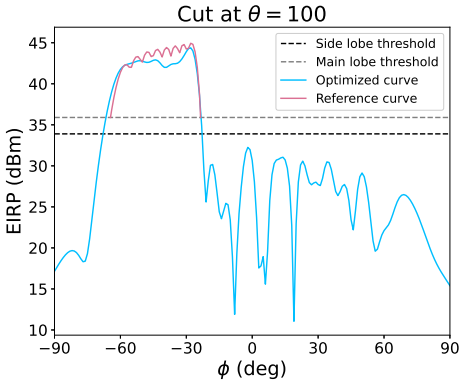
In this section, we compare our algorithms to the method currently employed at Ericsson, which will be referred to as the current method (CM). For this purpose we used a set of 24 wide beam BWs, optimized by CM, which covers the CAR. For a fair comparison, we optimized wide beam BWs with GA and PSO for the same antenna array as the beam weights received from Ericsson.

6.2.1 Example Beams from Current Method

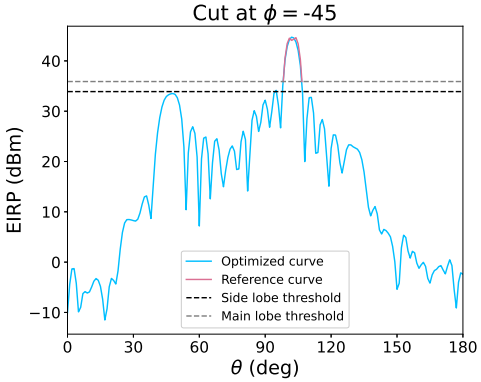
In Figure 6.4 we show a wide beam calculated using the BWs optimized by CM.



(a) Main beam of a radiation pattern optimized by CM.



(b) Azimuth cut of a radiation pattern optimized by CM.



(c) Elevation cut of a radiation pattern optimized by CM.

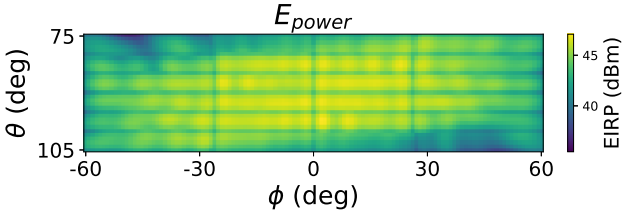
Figure 6.4: A wide beam optimized by CM. In b) and c) the pattern optimized by CM is blue and the radiation pattern of the reference field is pink. The grey and black dashed lines represent the side and main lobe thresholds respectively.

Table 6.3: Mean loss and runtime over 20 runs of GA and PSO when terminated using convergence criterion.

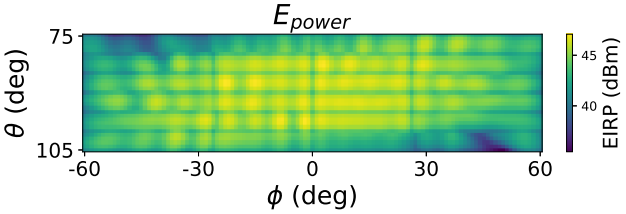
	Total loss	Runtime
GA	$1.40 \cdot 10^{-3}$	12 m 14 s
PSO	$2.59 \cdot 10^{-3}$	4 m 5 s

The investigated wide beams are arranged in a 4×6 GoB. We optimized the set of 24 wide beams five times until convergence for both GA and PSO, and a randomly

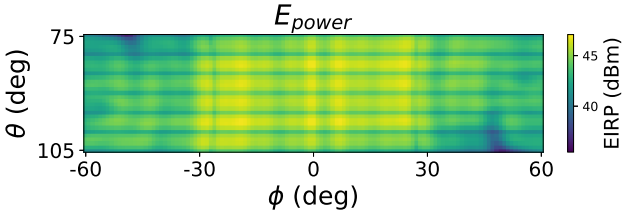
selected GoB from each of these methods is shown in Figure 6.5. For CM we only have one set, which is shown in Figure 6.5c.



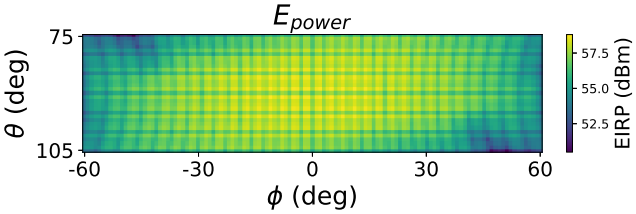
(a) Example GoB of wide beams optimized by GA.



(b) Example GoB of wide beams optimized by PSO.



(c) GoB of wide beams optimized by CM.



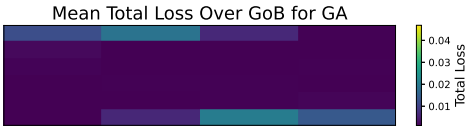
(d) GoB of traffic beams.

Figure 6.5: GoB of wide beams optimized by different methods and a GoB of the corresponding traffic beams.

The optimal GoB is one that is as similar as possible to the GoB of traffic beams shown in Figure 6.5d.

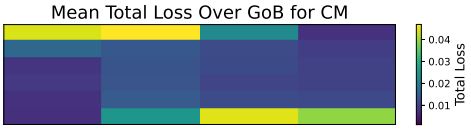
6.2.2 Differences in Loss to Current Method

The mean total losses for GA and PSO for each of the beams in the GoBs shown in Figure 6.5 are presented in Figures 6.6a-6.6b, and in Figure 6.6c we present the total loss of each corresponding beam from CM. Furthermore, in Figure 6.6d we show the difference between the mean total loss for GA and the total loss of CM. In Figure 6.6e we show the same result for PSO. Similar plots for the separate loss metrics can be found in Appendix B. From the Tables we understand that CM has a higher total loss than both GA and PSO for all beams in the GoB.

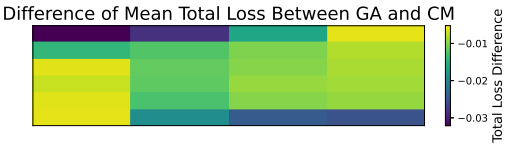


(a) Mean total loss of five runs for beams optimized by GA.

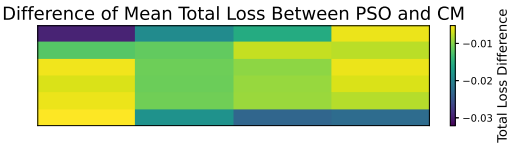
(b) Mean total loss of five runs for beams optimized by PSO.



(c) Total loss for beams optimized by current methods.



(d) Difference in total loss for each beam between GA and current method. Calculated as loss of GA minus loss of CM, i.e. negative value means GA has lower loss.



(e) Difference between total loss for each beam for PSO and CM. Calculated as loss of PSO minus loss of CM, i.e. negative value means PSO has lower loss.

Figure 6.6: Total loss for all beams in the GoB shown in Figure 6.5 and difference in total loss between CM and both GA and PSO. Each loss corresponds to the wide beam in the same position in the GoB.

To illustrate the difference between the methods more clearly, we calculated the mean difference between CM compared to both GA and PSO, that is, the mean of the values in Figure 6.6d and 6.6e. We present these values in Table 6.4 together with the corresponding values for taper loss, which was the only metric which increased. This is of interest since CM directly minimizes the taper loss as part of its objective function, while our method implicitly minimizes the taper loss as explained in Section 5.3.1. We once again see that the total loss for CM is higher than for both GA and PSO, but the difference is larger for GA.

Table 6.4: Mean difference in total loss and taper loss between GoBs optimized by CM compared to GA and PSO. For reference, taper loss is usually in the range two to three.

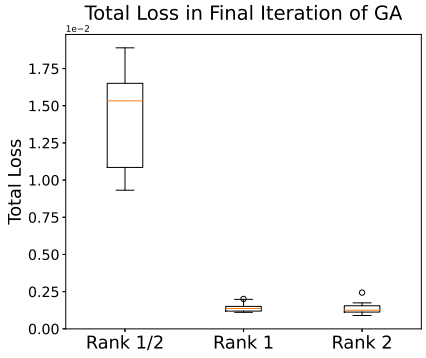
	Mean Total Loss Difference	Mean Taper Loss Difference
GA - CM	$-1.32 \cdot 10^{-2}$	$2.79 \cdot 10^{-1}$
PSO - CM	$-1.17 \cdot 10^{-2}$	$2.03 \cdot 10^{-1}$

6.3 Dimensionality Tests

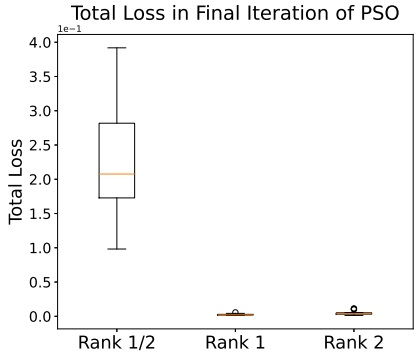
In this section we present a number of results illustrating the effect of the dimension of the search space on the runtime and performance of GA and PSO. There are a number of different factors that affect the dimension of the search space: the rank of the search space, the size of the antenna array, and whether we are performing single or dual polarized beamforming.

6.3.1 Ranks

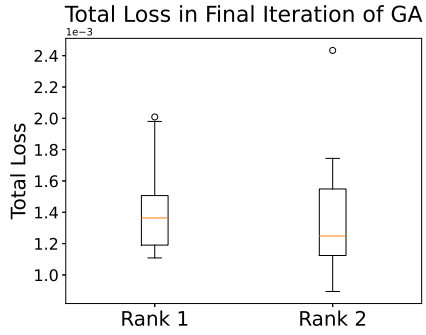
We investigated the differences in loss and runtime when optimizing with search spaces of different ranks. The tested ranks were rank 1, rank 2 and rank 1 with fixed amplitudes, which we refer to as rank $\frac{1}{2}$. For each case, the algorithms were run until convergence 20 times and we performed a two-sample t-test with rank 1 search space as the reference method. The loss values used in the two-sample t-tests are presented in Figure 6.7. Figures 6.7c and 6.7d are added to facilitate a visual comparison between rank 1 and 2.



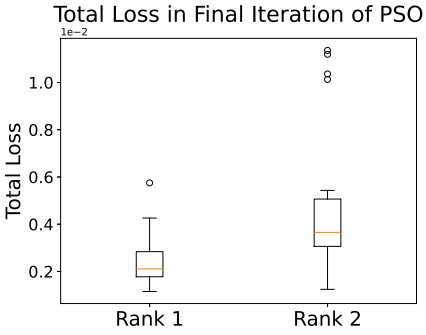
(a) Log final loss of 20 runs of GA for search spaces of rank 1/2, 1 and 2.



(b) Log final loss of 20 runs of PSO for search spaces of rank 1/2, 1 and 2.



(c) Final loss of 20 runs of GA or search spaces of rank 1 and 2.



(d) Final loss of 20 runs of PSO for search spaces of rank 1 and 2.

Figure 6.7: Distributions of final loss of 20 runs of GA and PSO for search spaces of rank 1/2, 1, and 2.

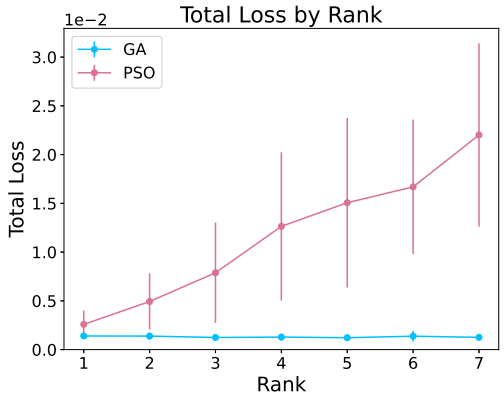
We discarded the null-hypothesis for rank $\frac{1}{2}$ for both GA and PSO with $p < 0.05$. It is a statistically significant increase of the mean loss. In the case of rank 2 we discarded the null hypothesis for PSO with an increase of loss, but we did not discard it for GA.

In Table 6.5 we show the mean runtimes for the three rank cases. We see an approximate doubling of the runtime with an increase in rank, with the exception of GA with rank $\frac{1}{2}$ which takes approximately the same amount of time as with rank 1.

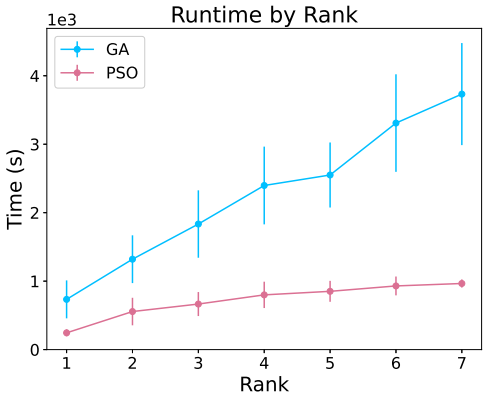
Table 6.5: Mean runtime over 20 runs for different search space ranks.

	Rank 1/2	Rank 1	Rank 2
Runtime GA	12 m 46 s	11 m 42 s	21 m 31 s
Runtime PSO	2 m 7 s	5 m 14 s	9 m 24 s

To study larger trends we broadened the scope to ranks 1-7. For this we ran each method 20 times until convergence for each rank. We show the resulting loss and runtime distributions in Figure 6.8.



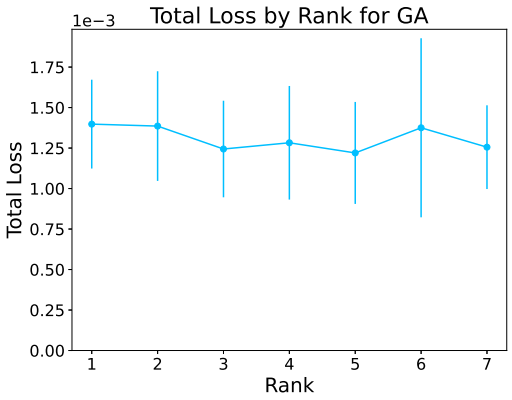
(a) Distribution over 20 runs of final loss for each method as a function of the rank of the search space.



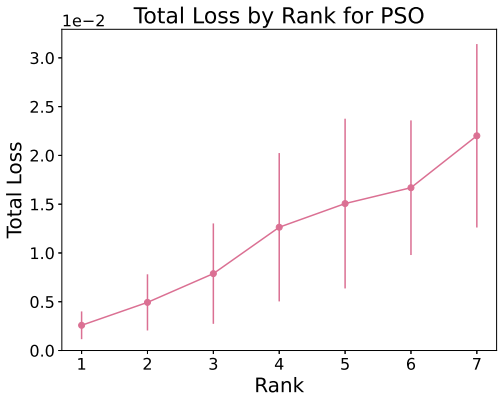
(b) Distribution over 20 runs of time until convergence for each method as a function of the rank of the search space.

Figure 6.8: Comparison between time and loss for different algorithms for different ranks. Results for GA are shown in blue, and results for PSO are shown in pink.

We see that PSO performs much worse than GA as the rank increases, but has a smaller increase in runtime when run until convergence. Since the loss distributions are so different between the methods, we present them each separately in Figure 6.9 so that they can be studied in isolation.



(a) Distribution over 20 runs of final loss for GA as a function of the rank of the search space.



(b) Distribution over 20 runs of final loss for PSO as a function of the rank of the search space.

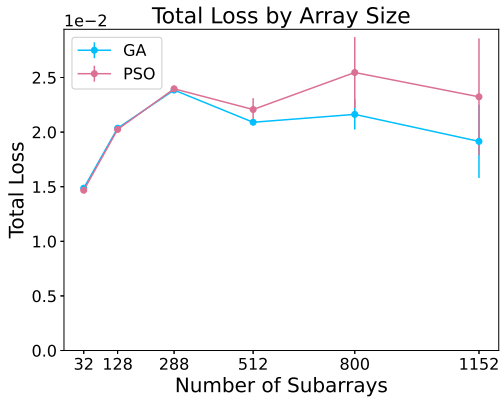
Figure 6.9: Loss distributions GA and PSO for different ranks. Results for GA are shown in blue, and results for PSO are shown in pink.

We see that as the rank increases, the loss returned by PSO tends to increase while the loss returned by GA shows no clear tendency in either direction.

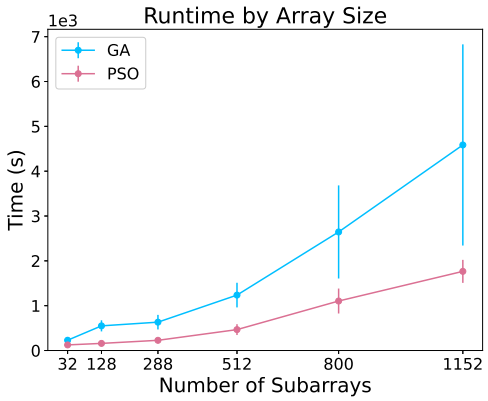
6.3.2 Array size

One important aspect of our algorithms is their scalability. The goal is an algorithm that scales well with antenna size, i.e. that the loss of the returned solution and the time taken to reach an acceptable solution does not increase too much for increased antenna size. For this reason, we investigated the distribution of the time until convergence and the loss of the returned solution for a number of antenna arrays of increasing size. We did not have access to realistic embedded subarray patterns for the range of antenna sizes we wanted to test, so we used ideal embedded subarray patterns.

For each antenna array size we calculated the ideal embedded subarray patterns and used these to create the reference beam. We chose to use the same traffic beams to build the reference beam for each array size for simplicity. The reference field was then used in 20 runs of each algorithm. The resulting mean total losses and mean runtimes are presented in Figure 6.10.



(a) Distribution over 20 runs of final loss for each method as a function of the number of subarrays.



(b) Distribution over 20 runs of time until convergence for each method as a function of the number of subarrays.

Figure 6.10: Comparison between time and loss for different algorithms for different array sizes. Results for GA are shown in blue, and results for PSO are shown in pink.

We see that GA and PSO scale similarly in loss, but PSO scales better in time when we have the same number of individuals as particles. We present the exact means for the different algorithms for the smallest and largest tested antenna array sizes in Table 6.6 and Table 6.7 respectively.

Table 6.6: Mean total loss and runtime for 20 runs of GA and PSO for an antenna array with 32 subarrays.

	Total Loss	Runtime
GA	$1.49 \cdot 10^{-2}$	3 min 49 s
PSO	$1.47 \cdot 10^{-2}$	2 min 5 s

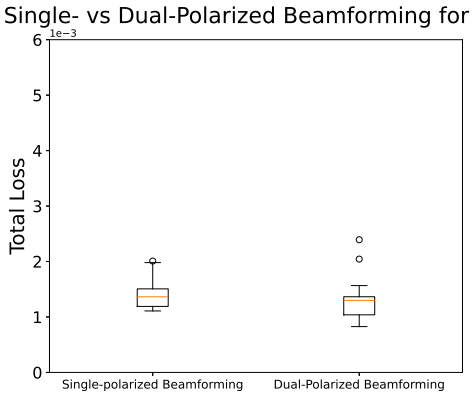
Table 6.7: Mean total loss and runtime for 20 runs of GA and PSO for an antenna array with 1152 subarrays.

	Total Loss	Runtime
GA	$1.92 \cdot 10^{-2}$	1h 12 min 25s
PSO	$2.32 \cdot 10^{-2}$	29 min 25s

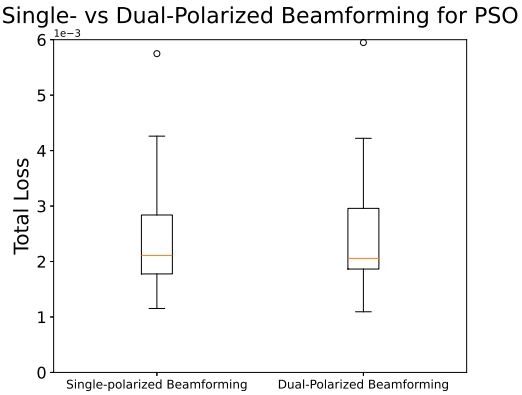
6.3.3 Single- vs Dual-Polarized Beamforming

Changing the rank or antenna size is not the only way to allow more degrees of freedom, it is also possible to perform dual-polarized beamforming, which doubles the number of beam weights to optimize. In Figure 6.11 a comparison between the final loss of single- and dual-polarized beamforming for GA and PSO is presented.

We performed a two-sample t-test on the data in Figure 6.11. We did not discard the null hypothesis for either GA or PSO as $p > 0.05$ in both cases, i.e, there was no significant difference in mean loss between the methods. Furthermore, in Table 6.8 the mean runtimes are shown, and it is clear that dual-polarized beamforming needs more time than single-polarized beamforming.



(a) Total loss for 20 runs of GA until convergence using single- and dual-polarized beamforming.



(b) Total loss for 20 runs of PSO until convergence using single- and dual-polarized beamforming.

Figure 6.11: Total loss for 20 runs of GA and PSO until convergence using single- and dual-polarized beamforming.

Table 6.8: Mean runtime over 20 runs for single- and dual-polarisation.

	Single-polarization	Dual-polarization
Runtime GA	11 m 42 s	38 m 14 s
Runtime PSO	5 m 14 s	8 m 12 s

6.4 Stability Analysis

During manufacturing, minute differences are created in antenna components which produce small variations in the shape of the electrical fields they produce compared to both other antennas and simulated data. Furthermore, beam weights are optimized in a continuous search space, but in practice only allow certain discrete values. For these reasons, the found optimized beams must be robust and stable against small variations in the beam weights. This is evaluated through a stability analysis consisting of adding noise to beam weights and evaluating the impact on the quality of the solution.

6.4.1 Beam Weight Error

By Ericsson's internal specifications, the error in the application of beam weights is not allowed to exceed a set threshold. To test whether an antenna exceeds this limit, the error in each beam weight is measured and the pairwise difference between these errors are calculated. The standard deviation of the differences of the amplitudes is σ_A and of the difference of the phases is σ_P, and they each have a threshold they must not exceed, defined as

$$3\sigma_A \leq A\,\text{dB} \qquad \text{for amplitudes,}$$
$$3\sigma_P \leq P^\circ \qquad \text{for phases,}$$

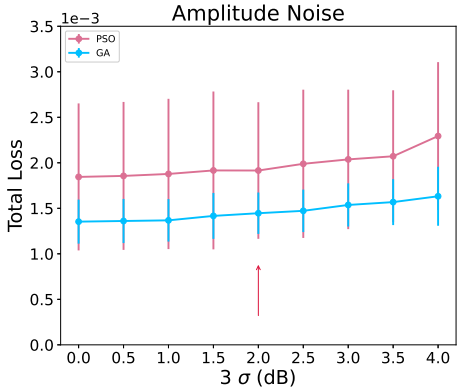
but exact values of A and P are not shared due to confidentiality reasons.

We assume the error caused by the antenna arrays is Gaussian with mean zero and standard deviation σ_A^n for amplitudes and σ_P^n for phases. The difference between two i.i.d. Gaussian distributions are $\mathcal{N}(0,\sigma^2) - \mathcal{N}(0,\sigma^2) = \mathcal{N}(0,2\sigma^2)$, which means that the standard deviation of the pairwise differences is $\sqrt{2}\sigma$. Therefore, we can write $\sqrt{2}\sigma_A^n = \sigma_A$ and $\sqrt{2}\sigma_P^n = \sigma_P$. The thresholds can then be rewritten as

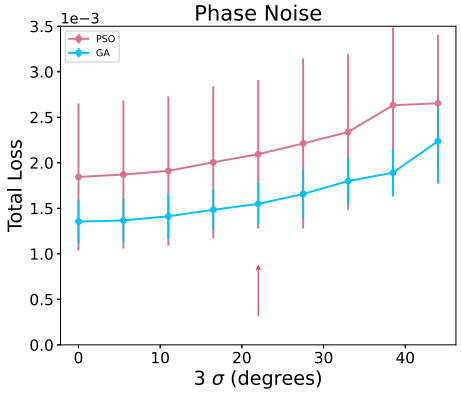
$$3\sqrt{2}\sigma_A^n \leq A\,\text{dB} \qquad \text{for amplitudes,}$$
$$3\sqrt{2}\sigma_P^n \leq P^\circ \qquad \text{for phases.}$$

Considering this, we added Gaussian noise with varying values of σ_A^n and σ_P^n corresponding to values of $3\sigma_A$ and $3\sigma_P$ between zero and twice the maximum allowed level onto the beam weights optimized in Section 6.1. Since the thresholds are in dB and degrees, the Gaussian noise was generated in these scales.

In Figure 6.12 the total losses for the 20 beam weights are shown for varying levels of noise. For the maximum allowed amplitude noise, $3\sigma_A = A$ dB, the loss increased with 7% for GA and 4% for PSO, and for the maximum allowed phase noise, $3\sigma_P = P^\circ$, the loss increased with 14% for both GA and PSO. Furthermore, in Figure 6.13 we regard two example beams optimized by GA and PSO and how they were affected by the added noise.

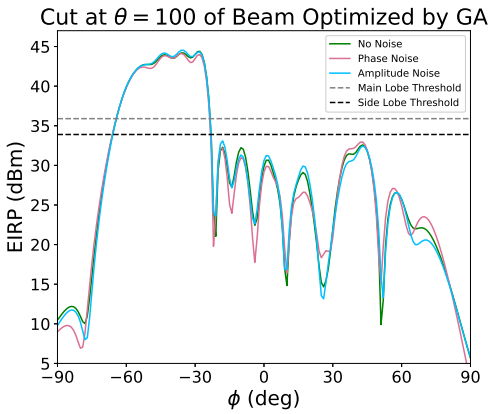


(a) Variation of total loss for 20 beam weights with noise added to the amplitudes.

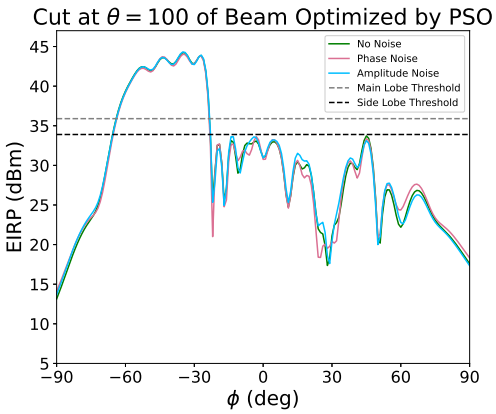


(b) Variation of total loss for 20 beam weights with noise added to the phases.

Figure 6.12: Variation of total loss for 20 beam weights with Gaussian noise added to either phase or amplitude. Red arrow indicates maximum allowed error according to internal Ericsson standards.



(a) Cut in azimuth of a beam optimized by GA which has no noise added or noise added to either amplitudes or phases.



(b) Cut in azimuth of a beam optimized by PSO which has no noise added or noise added to either amplitudes or phases.

Figure 6.13: Comparisons of beams with noise with $3\sigma_P = 22°$ added to phases and $3\sigma_A = 2$dB added to amplitudes. Also a beam with no noise added for reference.

6.5 Runtime in Real Application

When beamforming is performed for an antenna array, all wide beams in a GoB are optimized for several frequencies. For current methods, this takes approximately 1-2 days per frequency. The results from our methods suggest that such a process would take approximately two hours for PSO and 5 hours for GA, in the case of rank 1 single-polarized beamforming.

To further reduce runtimes, we can utilize similarities between the optimal beam weights for different beams. If we have optimized a wide beam for one frequency, it is possible that the optimal beam weights for the same wide beam in a similar frequency will be close in the search space. We therefore attempt to use the optimized beam weights for one frequency as warm starts when optimizing a similar frequency to speed up runtimes.

We also utilize approximate symmetries in the antenna array to create warm starts for wide beams with the same frequency. By mirroring the matrix of beam weights in a vertical line, we obtain a new matrix of beam weights which produces a radiation pattern very similar to the original radiation pattern mirrored in a vertical line. The same holds true for when we mirror in a horizontal line, or both. Given the beam weights of an optimized wide beam, it is then possible to create approximations of three other wide beams in the GoB by mirroring the beam weight matrix in different ways, and these can be used as warm starts to improve the optimization process. This idea is illustrated in Figure 6.14 which shows a GoB with one wide beam marked with an X. If we have optimized beam X, then by mirroring the beam weight matrix vertically we get a good warm start for beam one, by mirroring horizontally we get

a warm start for beam two, and by mirroring both ways we get a good warm start for beam three.

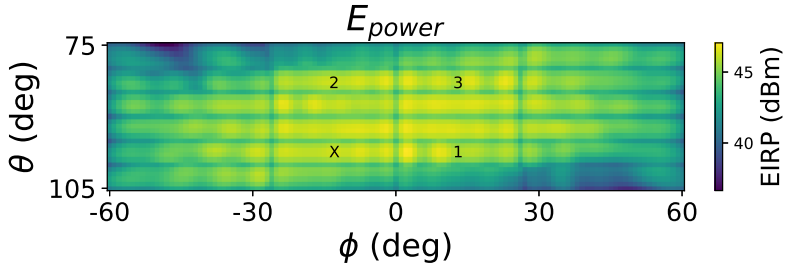


Figure 6.14: Beams marked 1, 2, and 3 can be approximately calculated using beam weights of the beam marked X by mirroring its amplitudes vertically, horizontally, or both.

To test the effect of these types of warm starts, we optimized each beam in the GoB for random frequencies. For each of these optimizations, we collected ten random individuals or particles from the final population or swarm. The corresponding beam weights of these were in turn mirrored horizontally, vertically, or both. These were then used as warm starts for the corresponding beams. A similar test was also performed where the ten individuals or particles were used as warm starts for the same beam but at a higher frequency. The averages of these are presented in Tables 6.9 and 6.10.

In Table 6.9 we observe that using symmetry warm starts results in a similar loss to optimizing without warm starts for GA, while PSO shows an increase of one magnitude for all symmetries except over frequencies. However, using warm start over frequencies resulted in a higher loss for both GA and PSO. Furthermore, in Table 6.10 we observe that all versions of warm starts are on average faster than no warm starts. However, using warm starts between frequencies is the worst of these cases.

Table 6.9: Mean total loss per beam over GoB for different kinds of warm starts.

	GA	PSO
No warm starts	$2.51 \cdot 10^{-3}$	$3.71 \cdot 10^{-3}$
Warm starts using vertical symmetries	$2.57 \cdot 10^{-3}$	$1.47 \cdot 10^{-2}$
Warm starts using horizontal symmetries	$2.65 \cdot 10^{-3}$	$1.72 \cdot 10^{-2}$
Warm start using diagonal symmetries	$2.50 \cdot 10^{-3}$	$1.98 \cdot 10^{-2}$
Warm start over frequencies	$3.07 \cdot 10^{-3}$	$6.56 \cdot 10^{-3}$

Table 6.10: Mean runtime per beam over GoB for different kinds of warm starts.

	GA	PSO
No warm starts	12.6 min	5.1 min
Warm starts using vertical symmetries	8.0 min	2.1 min
Warm starts using horizontal symmetries	8.9 min	2.1 min
Warm start using diagonal symmetries	7.0 min	2.1 min
Warm start over frequencies	8.2 min	4.0 min

From this data, we calculate the expected time needed to optimize all wide beams of an antenna array with five frequencies for three cases. The first case uses no warm starts, the second uses warm starts with the mirrored beam weights, and the last case uses warm starts with the mirrored beam weights and over frequencies. The expected times are presented in Table 6.11 and it is clear that both GA and PSO have a much lower runtime than CM and that PSO is faster than GA.

Table 6.11: Estimated times needed to optimize the entire GoB of an antenna with 288 subarrays for five frequencies using different warm start methods.

	CM	GA	PSO
No warm starts	5-10 days	25 h 15 m	10 h 10 m
Warm starts using symmetries	N/A	17 h 20 m	5 h 40 m
Warm starts using symmetries and over frequencies	N/A	15 h 35 m	5 h 15 m

7
Discussion

In this chapter. we begin with a discussion on whether the list of desiderata from Section 5.1 is fulfilled based on the results in Chapter 6. Next, we discuss the results from the dimensionality tests, before presenting the most significant limitations of this thesis project.

7.1 Fulfilment of Desiderata

A list of desiderata was presented in Section 5.1 and in this section we will, based on the results in Chapter 6, discuss whether these desiderata are fulfilled by our final algorithms.

7.1.1 Performance

The resulting BWs of GA and PSO both have a similar performance, as showcased by the examples in Figures 6.1 and 6.2 and by the mean total loss in Table 6.1. Our goal is not to prove that one of them is a better choice, but to show that both have potential to perform well in different situations, and that they both perform better than the currently used method. One indicator of their differing strengths is the main and side lobe loss in Table 6.1 which shows that GA focuses more on the side lobe loss and PSO on the main lobe loss.

In Table 6.4 we observe that the mean total losses of beam weights optimized by PSO and GA are both lower than the total loss of the beam weights optimized by CM. Thus, our algorithms perform better than CM for this antenna array.

In Figures 6.6d and 6.6e we observe that GA and PSO show the most improvement relative to CM in the top left and bottom right wide beams. This is very likely due to the fact that our methods use realistic simulated ESAPs and not ideal ESAPs, as CM does. In Figure 6.5d we note that the traffic beams corresponding to those wide beams have lower power than the rest. This is caused by the antenna construction and other realistic factors, which are reflected in realistic ESAPs. Since our methods optimize using realistic data, they can compensate for this and therefore naturally perform better in areas where antenna construction and related factors have a larger effect.

We see in Table 6.4 that the taper loss is higher for GA and PSO compared to CM.

As discussed in Section 5.3.1, our objective function does not directly try to force the taper loss to be low, but does so indirectly as the reference field is in EIRP, which is an absolute scale. This forces the amplitudes of the beam weights to increase, and thus the taper loss to decrease, as otherwise the corresponding radiation pattern of the beam weights will not reach the reference field. Since this method results in a higher taper loss but lower side and main lobe loss, this seems to indicate that the extra power used that causes the lower taper loss not only does not contribute to improving the shape of the main lobe, but also makes for higher side lobes. So although the taper loss is higher, this may in fact be a sign of more efficient energy usage.

7.1.2 Runtime and Convergence

In Figure 6.10 we present the convergence time and final loss as a function of the number of subarrays. We found that both PSO and GA appear to scale well with the size of the antenna array. PSO generally performs slightly worse than GA, but has a more favourable time complexity. For any array size, the choice between GA and PSO depends on whether the decrease in loss from using GA is worth the increase in time compared to using PSO. For larger antenna arrays the difference in both of these appears to grow larger.

The situation is similar when we optimize all beams for several frequencies of an antenna array. From Table 6.11 it is clear that PSO also performs better in this case in terms of runtimes. However, it also performs worse with regard to loss, as seen in Table 6.9.

It is faster to use both warm start versions for both GA and PSO. However, if the goal is to minimize the loss regardless of runtime, it is likely better to use only some or none of the warm start methods, since almost all cases result in a higher loss, especially symmetry warm starts for PSO.

The highest runtime in our results is for GA with no warm starts between beams, which is still 5-10 times faster than CM, while the lowest runtime is for PSO with both warm starts over frequencies and using symmetries which is 24-48 times faster than CM.

Validity of the Array Size Test

In section 6.3.2 we investigated how the runtime and performance were affected by the number of subarrays. In the test we chose to use the same traffic beam directions and the same number of traffic beams per wide beam. In real life, the size of the antenna array affects the width of traffic beams, meaning that larger antenna arrays create thinner traffic beams. For this reason, larger antenna arrays require a larger number of traffic beams to cover the CAR. Since we chose to use the same traffic beam directions regardless of the antenna size, we are not accurately reflecting what results the algorithms would yield if they were used to optimize beam weights for antennas of these sizes in real life. We chose this simplification

because we mainly wanted to focus on the differences between the algorithms when faced with a problem of a higher dimension. This means that the reference fields used in this test are simpler than the reference fields that would be used for antenna arrays of these sizes in real life, and it is possible that the algorithms would behave differently given a more intricate reference field. This is likely the reason why the mean total loss seems to decrease for the larger antenna sizes.

7.1.3 Stability Analysis

From Figure 6.12 we observe that the increase in loss is steeper for an increase in phase noise than for an increase in amplitude noise. For $3\sigma = A$ dB amplitude noise the increase in loss is 4% and 7% for GA and PSO respectively and for $3\sigma = P^\circ$ phase noise the increase in loss is 14% for both GA and PSO. We compare the results to the BWs produced using CM, which has a total loss of $7.6 \cdot 10^{-1}$. This is higher than all results in Figure 6.12, even when the noise is increased to twice the level of noise expected for an Ericsson product. We conclude that beam weights optimized using GA and PSO are robust against noise within the range that may be expected from Ericsson products. We further substantiate this claim through Figure 6.13 which shows the similarity between the beam shapes before and after noise was added to the beam weights, and find that there is no marked difference between the two.

7.2 Search Space Dimension

We investigated three main ways of changing the dimension of the search space. The first was by increasing the rank of the solutions. We found that the loss returned by PSO tends to increase as the rank increases, and that the loss returned by GA showed no clear trend in either direction. It is unclear how big the effect from the hyperparameter choice had on these results. Since the hyperparameters were tuned for rank 1, the results might have been better for higher ranks if the hyperparameters had been tuned separately for each case. Another factor that may have affected the loss distributions is the convergence criterion. The time until convergence did not increase much for PSO for larger ranks, and this might be an indicator that the convergence criterion used for PSO indicated convergence before the algorithm had truly converged.

The second way of increasing the search space dimension is by increasing the number of subarrays in the antenna array. We investigated the effect of increasing the number of subarrays on the loss and the runtime. We found that although PSO performs worse than GA as the dimension increases, the difference was not nearly as dramatic as when comparing different ranks. The runtime for GA increased faster than for PSO here as well, but as above this may have been a result of a poor choice of convergence criterion.

The third and final way is by switching from single-polarized to dual-polarized beamforming. This doubles the number of dimensions in the search space. There was

no statistical significance to show that dual polarized beamforming yielded a lower loss for either method, but this may have been affected by the fact that the hyperparameters were tuned for single-polarized beamforming. Since the dual-polarized beamforming had longer runtimes than the single-polarized beamforming, single polarized beamforming appears to be the superior choice in this case. These conclusions are also almost exclusively based on results from an antenna array with 288 subarrays, and much larger or smaller antenna arrays may yield different results.

In summary, for an antenna array of a set size, the best option appears to be rank 1 single-polarized beamforming, especially if one is wishes to keep the runtime low. In the standard setting, it is most likely better to use GA unless it is critical to keep the runtime low. However, since it is difficult to say exactly how large the effect of the hyperparameter choice has had on these results, we would recommend a more thorough analysis, including tuning hyperparameters individually for each case before making a comparison.

7.3 Limitations

There are some limitations of our project that were introduced during the process due to reasons such as time constraints, and that may have had an effect on our method and results. Here we attempt to outline as many as possible of these limitations.

7.3.1 Hyperparameter Selection

One such limitation is the hyperparameter selection which was only performed once for each algorithm. This was after all other decisions regarding the models had been made, i.e., after all the tests in Chapter 5. Before the selection, the used parameters were chosen based on recommendations from Wahde [9] and M L Shahab, F Azizi, B A Sanjoyo, *et al.* [10], which might not have been the optimal choices for this application and may have introduced a bias. Furthermore, during the generation of certain results, such as those discussed in Section 7.2, the hyperparameters were fitted for one of the tested cases, for example when rank $\frac{1}{2}$, 1, and 2 were compared, and the results indicated that rank 1 was the best option. The hyperparameters were fitted for rank 1 which likely created a bias. The extent of the effect of this bias on the results is unknown. The selection of parameters took around 40 hours per method and was therefore only performed once per method due to time constraints.

7.3.2 Number of Particles/Individuals

No thorough investigation of the number of particles/individuals used in the algorithms were performed. The reason this hyperparameter was not tuned using Bayesian optimization was because we used a set number of iterations during the hyperparameter tuning, and within a set number of iterations the expected loss decreases with an increase in the number of particles, but the time increases as well. Since we performed the Bayesian optimization based on the final loss after a

set number of iterations, the highest possible number of particles/individuals would always be selected by the optimizer.

7.3.3 Convergence Criterion

In Section 6.1.3 we presented the mean total loss and mean runtime over 20 runs of GA and PSO that were run until the convergence criterion was satisfied. When comparing these to the results in Section 6.1.2 where the algorithms were run for a set amount of iterations, we found that GA has a similar runtime and similar total loss regardless of method. For PSO however, the convergence criterion stops the algorithm much sooner and the loss is much higher, which seems to indicate that the convergence criterion does not work very well for PSO. Exploring different possible convergence criteria could likely improve performance for PSO when run until convergence. Improving the convergence criterion would most likely increase runtimes for PSO and decrease the mean loss returned by PSO. This would impact all results in Section 6 where the convergence criterion was used.

7.3.4 Limited Access to Data

In this report we use realistic data simulated using ANSYS HFSS for the ESAPs of an antenna array with 288 subarrays. It is time intensive to simulate ESAPs, and for this reason we only had one such set available to us. This means that all decisions and hyperparameter choices were based on this dataset. There is a risk that this introduces a bias as we have both tuned hyperparameters and tested our methods using the same data.

8

Conclusion

In this chapter we summarize our conclusions on whether this thesis project was successful before presenting some potential avenues for future investigations based on the results of this thesis.

8.1 Fulfilment of the Desiderata

In accordance with Section 7.1 we deem that the desiderata are fulfilled by our implementation. As a reminder, we briefly state these desiderata again here. The algorithm should:

- Consistently produce beam weights that correspond to a radiation pattern with desired metric values with respect to a given reference field.
- Run and converge in a shorter time than the current methods.
- Produce results that are robust against small variations resulting from the manufacturing of antenna and measurement components.
- Optimize beam weights using measured and simulated realistic ESAPs.

The first point is fulfilled, as both GA and PSO consistently produce beam weights that yield a lower total loss than CM. We also find that using the absolute EIRP scale and removing the taper loss component from the objective function, as opposed to using a relative scale, appears to work well.

The methods also produce results faster than CM. GA and PSO are 5-48 times faster when optimizing all beams for several frequencies for an antenna array, depending on which algorithm is used and which methods for warm starts are used. There is usually a trade-off between runtime and loss, with a lower loss for longer runtimes, and the choice ultimately depends on the user's needs. The time complexity for PSO appears to be especially favourable, though this might be due to a suboptimal choice of convergence criterion, as the loss generally is lower.

The stability analysis results are also positive and indicate that the optimized beam weights are robust against noise. When applying random noise within the levels that are accepted for Ericsson products, the loss levels rise but remain within acceptable limits.

The last point is fulfilled by the design of the algorithms, which were built to handle ESAP data of any type, regardless of how it was produced.

8.2 Further Studies

Due to time and resource constraints, we placed several limitations on this study. Here we propose some ways forward for potential future investigations and development of the algorithms presented in this thesis.

8.2.1 Hyperparameters

The choice of hyperparameters can greatly affect the performance of an algorithm. It is therefore of interest to investigate the choice of hyperparameters thoroughly.

Removing Hyperparameter Bias

As discussed in Section 7.3.1, when running tests to choose between two alternatives, it would be more fair to tune hyperparameters for both alternatives before selecting one. This would remove any potential bias from the hyperparameters used. One potential avenue of investigation would be to redo the different tests presented in Chapter 5 except with an additional step of tuning the hyperparameters separately for each case. It would also be interesting to see the effect on the results in Chapter 6 if the hyperparameters were tuned separately for each tested case.

PSO with Inertia Decay

Another related change is to let parameters change over iterations, for example, inertia, as a common version of PSO is for inertia to decrease over iterations. This is referred to inertia decay. Implementing inertia decay could potentially help PSO convergence stability.

Number of Particles/Individuals

Another hyperparameter which we discussed was the number of particles/individuals used by PSO/GA. Investigating the optimal number of these could be a way of further improving the algorithms, as a higher number may lead to an improved loss, but also to an increased runtime. One potential avenue for future investigations could be to find which number of particles gives the greatest decrease in loss per time unit for a set number of iterations. Another approach could be to investigate what number of particles that give the lowest loss within a set time limit.

8.2.2 Convergence Criterion

The selected convergence criteria worked well for GA, though not as well for PSO. This could be a reason why PSO generally had a worse loss than GA in our results,

and it is worth investigating alternative criteria, both other thresholds and completely different methods.

For example, it might be of interest to explore different thresholds than 10^{-5} for the difference in loss, or to use a relative threshold as the current absolute threshold may need to be adjusted for different cases. A threshold that is, for example, 10^{-2} of the current best loss could better adapt to the situation and work in more situations. A completely different convergence criteria could also be interesting to investigate, for example, one monitoring how close all particles are to one another in the search space, or the velocities of the particles. However, due to the curse of dimensionality it might be hard to define when the particles can be considered to be close to one another.

8.2.3 Testing on New Data

It would be interesting to get access to new data and be able to test if there is a bias due to training and testing on the same data or if the results are similar.

8.2.4 Optimizing Full Antenna

The tests performed in Section 6.5 are mainly a proof of concept to illustrate how one could speed up the optimization of an entire antenna by utilizing warm starts. It might be of interest to further study how altering implementation details pertaining to the selection of particles/individuals to use in the warm start for the optimization of the next beam might alter the efficiency of the next optimization. This could likely further improve the runtime for the optimization of an entire antenna.

Bibliography

[1] R. Frilund and Z. Olofsson, "Beamweight Prediction Using Complex Valued Convolutional Neural Networks," eng, 2024. [Online]. Available: http://hdl.handle.net/20.500.12380/308334 (visited on 11/04/2024).

[2] P. S. Kildal, *Foundations of Antenna Engineering*. Gothenburg: Kildal Antenn AB, Mar. 2017, Publisher: [Online]. Available: https://www.kildal.se/downloads/ (visited on 11/12/2024).

[3] D. K. Cheng, *Field and Wave Electromagnetics*, English, Second edition. Pearson, 1989, ISBN: 9781292026565. (visited on 11/12/2024).

[4] "IEEE Standard for Definitions of Terms for Antennas," *IEEE Std 145-2013 (Revision of IEEE Std 145-1993)*, 2014. DOI: 10.1109/IEEESTD.2014.6758443. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6758443&tag=1 (visited on 11/25/2024).

[5] C. A. Balanis, *Antenna Theory, Analysis and Design*, Second Edition. John Wiley & Sons, 1996. (visited on 11/22/2024).

[6] NIST, "Guide for the Use of the International system of Units," Tech. Rep. NIST Special Publication 811, 2008. [Online]. Available: https://physics.nist.gov/cuu/pdf/sp811.pdf (visited on 11/25/2024).

[7] M. Girnyk, H. Jidhage, and S. Faxér, "Broad beamforming technology in 5G Massive MIMO," English, Ericsson Technology Review 10, Oct. 2023, p. 6. [Online]. Available: https://www.ericsson.com/490a10/assets/local/reports-papers/ericsson-technology-review/docs/2023/broad-beamforming-technology-in-5g-massive-mimo.pdf (visited on 05/21/2025).

[8] C. A. Balanis, *Antenna theory: analysis and design* (New York Academy of Sciences Series), English, Fourth Edition. John Wiley & Sons, 2016. (visited on 04/15/2025).

[9] M. Wahde, *Biologically Inspired Optimization Methods: An Introduction*. WIT Press, 2008, ISBN: 9781845643447. [Online]. Available: https://research.ebsco.com/c/lu54te/search/details/g4jqalpsm5?q=biologically%20inspired%20optimization%20methods (visited on 11/29/2024).

[10] M L Shahab, F Azizi, B A Sanjoyo, M I Irawan, N Hidayat, and A M Rukmi, "A genetic algorithm for solving large scale global optimization problems," *Journal of Physics: Conference Series*, vol. 1821, no. 1, p. 012 055, Mar. 2021. DOI: 10.1088/1742-6596/1821/1/012055. [Online]. Available: https://dx.doi.org/10.1088/1742-6596/1821/1/012055 (visited on 04/17/2025).

[11] P. I. Frazier, *A Tutorial on Bayesian Optimization*, English, 2018. [Online]. Available: https://arxiv.org/abs/1807.02811 (visited on 04/29/2025).

[12] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, Nov. 2005, ISBN: 9780262256834. DOI: 10.7551/mitpress/3206.001.0001. [Online]. Available: https://doi.org/10.7551/mitpress/3206.001.0001.

[13] D. Boeringer and D. Werner, "Particle swarm optimization versus genetic algorithms for phased array synthesis," *IEEE Transactions on Antennas and Propagation*, vol. 52, no. 3, pp. 771–779, 2004. DOI: 10.1109/TAP.2004.825102.

[14] O. Duzel, B. Saoud, I. Shayea, and G. N. Tulepova, "Antenna optimization based on genetic algorithm," in *2024 IEEE 3rd World Conference on Applied Intelligence and Computing (AIC)*, 2024, pp. 511–517. DOI: 10.1109/AIC61668.2024.10730897.

[15] M. Xu, D. Fralick, and J. Z. Z Zheng, "The Differences and Similarities Between Two-Sample T-Test and Paired T-Test," *Shanghai Arch Psychiatry*, vol. 29, no. 3, pp. 184–188, Jun. 2017. DOI: 10.11919/j.issn.1002-0829.217070.

[16] A. Forsgren, "On Warm Starts for Interior Methods," in *System Modeling and Optimization*, F. Ceragioli, A. Dontchev, H. Futura, K. Marti, and L. Pandolfi, Eds., Boston, MA: Springer US, 2006, pp. 51–66, ISBN: 978-0-387-33006-8. (visited on 04/24/2025).

A

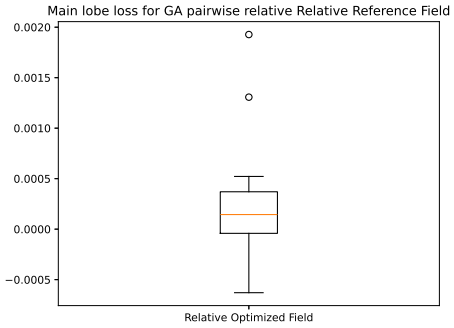
Method Appendix

Appendix containing Figures and Tables concerning topics in Chapter 5.

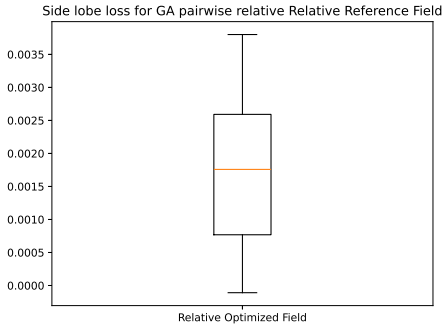
A.1 Paired t-Tests

Section containing results from paired t-tests that were not deemed important enough to show in the Method part of the thesis,

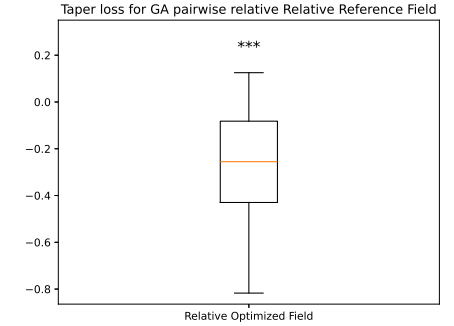
A.1.1 Side Lobe Thresholds



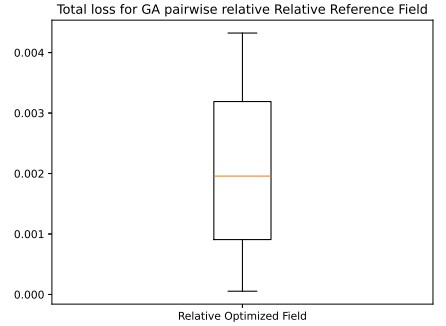
(a) Boxplot of difference between main lobe loss for optimization with side lobe thresholds relative to optimized field vs reference field.



(b) Boxplot of difference between side lobe loss for optimization with side lobe thresholds relative to optimized field vs reference field.

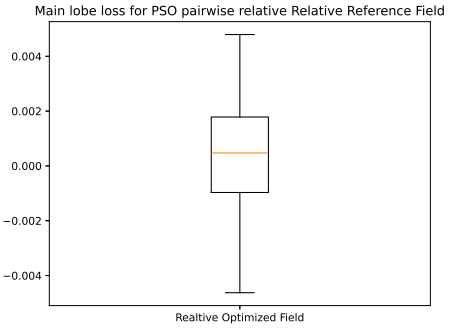


(c) Boxplot of difference between taper loss for optimization with side lobe thresholds relative to optimized field vs reference field.

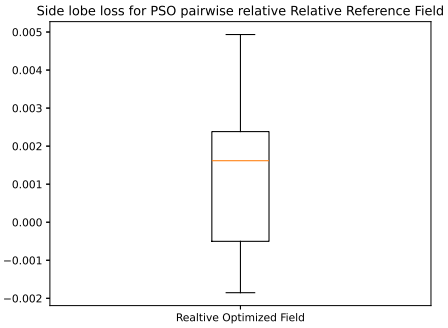


(d) Boxplot of difference between total loss for optimization with side lobe thresholds relative to optimized field vs reference field.

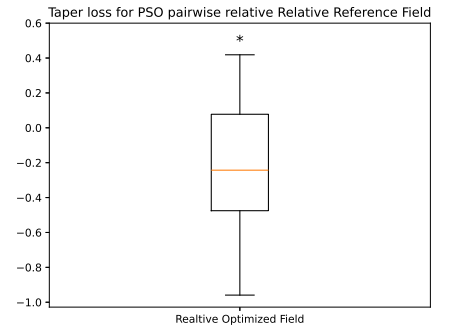
Figure A.1: Results from paired t-test for optimization using genetic algorithm and side lobe thresholds relative to optimized field vs reference field. Both methods were ran 20 times.



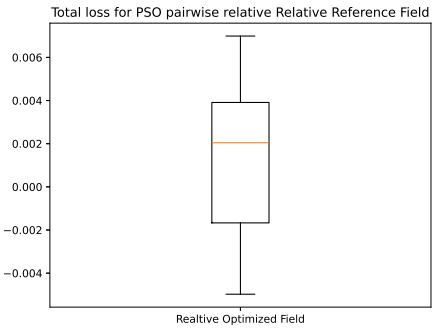
(a) Boxplot of difference between main lobe loss for optimization with side lobe thresholds relative to optimized field vs reference field.



(b) Boxplot of difference between side lobe loss for optimization with side lobe thresholds relative to optimized field vs reference field.



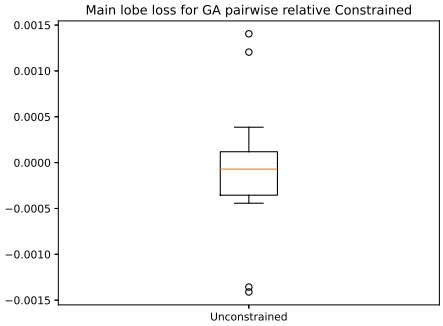
(c) Boxplot of difference between taper loss for optimization with side lobe thresholds relative to optimized field vs reference field.



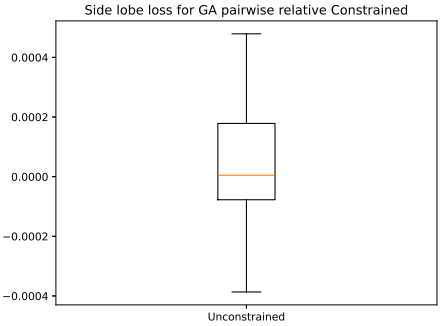
(d) Boxplot of difference between total loss for optimization with side lobe thresholds relative to optimized field vs reference field.

Figure A.2: Results from paired t-test for particle swarm optimization and side lobe thresholds relative to optimized field vs reference field. Both methods were ran 20 times.

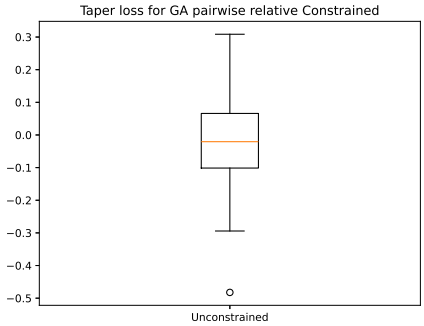
A.1.2 Constrained vs Unconstrained Optimization



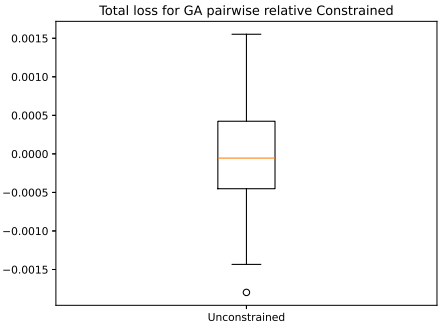
(a) Boxplot of difference between main lobe loss for constrained vs unconstrained optimization.



(b) Boxplot of difference between side lobe loss for constrained vs unconstrained optimization.

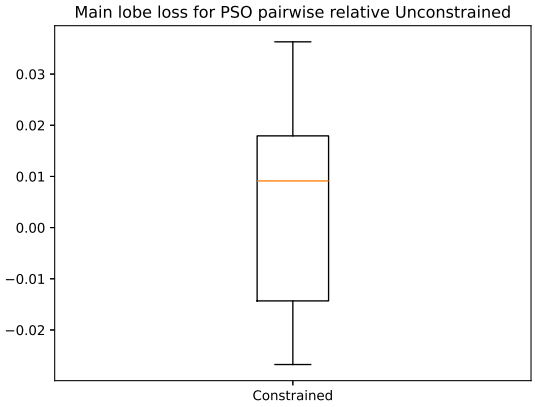


(c) Boxplot of difference between taper loss for constrained vs unconstrained optimization.

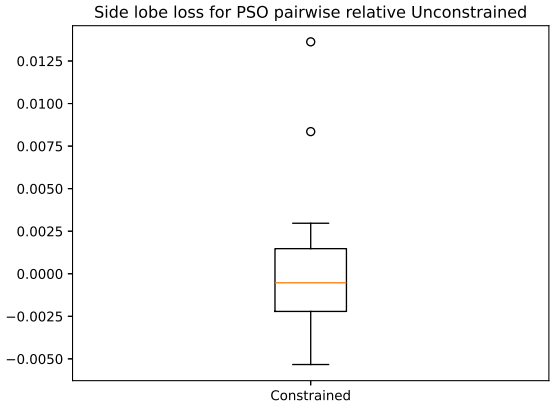


(d) Boxplot of difference between total loss for constrained vs unconstrained optimization.

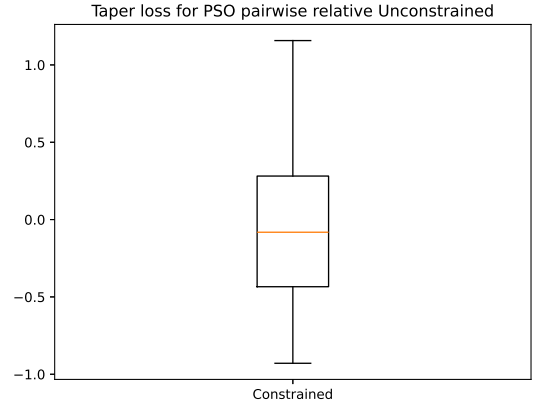
Figure A.3: Results from paired t-test for optimization using constrained vs unconstrained GA.



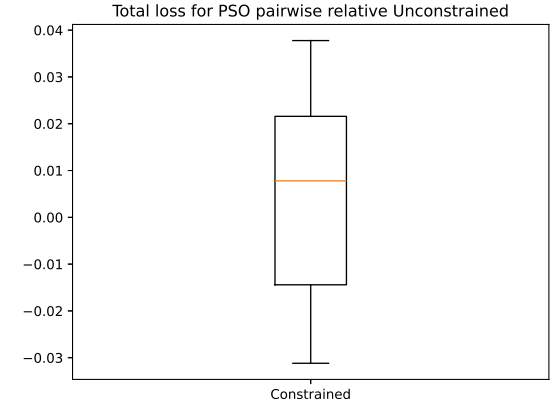
(a) Boxplot of difference between main lobe loss for constrained vs unconstrained optimization.



(b) Boxplot of difference between side lobe loss for constrained vs unconstrained optimization.



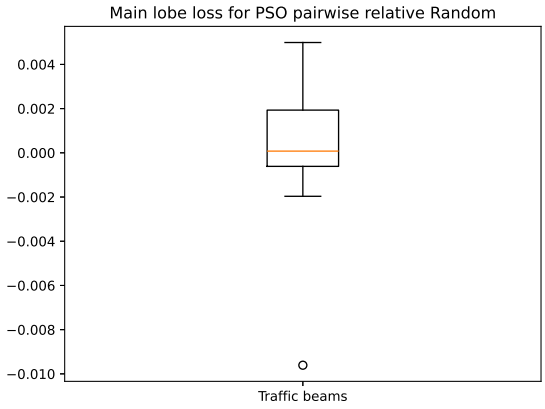
(c) Boxplot of difference between taper loss for constrained vs unconstrained optimization.



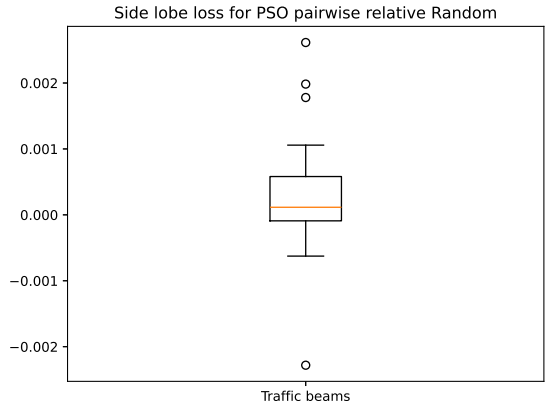
(d) Boxplot of difference between total loss for constrained vs unconstrained optimization.

Figure A.4: Results from paired t-test for constrained vs unconstrained PSO.

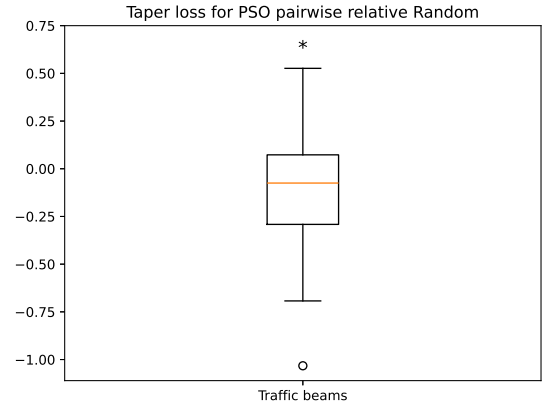
A.1.3 Warm Starts



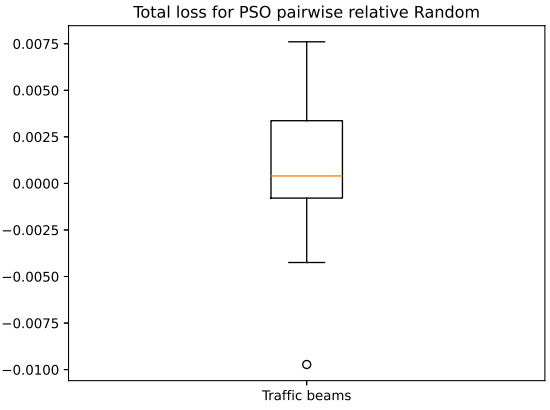
(a) Boxplot of difference between main lobe loss for initial positions completely random vs half random and half traffic beam beam weights.



(b) Boxplot of difference between side lobe loss for initial positions completely random vs half random and half traffic beam beam weights.

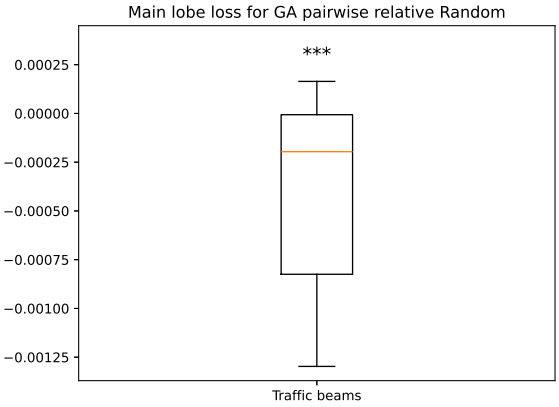


(c) Boxplot of difference between taper loss for initial positions completely random vs half random and half traffic beam beam weights.

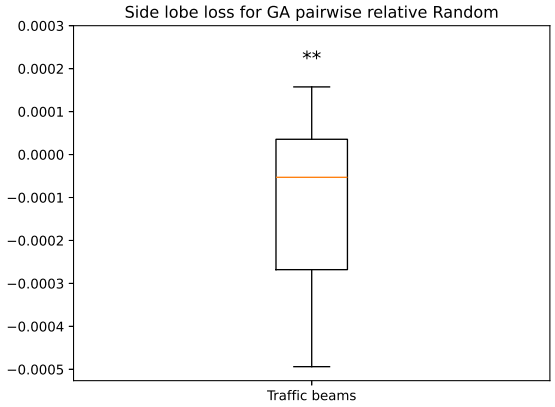


(d) Boxplot of difference between total loss for initial positions completely random vs half random and half traffic beam beam weights.

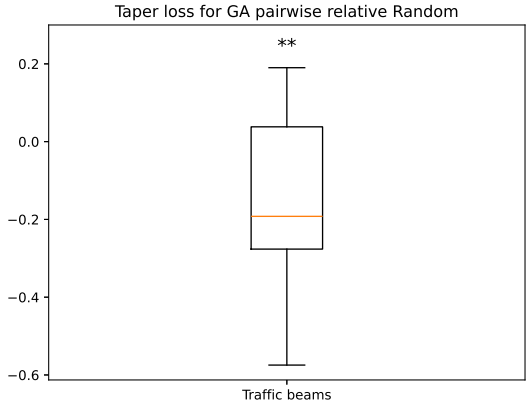
Figure A.5: Results from paired t-test for PSO with initial positions completely random vs half random and half traffic beam beam weights.



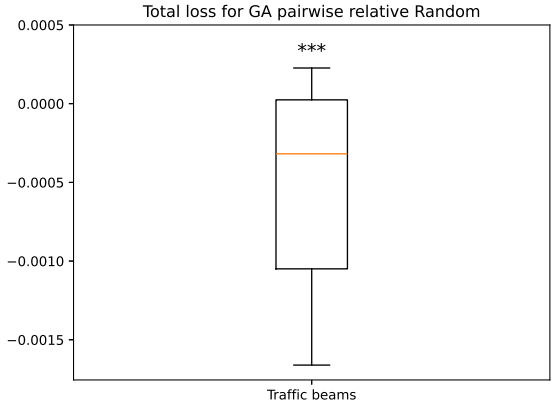
(a) Boxplot of difference between main lobe loss for initial positions completely random vs half random and half traffic beam beam weights.



(b) Boxplot of difference between side lobe loss for initial positions completely random vs half random and half traffic beam beam weights.

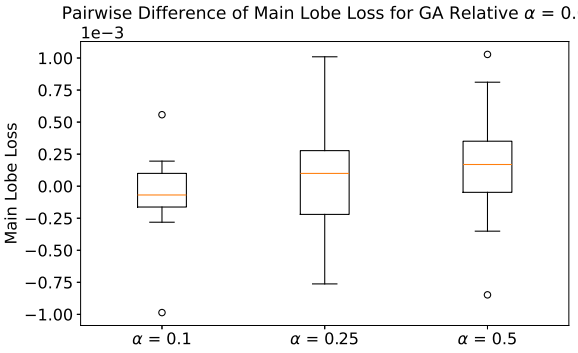


(c) Boxplot of difference between taper loss for initial positions completely random vs half random and half traffic beam beam weights.

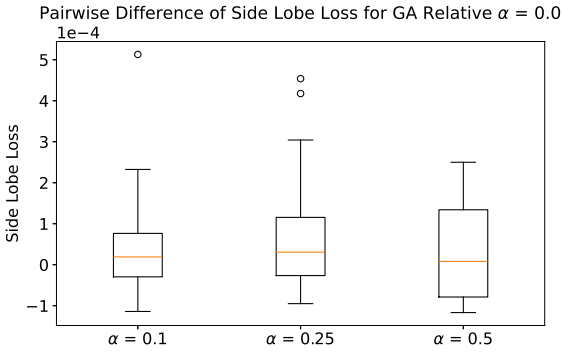


(d) Boxplot of difference between total loss for initial positions completely random vs half random and half traffic beam beam weights.

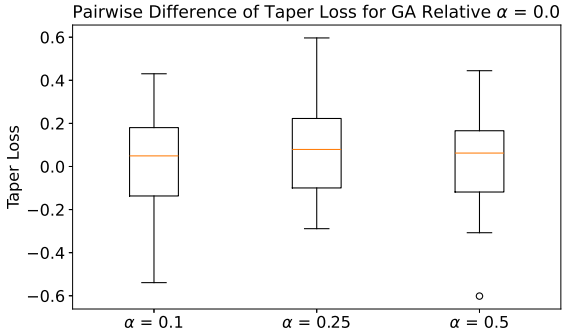
Figure A.6: Results from paired t-test for GA with initial positions completely random vs half random and half traffic beam beam weights.



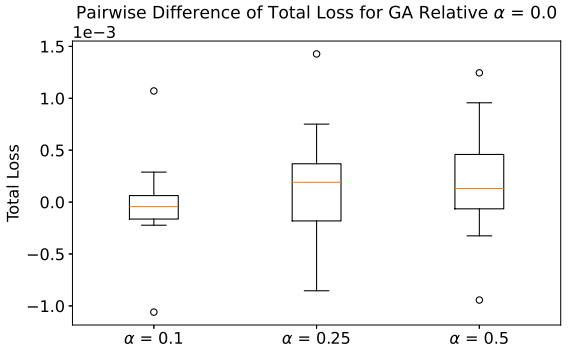
(a) Boxplot of difference between main lobe loss for initial positions that are half random and half traffic beam beam weights with different levels of noise added.



(b) Boxplot of difference between side lobe loss for initial positions that are half random and half traffic beam beam weights with different levels of noise added.



(c) Boxplot of difference between taper loss for initial positions that are half random and half traffic beam beam weights with different levels of noise added.



(d) Boxplot of difference between total loss for initial positions that are half random and half traffic beam beam weights with different levels of noise added.

Figure A.7: Result of paired t-test for GA for initial positions that are half random and half traffic beam weights. α is the level of noise added unto the traffic beams.

A.1.4 p-Values

p-values for all paired t-tests performed in Chapter 5.

Table A.1: p-values for all paired t-tests, rounded to two significant figures.

	Main lobe loss	Side lobe loss	Taper loss	Total loss
Warm start GA - Random vs TB	$7.1 \cdot 10^{-4}$	$6.3 \cdot 10^{-3}$	$1.6 \cdot 10^{-3}$	$7.5 \cdot 10^{-4}$
Warm start PSO - Random vs TB Random vs TB	0.73	0.91	0.052	0.81
Warm start GA - $\alpha = 0.0$ vs 0.1 Random vs TB	0.15	0.91	0.62	0.37
Warm start GA - $\alpha = 0.0$ vs 0.25	0.78	0.98	0.93	0.90
Warm start GA - $\alpha = 0.0$ vs 0.5	0.95	0.91	0.65	0.95
Unconstrained vs Constrained GA	0.33	0.75	0.21	0.44
Unconstrained vs Constrained PSO	0.79	0.66	0.34	0.79
Relative ref field vs Relative opt field GA	0.95	1	$7.1 \cdot 10^{-5}$	1
Relative ref field vs Relative opt field PSO	0.64	0.99	0.014	0.94

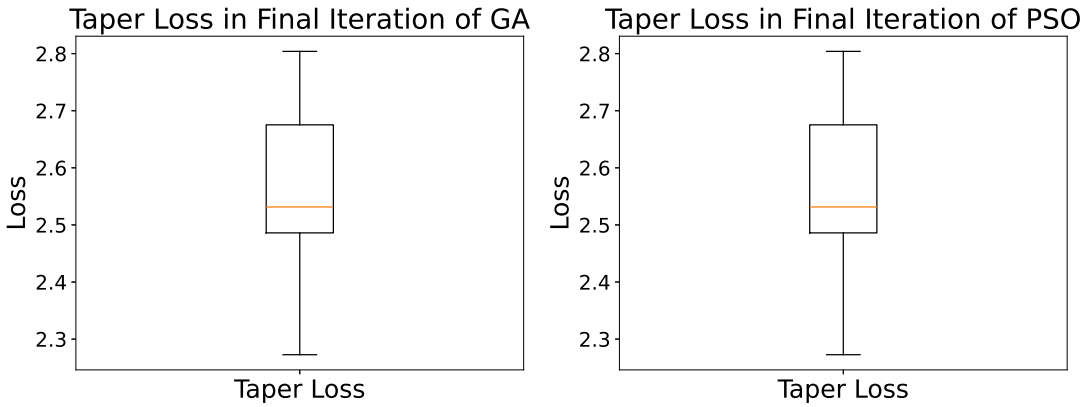
B

Result Appendix

Appendix containing Figures and Tables concerning topics in Chapter 6.

B.1 Performance

Figures and Tables relevant to Section 6.1.



(a) Best taper loss of final iteration from 20 runs of GA.

(b) Best taper loss of final iteration from 20 runs of PSO.

Figure B.1: Comparison of best taper loss of final iteration for 20 runs of GA and PSO.

Table B.1: Mean taper loss over 20 runs of GA and PSO.

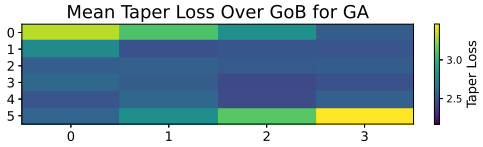
	Taper loss
GA	2.564
PSO	2.529

Table B.2: Standard deviation of final taper loss for GA and PSO.

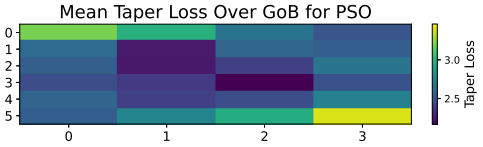
	Taper loss
GA	$1.462 \cdot 10^{-1}$
PSO	$2.364 \cdot 10^{-1}$

B.2 Comparison to Current Method

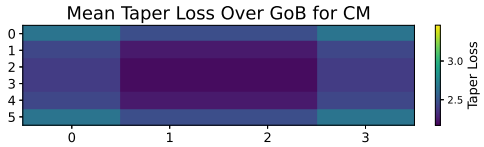
Figures and Tables relevant to Section 6.2.



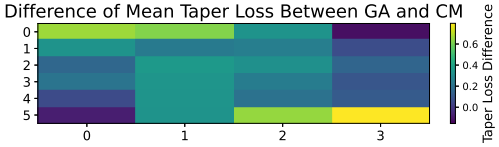
(a) Mean taper loss of five runs for beam weights optimized by GA.



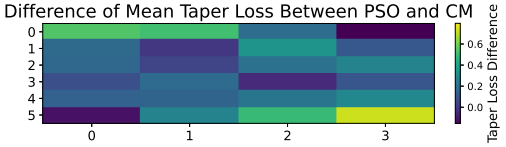
(b) Mean taper loss of five runs for beam weights optimized by PSO.



(c) Taper loss for beam weights optimized by current methods.

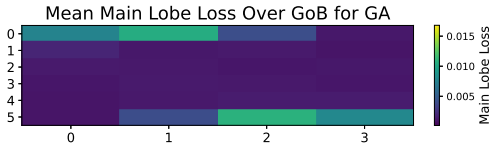


(d) Difference in taper loss for each beam weight between GA and current method. Calculated as loss of GA minus loss of CM, i.e. negative value means GA has lower loss.

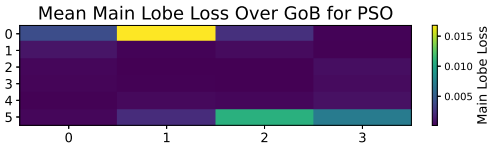


(e) Difference between taper loss for each beam weight for PSO and CM. Calculated as loss of PSO minus loss of CM, i.e. negative value means PSO has lower loss.

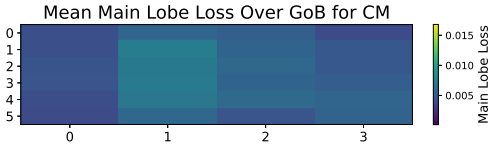
Figure B.2: Taper loss for all beams in the GoB shown in Figure 6.5, and difference in total loss between CM and both GA and PSO.



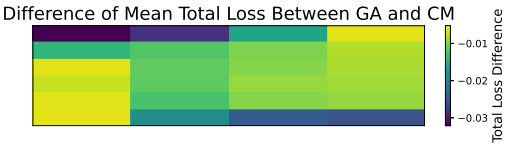
(a) Mean main lobe loss of five runs for beam weights optimized by GA.



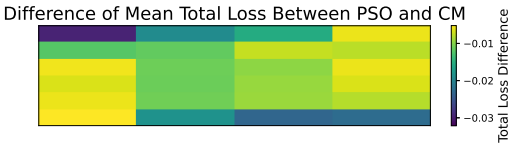
(b) Mean main lobe loss of five runs for beam weights optimized by PSO.



(c) Main lobe loss for beam weights optimized by current methods.

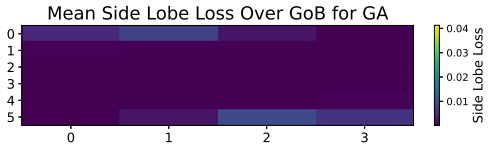


(d) Difference in main lobe loss for each beam weight between GA and current method. Calculated as loss of GA minus loss of CM, i.e. negative value means GA has lower loss.

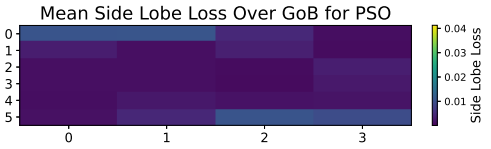


(e) Difference between main lobe loss for each beam weight for PSO and CM. Calculated as loss of PSO minus loss of CM, i.e. negative value means PSO has lower loss.

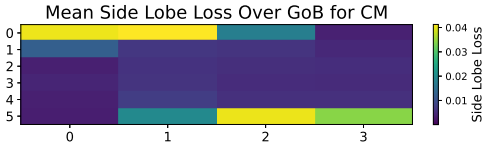
Figure B.3: Main lobe loss for all beams in the GoB shown in Figure 6.5, and difference in main lobe loss between CM and both GA and PSO.



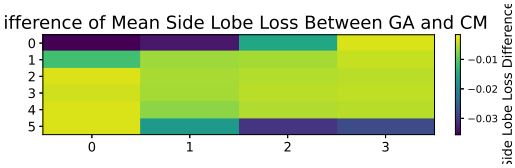
(a) Mean side lobe loss of five runs for beam weights optimized by GA.



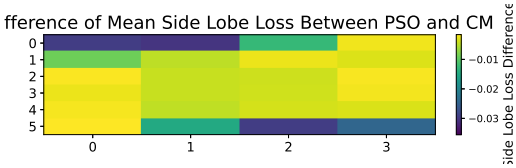
(b) Mean side lobe loss of five runs for beam weights optimized by PSO.



(c) Side lobe loss for beam weights optimized by current methods.



(d) Difference in side lobe loss for each beam weight between GA and current method. Calculated as loss of GA minus loss of CM, i.e. negative value means GA has lower loss.



(e) Difference between side lobe loss for each beam weight for PSO and CM. Calculated as loss of PSO minus loss of CM, i.e. negative value means PSO has lower loss.

Figure B.4: Side lobe loss for all beams in the GoB shown in Figure 6.5, and difference in main lobe loss between CM and both GA and PSO.

Table B.3: Mean difference in total loss and taper loss between all 24 wide beam optimized by current method compared to GA and PSO.

	Total Loss	Main Lobe Loss	Side Lobe Loss	Taper Loss
GA - Current method	$-1.32 \cdot 10^{-2}$	$-2.45 \cdot 10^{-3}$	$-1.08 \cdot 10^{-2}$	$2.79 \cdot 10^{-1}$
PSO - Current method	$-1.17 \cdot 10^{-2}$	$-3.14 \cdot 10^{-3}$	$-8.55 \cdot 10^{-3}$	$2.03 \cdot 10^{-1}$

B.3 p-Values

p-values for all two-sample t-tests performed in Chapter 6.

Table B.4: p-values for all two-sample t-tests, rounded to two significant figures.

	Total loss
Rank 1 vs Rank 1/2 GA	0.0
Rank 1 vs Rank 2 GA	$5.3 \cdot 10^{-1}$
Rank 1 vs Rank 1/2 PSO	$5.1 \cdot 10^{-15}$
Rank 1 vs Rank 2 PSO	$4.2 \cdot 10^{-3}$
Single-pol vs Dual-pol GA	$2.6 \cdot 10^{-1}$
Single-pol vs Dual-pol PSO	$9.2 \cdot 10^{-1}$

Snipp, snapp, snut, så var sagan slut.

DEPARTMENT OF MATHEMATICAL SCIENCES
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY