POLITECNICO DI TORINO

Master Degree course in Electronic Engineering

Master Degree Thesis

Inverse Design of Photonic Integrated Circuits

Supervisors
Prof. Paolo Bardella

Candidate
Jenny Capaldo

Academic Year 2025-2026

Inverse Design is a computational approach in which the desired optical functionality of a device is specified in advance, and the corresponding geometry or material distribution is automatically determined through numerical optimization. Inverse design iteratively modifies the device configuration to maximize a given performance metric, enabling the discovery of highly efficient and non-intuitive photonic structures.

In this thesis, different inverse-design methodologies for integrated photonic devices are investigated through the design and optimization of passive nanophotonic components. All the electromagnetic simulations and optimization procedures are performed using the open-source Finite-Differences Time-Domain (FDTD) software **MEEP**.

Optimization of a 1×2 Power Splitter. I designed and optimized a conventional 1×2 MMI power splitter in order to maximize the optical transmission to the two output waveguides. I performed the optimization by varying only two geometrical parameters: the MMI length and the separation between the output waveguides. For the optimization, I first generated heatmaps through parametric sweeps over progressively smaller parameter ranges, followed by refinement using the `SciPy` optimization routines integrated with MEEP. The procedure was carried out both at the operating wavelength of 1550 nm and over the broadband range from 1450 nm to 1650 nm.

For the single-wavelength case, the optimal geometry corresponds to an MMI length of $L_{\mathrm{MMI}} = 7.0\,\mu\mathrm{m}$ and a center-to-center output waveguide separation of 1.30 µm. At the output, the optimized structure achieves a total transmission of approximately 97.5 %.The optimal broadband dimensions were found to be an MMI length of 6.8 µm and an output waveguide separation of 1.34 µm, yielding an average transmission of approximately 93.8 % across the considered bandwidth. Compared with the single-wavelength design, the broadband optimization produces a flatter spectral response, as shown in Fig. 1.

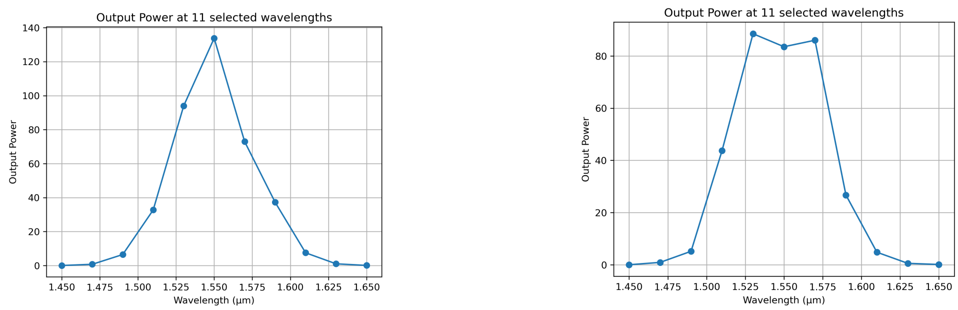
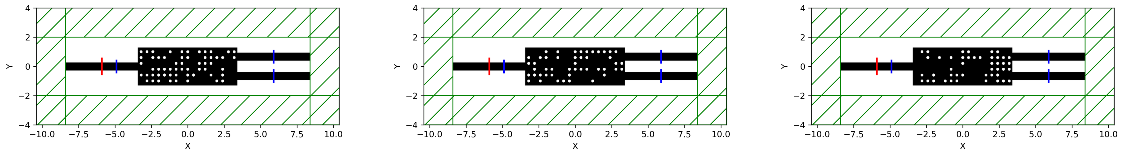
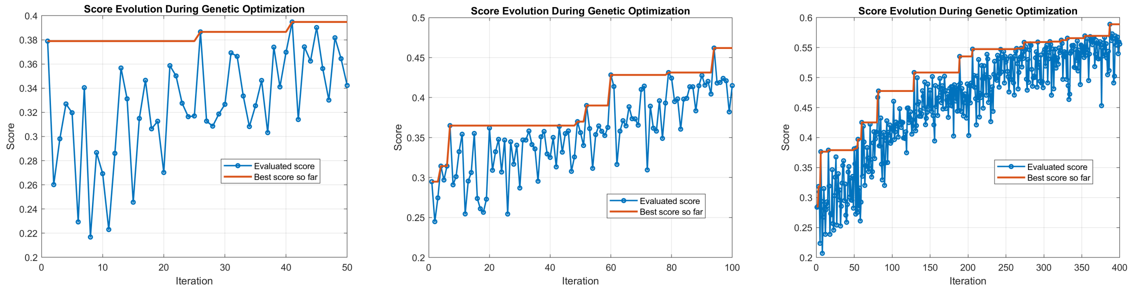


Figure 1: Spectral responses of the MMI splitter before (a) and after (b) the broadband optimization.

Pixel Placement and Optimization using MEEP Genetic Algorithm. I then introduced a pixelated design region inside the MMI section. I performed the optimization using a **Genetic Algorithm** (GA) implemented through the **pymoo** MEEP library. The algorithm searches for the optimal pixel configuration that maximizes the average transmission in the range **1450–1650 nm**. I carried out three optimization runs by varying the population size and the number of generations of the GA. Results are shown in Fig. 2.



(a) `pop_size = 10`, `n_gen = 5`, `score = 0.395`. (b) `pop_size = 10`, `n_gen = 10`, `score = 0.462`. (c) `pop_size = 20`, `n_gen = 20`, `score = 0.589`.

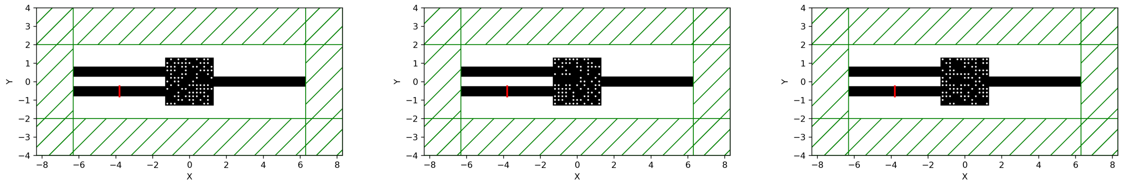


(d) `pop_size = 10`, `n_gen = 5`. (e) `pop_size = 10`, `n_gen = 10`. (f) `pop_size = 20`, `n_gen = 20`.

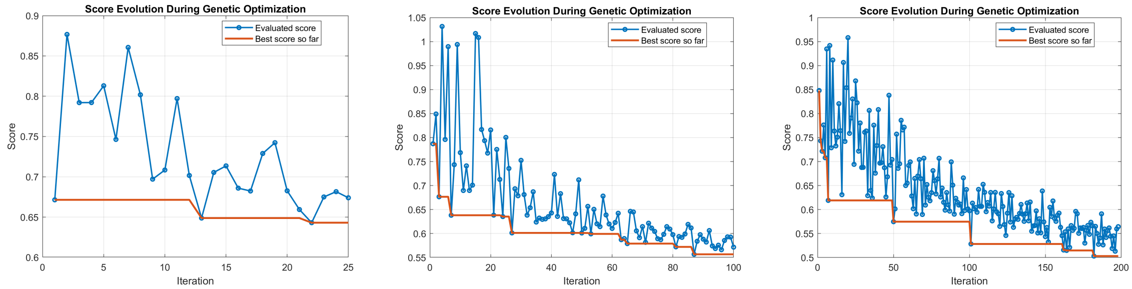
Figure 2: Best pixel configurations for the three GA optimizations (top row), with the corresponding evolution of the score values vs. the iteration number (bottom row).

Pixel Placement and Optimization for an AND Logic Gate. I then extended the pixel-based optimization methodology to the design of an AND logic gate with two optical inputs. The fitness function that I adopted is $\frac{1}{N}\sum_{i=1}^{N}(|1-T_{14,i}|+|T_{03,i}|)$, where $N = 50$, and T_{14} and T_{03} denote the normalized output optical powers for the (1,1) and (1,0) input configurations, respectively, both normalized to the power of a single input waveguide. The optimization aims to obtain T_{14} as close as possible to **1** while approaching T_{03} to **0**. I carried out three optimization runs. Results are reported in Fig. 3. The longest optimization run achieved the best result, with a minimum score of approximately **0.5**, corresponding to $T_{14} \approx 1$ and $T_{03} \approx 0.4$. Further improvements are difficult, as the optimization must simultaneously suppress the output for the (1,0) state while achieving an output power equal to that of a single input for the (1,1) state.

Design and Topology Optimization of a 1×2 Optical Splitter Using an Adjoint Solver. In the last part of my thesis, I investigated the inverse design of a 1×2 optical splitter using the **MEEP Adjoint Solver**. With this approach, a continuation strategy based on an increasing projection parameter β progressively binarizes the design, while fabrication constraints are introduced in the final stages to obtain a manufacturable structure. I used two objective functions that simultaneously minimize insertion loss and reflection while enforcing an equal power split between the two output waveguides. I studied a broadband splitter **(1260 nm–1300 nm)** and two single-wavelength splitters operating at **1300 nm** and **1550 nm**. The activation of the fabrication constraints slightly degrades the optical performance, illustrating the expected trade-off between manufacturability and efficiency. Nevertheless, the final devices remain nearly symmetric, with reflection below 1 % and an almost balanced power split (Fig. 4).

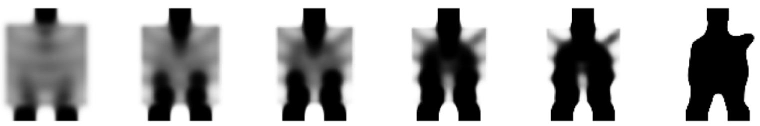


(a) `pop_size = 5, n_gen = 5, score = 0.643` (b) `pop_size = 10, n_gen = 10, score = 0.556` (c) `pop_size = 20, n_gen = 10, score = 0.503`



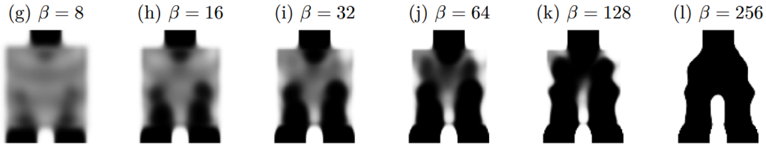
(d) `pop_size = 5, n_gen = 5.` (e) `pop_size = 10, n_gen = 10.` (f) `pop_size = 20, n_gen = 10.`

Figure 3: Best pixel configurations obtained for the three GA optimizations, with the corresponding evolution of the score values vs. the iteration number.



(a) $\beta = 8$ (b) $\beta = 16$ (c) $\beta = 32$ (d) $\beta = 64$ (e) $\beta = 128$ (f) $\beta = 240$

(g) $\beta = 8$ (h) $\beta = 16$ (i) $\beta = 32$ (j) $\beta = 64$ (k) $\beta = 128$ (l) $\beta = 256$



(m) $\beta = 8$ (n) $\beta = 16$ (o) $\beta = 32$ (p) $\beta = 64$ (q) $\beta = 128$ (r) $\beta = 256$

Figure 4: Evolution of the optimized splitter geometry for the broadband optimization (top row) and the single-wavelength optimizations at 1300 nm (middle row) and 1550 nm (bottom row). Final transmission and reflection coefficients: **Broadband:** $T_1 = 0.34946$, $T_2 = 0.46380$, $R = 6.23 \times 10^{-4}$. **1300 nm:** $T_1 = 0.43513$, $T_2 = 0.35669$, $R = 3.37 \times 10^{-3}$. **1550 nm:** $T_1 = 0.49569$, $T_2 = 0.41914$, $R = 4.31 \times 10^{-3}$

Acknowledgements

Il primo ringraziamento al **Politecnico di Torino**, che mi ha dato la possibilità di vivere questi anni. Non è stato soltanto il luogo in cui ho studiato: è stato il posto in cui sono cresciuta, ho trovato amicizie incredibili e ho costruito ricordi che porterò sempre con me.

Grazie ai miei amici del Poli: **Federica, Marco, Luca A., Gaetano, Luca S., Emanuele, Peppe BNN** e a tutti gli altri ragazzi che ho avuto il piacere di incontrare. Senza di voi questo percorso sarebbe stato soltanto una lunga serie di lezioni, esami e ore di studio.

Un ringraziamento particolare va a **Federica** e **Riccardo**, che sono stati abbastanza masochisti da scegliere me come compagna di laboratorio più volte, e per di più di loro spontanea volontà.

Grazie anche ai ragazzi di **UniTo**, che probabilmente non avrei mai conosciuto senza **Greta**. Siete davvero troppi da elencare tutti, ma un ringraziamento speciale va a **Momo**, per la sua infinita pazienza e la sua toposità, e soprattutto al mio **Bibi**, che mi è sempre accanto e che ha riempito il mio cuore.

Ai miei fratelli in Cristo — **Diego, Loris e Dani** — grazie per aver visto tutto il mio percorso, da quando ero semplicemente **Gennj** fino a diventare **Dott. Dott. Jenn Jëngïer**. Possiamo finalmente dire che ho raggiunto l'inizio della Strada della Goduria.

Infine, grazie alla mia famiglia. A **mamma e papà**, per aver creduto in me sempre, anche quando magari facevo più fatica a crederci io. E un grazie anche al mio micione **Romeo**, che ha passato infinite ore a fare la nanna sulle mie gambe o sul mio letto durante le mie lunghe (ma neanche troppo) sessioni di studio. La sua compagnia ha reso tutto un po' più bello.

A tutti voi, grazie. Questa tesi porta il mio nome, ma senza le persone che hanno fatto parte di questo percorso non sarebbe mai esistita.

Grazie.

Jenny Catapaldo

Contents

Chapter 1

Introduction

The increasing complexity of integrated photonic devices has made the design process progressively more challenging. Traditional design approaches are typically based on physical intuition and repeated parameter tuning, requiring numerous simulations before reaching satisfactory device performance. As the number of design variables increases, these methods become increasingly time-consuming and may fail to identify the best possible solution.

Inverse design addresses this limitation by reversing the conventional design process. Instead of starting from a predefined geometry and evaluating its optical response, the desired device performance is specified first, while numerical optimization algorithms automatically determine the geometry that best satisfies the design objectives. In this way, the design problem is formulated as an optimization task, where the material distribution or the device geometry is iteratively modified until the target optical behavior is achieved.

This approach has become one of the most powerful tools in modern nanophotonics, enabling the realization of compact and high-performance devices that would be extremely difficult to obtain through conventional design techniques. Depending on the application, different optimization strategies can be adopted, including evolutionary algorithms, gradient-based adjoint optimization, and, more recently, machine learning methods.

The following sections present the main inverse-design techniques adopted in integrated photonics and introduce the methods employed in this work.

1.1 Introduction to Inverse Design

The continuous growth of integrated photonics has led to increasingly compact and sophisticated optical devices. As the required functionalities become more complex, designing photonic components through conventional approaches becomes progressively more challenging. Traditional photonic design, often referred to as *forward design*, relies on the designer's physical intuition and experience. Starting from an initial geometry, the device is repeatedly modified and simulated until the desired optical response is achieved. This methodology becomes inefficient when the number of design parameters is large or when the optimal geometry is not intuitively evident.

To overcome these limitations, inverse design has emerged as a powerful alternative. Instead of starting from a predefined geometry, the desired optical functionality is specified first, while the geometry is automatically determined through an optimization process. The design problem is therefore reformulated as an optimization task, where the objective is to minimize (or maximize) a suitable performance metric by modifying the device geometry or the material distribution inside a predefined design region.

One of the major advantages of inverse design is its ability to explore extremely large design spaces that would be impractical to investigate manually. This makes it possible to discover compact and high-performance photonic structures whose geometries are often highly non-intuitive. As a result, inverse design has become one of the most powerful computational tools in modern nanophotonics and integrated photonics, enabling the realization of devices that would be difficult, or even impossible, to obtain using traditional design methodologies.

1.2 General Inverse Design Workflow

Although different inverse-design methodologies employ different optimization algorithms, they all follow the same general workflow. The optimization process can be summarized by the following steps:

- **Definition of the design region.** A portion of the photonic device is selected as the design region, namely the area whose geometry or material distribution can be modified during the optimization. The remaining parts of the structure, such as the input and output waveguides, are kept fixed.
- **Definition of the design variables.** The design region is parameterized through a set of design variables. Depending on the adopted methodology, these variables may represent binary pixels, continuous material densities, geometric parameters, or more complex shape descriptions.
- **Electromagnetic simulation.** For each design configuration, the optical response of the device is evaluated by numerically solving Maxwell's equations, typically using methods such as the Finite-Difference Time-Domain (FDTD) technique. From the computed fields, quantities such as transmission, reflection, insertion loss, or field confinement are extracted.
- **Evaluation of the objective function.** The simulated optical quantities are combined into one or more objective functions, also known as figures of merit (FoMs), which quantify how well the current design satisfies the desired specifications.
- **Design update.** The optimization algorithm modifies the design variables according to the value of the objective function. Depending on the adopted methodology, this update may exploit gradient information, evolutionary operators, heuristic search strategies, or machine-learning models.
- **Convergence.** The optimization proceeds iteratively by alternating simulations and design updates until a stopping criterion is reached, such as a maximum number

of iterations, convergence of the objective function, or the achievement of a target optical performance. The resulting geometry represents the optimized device while satisfying any imposed fabrication constraints.

Although the optimization algorithm may vary significantly from one inverse-design methodology to another, these fundamental steps are common to most approaches employed in integrated photonics.

1.3 Inverse Design Methods and Applications

Over the past decade, several optimization strategies have been proposed for inverse design in integrated photonics. Although they all follow the general workflow described in the previous section, they differ in the way the design variables are updated during the optimization process.

The earliest inverse-design approaches mainly relied on gradient-free optimization algorithms. These methods do not require gradient information and instead explore the design space by evaluating a large number of candidate geometries according to a predefined figure of merit. Among the most widely adopted techniques are Genetic Algorithms (GA), Particle Swarm Optimization (PSO), and Direct Binary Search (DBS). Although computationally expensive, these methods are relatively robust and can optimize highly complex design problems.

Particle Swarm Optimization has been successfully employed for the design of ultra-compact wavelength multiplexers and demultiplexers. For example, a silicon-on-insulator wavelength multiplexer/demultiplexer supported by subwavelength gratings (SWGs) was proposed, where the optimization algorithm automatically determined the optimal geometrical parameters of the device to minimize insertion loss while reducing its footprint [8].

Another popular gradient-free approach is the Direct Binary Search algorithm, in which the design region is discretized into binary pixels that are iteratively modified to improve device performance. An improved Rotation Direct Binary Search (RDBS) algorithm was introduced, extending the conventional DBS approach by allowing pixel rotation in addition to binary state changes (Fig. 1.1). This additional degree of freedom significantly improves design flexibility, enabling the realization of ultra-compact multi-functional wavelength demultiplexers capable of simultaneously performing wavelength demultiplexing and mode conversion [7].

Despite their flexibility, heuristic optimization algorithms generally require a very large number of electromagnetic simulations, making them computationally demanding for devices characterized by thousands of design variables.

A major progress in inverse photonic design was the introduction of gradient-based optimization through the adjoint method. Unlike heuristic approaches, the adjoint method computes the sensitivity of the objective function with respect to all design variables using only two electromagnetic simulations per optimization iteration, independently of the number of optimization variables (Fig. 1.2) [4]. This reduction in computational cost enables the optimization of significantly larger and more complex nanophotonic structures.

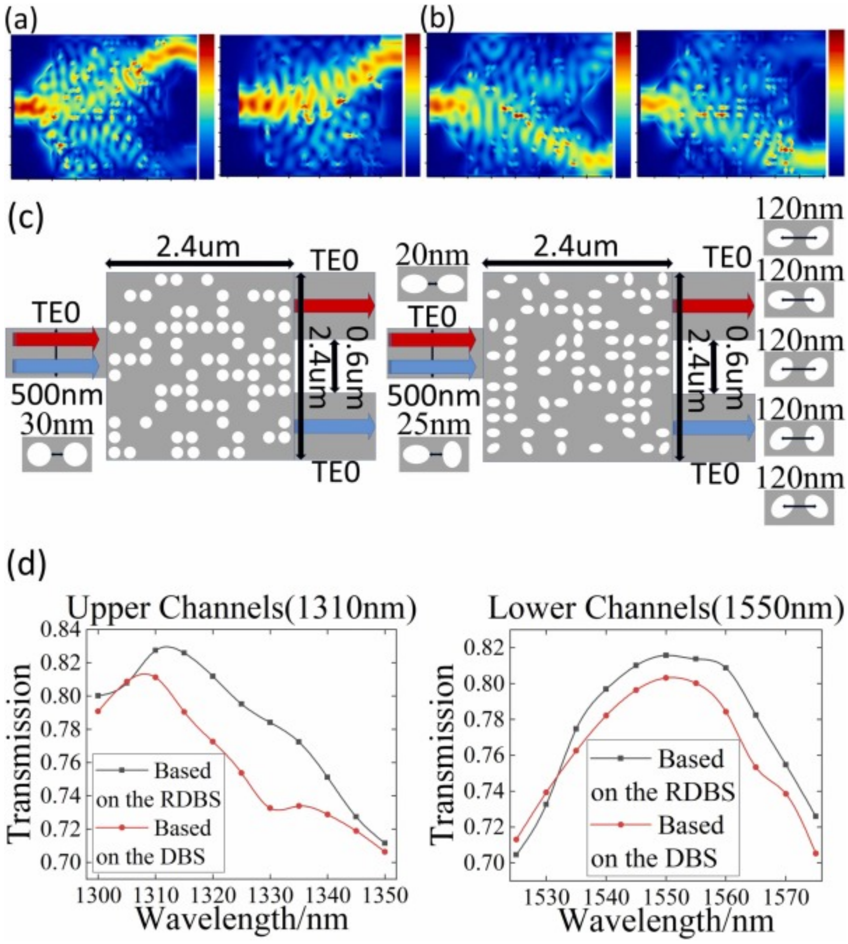


Figure 1.1: Comparison of the optimized device geometries and optical performance obtained using the conventional Direct Binary Search (DBS) algorithm and the proposed Rotation Direct Binary Search (RDBS) algorithm. Reproduced from [7].

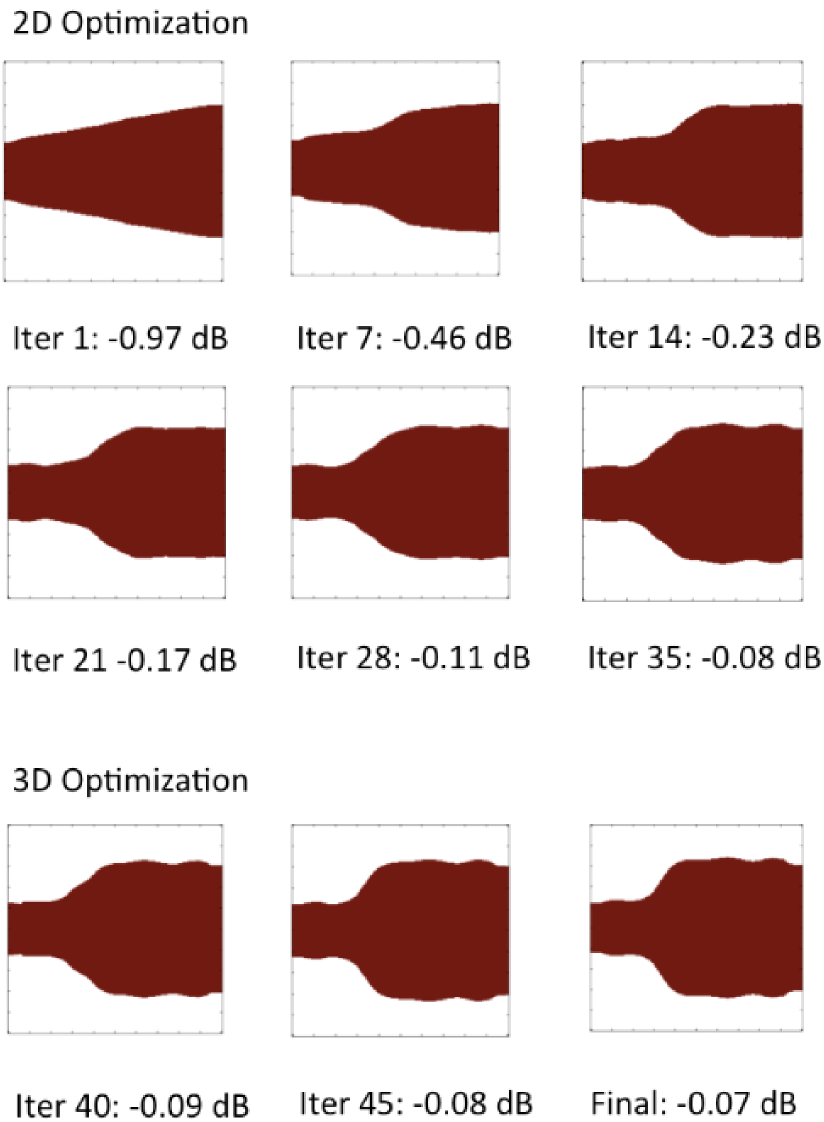


Figure 1.2: Evolution of the Y-junction geometry of a splitter designed using the adjoint optimization process. The iteration number (Iter) and the corresponding insertion loss (in dB) are reported for each intermediate design. Reproduced from [4].

More recently, the adjoint method has been extended to directly incorporate fabrication constraints into the optimization process. Rather than optimizing only the optical performance, the algorithm simultaneously takes into account practical fabrication limits, such as minimum feature size and shape constraints, producing geometries that can be fabricated using standard silicon photonics technologies [5].

Machine learning has recently become a promising alternative for inverse photonic design. Instead of repeatedly solving Maxwell's equations during optimization, deep neural networks are trained to learn the relationship between device geometry and optical response from a large dataset of simulated structures. Once trained, these models can predict both the optical response of a given device (forward design) and the geometry required to achieve a desired optical functionality (inverse design) almost instantaneously.

For instance, deep learning has been employed for the inverse design of all-optical nonlinear plasmonic ring resonator switches (Fig. 1.3), obtaining accurate spectral predictions while reducing the computational cost associated with conventional electromagnetic simulations [1].

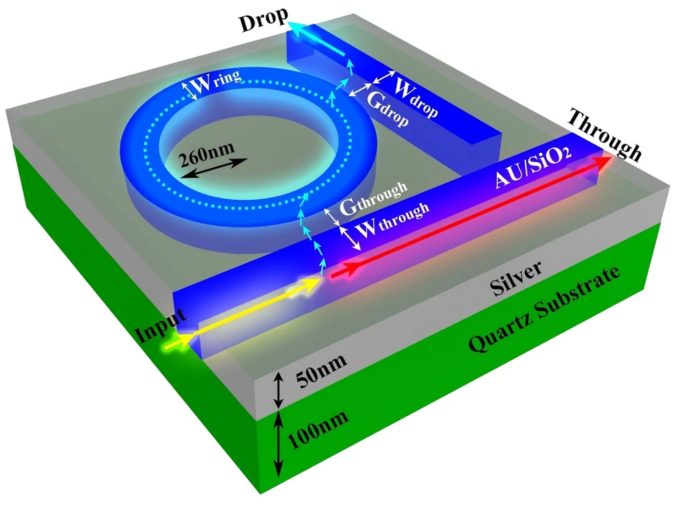


Figure 1.3: Schematic of the circular all-optical Nonlinear Plasmonic Ring Resonator Switch configuration. Reproduced from [1].

Similarly, deep neural networks have been employed to design ultra-compact silicon-on-insulator 1×2 power splitters with arbitrary splitting ratios, achieving transmission efficiencies above 90 % while generating optimized device layouts in a fraction of a second (Fig. 1.4) [6].

Among the most common applications are optical power splitters, wavelength multiplexers and demultiplexers, optical switches, mode converters, waveguide crossings, and directional couplers. In many cases, inverse design has demonstrated the ability to simultaneously improve multiple performance metrics, including transmission efficiency, insertion loss, reflection, crosstalk, bandwidth, and device footprint.

The examples presented in the literature show that inverse design is no longer used only for simple demonstrations but has become a useful method for designing complex

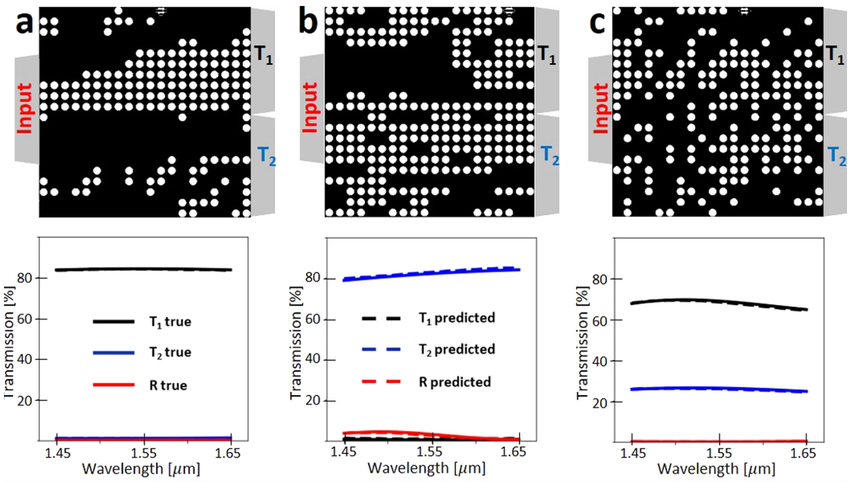


Figure 1.4: Predicted transmission values at the output ports of the 1×2 optical splitter obtained using a deep neural network. Reproduced from [6].

photonic devices. Continuous improvements in optimization algorithms, machine learning, and techniques that take fabrication into account are expected to make it even more important for future photonic integrated circuits.

1.4 Simulation Algorithms for Photonic Devices

The numerical simulation of photonic devices relies on different computational methods, each characterized by specific assumptions, advantages, and limitations. Among the most widely used techniques are the Beam Propagation Method (BPM), the Eigenmode Expansion Method (EME), and the Finite-Difference Time-Domain (FDTD) method [2].

The **Beam Propagation Method (BPM)** computes the evolution of the slow components of the electromagnetic field along a preferred propagation direction. Because of this approximation, it is computationally efficient and well suited for smoothly varying waveguide structures. However, it cannot accurately model strong reflections, multiple scattering, or complex geometries. BPM is mainly applicable to relatively large devices with low refractive-index contrast Δn, where the approximation remains valid.

The **Eigenmode Expansion Method (EME)** represents the electromagnetic field as a superposition of the eigenmodes supported by consecutive sections of the device. This approach is particularly efficient for structures that can be naturally divided into uniform segments, such as tapered waveguides. EME can accurately handle larger refractive-index contrasts, and it is also well suited for modeling relatively large devices.

The **Finite Difference Time Domain (FDTD)** method directly solves Maxwell's equations by discretizing both space and time, allowing the electromagnetic field to evolve throughout the computational domain. Unlike BPM and EME, it does not rely on approximations regarding the propagation direction or the device geometry, making it suitable for modeling reflections, scattering, high-index-contrast structures, and broadband devices. FDTD offers the widest range of applicability, remaining accurate even for compact nanophotonic devices characterized by large refractive-index contrasts.

Figure 1.5 shows the applicability domain of the three simulation methods as a function of the refractive-index contrast Δn and the device length. As can be observed, BPM is mainly suitable for relatively long devices with low Δn, EME extends the range of applicability to larger index contrasts and relatively large structures, while FDTD covers the widest region, including compact devices with high refractive-index contrast.

For these reasons, all the simulations presented in this thesis were performed using MEEP, an open-source electromagnetic simulation software based on the FDTD method. Its flexibility, together with its native support for optimization algorithms and adjoint-based methods, makes it particularly suitable for the inverse-design methodologies investigated in this work [4].

1.5 Scope of this Thesis

As explained before, a wide variety of optimization strategies for inverse design are currently available, each characterized by different computational costs, levels of complexity,

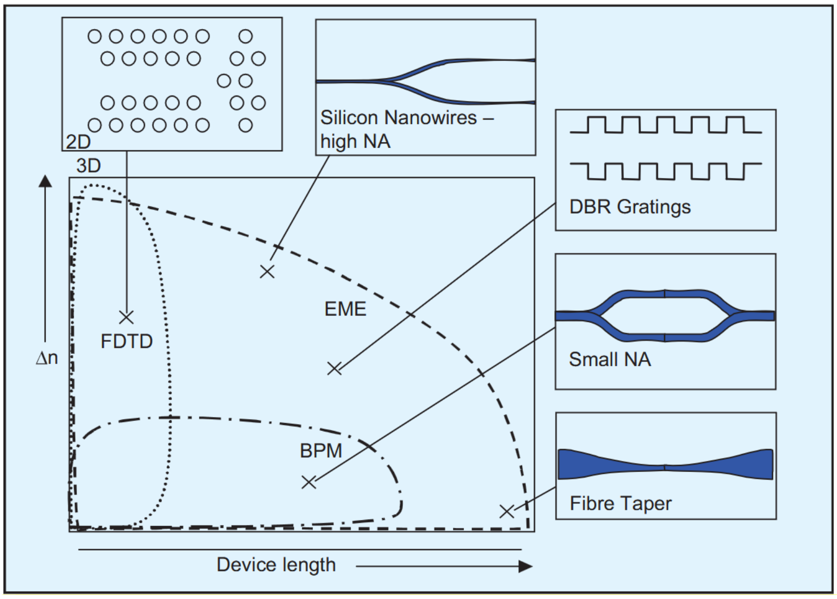


Figure 1.5: Applicability domain of BPM, EME, and FDTD as a function of device length and refractive-index contrast Δn. Reproduced from [2].

and design capabilities. For this reason, it is interesting to investigate how these methodologies perform when applied to similar nanophotonic structures and to evaluate their respective advantages and limitations.

The main objective of this thesis is, therefore, to experimentally investigate different inverse-design approaches, ranging from relatively simple optimization techniques to more advanced and computationally demanding methodologies. Rather than focusing on a single optimization algorithm, this work aims to gradually explore the inverse-design process by applying different optimization strategies to closely related photonic devices.

A fundamental aspect of this work is that the entire design process is carried out using the open-source electromagnetic simulation software **MEEP**, which provides a common simulation environment for all the investigated devices. Depending on the optimization problem, MEEP is combined with different optimization libraries and algorithms while maintaining the same FDTD simulation framework.

The following section presents in detail how the thesis is structured and the analyzes conducted in this work, focusing on the structures used and the type of inverse-design approach applied in each case.

1.6 Structure of the Thesis

The selected test devices are all based on the same fundamental MMI platform and share a similar optical functionality. Starting from the optimization of a conventional 1×2 splitter, the complexity of the design problems is gradually increased by introducing pixel-based inverse design, different optimization algorithms, and, eventually, a different device functionality through the realization of an all optical AND logic gate.

The thesis is organized as follows:

- **Chapter 2** introduces the simulation software **MEEP**, describing its working principles based on the Finite-Difference Time-Domain (FDTD) method, its main functionalities, and its role in electromagnetic modeling. The chapter also outlines the installation procedure and the specific features and tools used throughout this thesis for simulation and optimization tasks.

- **Chapter 3** presents the optimization of a conventional 1×2 MMI power splitter by varying only two geometrical parameters: the MMI length and the separation between the output waveguides. The optimization is carried out using a parametric sweep visualized through a heatmap and the routines available in the **MEEP SciPy** library, where a figure of merit (FOM) is defined and used to determine the optimal parameter values. The problem is analyzed both at a single wavelength and over a range of wavelengths.

- **Chapter 4** introduces a pixel-based inverse-design approach for the same splitter structure. The design region is discretized into binary pixels whose material distribution is optimized using a **Genetic Algorithm** implemented through the **MEEP pymoo** optimization library, where a fitness function is defined to guide the optimization process.

- **Chapter 5** applies the same pixel-placement approach using a Genetic Algorithm implemented with **pymoo** to the design of an optical **AND logic gate**. Although the optimization methodology remains essentially unchanged, the fitness function and the device functionality are modified to implement a logic operation.

- **Chapter 6** investigates topology optimization through the **MEEP Adjoint Solver**. The Adjoint Inverse Design Method is employed to optimize 1×2 optical splitters operating in both broadband and single-wavelength regimes while incorporating fabrication constraints into the optimization process.

- **Chapter 7** presents the conclusions of this work. The obtained results are discussed by comparing the investigated inverse-design methodologies, highlighting their respective advantages and limitations. The chapter also outlines possible future developments and concludes with final remarks on the role of inverse design in integrated photonics.

In each chapter, the considered device is first introduced together with its specifications, followed by a description of the optimization algorithm employed. The obtained results are then analyzed in detail, allowing for a clear evaluation of the performance of each inverse-design methodology.

Chapter 2

MIT Electromagnetic Equation Propagation

The software chosen for this work is MIT Electromagnetic Equation Propagation (MEEP), an open-source simulation tool for computational electromagnetics based on the Finite-Difference Time-Domain (FDTD) method. It was developed at the Massachusetts Institute of Technology and is widely used to model the propagation of electromagnetic waves in complex structures, such as waveguides, photonic crystals, and optical devices.

MEEP enables the numerical solution of Maxwell's equations in the time domain, providing detailed information about the behavior of electric and magnetic fields in materials with different physical properties.

The choice of MEEP for this work is motivated by several factors:

- **Integration with Python**: provides a comprehensive Python interface, allowing simulations to be implemented in a flexible, modular, and reproducible way.
- **Open-source software**: being freely available, it ensures transparency and allows the code to be customized according to the specific needs of the research.
- **Scientific reliability**: MEEP is widely adopted in the photonics and computational electromagnetics communities, providing accurate and well-validated simulation results.
- **Versatility**: it supports the simulation of a wide range of electromagnetic phenomena, including complex geometries, dispersive and anisotropic materials, and both two-dimensional and three-dimensional structures.
- **Automation and scripting**: thanks to its Python interface, simulations can be easily automated, enabling parametric analyzes, optimization workflows, and seamless integration with scientific libraries such as `NumPy`, `SciPy`, and `pymoo`.
- **Broadband simulations**: a single time-domain simulation can provide the optical response over a wide wavelength range through Fourier analysis, making MEEP particularly efficient for spectral characterization.

- **Built-in analysis tools**: MEEP includes flux monitors and DFT monitors for evaluating transmitted and reflected power, as well as electric and magnetic field distributions at selected frequencies/wavelengths.
- **Inverse design capabilities**: MEEP provides dedicated optimization libraries, such as the `MEEP.adjoint` module, which enable gradient-based inverse design of photonic devices through the adjoint optimization method.

2.1 Finite-Difference Time-Domain Method

The FDTD method is a numerical technique used to solve Maxwell's equations, which describe the behavior of electromagnetic fields. It is widely used to simulate how electromagnetic waves (such as light or radio waves) propagate in space and interact with materials and structures.

The main idea behind the FDTD method is to discretize both space and time:

- the computational domain is divided into a **grid** (a lattice of small cells);
- time is divided into **discrete steps**;
- the electric and magnetic fields are computed **step by step in time**.

At each point of the grid:

- the electric field E and the magnetic field H are updated alternately in time;
- each update depends on the values of the fields at neighboring points.

This process can be interpreted as a frame-by-frame simulation of the evolution of the electromagnetic wave.

From a more technical point of view, the FDTD method:

- discretizes Maxwell's equations using finite-difference approximations;
- employs a staggered grid scheme, known as the **Yee grid**;
- updates the fields through a time-stepping procedure.

The algorithm follows a cyclic process:

1. update the electric field E using the magnetic field H;
2. update the magnetic field H using the electric field E;
3. repeat the process over successive time steps.

This iterative procedure continues until the desired simulation time or convergence is reached.

The FDTD method is widely used due to several advantages:

- **Time-domain formulation**: it allows for broadband analysis, meaning multiple wavelengths can be obtained from a single simulation;
- **Geometrical flexibility**: it can handle complex structures and material distributions;
- **Physical intuition**: it provides a direct visualization of wave propagation over time.

2.2 Installation and Configuration of the Simulation Environment

For the development of the electromagnetic simulation code, MEEP was used, an open-source software based on the Finite-Difference Time-Domain (FDTD) method, executed on Windows 11 through WSL2 and Python managed with Miniconda.

Choice of the Environment

MEEP is no longer natively supported on Windows; therefore, the most stable solution is the use of WSL2 (Windows Subsystem for Linux 2). This approach allows MEEP to run in a Linux environment, providing performance comparable to a native Linux system and full compatibility with Python.

Installation of WSL2 and Ubuntu

WSL2 was installed using PowerShell with administrative privileges:

```
wsl --install
```

After restarting the system, an Ubuntu distribution (version 22.04 or 24.04) was installed from the Microsoft Store. The system packages were then updated within the Linux shell:

```
sudo apt update && sudo apt upgrade -y
```

Installation of Miniconda

Miniconda is a lightweight Python environment manager used to create isolated environments and automatically handle dependencies.

It was installed within WSL as follows:

```
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
bash Miniconda3-latest-Linux-x86_64.sh
```

During installation, automatic initialization of Conda was enabled. The configuration was then activated:

```
source ~/.bashrc
conda config --set auto_activate_base false
```

Creation of the Python Environment

To isolate dependencies, a dedicated environment was created:

```
conda create -n meep-env python=3.11 -y
conda activate meep-env
```

Installation of MEEP

The correct Python package for MEEP is `pyMEEP`, while the module is imported as `MEEP`. Installation was performed via the `conda-forge` channel:

```
conda install -c conda-forge pymeep mpich mpb h5utils
```

The `mpb` and `h5utils` libraries are auxiliary components required by the software. Conda automatically manages native dependencies such as MPI, HDF5, and FFTW.

The installation was verified using:

```
python3 -c "import meep as mp; print(mp.__version__)"
```

Execution of Official Examples

In recent Conda-based installations, MEEP examples are not included as Python modules. Therefore, it is necessary to download the official repository.

The standard procedure consists of cloning the repository:

```
git clone https://github.com/NanoComp/meep.git
cd meep/python/examples
```

This directory contains several complete examples (e.g., waveguides, resonant cavities, and photonic crystals), which can be executed directly:

```
conda activate meep-env
python3 straight-waveguide.py
```

If the `git` command is not available, it can be installed using:

```
sudo apt install git
```

Alternatively, the repository can be downloaded manually as a `.zip` file from the official website, extracted, and accessed through the `python/examples` directory.

This approach provides access to the same scripts used in the official documentation, which can be executed, analyzed, and modified according to specific simulation needs. In particular, the examples `bent-waveguide.py` and `straight-waveguide.py` were used as initial references to understand the structure of MEEP scripts and the setup of simulations, forming the basis for the development of the codes used in this work.

2.3 MEEP Features Used in This Work

In this thesis, several functionalities provided by MEEP were employed for the simulation, analysis, and optimization of integrated photonic devices. The main features used during the different stages of the work are summarized below.

- **Geometry definition**: the photonic devices were modeled by combining simple geometric primitives, such as rectangular waveguides and cylindrical inclusions. This approach was used to realize conventional MMI splitters, pixel-based structures, and the design regions employed for inverse design.
- **Eigenmode excitation**: all simulations employed `EigenModeSource` to excite the fundamental guided mode of the input waveguide, ensuring an accurate representation of the optical signal propagating through the device.
- **Broadband simulations**: Gaussian pulse sources were adopted to excite a wide wavelength range, allowing the transmission spectra to be evaluated over multiple wavelengths with a single FDTD simulation.
- **Boundary conditions**: Perfectly Matched Layers (PMLs) were used at the boundaries of the computational domain to suppress artificial reflections and emulate an open space.
- **Power and field monitoring**: flux monitors were used to compute the transmitted optical power and the transmission coefficients, while DFT monitors were employed to reconstruct the electric field distribution and the field intensity at selected wavelengths.
- **Mode decomposition**: MEEP's eigenmode decomposition was exploited to evaluate the power carried by the fundamental guided mode, providing a more accurate estimation of the transmission.
- **Automation and scripting**: thanks to its Python interface, all the simulations were fully automated, enabling parametric sweeps, heatmap generation, data analysis, and integration with external optimization libraries.
- **Optimization**: MEEP was coupled with external optimization libraries, including `SciPy` for the optimization of the MMI dimensions and `pymoo` for the genetic-algorithm optimization of the pixel configurations.
- **Inverse design**: the `meep.adjoint` library was employed to perform gradient-based inverse design using the adjoint optimization method. In particular, the `MaterialGrid` representation of the design region, together with spatial filtering, projection techniques, and fabrication constraints, was used to automatically synthesize optimized photonic devices.

Overall, MEEP proved to be the key software tool throughout this thesis. Its flexibility, extensive Python interface, and advanced optimization capabilities made it possible to perform all the simulation, analysis, and optimization tasks presented in this work, from the modeling of conventional MMI devices to the implementation of genetic algorithms and gradient-based inverse design methods. The optimization strategies and inverse design approaches adopted in this work will be discussed in detail in the following chapters.

Chapter 3

Optimization of the MMI Length and Output Waveguide Separation in a 1×2 Power Splitter

The first step of the design process consisted of the development of a 1×2 power splitter based on a multimode interference section. Starting from the initial device geometry, MEEP was employed to simulate the device and perform a parametric optimization aimed at maximizing the optical power delivered to the two output waveguides while ensuring equal power distribution.

The optimization focused exclusively on two geometrical parameters, namely the length of the MMI section and the separation between the output waveguides. By systematically varying these parameters, the configuration providing the highest transmission efficiency and the most balanced power splitting was identified.

The simulation parameters and device specifications adopted in this study are reported below, followed by a detailed description of the optimization procedure used to identify the optimal values of the MMI length and the output waveguide separation.

- Working wavelength: 1550 nm
- Core refractive index: $n_{\mathrm{core}} = 3.23$
- Cladding refractive index: $n_{\mathrm{clad}} = 1.44$
- Width of the input and output waveguides: 0.55 µm
- Length of the input and output waveguides: 5 µm
- MMI width: 2.6 µm
- MMI length: optimized parameter
- Center-to-center separation between the output waveguides: optimized parameter
- Spatial resolution: 40 points/um

- Perfectly matched layer (PML) thickness: 2 µm

The simulations were performed considering a silicon-on-insulator (SOI) platform. A refractive index of $n_{\text{core}} = 3.23$ was assigned to the waveguide core, corresponding to silicon at the operating wavelength of 1550 nm, while the surrounding cladding was modeled with a refractive index of $n_{\text{clad}} = 1.44$, representative of silicon dioxide. This material platform was selected because of its high index contrast, which enables strong optical confinement and allows the realization of compact photonic devices with reduced footprint.

A Gaussian source was employed in all simulations. The source bandwidth was set equal to $0.15\, f_c$, where the center frequency f_c is defined as the inverse of the operating wavelength,

$$f_c = \frac{1}{\lambda_0},$$

with $\lambda_0 = 1550$ nm. Therefore, the source frequency width was chosen as

$$\Delta f = 0.15\, f_c.$$

The source was configured to excite the fundamental transverse electric (TE) mode of the input waveguide. Since the fundamental mode is the mode of interest in this work, the excitation was restricted to the first guided eigenmode of the structure.

This choice ensures adequate spectral confinement around the operating wavelength while maintaining a sufficiently broad excitation for accurate FDTD simulations.

It is important to specify that in MEEP the frequency is calculated as the inverse of the wavelength, since the software uses normalized units in which the speed of light is set to one. In this unit system, the frequency therefore represents the fundamental quantity used in the simulations, while the wavelength is derived directly as its inverse. Consequently, the definition of the spectral bandwidth and the sampling of points must be carried out in frequency space rather than directly in wavelength space.

An example of the device geometry adopted in this work is shown in Fig. 3.1.

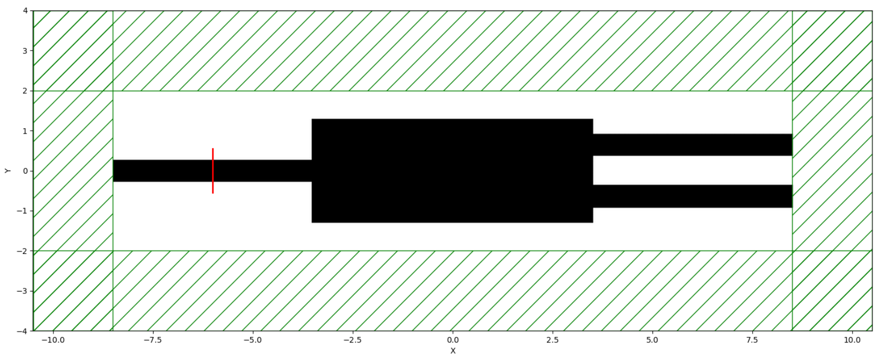


Figure 3.1: Representative layout of the 1×2 MMI power splitter

3.1 MMIs operation principles

A Multimode Interferer (MMI) device is a key photonic integrated circuit component, acting as a waveguide-based optical power splitter or combiner. It uses a broad, multimode waveguide to create single or multiple self-images of an input field at periodic intervals through interference, providing superior performance over directional couplers.

These devices operate based on self-imaging, where the different propagation constants of a multimode region cause input light to form periodic interference patterns. MMI is commonly used for 1×N/N×N power splitters, N×N combiners, Mach-Zehnder interferometers, and optical switches.

The MMI chosen for this thesis is a 1×2 power splitter. The ideal 1×2 MMI would be a 50-50 splitter such that light enters along one path and exits along two paths, with half the power in each exit path.

An ideal 50-50 splitter is nearly impossible in practice. Optical loss will always occur as light travels through an MMI, resulting in total output power of a real MMI being smaller than the total input power. Additionally, the light propagates through the rectangular box in multiple modes while also experiencing reflection and interference. This leads to mathematical models and equations that are too complex to solve by hand; thus, MMIs are typically designed through computer simulations.

3.1.1 Self Imaging Principle

The input field profile is replicated, both in amplitude and phase, at specific positions along the MMI section, known as self-imaging lengths L_{si}. This self-imaging effect is caused by the constructive interference of the excited modes, which occurs when their relative phases periodically realign.

If the **excitation** is **not symmetric**, at lengths equal to submultiples of L_{si}, it is possible to observe:

- self image at $z = L_{\mathrm{si}}$
- mirrored image at $z = L_{\mathrm{si}}/2$
- self and mirrored images at $z = L_{\mathrm{si}}/4$ and $z = 3L_{\mathrm{si}}/4$.

An asymmetrically excited field also exhibits regions where approximate multiple replicas of the initial field are formed (Fig. 3.2).

If the **excitation** is **symmetric**, only even order harmonics are generated in the MMI (m= 0,2,4,...). In this situation, self and mirrored image correspond to the self-image, located in position $z = kL_{\mathrm{si}}/8$, with k integer (1,2,3, ...).

In this case, multiple image peaks will occur (Fig. 3.3) at

$$z = \frac{1}{N}\frac{L_{\mathrm{si}}}{8}, \quad \text{with N integer (2,3,4,...)}.$$

The theoretical background presented in this section, including the self-imaging equations and the illustrative figures, is based on the lecture material provided by Prof. Paolo Bardella.

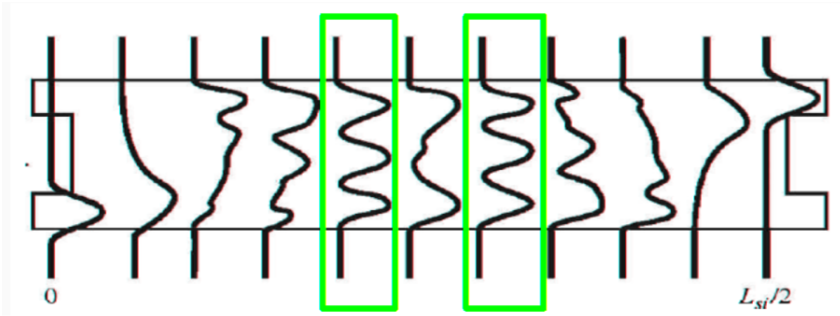


Figure 3.2: Example of asymmetric self-imaging in the MMI section. Reproduced from the lecture notes of Prof. Paolo Bardella.

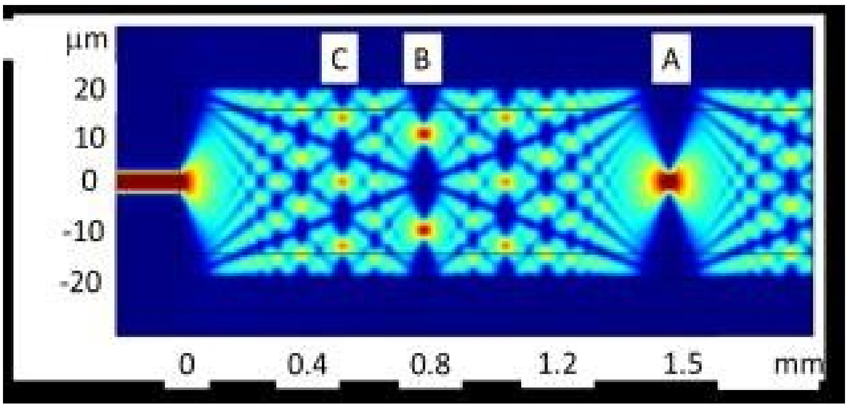


Figure 3.3: Example of symmetric self-imaging in the MMI section. Reproduced from the lecture notes of Prof. Paolo Bardella.

3.2 Theoretical MMI Length

The first step was the calculation of the theoretical MMI length in order to determine the optimal truncation point that maximizes the output power at the two ports.

Since the structure is symmetric, the first position at which the two multiple images peaks can be seen occur at:

$$z = \frac{1}{N}\frac{L_{\mathrm{si}}}{8}, \quad \text{with } N = 2$$

The corresponding value for L_{si} is:

$$L_{\mathrm{si}} = \frac{8 n_{\mathrm{eff}} W^2}{\lambda_0}$$

with:

- $n_{\mathrm{eff}} = 2.8$
- $W = 2.6\,\mu\mathrm{m}$
- $\lambda_0 = 1550\,\mathrm{nm}$

where n_{eff} is the effective refractive index of the fundamental TE mode supported by the MMI section, obtained through an eigenmode analysis performed with RSoft.

Resulting in $L_{\mathrm{si}} = 97.7\,\mu\mathrm{m}$ and $L_{\mathrm{MMI}} = 6.2\,\mu\mathrm{m}$.

The effective real value of L_{MMI} has been evaluated more accurately using optimization methods in MEEP. All details will be explained in 3.4

3.3 Results Obtained with the Theoretical MMI Length

A preliminary simulation of the MMI structure was carried out using the theoretical value of the MMI length obtained from the self-imaging equations. The output waveguide offset was initially set to Y_{offset}=0.5 µm, while all the remaining simulation parameters and device specifications were kept equal to those described in the previous section. The purpose of this analysis was to assess the performance of the initial design before applying the optimization procedure. The corresponding results are presented below.

Several flux monitors were placed along the splitter structure in order to evaluate the optical power distribution, transmission, and field evolution throughout the device. This arrangement allowed the corresponding quantities to be monitored at different positions along the propagation direction. A qualitative representation of the monitor placement is shown in Fig. 3.4.

As shown in Fig. 3.5, the total transmitted power gradually decreases along the structure, reaching approximately 75 % of the input power at the output. This behavior could be due to radiation losses caused by the abrupt transition between the input waveguide and the multimode section, as well as by the imperfect coupling between the self-images generated inside the MMI and the output waveguides. Consequently, a fraction of the

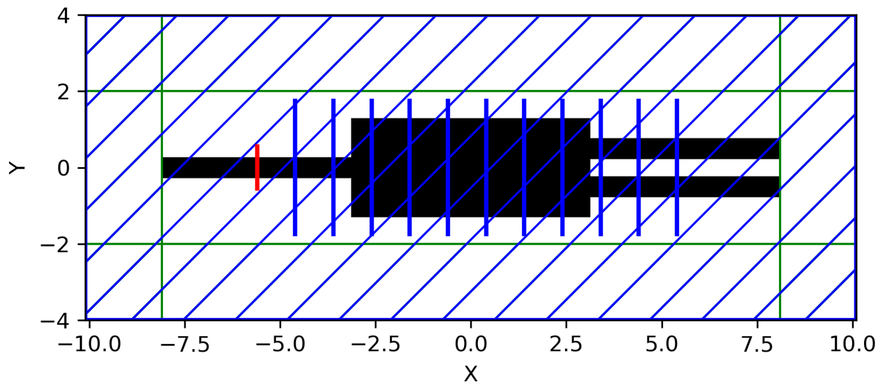


Figure 3.4: Qualitative representation of the flux monitor placement along the 1×2 MMI power splitter.

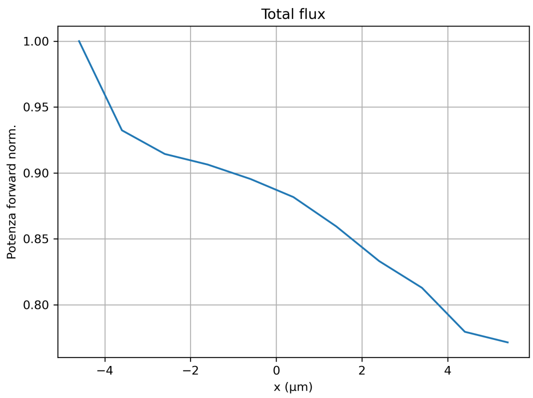
optical power is not guided through the device and is eventually absorbed by the perfectly matched layers.

A different trend is observed when considering only the power associated with the fundamental TE mode.

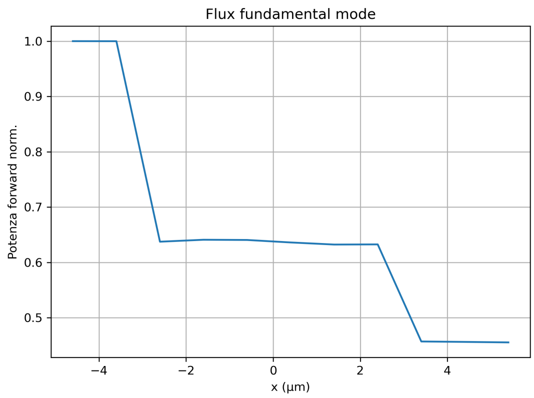
Before entering the MMI section, the input waveguide operates in a single-mode regime, so all the optical power is carried by the fundamental mode. When the waveguide abruptly widens into the multimode region, the input field no longer matches perfectly with the fundamental mode of the wider waveguide. As a result, part of the power is transferred to higher-order modes supported by the MMI, causing the power carried by the fundamental mode to decrease to approximately 0.65.

Inside the MMI section, the excited modes propagate with negligible losses. The power associated with the fundamental mode remains approximately constant, while the relative phases of the excited modes evolve along the propagation direction. Their interference gives rise to the self-imaging phenomenon. Therefore, although several modes are simultaneously present inside the MMI section, the amount of power carried by the fundamental mode does not vary significantly.

At the output of the MMI section, the optical field is coupled into the two single-mode output waveguides. However, only a fraction of the total field is coupled into the fundamental mode of each output waveguide, while the remaining power is either transferred to higher-order components or radiated away. Consequently, the power carried by the fundamental mode decreases further, falling below 50 % at the output.



(a) Total transmitted power.



(b) Fundamental mode power.

Figure 3.5: Comparison between the total transmitted power and the power carried by the fundamental mode along the MMI structure.

Moreover, since the MMI length used in this analysis corresponds to the theoretical value, it seems that the truncation point does not coincide with the optimal self-imaging position. In addition, the relatively poor performance may also be attributed to the arbitrary choice of the output waveguide separation, which was selected without any optimization. This motivates the optimization procedure presented in the following sections.

In order to learn more about the location of the self-imaging positions, an additional simulation was performed using the same device specifications described previously, but with an extended MMI length of 25 µm. The field intensity distribution along the structure was then evaluated and plotted. Although the resulting intensity map does not exhibit a perfectly clear self-imaging pattern, it still provides useful qualitative information regarding the approximate position at which the images are formed. This allows for a more accurate estimate of the optimal MMI length to be obtained.

As can be observed from Fig. 3.6, the first single self-image appears to be located at approximately x=2 µm-2.5 µm, corresponding to an MMI length of about 14.5 µm to 15 µm. Therefore, considering the symmetry of the self-imaging process and the field intensity distribution shown in the figure, it is reasonable to expect that the two-fold image required for the 1×2 splitter is formed at approximately half this distance. Consequently, the optimal MMI length is expected to lie in the range of 7 µm to 7.5 µm, which is consistent with the observed field pattern.

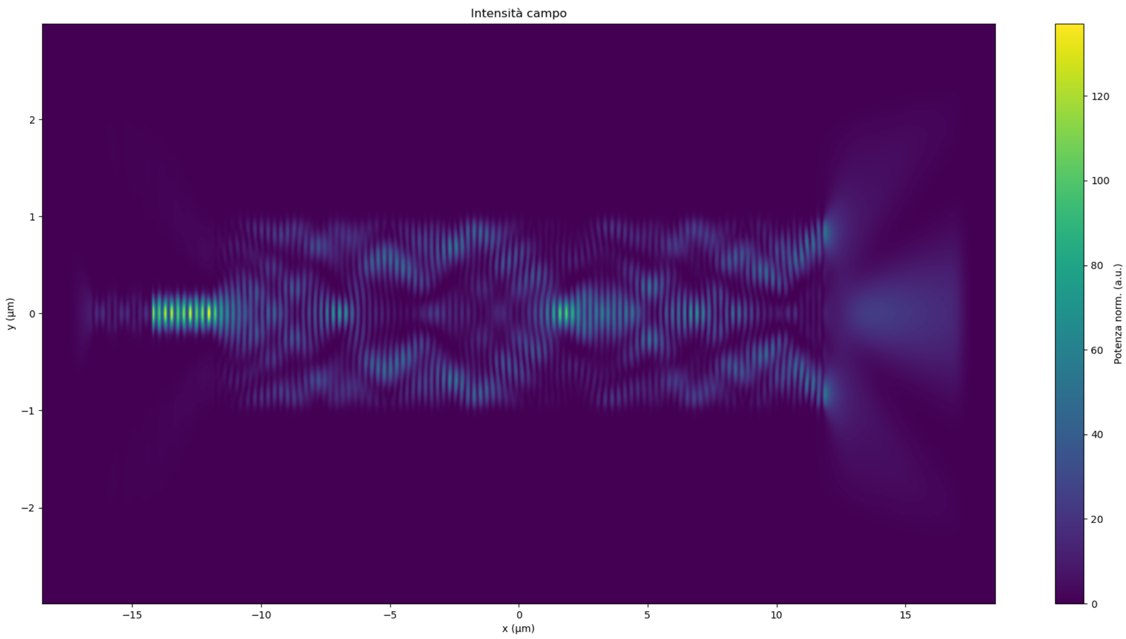


Figure 3.6: Field intensity distribution obtained for an MMI length of 25 µm.

The discrepancy between the theoretical and optimized values of the MMI length can be attributed to:

- The analytical expression for the self-imaging length is derived under simplifying assumptions and therefore provides only an approximate estimate of the optimal MMI length.
- A single effective refractive index was used in the calculation, whereas the self-imaging mechanism arises from the interference of multiple guided modes, each characterized by a different propagation constant.
- The analytical model neglects the effects introduced by the transitions between the

input waveguide, the MMI section, and the output waveguides. These discontinuities modify the phase evolution inside the multimode region and shift the image positions.

- The self-images are formed over a finite spatial region rather than at a single, well-defined position, making the device performance relatively insensitive to small variations in the MMI length.

3.4 Actual MMI length and distance between output branches

The actual optimal length of the MMI turned out to be slightly different from the theoretical prediction. In addition, it was essential to determine the optimal center-to-center spacing between the two output waveguides. These outputs must have to be located exactly at the same vertical axes positions of the two multiple images peaks.

To determine these parameters, two different approaches were employed:

1. Generation of a heatmap to investigate the dependence of device performance on the geometric parameters.

2. Use of the built-in optimizer in MEEP, based on the `SciPy` optimization algorithms, to systematically refine the optimal values.

Both approaches aim to identify the optimal values of L_{MMI} and Y_{offset} that enable maximum power transfer by maximizing the ratio between the total output power from the two output branches and the input power coming from the input waveguide.

It should be noted that the parameter Y_{offset} does not represent the center-to-center separation between the two output waveguides. Instead, it denotes the distance along the y-axis between the center of the structure and the center of the corresponding output waveguide. Consequently, the center-to-center distance between the two output waveguides is equal to $2Y_{\mathrm{offset}}$.

The function invoked by the heatmap and by `SciPy`, which returns the score value as output, operates with the same structure described in 3. It uses the same Gaussian source and includes one monitor positioned at the input and one at the output. In this way, the obtained results are practically identical to those that would be achieved by simulating the main MMI file.

The figure of merit that describes the ratio between output power and input power is:

$$score = \frac{Pout}{Pin}$$

The quantities P_{in} and P_{out} were obtained from the input and output flux monitors, respectively. These power values correspond to the optical total power. By definition, the score ranges between 0 and 1, with values closer to unity corresponding to lower propagation losses and, therefore, to a more efficient power transfer through the device. The optimization process was aimed at maximizing this quantity, i.e., obtaining a score as close as possible to 1.

3.4.1 Heatmap

The program computes the score value for all combinations of L_{MMI} and Y_{offset}, considering a user-defined number of points and range, and then generates the corresponding heatmap. The heatmap is useful for identifying the L_{MMI} and Y_{offset} values that yield the best possible score. In addition, the program reports to the user the point within the generated heatmap that corresponds to the highest score. This makes it easier to identify the region of the parameter space where the optimal solution is located and to define a narrower range for the subsequent simulations.

The first test was carried out over range 1, then narrowed to interval 2, and finally concluded with an even more accurate simulation in range 3. The ranges adopted for the optimization were selected based on the information extracted from the field intensity distribution obtained for the extended MMI structure.

Ranges:

1. $L_{\text{MMI}} \in 6\,\mu\text{m} - 8\,\mu\text{m}$, 5 points; $Y_{\text{offset}} \in 0.5\,\mu\text{m} - 0.7\,\mu\text{m}$, 5 points.
   Optimal values: $L_{\text{MMI}} = 7\,\mu\text{m}$, $Y_{\text{offset}} = 0.65\,\mu\text{m}$, score =0.97.

2. $L_{\text{MMI}} \in [6.5{,}7.5]\ \mu\text{m}$, 5 points; $Y_{\text{offset}} \in [0.6{,}0.7]\ \mu\text{m}$, 5 points.
   Optimal values: $L_{\text{MMI}} = 7\ \mu\text{m}$, $Y_{\text{offset}} = 0.625\ \mu\text{m}$, $score = 0.971$.

3. $L_{\text{MMI}} \in [6.9{,}7.1]\ \mu\text{m}$, 5 points; $Y_{\text{offset}} \in [0.62{,}0.64]\ \mu\text{m}$, 5 points.
   Optimal values: $L_{\text{MMI}} = 7\ \mu\text{m}$, $Y_{\text{offset}} = 0.635\ \mu\text{m}$, $score = 0.972$.

The heatmaps figure are illustrated in 3.7

To obtain further confirmation and even more accurate results, two additional simulations were subsequently performed using the `SciPy` optimizer within ranges 2 and 3.

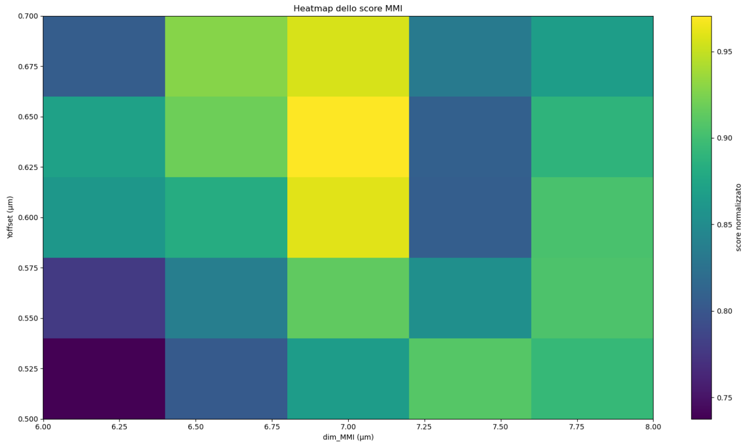
3.4.2 SciPy optimizer

SciPy is the reference library for numerical scientific computing in Python. The function used for the case under consideration is *scipy.optimize*. *scipy.optimize* is used to find variable values that minimize or maximize a function, or to solve equations and systems of nonlinear equations.

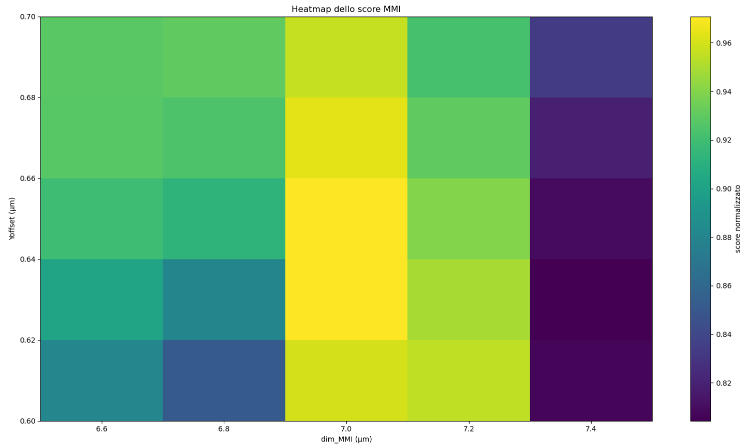
The MMI length and the output waveguide offset were selected as the optimization variables and constrained within predefined ranges. Starting from an initial guess chosen according to the results of the heatmap analysis, the performance of the device was evaluated through the score function implemented in the simulation routine.

Since the optimization algorithm employed by `SciPy` performs a minimization, the opposite of the score value was supplied as the objective function. In this way, maximizing the score corresponds to minimizing the objective function.

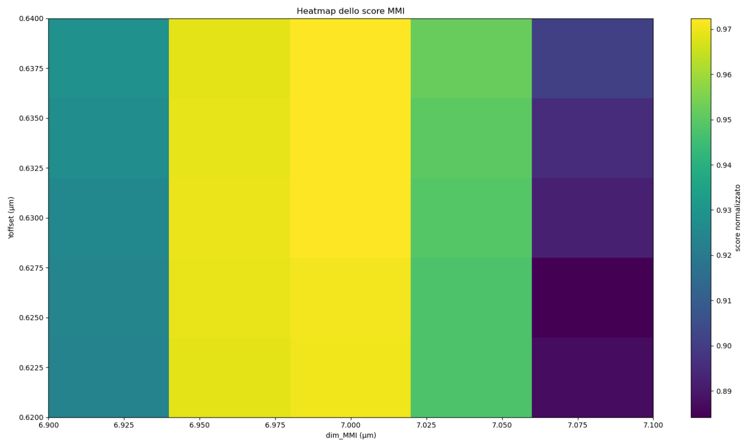
The search for the optimal parameters was performed using the Nelder–Mead method: during the optimization, the algorithm progressively updates the values of the MMI length and output waveguide offset in order to identify the configuration providing the highest score. A maximum number of 20 iterations was imposed, meaning that the algorithm



(a) First range.



(b) Second range.



(c) Third range.

Figure 3.7: Heatmaps obtained for the three parameter ranges investigated during the optimization process. The horizontal axis represents the MMI length, while the vertical axis corresponds to (Y_{offset}). The color scale indicates the value of the score function for the different combinations of parameter values considered during the optimization process

was allowed to perform up to 20 successive updates of the search region. This value was found to be sufficient to ensure convergence to the optimal solution.

The `SciPy` optimizer code starts from a point chosen by the user, and finds the maximum score point within the specified interval.

As said before, `SciPy` optimizer performed for intervals 2 and 3, with the following starting points:

- Range 2: $x0 = [7.0\,\mu\text{m}, 0.65\,\mu\text{m}]$
- Range 3: $x0 = [7.0\,\mu\text{m}, 0.63\,\mu\text{m}]$

The results obtained from the optimizer were very similar to those obtained with the heatmap, but slightly more accurate:

Range 2:
$L_{\text{MMI}} = 6.989\,\mu\text{m}$
$Y_{\text{offset}} = 0.631\,\mu\text{m}$
$score = 0.974$

Range 3:
$L_{\text{MMI}} = 6.988\,\mu\text{m}$
$Y_{\text{offset}} = 0.631\,\mu\text{m}$
$score = 0.974$

In conclusion, the optimal values were chosen as follows:

$$L_{\text{MMI}} = 7\,\mu\text{m} \quad \text{and} \quad Y_{\text{offset}} = 0.65\,\mu\text{m}$$

since they allow for achieving a higher score, approximately equal to 0.975.

In the following section, the MMI structure will be simulated using the optimized values obtained through the procedure described above in order to verify that the optimization effectively leads to a maximization of the transmitted power. The results will then be compared with those obtained using the theoretical MMI length, and the differences in device performance will be discussed.

3.5 Results Obtained with the Optimized MMI Length

A final simulation was carried out using the values of the MMI length and output waveguide offset obtained through the optimization procedure. The corresponding results are presented below.

The optimized design exhibits a significant improvement compared to the preliminary configuration based on the theoretical MMI length. As shown in Fig. 3.9, the total transmitted power decreases only slightly along the structure, reaching a value of approximately 0.975 at the output. This indicates that radiation losses are very limited and that the optimization procedure successfully identified a configuration with high transmission efficiency.

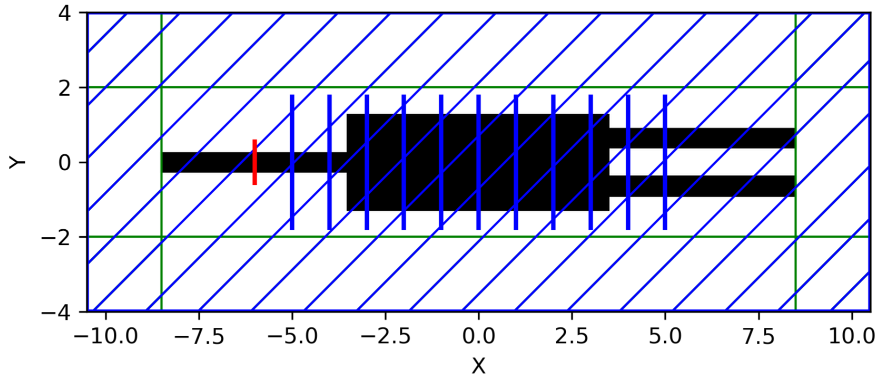
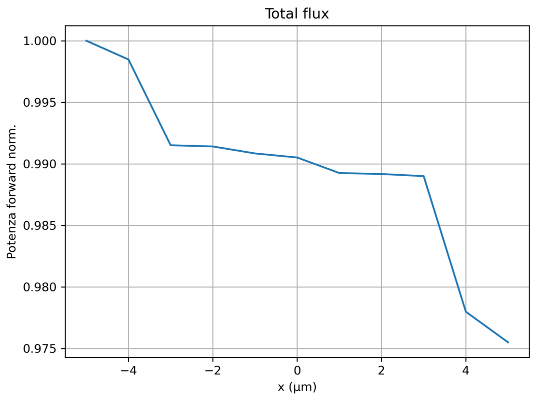
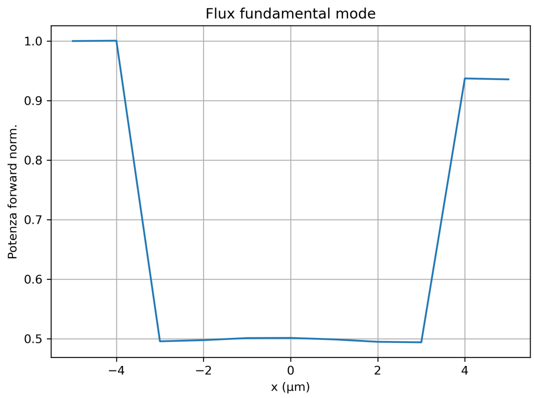


Figure 3.8: Qualitative representation of the flux monitor placement along the 1×2 MMI power splitter.



(a) Total transmitted power.



(b) Fundamental mode power.

Figure 3.9: Comparison between the total transmitted power and the power carried by the fundamental mode along the MMI structure.

The behavior of the power carried by the fundamental mode is different. At the entrance of the MMI section, a sudden decrease to approximately 0.5 is observed. This effect is due to the modal mismatch between the single-mode input waveguide and the multimode region: the input field excites several modes supported by the MMI, and therefore only a fraction of the power remains associated with the fundamental mode of the wider waveguide. Inside the MMI section, the power carried by the fundamental mode remains nearly constant, while the interference among the excited modes gives rise to the self-imaging phenomenon. At the output of the MMI, the optical field is efficiently coupled back into the fundamental modes of the two single-mode output waveguides. As a consequence, almost all the transmitted power is recovered in the fundamental mode, with a normalized value close to 0.94.

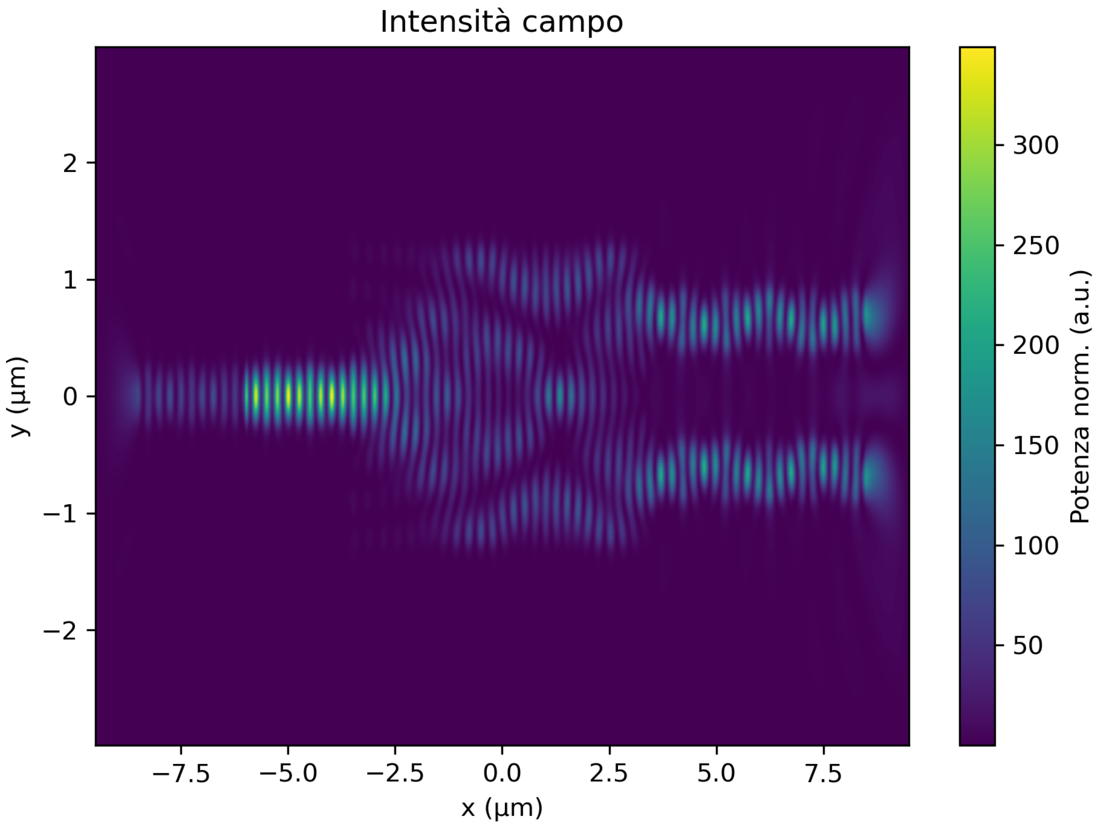


Figure 3.10: Field intensity distribution

The field intensity distribution shown in Fig. 3.10 clearly highlights the formation of the two self-images and the subsequent coupling into the output waveguides. Although the wavefronts propagating inside the output guides are not perfectly straight, indicating the presence of a small residual excitation of higher-order components, the overall field distribution confirms the good performance of the device.

Overall, the optimized splitter exhibits excellent characteristics. The total power losses are very small, and, most importantly, almost all the transmitted power is coupled into the fundamental modes of the output waveguides, demonstrating the effectiveness of the optimization procedure.

3.6 Bandwidth Extension

The next step was to analyze the behavior of the MMI over a small wavelength range around the central wavelength. The simulation of the MMI, which previously operated at a Working wavelength: 1550 nm, was extended over a wavelength range from 1450 nm to 1650 nm; therefore, an appropriate bandwidth for the Gaussian Source was chosen.

It is important to specify again that in MEEP, the frequency is calculated as the inverse of the wavelength since the software uses normalized units in which the speed of light is set to one. In this unit system, the frequency, therefore, represents the fundamental quantity used in the simulations, while the wavelength is derived directly as its inverse. Consequently, the definition of the spectral bandwidth and the sampling of points must be carried out in frequency space rather than directly in wavelength space.

The Working wavelength was

$$\lambda_c = 1.55\,\mu\text{m}$$

The corresponding normalized working frequency is

$$f_{cen} = \frac{1}{\lambda_c} = \frac{1}{1.55} \approx 0.64516$$

and the chosen bandwidth for the Gaussian Source is

$$df = 0.15\, f_{cen} \approx 0.15 \times 0.64516 \approx 0.09677$$

In this way, the obtained minimum and maximum covered wavelengths and frequencies are

$$f_{min} = f_{cen} - \frac{df}{2} \approx 0.59678$$

$$f_{max} = f_{cen} + \frac{df}{2} \approx 0.69354$$

$$\lambda_{min} = \frac{1}{f_{max}} = \frac{1}{0.69354} \approx 1.442\,\mu\text{m} = 1442\,\text{nm}$$

$$\lambda_{max} = \frac{1}{f_{min}} = \frac{1}{0.59678} \approx 1.676\,\mu\text{m} = 1676\,\text{nm}$$

$$\lambda \in [1442,\ 1676]\,nm$$

In this way the bandwidth covered by the Gaussian Source is slightly larger than the desired one. This is because:

- The amplitude of the Gaussian decreases gradually at the edges. Therefore, if the desired range had coincided exactly with the target values, the energy at the extremes would have been too small,

- The relationship between frequency and wavelength is not linear; therefore, a bandwidth that is symmetric in frequency becomes asymmetric in wavelength. A small margin prevents this issue.

The considered bandwidth was sampled using 50 uniformly spaced frequency points.

3.7 Results Obtained with the previous MMI Length

The chosen MMI length is the one previously calculated from the heatmap and the `SciPy` optimizer to optimize transmission at 1550 nm:

$$L_{\text{MMI}} = 7\,\mu\text{m} \quad \text{and} \quad Y_{\text{offset}} = 0.65\,\mu\text{m}.$$

Therefore, it is expected that the highest output power (Fig. 3.11a) will be obtained in the vicinity of 1550 nm, also because the source employed in the simulations is Gaussian and thus exhibits its maximum spectral power around the central wavelength. As for the transmission (Fig. 3.11b), the values remain relatively high over the entire wavelength range considered, which is not surprising since the relative spectral interval under investigation is very limited.

The objective is to further improve the device in order to achieve a broader operating bandwidth. Consequently, once the broadband optimization is completed, the following outcomes are expected:

- a lower peak output power at the central wavelength, accompanied by higher power values in the vicinity of 1550 nm, resulting in a broader spectral response;
- a higher average transmission over the considered wavelength range;
- values of L_{MMI} and Y_{offset} not significantly different from those previously obtained, since the optimization is performed over a relatively limited wavelength range.

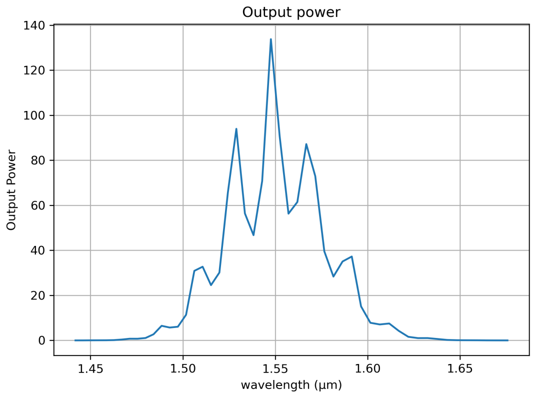
By looking at the plot of the total output power and the transmission values, fluctuations in the trend can be observed. When the single-mode signal from the waveguide enters the MMI, the guided modes recombine (Multimode Interference), and the phase differences between the modes generate interference patterns: the incoming field decomposes into a combination of the MMI modes. The modes propagate at different velocities because each mode has a different propagation constant; therefore, as they propagate, they accumulate different phases.

The propagation constant depends on the wavelength, so when it changes, the phases change as well, the interference pattern changes, and consequently, the amount of light that reaches the outputs changes.

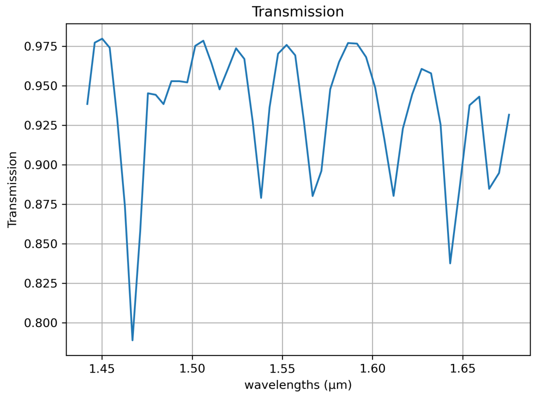
To make the trend clearer, only 11 equispaced wavelengths were selected in the range from 1450 nm to 1650 nm, thus spaced 20 nm apart (Fig. 3.12).

As shown in Fig. 3.12, the output power exhibits a pyramidal shape centered around the design wavelength of 1550 nm as expected.

On the other hand, the normalized transmission remains relatively high over the entire wavelength interval. In particular, the transmission varies between approximately

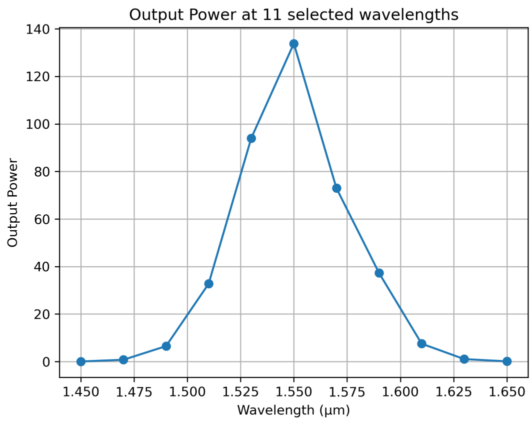


(a) Output power as a function of wavelength.

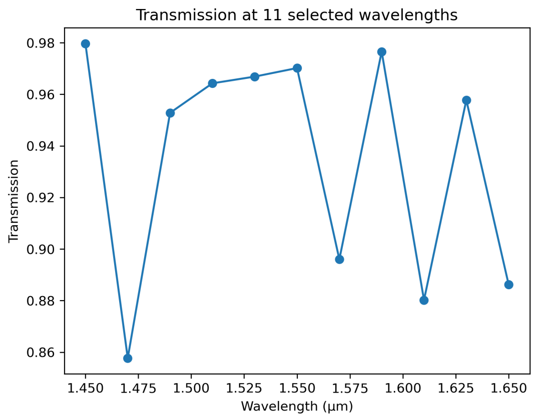


(b) Transmission spectrum of the optimized MMI splitter.

Figure 3.11: Spectral response of the optimized MMI splitter over the wavelength range from 1450 nm to 1650 nm. (a) Output power spectrum and (b) normalized transmission spectrum.



(a) Output power as a function of wavelength.



(b) Transmission spectrum of the optimized MMI splitter.

Figure 3.12: Spectral response of the optimized MMI splitter over the wavelength range from 1450 nm to 1650 nm. (a) Output power spectrum and (b) normalized transmission spectrum.

0.88 and 0.98. Although some fluctuations are present, the device maintains a high transmission efficiency throughout the considered spectral range.

Interestingly, the maximum transmission is not necessarily achieved at the design wavelength. This is expected since the MMI splitter is not a resonant device, and its operation relies on multimode interference. Small variations in wavelength modify the relative phase accumulation among the propagating modes and may slightly improve or degrade the coupling efficiency at the output waveguides. Therefore, wavelengths other than 1550 nm may exhibit transmission values slightly higher than those obtained at the design wavelength.

Nevertheless, the differences are small, and the transmission remains high throughout the entire spectral range, indicating that the device performance is only weakly dependent on wavelength.

The transmission values obtained at the 11 selected wavelengths are reported in Table 3.1.

Table 3.1: Transmission values obtained at the 11 selected wavelengths.

Wavelength (nm)	Transmission
1450	0.9683
1470	0.8406
1490	0.9422
1510	0.9602
1530	0.9484
1550	0.9643
1570	0.8775
1590	0.9723
1610	0.8650
1630	0.9488
1650	0.8631

In order to quantitatively evaluate the broadband performance of the device, an overall score was defined as the average transmission over the considered wavelength range. Specifically, the score is computed as

$$\text{score} = \frac{1}{N}\sum_{i=1}^{N} T_i = 0.923$$

where T_i is the transmission at the i-th wavelength and $N = 11$ is the total number of sampled wavelengths. Therefore, the score represents the mean transmission of the device over the entire spectral range and provides a convenient figure of merit for broadband optimization.

3.8 Actual MMI length and distance between output branches in the extended bandwidth

Subsequently, an attempt was made to find the optimal values of the MMI length and the distance between the outputs, which could allow for obtaining the best possible score across the entire wavelength range. The two methods described in 3.4.1 and 3.4.2 were used. The chosen bandwidth for the Gaussian Source was $df = 0.15\, f_{cen}$ as before, and the considered bandwidth was sampled using 50 uniformly spaced frequency points. The score value was calculated as the average of the ratio between the output and input powers at the same 11 wavelengths previously selected.

$$\text{score} = \frac{1}{N}\sum_{i=1}^{N} T_i$$

Heatmap in bandwidth 1450 nm - 1650 nm

The first test was carried out over the range 1, then narrowed to the interval 2.

Ranges:

1. $L_{\text{MMI}} \in [6.5\,\mu\text{m}, 7.5\,\mu\text{m}]$, 5 points; $Y_{\text{offset}} \in [0.6\,\mu\text{m}, 0.7\,\mu\text{m}]$, 5 points.
   Optimal values: $L_{\text{MMI}} = 6.75\,\mu\text{m}$, $Y_{\text{offset}} = 0.675\,\mu\text{m}$, $score = 0.936$.

2. $L_{\text{MMI}} \in [6.65\,\mu\text{m}, 6.85\,\mu\text{m}]$, 5 points; $Y_{\text{offset}} \in [0.665\,\mu\text{m}, 0.685\,\mu\text{m}]$, 5 points.
   Optimal values: $L_{\text{MMI}} = 6.8\,\mu\text{m}$, $Y_{\text{offset}} = 0.67\,\mu\text{m}$, $score = 0.938$.

The heatmap figure is illustrated in 3.13

To obtain further confirmation and even more accurate results, two additional simulations were subsequently performed using the `SciPy` optimizer within ranges 1 and 2.

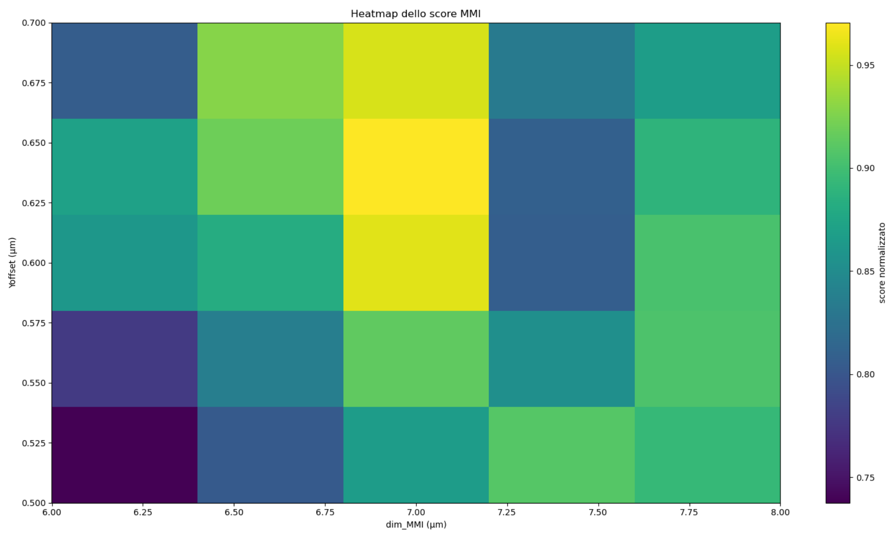
SciPy optimizer between 1450 nm and 1650 nm

As mentioned before, the `SciPy` optimizer was performed for intervals 1 and 2, with the following starting points:

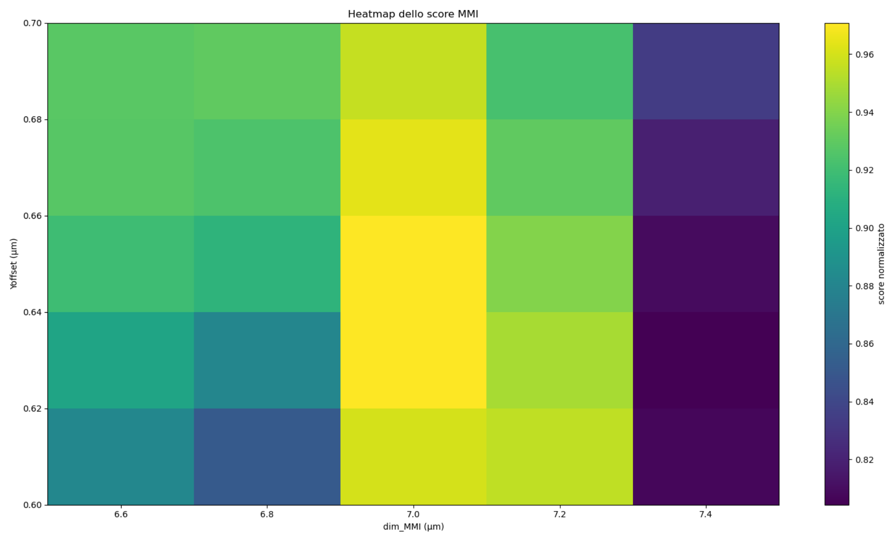
- Range 1: $x0 = [6.75\,\mu\text{m}, 0.675\,\mu\text{m}]$
- Range 2: $x0 = [6.8\,\mu\text{m}, 0.67\,\mu\text{m}]$

The results obtained from the optimizer were very similar to those obtained with the heatmap, but slightly more accurate:

Range 1:
$L_{\text{MMI}} = 6.786\,\mu\text{m}$
$Y_{\text{offset}} = 0.669\,\mu\text{m}$
$score = 0.942$



(a) First range.



(b) Second range.

Figure 3.13: Heatmaps obtained for the two parameter ranges investigated during the optimization process. The horizontal axis represents the MMI length, while the vertical axis corresponds to (Yoffset). The color scale indicates the value of the score function for the different combinations of parameter values considered during the optimization process

Range 2:
$L_{\mathrm{MMI}} = 6.784\,\mu\mathrm{m}$
$Y_{\mathrm{offset}} = 0.658\,\mu\mathrm{m}$
$score = 0.942$

In conclusion, the optimal values were chosen as follows:

$$L_{\mathrm{MMI}} = 6.8\,\mu\mathrm{m} \quad \text{and} \quad Y_{\mathrm{offset}} = 0.67\,\mu\mathrm{m}$$

since this choice allows us to achieve a higher mean score, approximately between 0.938 and 0.942.

The values of L_{MMI} and Yoffset obtained do not differ significantly from those previously found through single-wavelength optimization, as expected.

3.9 Results Obtained with the Actual MMI Length

The MMI simulation was repeated while adopting the optimal values of L_{MMI} and Y_{offset} obtained from the optimization procedure.

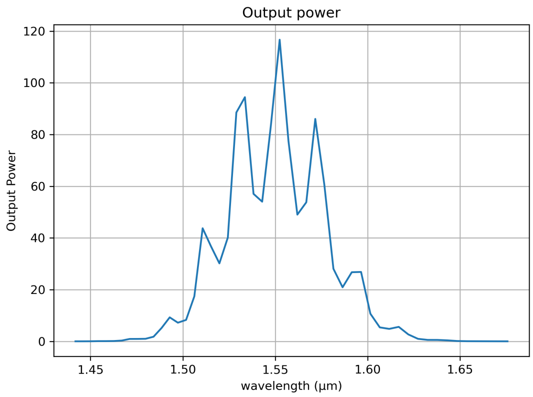
It can be observed that after the optimization, the peak value of the output power slightly decreases around the central wavelength (Fig. 3.14c). The optimizer slightly sacrifices the peak value at the central wavelength in order to improve the power at neighboring wavelengths, leading to a flatter spectral response.

The transmission values obtained at the 11 selected wavelengths are reported in Table 3.2.

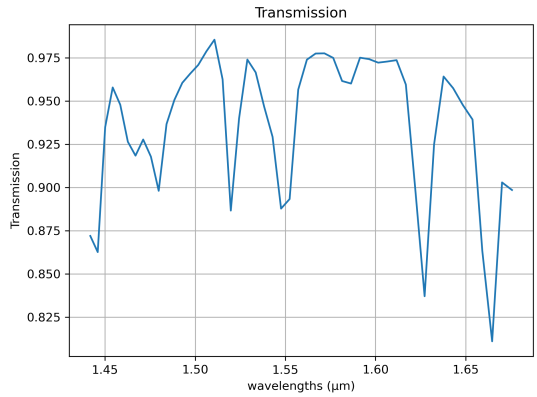
Table 3.2: Transmission values obtained at the 11 selected wavelengths after broadband optimization.

Wavelength (nm)	Transmission
1450	0.9241
1470	0.9149
1490	0.9307
1510	0.9746
1530	0.9596
1550	0.8643
1570	0.9727
1590	0.9688
1610	0.9678
1630	0.9073
1650	0.9307

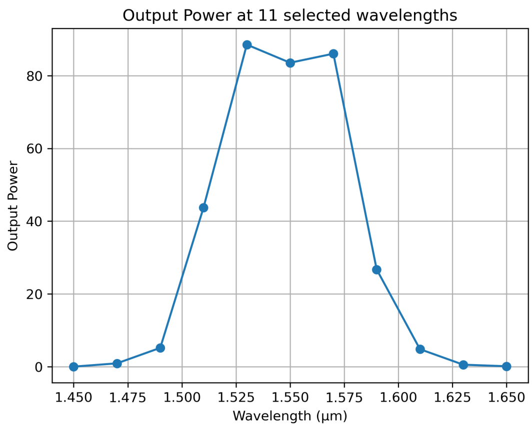
Although the transmission at the design wavelength of 1550 nm slightly decreases after the broadband optimization, this behavior is expected. Indeed, the optimization objective is no longer the maximization of the transmission at a single wavelength, but rather the maximization of the average transmission over the entire spectral range. Consequently, a



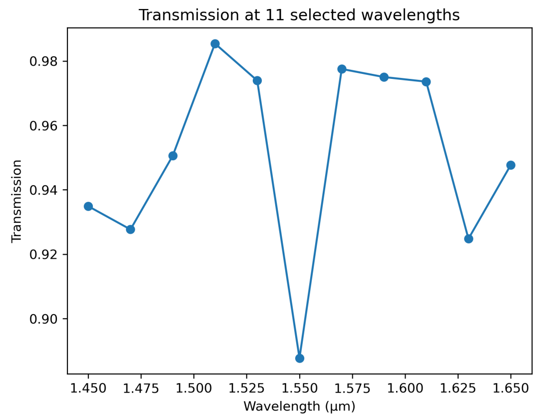
(a) Output power over the considered wavelength range.



(b) Transmission over the considered wavelength range.



(c) Output power sampled at the 11 selected wavelengths.



(d) Transmission sampled at the 11 selected wavelengths.

Figure 3.14: Spectral response of the MMI splitter obtained using the geometrical parameters resulting from the broadband optimization. The top row shows the output power and transmission evaluated over the whole wavelength range, while the bottom row reports the corresponding quantities sampled at the 11 selected wavelengths

small reduction of the transmission at 1550 nm is compensated by improved transmission values at neighboring wavelengths, resulting in a higher average score and a more uniform spectral response.

In order to quantitatively assess the broadband performance of the device, the average score is evaluated:

$$\text{score} = \frac{1}{N}\sum_{i=1}^{N} T_i = 0.938$$

The average score obtained confirms the improvement achieved through the broadband optimization. It demonstrates that a better compromise among the transmission values at the different wavelengths has been achieved.

Chapter 4

Pixel Placement and Optimization using MEEP Genetic Algorithm

In this chapter, a different approach is investigated. The MMI section is filled with an array of small circular pixels.

The introduction of pixels inside the MMI region provides additional degrees of freedom for controlling the propagation of light. Each pixel locally modifies the refractive index distribution and consequently affects the interference pattern responsible for the self-imaging process. By properly selecting which pixels are present and which are removed, it is possible to tune the optical field inside the multimode section and influence the coupling efficiency at the output waveguides.

Since the number of possible configurations is extremely large, manually determining the optimal arrangement is impractical. For this reason, a genetic algorithm is employed to automatically search for suitable pixel distributions. The purpose of this study is to investigate how these local perturbations affect device behavior and to determine whether a specific pixel arrangement can provide better performance than a fully populated structure.

The device specifications, geometrical dimensions, pixel size, and the optimization algorithm employed are described in detail below and in the following sections.

The operating conditions and dimensions of the structure are as follows:

- Working bandwidth: 1450 nm-1650 nm
- Core refractive index: $n_{\text{core}} = 3.23$
- Cladding refractive index: $n_{\text{clad}} = 1.44$
- Width of the input and output waveguides: 0.55 µm
- Length of the input and output waveguides: 5 µm
- Width of the MMI: 2.6 µm
- Length of the MMI: 6.8 µm

- Distance from center to center in the vertical direction between the two outputs: 1.34 µm
- Spatial resolution: 40 points/µm
- Perfectly matched layer (PML) thickness: 2 µm

The dimensions and simulation specifications considered in this section are the same as those obtained from the broadband optimization. In particular, the values of L_{MMI} and Y offset correspond to the optimized values previously obtained.

As described in 3.6, the chosen bandwidth for the Gaussian Source: $df = 0.15\, f_{cen}$. The considered bandwidth was sampled using 50 uniformly spaced frequency points.

The specifications of the pixels are as follows:

- pixel diameter: 200 nm
- center-to-center distance between pixels in both directions: 400 nm
- number of rows: 6
- number of columns: 17
- total number of pixels: 102

The diameter of the pixels was chosen to be 200 nm, which is significantly smaller than the operating bandwidth, to introduce local perturbations of the refractive index without disrupting the propagation of the optical field within the multimode interference region.

The center-to-center spacing between the cylinders was set to 400 nm to avoid geometric overlap and to maintain a regular distribution of possible perturbations within the MMI region.

A qualitative representation of the structure, with all the pixels of the matrix present, is shown in Fig. 4.1.

4.1 Optimization

The insertion of pixels allows precise local control of the refractive index within the MMI region. By selectively activating or deactivating individual pixels, the optical field distribution can be shaped to optimize power splitting or transmission. This discrete pixel-based approach is compatible with optimization algorithms, enables fine tuning without disrupting the overall propagation, and provides a flexible and scalable design methodology.

To predict the optimal pixel configurations, a **Genetic Algorithm (GA)** available in the **pymoo** Python library was adopted.

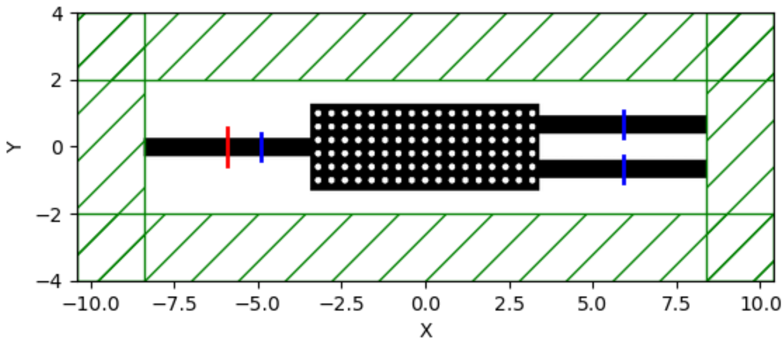


Figure 4.1: Qualitative representation of the MMI splitter with all the pixels of the matrix present.

pymoo

pymoo is a Python library for numerical optimization, particularly focused on evolutionary algorithms such as Genetic Algorithms (GA).

In general, `pymoo`:

1. Allows the definition of an optimization problem (variables, objective function, constraints)
2. Automatically generates candidate solutions
3. Applies an algorithm (GA in this case) to progressively improve the solutions
4. Returns the best solution found

In this work, a Genetic Algorithm is implemented to explore different configurations of the pixels inside the MMI region. Each configuration is represented as a 102 bit long binary vector, where each element corresponds to the presence or absence of a cylinder ('1' indicates presence, while '0' indicates absence). For each candidate solution, `pymoo` calls a custom evaluation function that passes the current configuration to the fitness function and returns the corresponding performance score.

Based on these evaluations, the algorithm iteratively improves the population by selecting, recombining, and mutating the best-performing configurations, with the goal of maximizing the transmitted optical power.

GA

A Genetic Algorithm (GA) is an optimization method inspired by natural selection and biological evolution. It is used to find good solutions in very large and complex search spaces without checking every possibility one by one.

How it works:

1. **Initial population:** a set of candidate solutions is created, usually at random. The number of candidates for each generation is defined by the variable "Population Size" *pop_size*.
2. **Fitness evaluation:** each individual is scored using a *fitness function*, which measures how well the solution meets the goal. The specific used *fitness function* is described in 4.1.
3. **Selection:** the best individuals are selected to reproduce those with higher scores.
4. **Crossover (recombination):** two selected individuals are combined to create one or more children, mixing the features of the parents.
5. **Mutation:** some individuals are randomly changed slightly to explore new parts of the solution space.
6. **Replacement:** the new individuals (children) replace the less fit ones in the population, and the cycle repeats.

7. **Iteration:** the process continues for a "Number Of Generations" *n_gen* or until a stopping criterion is reached.

Fitness Function

The fitness function simulates the structure described in 4.

The fitness function evaluates the performance of each candidate pixel configuration using a full electromagnetic simulation:

- 1 → cylinder present ($n = n_{\text{clad}}$)
- 0 → cylinder absent ($n = n_{\text{core}}$)

For each configuration provided by the optimization algorithm, the function builds the corresponding geometry by activating or deactivating cylinders inside the MMI region according to the binary vector.

During the simulation, optical power is monitored using three flux monitors: one placed at the input waveguide and one at each output branch. Unlike the previous analyses, no additional monitors are placed along the structure, since the large number of simulations required by the optimization procedure would result in prohibitively long computation times. A qualitative representation of the structure, showing the positions of the source and the input and output flux monitors, is presented in Fig. 4.2.

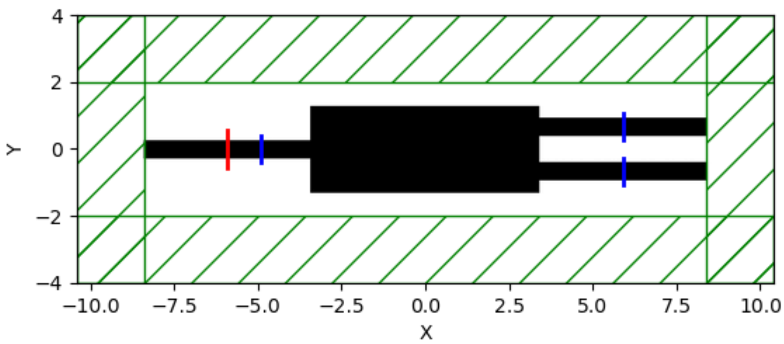


Figure 4.2: Qualitative representation of the MMI splitter showing the positions of the input and output flux monitors used during the genetic algorithm optimization.

The monitors record the total optical power over the entire range of interest, sampled at 50 frequency points.

The fitness value used by the genetic algorithm is then defined as the average transmission over the 50 sampled frequencies. Therefore, the score can be expressed as

$$score = \frac{1}{N} \sum_{i=1}^{N} \frac{P_{\text{out,top},i} + P_{\text{out,bottom},i}}{P_{\text{in},i}}, \qquad N = 50, \tag{4.1}$$

where $P_{\text{in},i}$, $P_{\text{out,top},i}$, and $P_{\text{out,bottom},i}$ denote the total optical powers measured at the input, top output, and bottom output monitors, respectively, at the (i)-th frequency point.

The function returns a scalar score representing the transmission efficiency of the given configuration. Since `pymoo` performs minimization by default, the negative of this score is passed to the optimizer.

The fitness function is called by pymoo every time a new candidate solution needs to be evaluated for each individual in the population at every generation of the Genetic Algorithm.

4.2 Results Before and After the Optimization

Since the optimization starts from the MMI dimensions previously found to be optimal, no significant improvement with respect to the transmission score obtained for the empty structure is expected.

The objective of this study is not necessarily to outperform the original empty structure, but rather to investigate the behavior of the genetic algorithm and verify its ability to identify increasingly better pixel configurations. In particular, the goal is to achieve transmission values above 50 % and to observe the progressive improvement of the score throughout the optimization process.

By generating and evaluating different pixel arrangements, the genetic algorithm explores the design space and iteratively selects the most promising configurations. A gradual increase in the score over successive generations is therefore expected, confirming that the optimization procedure is effectively working as intended.

As a first step, the structure with all the pixels present was simulated in order to evaluate its transmission score. This preliminary simulation was performed mainly to provide a reference point and to obtain an estimate of the expected order of magnitude of the score value:

$$\text{score} = 0.399$$

This value serves to provide a rough estimate of the transmission values that can be expected and to evaluate the improvements achieved by the optimized pixel configurations.

Three optimization runs were performed using different values of the population size `pop_size` and the number of generations `n_gen`:

1. `pop_size = 10,   n_gen = 5`

2. `pop_size = 10,   n_gen = 10`

3. `pop_size = 20,   n_gen = 20`

Increasing both the population size and the number of generations allows the genetic algorithm to explore a larger number of possible configurations. A more extensive exploration of the design space generally leads to better solutions and higher score values.

During the optimization process, all the tested pixel configurations and their corresponding score values were saved. By analyzing these results, a progressive increase in the score was observed over successive generations, confirming the ability of the genetic algorithm to identify increasingly better solutions.

Evolution of the Score Value as a Function of the Iterations

The evolution of the score values as a function of the iteration number for the three optimization cases is shown in Fig. 4.3.

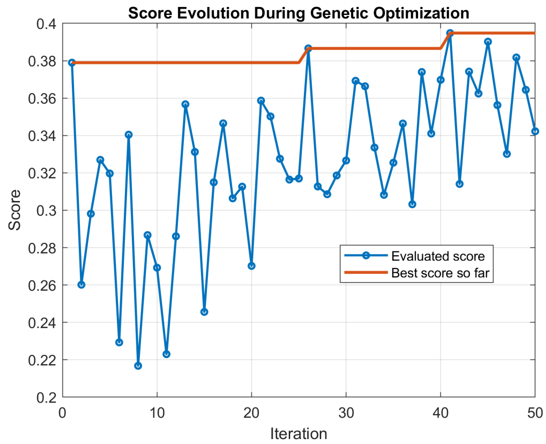
- **`pop_size = 10, n_gen = 5`.** As shown in Fig. 4.3(a), the best score is reached at iteration 41 out of 50 and is equal to **0.395**, starting from an initial value of approximately 0.38. The evaluated scores exhibit significant fluctuations, as expected from the nature of the genetic algorithm, which continuously explores different pixel configurations. After roughly half of the iterations, the average level around which the score oscillates begins to increase, indicating that better solutions are being identified. However, the total number of iterations is relatively small, so the improvement is limited.

- **`pop_size = 10, n_gen = 10`.** As shown in Fig. 4.3(b), the best score is reached at iteration 94 out of 100 and reaches a value of **0.462**. In this case, the initial random configuration yields a score below 0.3. As the optimization progresses, the algorithm explores different configurations, and a gradual increase in the best score can be observed. Compared with the previous case, the larger number of generations allows the genetic algorithm to achieve a more significant improvement.

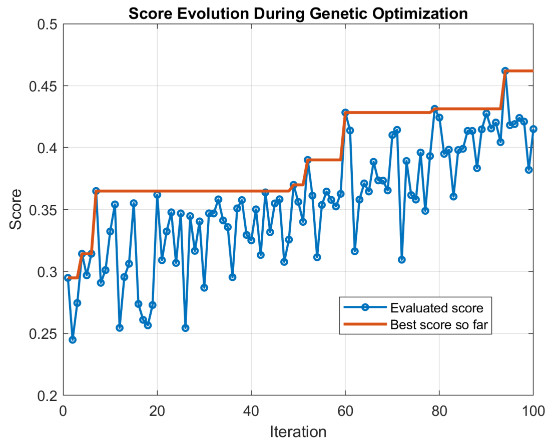
- **`pop_size = 20, n_gen = 20`.** As shown in Fig. 4.3(c), the best score is obtained at iteration 387 out of 400 and reaches a value of **0.589**. Starting again from an initial score below 0.3, the optimization exhibits a clear and gradual increase of the best score throughout the iterations. The larger population size and number of generations allow the algorithm to explore the design space more effectively, leading to the best performance among the three cases. The gradual rise of the score demonstrates that the genetic algorithm is successfully converging towards increasingly better pixel configurations.

Best Configurations for the Three Optimization Runs

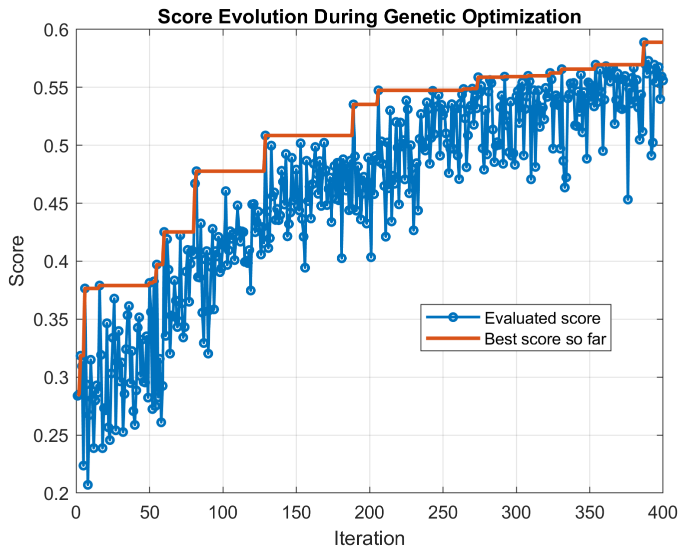
The three best configurations obtained for the different values of `pop_size` and `n_gen`, together with their corresponding score values, are shown in Fig. 4.4.



(a) `pop_size = 10`, `n_gen = 5`.

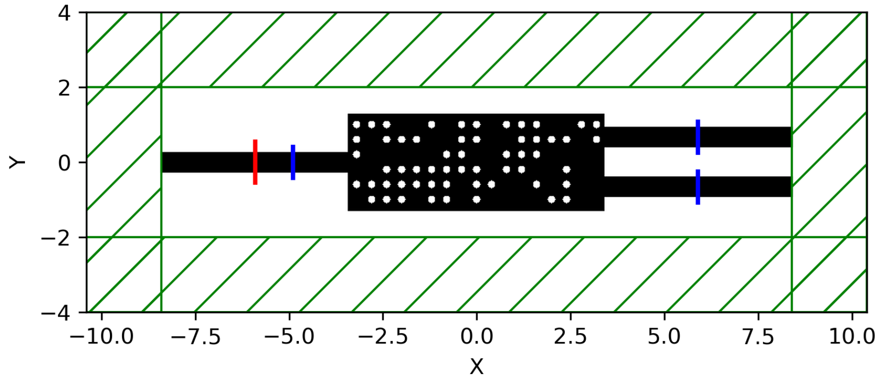


(b) `pop_size = 10`, `n_gen = 10`.

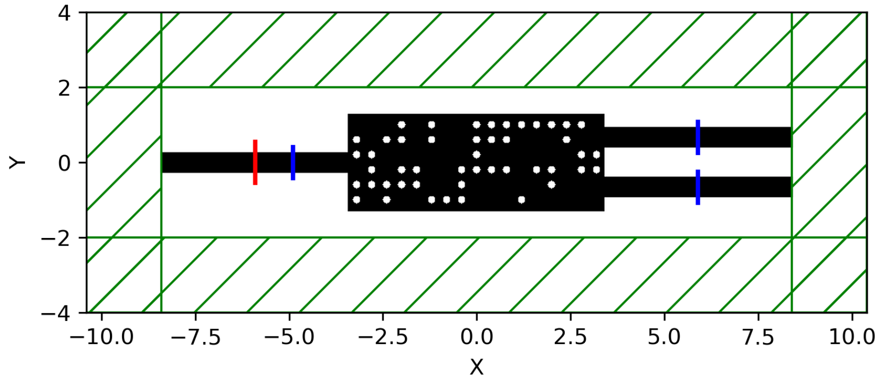


(c) `pop_size = 20`, `n_gen = 20`.

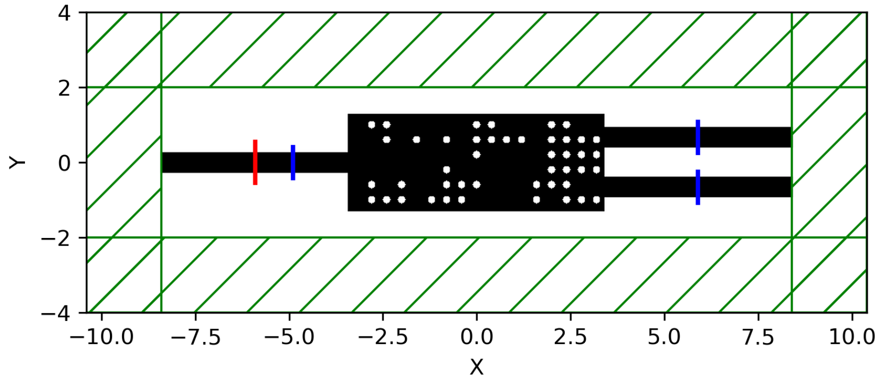
Figure 4.3: Evolution of the score values as a function of the iteration number for the three genetic algorithm optimization cases.



(a) `pop_size = 10`, `n_gen = 5`, `score = 0.395`



(b) `pop_size = 10`, `n_gen = 10`, `score = 0.462`



(c) `pop_size = 20`, `n_gen = 20`, `score = 0.589`

Figure 4.4: Best pixel configurations obtained for the three genetic algorithm optimization cases, together with their corresponding score values.

Field Evolution at the central wavelength for the Three Optimization Runs

Finally, the field evolution at the central wavelength of 1550 nm was analyzed for the three optimized configurations. The corresponding field intensity distributions are presented in Fig. 4.5, allowing a qualitative comparison of the optical behavior associated with the different solutions.

Although the field intensity distributions shown in Fig. 4.5 do not allow a detailed quantitative analysis, some qualitative observations can still be made:

- **`pop_size = 10, n_gen = 5`.** As shown in Fig. 4.5(a), corresponding to a score of 0.395, the optical power appears to be preferentially directed towards the upper output arm. However, only a limited amount of power reaches the output waveguide, since a significant fraction of the field does not propagate straight towards the output branches and is instead scattered inside the MMI region. This behavior is consistent with the relatively low score obtained.

- **`pop_size = 10, n_gen = 10`.** As shown in Fig. 4.5(b), corresponding to a score of 0.462, the field distribution appears more balanced between the two output arms. Nevertheless, considerable losses are still present, as indicated by the power radiated outside the output waveguides.

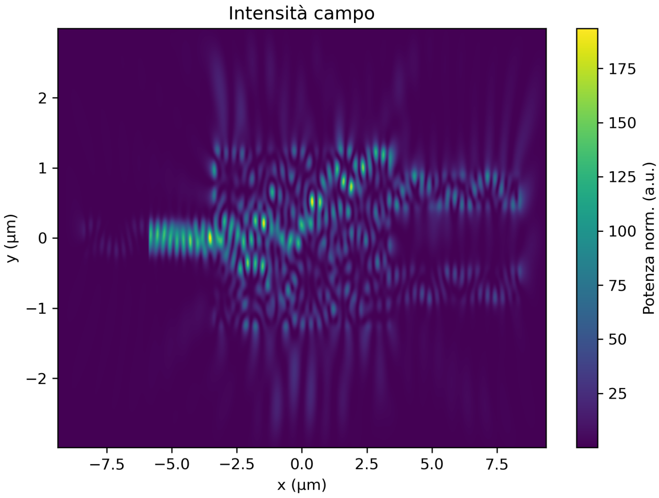
- **`pop_size = 20, n_gen = 20`.** As shown in Fig. 4.5(c), corresponding to a score of 0.589, the field distribution suggests a lower amount of radiated power and a more efficient guiding of light towards the output waveguides. A slightly larger fraction of the optical power appears to be coupled into the lower arm, although the overall distribution remains reasonably balanced.

Overall, only limited information can be extracted from the field maps. Their main purpose is to provide a qualitative visualization of how the different pixel configurations modify the propagation of light inside the MMI region. The improvement in performance is more clearly highlighted by the score evolution and by the corresponding transmission values rather than by the field distributions themselves.

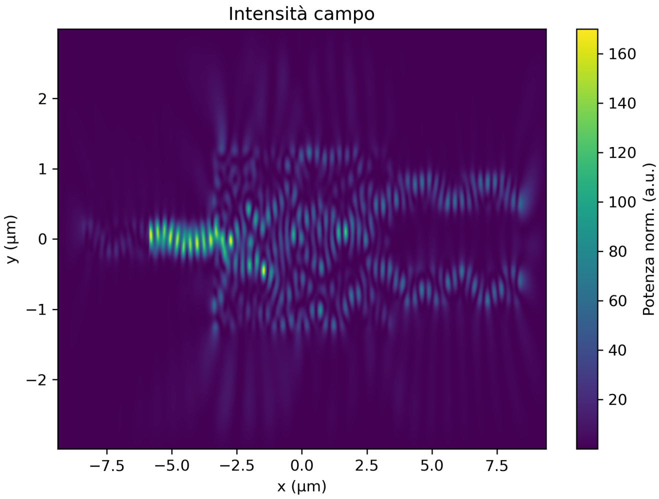
Among the three cases, the best result is clearly achieved for `pop_size = 20` and `n_gen = 20`. The continuous increase of the best score suggests that further improvements could likely be obtained by performing longer optimization runs. However, due to the considerable computational cost associated with each genetic algorithm simulation and the limited time available, additional runs were not investigated.

4.3 Possible Improvements

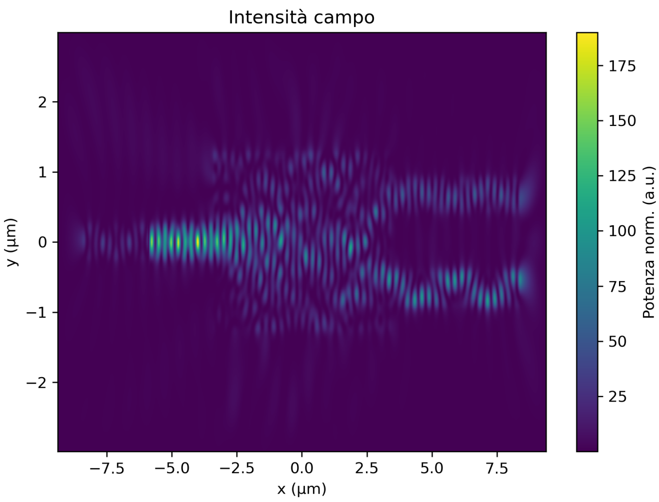
The optimization presented in this chapter was performed starting from the MMI dimensions previously found to be optimal. As a consequence, the insertion of pixels inside the multimode region is not expected to significantly outperform the original empty structure, which already provides near-optimal transmission.



(a) `pop_size = 10`, `n_gen = 5`.



(b) `pop_size = 10`, `n_gen = 10`.



(c) `pop_size = 20`, `n_gen = 20`.

Figure 4.5: Field intensity distributions at 1550 nm for the three optimized pixel configurations.

A possible improvement would be to intentionally reduce the MMI dimensions, thereby obtaining a smaller device footprint at the expense of a lower transmission efficiency. The introduction of the pixels could then be exploited to compensate for this degradation and recover the lost performance. In this way, the optimization process could potentially lead to a more compact device while maintaining satisfactory transmission characteristics.

This approach was not investigated in the present work due to time constraints, since the simulations required by the genetic algorithm are computationally demanding and involve long execution times.

Similarly, increasing the population size `pop_size` and the number of generations `n_gen` of the genetic algorithm would have resulted in better solutions by allowing a more extensive exploration of the design space. However, only small improvements are expected, while the computational time would increase significantly.

Finally, a more detailed analysis of the field distributions could have been performed by extracting and analyzing the optical power coupled into the upper and lower output waveguides for the three optimized configurations. Such an analysis would have allowed for a better understanding of the apparent asymmetries and losses visible in Fig. 4.5. However, this additional investigation was not carried out within the scope of the present work.

Chapter 5

Pixel Placement and Optimization of an AND Logic Gate using MEEP Genetic Algorithm

In this chapter, the genetic algorithm previously used for the pixel placement optimization is applied to a different problem: the design of an optical AND gate.

As in the previous experiment, the MMI region is filled with a matrix of small dielectric pixels whose position is optimized automatically. The pixel configuration is not selected manually; instead, a genetic algorithm implemented through the `pymoo` library is used to test different arrangements and identify the most suitable ones.

The objective is to find a pixel distribution capable of reproducing the behavior of an AND logic gate.

The structure of the MMI in which the pixels will be embedded slightly differs from the initial configuration. Specifically, instead of having a single input waveguide and two output waveguides, it is characterized by two input waveguides and a single output waveguide. This configuration was obtained through a simple rotation of the original structure.

This modification is motivated by the need for the device to operate as a logic gate, which requires two inputs and a single output.

As a consequence, the geometric parameters of the MMI, such as the MMI length and the center-to-center distance between the waveguides (originally outputs and now inputs), are no longer selected to maximize optical transmission. The design objective is in fact different: since the device operates as a logic gate, it is essential to ensure a high level of distinguishability between the output logic states. In particular, the optical output power associated with the logical state "1" must be significantly different from that corresponding to the logical state "0", to allow a clear discrimination between the two logic levels.

The truth table of the AND logic gate is reported below 5.1.

The operating conditions and dimensions of the structure are the following:

- Working bandwidth: 1450 nm-1650 nm

Input A	Input B	Output	Output Power
0	0	0	P_{01}
0	1	0	P_{02}
1	0	0	P_{03}
1	1	1	P_{14}

Table 5.1: Truth table of the AND gate with two inputs, showing the binary output and the corresponding output powers.

- Core refractive index: $n_{\text{core}} = 3.23$
- Cladding refractive index: $n_{\text{clad}} = 1.44$
- Width of the input and output waveguides: 0.55 µm
- Length of the input and output waveguides: 5 µm
- Width of the MMI: 2.6 µm
- Length of the MMI: 2.6 µm
- Distance from center to center in the vertical direction between the two outputs: $0.5 \times 2 = 1$ µm
- Spatial resolution: 40 points/um
- Perfectly matched layer (PML) thickness: 2 µm

As described in 3.6, the chosen bandwidth for the Gaussian Source: $df = 0.15\, f_{cen}$. The considered bandwidth was sampled using 50 uniformly spaced frequency points.

The specifications of the pixels are as follows:

- pixel diameter: 100 nm
- center-to-center distance between pixels in both directions: 200 nm
- number of rows: 13
- number of columns: 13
- total number of pixels: 169

A qualitative representation of the structure with all the pixels of the matrix present is shown in Fig. 5.1

Compared to the previous pixel-placement experiment, smaller pixels are used in this study. This choice is motivated by the reduced dimensions of the MMI region considered for the AND gate. Consequently, the pixel size was scaled accordingly to maintain a sufficiently fine discretization of the design area.

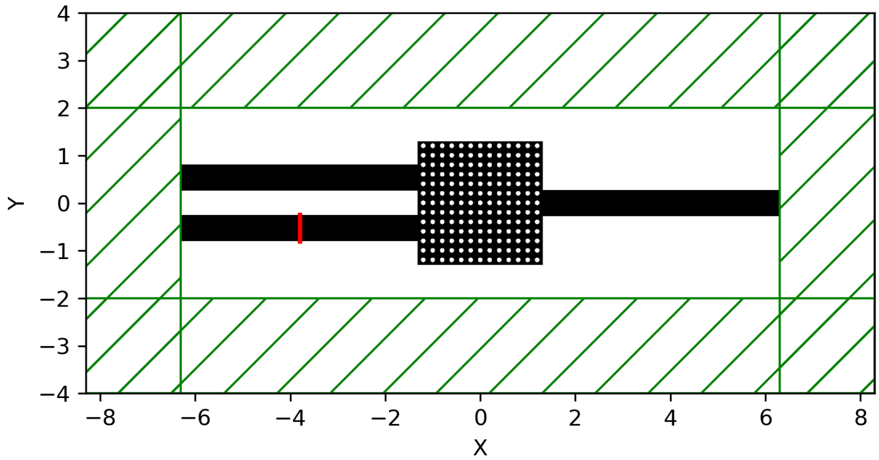


Figure 5.1: Qualitative representation of the structure with all the pixels of the matrix present.

The diameter of the pixels was chosen to be 100 nm, which is significantly smaller than the operating bandwidth, to introduce local perturbations of the refractive index without disrupting the propagation of the optical field within the multimode interference region.

The center-to-center spacing between the cylinders was set to 200 nm to avoid geometric overlap and to maintain a regular distribution of possible perturbations within the MMI region.

5.1 Optimization

As in the previous work, **pymoo** library is used to implement a Genetic Algorithm that explores different configurations of the pixels inside the MMI region. Each configuration is represented as a 169 bit long binary vector, where each element corresponds to the presence or absence of a cylinder ('1' indicates presence, while '0' indicates absence). For each candidate solution, `pymoo` calls a custom evaluation function, which passes the current configuration to the fitness function and returns the corresponding performance score.

Based on these evaluations, the algorithm iteratively improves the population by selecting, recombining, and mutating the best-performing configurations.

Fitness Function

The fitness function evaluates the performance of each candidate pixel configuration using a full electromagnetic simulation:

- $1 \rightarrow$ cylinder present ($n = n_{\text{clad}}$)
- $0 \rightarrow$ cylinder absent ($n = n_{\text{core}}$)

For each configuration provided by the optimization algorithm, the function builds the corresponding geometry by activating or deactivating cylinders inside the MMI region according to the binary vector.

To evaluate how distinguishable the logical values `1` and `0` are at the output, two different simulations are performed for each candidate pixel configuration. In the first case, both input waveguides are excited simultaneously (1,1), while in the second case, only one input is active (1,0). The corresponding values are denoted as T_{14} and T_{03}, respectively.

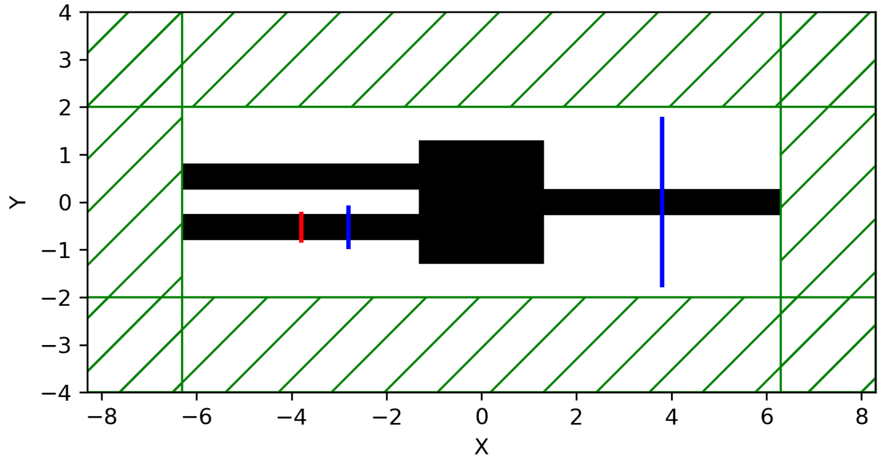
For both simulations, the T value is computed as

$$T = \frac{P_{\text{out}}}{P_{\text{in}}},$$

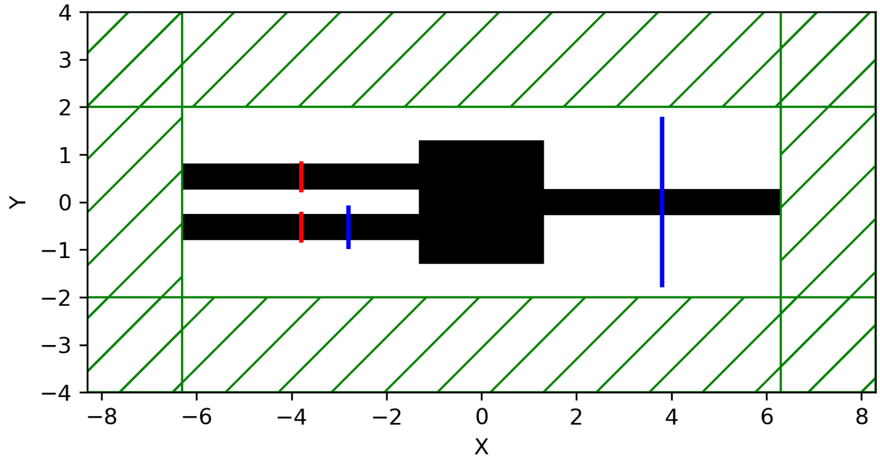
where P_{out} and P_{in} are the optical powers measured by the output and input flux monitors. In the (1,0) case, the input monitor is placed on the active input waveguide. In the (1,1) case, both sources are identical and active simultaneously, but the transmission is still normalized using a single input monitor.

It should be noted that T_{03} and T_{14} do not represent conventional transmission coefficients. In the (1,1) case, two identical input sources are simultaneously excited, whereas the normalization is performed using the power measured by a single input monitor. As a consequence, values of T_{14} greater than unity are expected and do not violate power conservation. In the ideal case, the objective is to obtain an output power equal to that generated by a single active input, even when both inputs are excited simultaneously, consistently with the behavior of an AND gate.

A qualitative representation of the structure, showing the positions of the sources and the input and output flux monitors for the two cases, is presented in Fig. 5.2



(a) Sources and monitor configuration for the (1,0) input state.



(b) Sources and monitor configuration for the (1,1) input state.

Figure 5.2: Qualitative representation of the score and the monitor positions used.

The values T_{14} and T_{03} are evaluated over 50 frequency points, resulting in two vectors. The score function is then defined as

$$score = \frac{1}{N}\sum_{i=1}^{N}\left(|1 - T_{14,i}| + |T_{03,i}|\right), \qquad N = 50 \tag{5.1}$$

where $T_{14,i}$ and $T_{03,i}$ are the values for the two cases at the (i)-th frequency point.

This definition encourages value T14 close to unity for the (1,1) input state and T03 close to zero for the (1,0) input state, which is consistent with the desired behavior of an optical AND gate. The fitness function therefore returns a single scalar value corresponding to the average score over the considered frequency range.

Only these two cases were simulated because:

- the case (0,0) does not produce any output power,
- the case (1,0) is equivalent, by symmetry, to the case (0,1).

Since the objective of this optimization is to minimize the score function, the score is passed directly to the optimizer without changing its sign.

5.2 Results Before and After the Optimization

An initial simulation of the structures without pixels was performed to evaluate the baseline values and the corresponding scores for the two considered cases before optimization. The results are reported below 5.2:

Table 5.2: Values obtained over the 50 frequency points considered in the optimization.

Quantity	Mean	Minimum	Maximum
T_{03}	0.6004	0.4966	0.6848
T_{14}	1.4095	1.1196	1.6882

$$score = \frac{1}{N}\sum_{i=1}^{N}\left(|1 - T_{14,i}| + |T_{03,i}|\right) = 0.99 \tag{5.2}$$

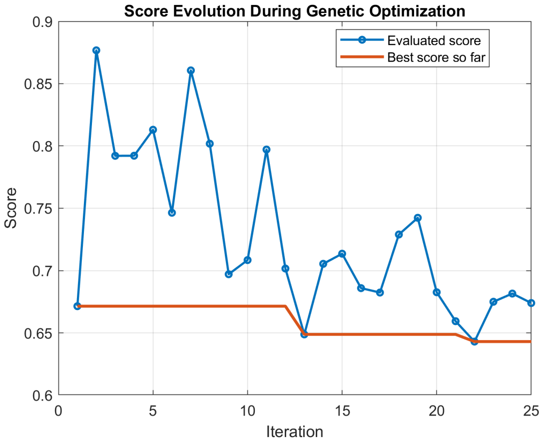
These results are consistent with the fact that the structure has not yet been optimized to perform the AND-gate operation. In particular, for the (1,0) case, it is reasonable that a non-negligible amount of optical power reaches the output, since no optimization has yet been performed to suppress this contribution. Similarly, for the (1,1) case, it is expected that the output power is higher than the value obtained with a single active input, as power is injected into both inputs and the structure has not yet been optimized to control or redistribute it according to the desired logical behavior.

Three optimization runs were performed using different values of the population size `pop_size` and the number of generations `n_gen`:

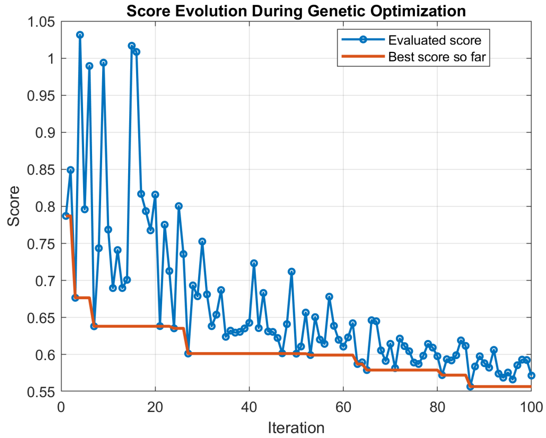
1. `pop_size = 5,  n_gen = 5`
2. `pop_size = 10,  n_gen = 10`
3. `pop_size = 20,  n_gen = 10`

Evolution of the Score Value as a Function of the Iterations

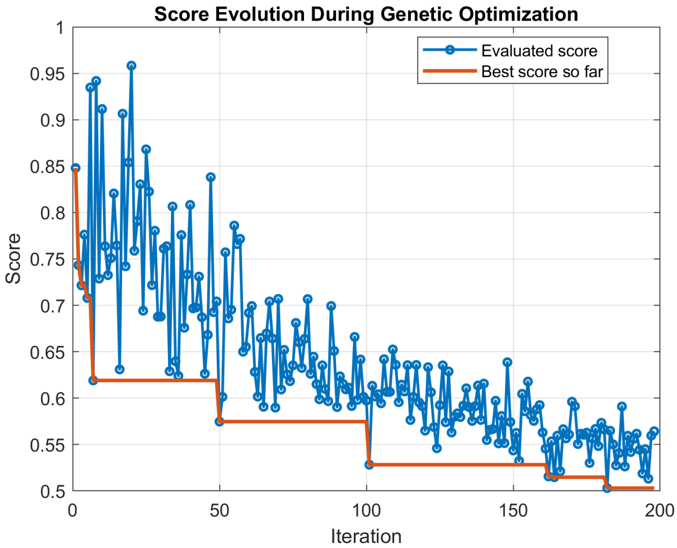
The evolution of the score values as a function of the iteration number for the three optimization cases is shown in Fig. 5.3.



(a) `pop_size = 5`, `n_gen = 5`.



(b) `pop_size = 10`, `n_gen = 10`.



(c) `pop_size = 20`, `n_gen = 10`.

Figure 5.3: Evolution of the score values as a function of the iteration number for the three genetic algorithm optimization cases.

- **`pop_size = 5, n_gen = 5`.** As shown in Fig. 5.3(a), the best score is reached at iteration 22 out of 25 and is equal to **0.643**. Although only a limited number of iterations is performed, the optimization already produces a noticeable improvement with respect to the initial score of approximately 0.99 obtained for the empty structure. As in the previous optimization experiments, the evaluated score fluctuates as different pixel configurations are explored, while the average value around which the score oscillates gradually decreases, indicating that the genetic algorithm is progressively identifying better solutions.

- **`pop_size = 10, n_gen = 10`.** As shown in Fig. 5.3(b), the best score is reached at iteration 87 out of 100 and is equal to **0.556**. Compared with the previous case, the larger population and number of generations allow the algorithm to explore the design space more effectively. During the first iterations, the score exhibits large fluctuations as many different configurations are evaluated. As the optimization progresses, the evaluated scores become concentrated around lower values, while the best score decreases. This behavior suggests that the genetic algorithm is successfully converging towards increasingly better pixel configurations and is effectively performing the intended optimization.

- **`pop_size = 20, n_gen = 10`.** As shown in Fig. 5.3(c), the best score is reached at iteration 182 out of 200 and is equal to **0.503**. The evolution of the score follows the same trend observed for the previous configurations. Starting from an initial random solution with a score of approximately 0.85, the genetic algorithm progressively explores better solutions, and the evaluated scores gradually shift towards lower values. Consequently, the best score found during the optimization decreases step by step until reaching its final value of approximately 0.50.

  Compared to the previous case, only a marginal improvement is achieved despite the larger population size. This suggests that, under the adopted optimization setup, the algorithm is approaching a performance plateau.

T_{03} and T_{14} values for the Three Optimization Runs

The minimum, mean, and maximum values of the T_{03} and T_{14} vectors, together with the corresponding score obtained for the three optimization cases, are reported in Table 5.3.

Table 5.3: Transmission statistics and score obtained for the best solutions found using the three genetic algorithm configurations.

Configuration	T_{03}			T_{14}			Score
	Mean	Min	Max	Mean	Min	Max	
`pop_size` = 5, `n_gen` = 5	0.4758	0.3717	0.5869	1.1188	0.8665	1.7068	0.643
`pop_size` = 10, `n_gen` = 10	0.4516	0.3406	0.6422	1.0084	0.7323	1.3105	0.556
`pop_size` = 20, `n_gen` = 10	0.3752	0.2747	0.4793	1.0640	0.7696	1.3910	0.503

As can be observed from Table 5.3, the second optimization run already provides a noticeable improvement over the first one. In particular, the mean value of T_{14} moves closer to the target value of 1, while the mean value of T_{03} decreases.

As already observed from the score evolution discussed in the previous section, only marginal improvements are obtained between the second and the third optimization runs. This suggests that, for the considered optimization setup, the genetic algorithm approaches a performance plateau. The design of an optical AND gate through pixel placement is an inherently challenging optimization problem. While driving T_{14} towards the ideal value of 1 is relatively achievable, forcing T_{03} towards zero is considerably more difficult, as it requires suppressing almost all the optical power at the output while one input waveguide remains excited. Simultaneously satisfying both requirements is therefore a highly demanding task.

For this reason, a significant further reduction of T_{03}, and consequently of the overall score, is not expected within the current optimization framework. Nevertheless, the obtained results can be considered satisfactory. The genetic algorithm successfully reduced the score from an initial value close to 1 to approximately 0.5, while producing devices with T_{14} values close to the desired target and a substantial reduction of the undesired transmission T_{03}. This confirms the effectiveness of the adopted optimization strategy despite the complexity of the problem.

Best Configurations for the Three Optimization Runs

The three best configurations obtained for the different values of `pop_size` and `n_gen`, together with their corresponding score values, are shown in Fig. 5.4.

Field Evolution at the central wavelength for the Three Optimization Runs

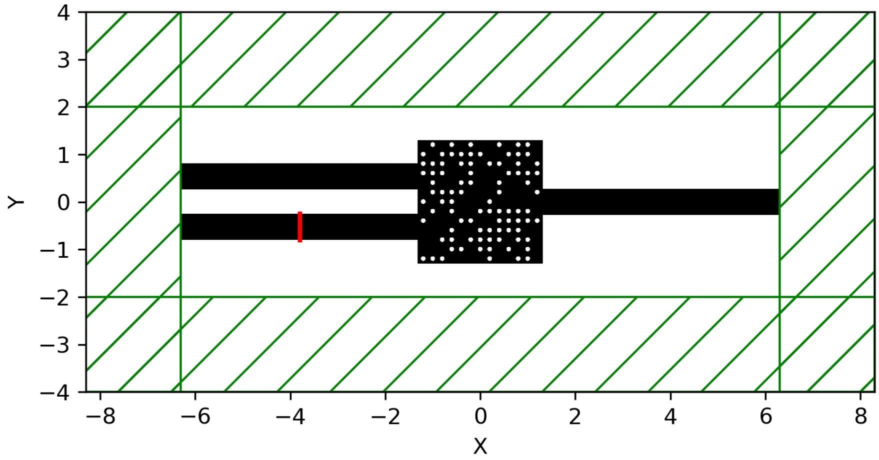
The field intensity distributions at the central wavelength of 1550 nm were analyzed for the optimized configurations. Since the device response depends on the input state, the fields were evaluated separately for the (1,0) and (1,1) cases. The corresponding results are shown in Fig. 5.5 and Fig. 5.6, respectively.

The electric field distributions obtained for the three optimization cases are reported in Figures 5.5 and 5.6, corresponding to the logical inputs (1,0) and (1,1), respectively. Each figure includes the field maps for the three different optimizations.

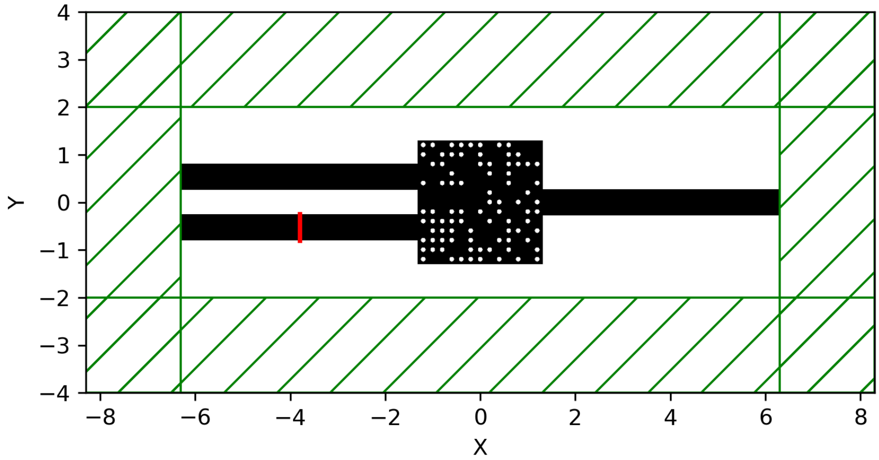
In all cases, the field evolution clearly exhibits the self-imaging behavior characteristic of multimode interference (MMI) devices, although this aspect is not of primary interest in the present analysis. The field maps provide only a qualitative confirmation of the device operation; the numerical results discussed in the previous sections offer a much more informative evaluation of the device performance.

As expected, for the logical input (1,0) only one input waveguide is excited, whereas for the logical input (1,1) both input waveguides simultaneously inject light into the device. In the (1,0) configuration, the optical intensity measured at the output is noticeably lower than the intensity launched into the active input waveguide, consistently with the behavior expected for the logical FALSE state of the AND gate. Conversely, in the (1,1) case, the output intensity is comparable to the intensity carried by a single input waveguide, indicating that the device successfully combines the contributions from both inputs and generates the desired logical TRUE output.

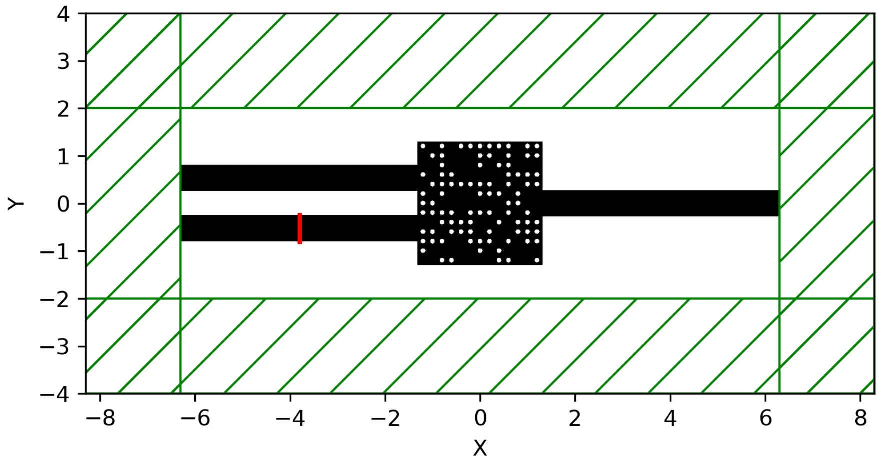
Overall, the genetic algorithm proved capable of effectively optimizing the pixel distribution of the device. Starting from an initial random population with score values



(a) `pop_size = 5`, `n_gen = 5`, `score = 0.643`

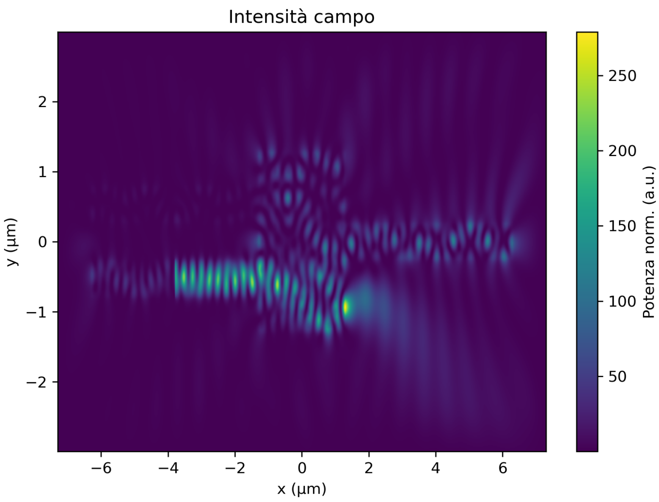


(b) `pop_size = 10`, `n_gen = 10`, `score = 0.556`

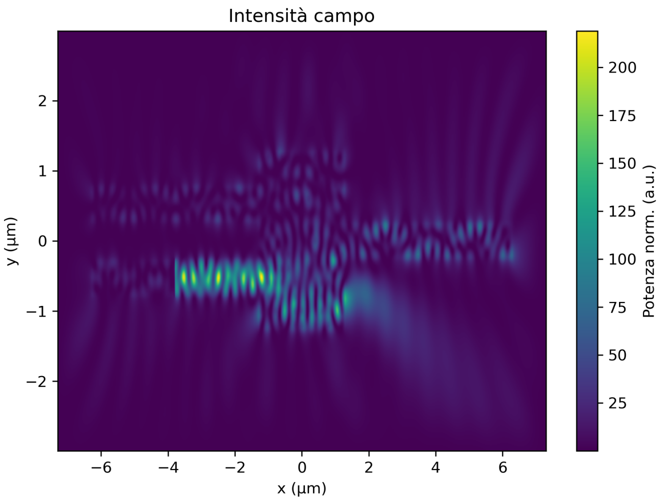


(c) `pop_size = 20`, `n_gen = 10`, `score = 0.503`

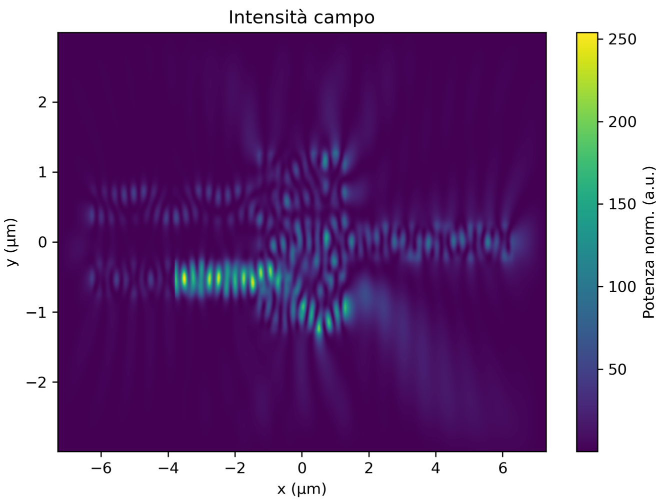
Figure 5.4: Best pixel configurations obtained for the three genetic algorithm optimization cases, together with their corresponding score values.



(a) `pop_size = 5`, `n_gen = 5`.

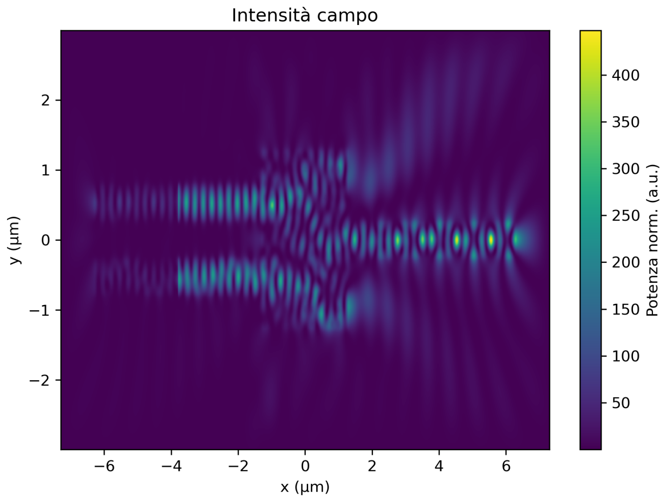


(b) `pop_size = 10`, `n_gen = 10`.

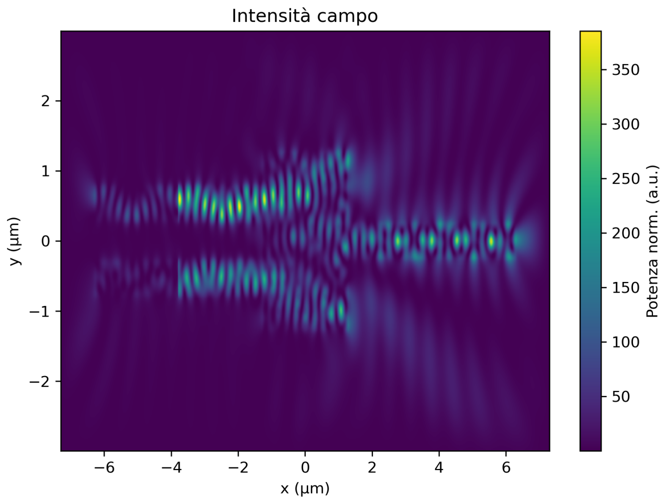


(c) `pop_size = 20`, `n_gen = 10`.

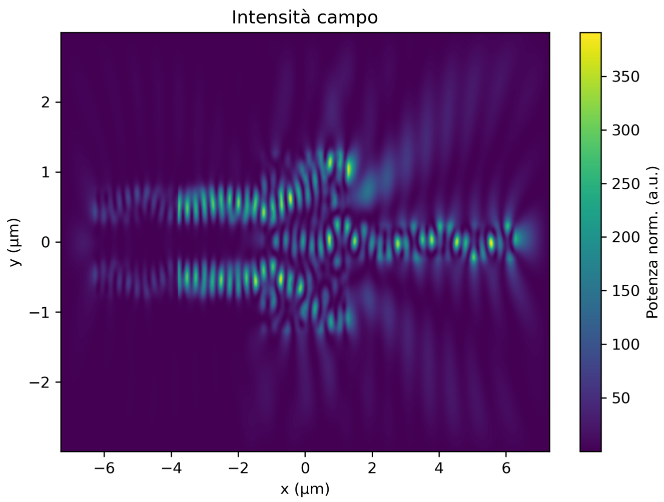
Figure 5.5: Field intensity distributions at λ= 1550 nm for the (1,0) input state and the three optimized pixel configurations.



(a) `pop_size = 5`, `n_gen = 5`.



(b) `pop_size = 10`, `n_gen = 10`.



(c) `pop_size = 20`, `n_gen = 10`.

Figure 5.6: Field intensity distributions at $\lambda = 1550$ nm for the (1,1) input state and the three optimized pixel configurations.

close to 1, the optimization consistently reduced the objective function down to approximately 0.5, demonstrating a significant improvement in the device performance. At the same time, the optimized structures achieved promising optical characteristics, with the desired transmission coefficient T_{14} approaching unity and the undesired transmission T_{03} decreasing below 0.4.

These results confirm that the adopted optimization strategy is able to identify meaningful device topologies despite the extremely large design space associated with the pixel-placement approach.

5.3 Possible Improvements

It should be noted that the output monitor used during the optimization was intentionally chosen to be larger than the output waveguide. This choice was made to ensure that all the transmitted power reaching the output region was collected.

However, this could represent a limitation of the analysis since part of the power collected by the monitor may correspond to radiation or field components not efficiently coupled into the output waveguide.

A more accurate evaluation of the device performance would require a smaller monitor aligned with the output guide. Due to the high computational cost of the genetic algorithm simulations, the full optimization was not repeated with a different monitor configuration.

Another possible improvement concerns the optimization strategy itself. The genetic algorithm required considerable computational effort. For instance, the longest optimization run `pop_size = 20`, `n_gen = 10` required almost four days to complete. Alternative optimization algorithms could therefore be investigated in order to reduce the computational time while maintaining comparable performance.

Furthermore, the score function adopted in this work could potentially be reformulated. A different definition of the objective function might lead to faster convergence. However, it should also be considered that the design of an optical AND gate through pixel placement is a highly complex optimization problem involving a very large design space.

Before concluding that the optimization has reached its practical performance limit, it would also be worthwhile to extend the longest optimization run `pop_size = 20`, `n_gen = 10` by increasing the number of generations. This could provide additional evidence on whether the score is able to decrease below the value of approximately 0.5 or not. Such an extended optimization was not performed in the present work due to the limited time available, considering that the longest simulation already required days of computation.

Based on the results obtained in this work, the current optimization framework appears to approach a score of approximately 0.5. If future simulations with a larger number of generations confirm that the score does not decrease further, improving the results will probably require more than just running the optimization for a longer time, such as adopting a different score formulation, a different device topology, or a more general inverse-design methodology.

Chapter 6

Inverse design using Adjoint Optimization

While the optimization techniques presented in the previous chapters were based on the adjustment of a limited set of geometrical parameters, the adjoint optimization method constitutes a true inverse design approach, allowing a much larger design space to be explored.

The adjoint method is a technique used to efficiently compute the sensitivity of an objective function with respect to a large number of design parameters. The method finds applications in several fields of engineering and computational physics, including fluid dynamics, structural optimization, and computational photonics.

Consider a physical system described by a set of equations that depend on a vector of design parameters $\mathbf{p}$. The objective is to determine how a quantity of interest F, which depends on the state of the system, varies as the parameters change. Mathematically, the goal is to compute the gradient

$$\nabla_{\mathbf{p}} F.$$

A direct approach consists of perturbing each parameter individually and evaluating the variation of the objective function using finite differences. However, when the number of parameters is large, this procedure requires a number of evaluations proportional to the number of variables and quickly becomes computationally prohibitive.

The fundamental idea behind the adjoint method is to reformulate the derivative computation problem by introducing an adjoint system. Instead of determining separately the effect of each parameter on the objective function, the method exploits the structure of the equations describing the system to obtain the entire gradient much more efficiently.

From an operational point of view, the procedure consists of two main steps. In the first step, the original problem is solved through a *forward simulation*, using the current design parameters. This simulation determines the state of the system and allows the objective function, i.e., the quantity to be optimized, to be evaluated.

Subsequently, an *adjoint simulation* is performed, consisting of the solution of an auxiliary problem constructed from the objective function. While the forward simulation describes the behavior of the system for a given configuration of parameters, the adjoint simulation provides information on how the objective function responds to variations of the system itself.

By combining the results of the forward and adjoint simulations, it is possible to compute the gradient of the objective function with respect to all design parameters simultaneously. The main advantage of this approach is that the computational cost of evaluating the gradient remains essentially independent of the number of optimization variables, making it possible to efficiently tackle problems characterized by thousands or even tens of thousands of degrees of freedom.

6.1 Adjoint Method in MEEP

The description presented in this section is based on the official MEEP Adjoint Solver documentation [3].

In MEEP, the adjoint method is implemented through the `MEEP.adjoint` module, which provides a framework for the gradient-based optimization of electromagnetic devices. The method is based on the solution of Maxwell's equations using the Finite-Difference Time-Domain (FDTD) method and enables the efficient computation of the sensitivity of an objective function with respect to the material distribution within a design region.

The design region is generally represented by a `MaterialGrid`, i.e., a discretized grid in which each cell is associated with a design variable. These variables control the permittivity distribution within the structure and constitute the degrees of freedom of the optimization problem.

The objective function can be defined in terms of different electromagnetic quantities, such as modal coefficients (S-parameters), field intensities, local density of states (LDOS), or near- and far-field quantities. The goal of the optimization process is to modify the material distribution in order to maximize or minimize the quantity of interest.

Similarly to the general formulation of the adjoint method, the gradient evaluation requires two simulations. The first is the *forward simulation*, in which the device is excited by the physical source of the problem. This simulation is used to compute the electromagnetic fields and evaluate the objective function.

Subsequently, an *adjoint simulation* is performed. In this case, the source no longer represents the physical excitation of the device but is automatically constructed from the derivative of the objective function with respect to the electromagnetic fields. The adjoint simulation therefore provides information on how a local variation of the material distribution affects the objective function.

Once the fields from the forward and adjoint simulations have been obtained, the gradient is computed by combining the information from both simulations within the design region. This step is performed automatically by MEEP's adjoint module.

The *forward simulation* provides the electromagnetic fields associated with the physical excitation of the device. The *adjoint simulation*, constructed from the objective function, provides information about the sensitivity of the objective function with respect to the electromagnetic fields. By combining the forward and adjoint fields within the design region, a sensitivity map is obtained, which allows the gradient of the objective function with respect to all design variables to be computed simultaneously.

The main advantage of this approach is that the computational cost of the gradient evaluation does not depend on the number of design variables. Even in the presence of thousands of independent parameters, only one forward simulation and one adjoint simulation are required for each iteration of the optimization algorithm.

6.2 Design and Topology Optimization of a 1×2 Optical Splitter Using MEEP's Adjoint Solver

The design of the optical splitter was carried out using the topology optimization capabilities provided by MEEP's adjoint solver. As a starting point for the development of the optimization framework, the *waveguide mode converter* example provided in the MEEP adjoint solver tutorials was used [3].

This example was chosen because it already includes the fundamental elements required for topology optimization, such as the definition of a `MaterialGrid`, the computation of gradients through the adjoint method, and the use of a *minimax* optimization strategy.

Starting from this implementation, the code was modified and adapted for the design of an optical splitter. While preserving the overall structure of the optimization algorithm, several aspects of the problem were redefined, including the device geometry, the objective function, and the corresponding design constraints.

In the following sections, all the main components of the optimization framework will be described in detail. Particular attention will be given to the algorithms and techniques employed in the optimization process, including the filtering and projection procedures, the epigraph formulation, the *minimax* optimization strategy, the definition of the objective functions, and the computation of gradients through the adjoint method. In addition, the physical and numerical configuration of the problem will be presented, including the splitter geometry, the design region, the material distribution, the excitation source, and the monitoring regions used to evaluate the device performance.

After introducing and discussing each component individually, the overall workflow of the optimization procedure will be described, illustrating the sequence of operations performed by the main program during each optimization iteration.

6.2.1 Optimization Problem Setup

Main Parameters

The main geometrical, material, and simulation parameters used for the optimization of the splitter are summarized in Table 6.1. These values define the physical configuration of the device as well as the numerical settings adopted throughout the optimization process.

The simulation resolution was set to 30 pixels/um to reduce computational costs and simulation time during the optimization process. A higher resolution could provide a more accurate representation of the device geometry and electromagnetic fields, at the expense of a significantly increased computational effort.

Table 6.1: Main parameters of the splitter optimization problem.

Parameter	**Value**
Simulation resolution	30 pixels/µm
Operating wavelength range	1.26 µm - 1.30 µm
Optimization wavelengths	1.265, 1.270, 1.275, 1.285, 1.290, 1.295 um
Waveguide dimensions	3.0 um × 0.32 µm
Design region dimensions	1.6 µm × 1.6 µm
Center-to-center separation of output waveguides	0.6 µm
Waveguide and design region refractive index	2.96
Background refractive index	1.0
PML thickness	1.0 µm
Padding above and below the design region	0.6 µm

The wavelength range and the optimized wavelengths were inherited from the original waveguide mode converter example provided in the MEEP adjoint solver tutorials. These wavelengths belong to the O-band, which is commonly used in optical telecommunications.

The waveguide length was also kept equal to that of the original example. Since the optimization is performed only within the design region, the exact waveguide length is not a critical parameter, provided that sufficient space is available for mode excitation and monitoring.

The perfectly matched layers (PMLs) and the padding regions surrounding the design area were inherited from the original MEEP waveguide mode converter example and retained in the present work. A PML thickness of 1.0 µm was used on all boundaries of the simulation domain to absorb outgoing electromagnetic waves and suppress artificial reflections from the edges of the computational cell. Similarly, a padding region of 0.6 µm was maintained above and below the design region, providing sufficient separation between the device and the absorbing boundaries.

Some of the parameters adopted in this work differ from those used in the original waveguide mode converter example. These modifications were introduced to better represent the characteristics of the considered optical platform and the specific requirements of the splitter design.

In particular, the waveguide width was set to 320 nm. This value was selected to ensure proper confinement of the fundamental guided mode at the operating wavelength around 1300 nm while maintaining a compact device geometry. Although larger widths could also support guided propagation, 320 nm represents a suitable compromise between optical confinement and device dimensions.

The refractive index of the waveguide material was set to $n_{\mathrm{Si}} = 2.96$, while the surrounding medium was assigned a refractive index of $n = 1$. These values differ from those used in the original tutorial and were chosen to better represent the considered device. The background medium therefore corresponds to air rather than silicon dioxide.

The considered device is based on a silicon waveguide with a thickness of 220 nm. This dimension does not explicitly appear in the optimization code because the simulations were performed using a two-dimensional FDTD model. As a result, only the in-plane geometry is represented, while the finite thickness of the silicon layer is neglected in the numerical model.

The geometry of the optimized splitter used in the simulations is shown in Fig. 6.1.

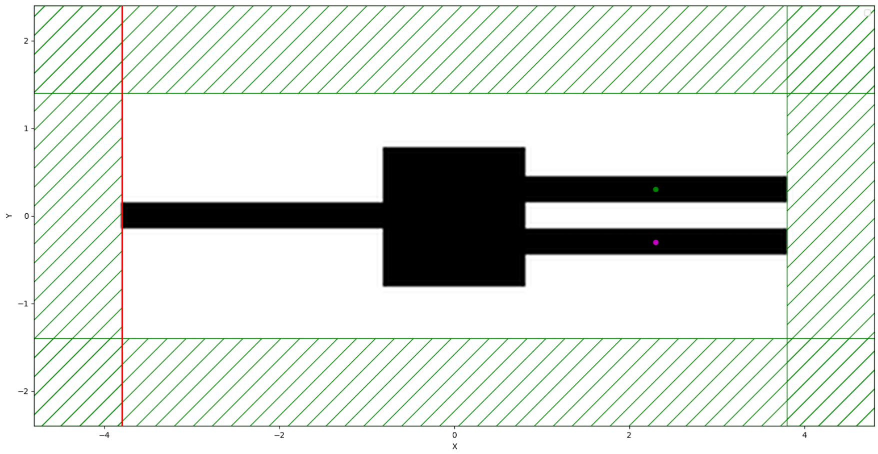


Figure 6.1: Geometry of the splitter

Minimax Optimization and Epigraph Formulation

The design of an optical splitter involves multiple performance objectives that must be simultaneously satisfied over the entire wavelength range of interest. In the present work, two objective functions are evaluated at each of the six design wavelengths. The first objective function, J_1, is associated with the reflected power, while the second objective function, J_2, measures the quality of the power splitting between the two output waveguides. A detailed description of these objective functions and their physical meaning will be provided in a subsequent section. Consequently, each optimization iteration produces a total of twelve objective values.

A straightforward approach would consist of minimizing the average value of these objectives. However, such a strategy may lead to designs that perform very well at some wavelengths while exhibiting poor performance at others. Since the goal of this work is to obtain a splitter with uniform behavior across the entire operating bandwidth, a different optimization strategy is required.

For this reason, a *minimax* formulation is adopted. Rather than minimizing the average performance metric, the optimization seeks to minimize the worst objective value among all wavelengths and all objective functions. Mathematically, the problem can be expressed as

$$\min \max_i f_i,$$

where f_i represents the set of objective values evaluated during the optimization. In this way, the optimization process focuses on improving the worst-performing condition, leading to a more robust and broadband device.

Directly solving a *minimax* problem is not straightforward within the optimization framework employed in this work. Therefore, an auxiliary variable, called the *epigraph variable* and denoted by t, is introduced. This variable is constrained to be greater than or equal to every objective value:

$$f_i \leq t, i = 1, \ldots, 12.$$

If all these constraints are satisfied, the value of t corresponds to the largest objective value among the entire set of objectives. The original *minimax* problem can therefore be reformulated as

$$\min t$$

subject to

$$f_i \leq t.$$

This approach is commonly known as the epigraph formulation of the *minimax* problem.

From a physical point of view, the epigraph variable can be interpreted as a measure of the worst performance of the device.

During the optimization process, the algorithm continuously attempts to reduce the value of t, thereby reducing the largest objective value and improving the overall performance uniformity across the considered wavelength range.

The implementation of this formulation is carried out through the `epigraph_constraint()` function.

At each iteration, the current design weights are first processed through the filtering and projection stages, and then used to construct the material distribution within the design region. The resulting geometry is subsequently analyzed through electromagnetic simulations, which evaluate both objective functions at all design wavelengths.

The objective values are collected into a single vector containing all reflectance and splitting-performance evaluations. The epigraph constraints are then constructed by comparing each objective value with the current value of the epigraph variable. In particular, the constraint function evaluates

$$f_i - t$$

A negative value indicates that the corresponding objective satisfies the epigraph constraint, whereas a positive value indicates that the objective exceeds the current value of t and therefore violates the minimax condition.

The optimization process proceeds by simultaneously adjusting the design weights and the epigraph variable in order to satisfy all constraints while progressively reducing the value of t. As a result, the final optimized design is not obtained by improving only a subset of wavelengths, but rather by balancing the performance over the entire operating band and minimizing the worst-case behavior of the splitter.

Continuation Strategy and Projection Parameter

A key element of the optimization procedure is the projection parameter β, which controls the transition between a continuous and a binary material distribution within the design region. As discussed later in the description of the filtering and projection process 6.2.1, the projection function maps the design weights toward either air or silicon. The sharpness of this mapping is directly determined by the value of β.

For small values of β, the projection function is relatively smooth and allows a large number of intermediate material values to exist within the design region. This provides the optimizer with greater flexibility during the initial stages of the optimization, facilitating the exploration of the design space. As β increases, the projection becomes progressively sharper, and the design weights are pushed toward binary values, corresponding to either air or silicon. Consequently, the geometry gradually evolves from a smooth distribution containing many gray regions to a nearly binary and physically realizable structure.

Rather than using a fixed projection parameter, a continuation strategy is adopted. The optimization is divided into a sequence of epochs characterized by increasing values of β.

For each value of β, the optimization is performed for a prescribed number of iterations before proceeding to the next stage.

This gradual increase in the projection parameter allows the optimization to first identify promising geometrical configurations and only later enforce a nearly binary material distribution.

The fabrication constraints described in the following section are not applied throughout the entire optimization process. Instead, they are activated only when

$$\beta \geq 224.$$

This choice prevents the fabrication constraints from excessively restricting the design space during the early stages of the optimization, while still ensuring the manufacturability of the final structure.

The optimization framework also includes additional numerical techniques aimed at improving convergence and stability. In particular, a damping term is introduced in the MaterialGrid formulation 6.2.1. The purpose of this damping is to penalize intermediate material values and promote the gradual formation of a binary design. The damping mechanism remains active throughout the optimization process.

Furthermore, the code supports the use of subpixel smoothing.This technique becomes particularly useful at high values of β, where the material interfaces become increasingly sharp. By averaging the permittivity distribution at subpixel resolution, the method can improve numerical stability and produce smoother gradients during the adjoint optimization. However, preliminary tests performed for the present splitter design showed that enabling subpixel smoothing degraded the optical performance of the optimized device. For this reason, the final simulations were carried out with this option disabled.

Overall, the continuation strategy based on the progressive increase of β, together with damping, optional subpixel smoothing, and the late introduction of fabrication constraints, provides a robust framework for obtaining a manufacturable binary design while maintaining good optical performance.

Linewidth and Spacing Constraints

In addition to the optical objectives, fabrication constraints were introduced in the optimization process to ensure that the final splitter geometry can be realistically manufactured. Without such constraints, the optimizer could generate extremely small silicon features or very narrow air gaps, which would be difficult or impossible to fabricate using standard lithographic processes.

In this work, a minimum feature size of 150 nm was imposed by setting
`MIN_LENGTH_UM`=0.15.

This value defines the minimum acceptable dimension for both silicon regions and air regions within the optimized device.

The fabrication constraints are not enforced during the initial stages of the optimization. Instead, they are introduced only when the projection parameter satisfies

$$\beta \geq 224.$$

The rationale behind this choice is that, during the early stages of the optimization, the algorithm must be allowed to explore the design space as freely as possible. Only once the structure has evolved toward a nearly binary distribution of materials are the fabrication constraints activated.

Two different constraint functions are employed. The first one is the solid constraint, implemented through `mpa.constraint_solid()`, which controls the minimum linewidth of the silicon regions. The second one is the void constraint, implemented through `mpa.constraint_void()`, which controls the minimum spacing of the air regions. Together, these two constraints ensure that both the silicon features and the air gaps remain larger than the prescribed minimum feature size.

Whenever the constraint function is evaluated, the current design weights are first filtered using the same conic filter adopted in the mapping procedure and subsequently projected through the hyperbolic tangent projection.

The resulting filtered and projected design is then analyzed through the two constraint functions, producing the quantities M_1 and M_2. The quantity M_1 measures possible violations of the minimum linewidth requirement, while M_2 measures possible violations of the minimum spacing requirement. Larger values of these quantities indicate a greater degree of violation of the corresponding fabrication constraint.

The constraints are formulated as

$$M_1 - a_1 t - b_1 \leq 0,$$

$$M_2 - a_1 t - b_1 \leq 0,$$

where t is the epigraph variable and a_1 and b_1 are hyperparameters used to scale the constraint. In the implemented code, the values

$$a_1 = 1 \times 10^{-4}, b_1 = 0$$

were adopted. The additional scaling parameter

$$c_0 = 1 \times 10^7 \left(\frac{filter\_radius}{\text{resolution}}\right)^4$$

is used internally by the constraint functions to control the penalty associated with violations of the minimum feature size.

From M_1 and M_2, two auxiliary quantities are computed:

$$t_1 = \frac{M_1 - b_1}{a_1}, t_2 = \frac{M_2 - b_1}{a_1}.$$

These values represent the minimum value of the epigraph variable required to satisfy the linewidth and spacing constraints, respectively. During the optimization process, they provide a useful indication of the severity of the constraint violations.

The erosion and dilation thresholds employed in the constraint evaluation were also modified with respect to the original implementation. In particular, the values

$$\eta_{\text{erosion}} = 0.55,\ \eta_{\text{dilation}} = 0.45$$

were selected in order to obtain softer fabrication constraints. Compared to more aggressive threshold values, this choice reduces the tendency of the constraints to excessively penalize the design, allowing the optimizer greater flexibility while still enforcing the desired minimum feature size.

Finally, the function computes the gradients of both constraint functions with respect to all design weights. These gradients are obtained through automatic differentiation and are passed to the NLopt optimizer together with the constraint values. As a result, the optimizer not only determines whether the current design satisfies the fabrication requirements but also receives information on how each design variable should be modified in order to reduce any linewidth or spacing violations while maintaining the optical performance of the device.

Filtering and Projection

The design variables generated by the optimizer are processed through a filtering and projection procedure. This approach improves the robustness of the optimization process and helps ensure that the resulting structures are physically meaningful and compatible with fabrication constraints.

The first operation performed by the `filter_and_project()` function consists of applying a set of border masks. These masks define regions of the design domain whose material composition must remain fixed throughout the optimization. In particular, the portions of the design region corresponding to the input and output waveguides are forced to remain silicon, while other boundary regions are constrained to remain air. This guarantees the continuity of the waveguides and prevents the optimizer from modifying areas that must preserve a predefined material distribution.

After applying the border masks, the design weights are processed using a conic filter through the `mpa.conic_filter()` function. The purpose of this filter is to remove very small geometrical features and enforce the minimum length scale prescribed by the geometric constraints.

The filtering operation replaces the value of each pixel with a weighted average of the neighboring pixels contained within a given filter radius. The weights assigned to neighboring pixels decrease linearly with distance from the central pixel, resulting in a conical weighting function.

Consequently, isolated pixels and abrupt local variations are smoothed, producing a more regular material distribution.

The filter radius is computed from the minimum feature size specified for the optimization and therefore directly controls the smallest geometrical detail that can appear in the final design.

Although the filtering operation improves the smoothness of the geometry, it introduces a large number of intermediate pixel values between 0 and 1. These so-called gray pixels correspond to intermediate material states which are not physically meaningful, since the final device should consist only of air and silicon. To gradually eliminate these intermediate values, a projection operation is applied using the `mpa.tanh_projection()` function.

The projection is based on a hyperbolic tangent function, which pushes the filtered weights toward either 0 or 1 according to the expression

$$\tilde{w} = \frac{\tanh(\beta\eta) + \tanh\left[\beta(w - \eta)\right]}{\tanh(\beta\eta) + \tanh\left[\beta(1 - \eta)\right]} \tag{6.1}$$

where w is the filtered design weight, η is the projection threshold, which is set to 0.5 in this work, and β controls the steepness of the projection function.

For small values of β, the transition between air and silicon is gradual and a significant number of gray pixels remain in the design.

As β increases, the projection becomes progressively sharper, driving the weights toward binary values. This continuation strategy allows the optimizer to explore the design space more freely during the initial stages of the optimization while gradually enforcing a manufacturable binary structure.

The filtered and projected weights obtained through this process are finally passed to the `MaterialGrid` object, which converts them into the corresponding permittivity distribution used by the electromagnetic simulations.

Design Region and MaterialGrid

The design region defines the portion of the computational domain in which the material distribution can be modified during the optimization process. In the present work, the dimensions of the design region were kept identical to those adopted in the original waveguide mode converter example, namely 1.6 µm × 1.6 µm. This area provides sufficient space for the optimization algorithm to generate the splitter geometry while maintaining a compact device footprint.

Two different spatial discretizations are employed in the optimization framework. The first is the Yee grid used by the FDTD solver to compute the electromagnetic fields, whose resolution was set to 30 pixels/µm. The second is the design grid associated with the MaterialGrid, whose resolution was chosen to be twice as large. While the Yee grid determines the numerical accuracy of the electromagnetic simulation, the design grid defines the number of optimization variables and the level of detail with which the material distribution can be represented. The higher design resolution allows for a finer description of the geometry and provides additional flexibility during the optimization process. The resulting material distribution is subsequently mapped onto the FDTD grid through MEEP's subpixel averaging procedure.

The design region is represented by a rectangular grid of design variables, also referred to as design weights. Given the dimensions of the design region and the adopted design resolution, the grid contains $N_x = 97$ pixels along the horizontal direction and $N_y = 97$ pixels along the vertical direction, resulting in a total of $N = N_x \times N_y = 9409$ design variables.

At each optimization iteration, the MaterialGrid object receives the filtered and projected design weights and converts them into a spatial distribution of permittivity within the design region. Each weight is associated with a pixel of the design grid and determines the local interpolation between air and silicon. More specifically, the permittivity assigned to each pixel is given by

$$\varepsilon = \varepsilon_{\text{air}} + \tilde{w}\left(\varepsilon_{\text{Si}} - \varepsilon_{\text{air}}\right) \tag{6.2}$$

where $\tilde{w}(\mathbf{r})$ denotes the filtered and projected design weight, while ε_{air} and ε_{Si} are the permittivities of air and silicon, respectively.

As a result, pixels with $\tilde{w} = 0$ correspond to air, pixels with $\tilde{w} = 1$ correspond to silicon, and intermediate values represent a continuous interpolation between the two materials. The resulting permittivity map is then used by the forward and adjoint simulations to evaluate the objective function and its gradient. In this way, the MaterialGrid acts as the link between the optimization variables and the electromagnetic model of the device.

Excitation Source and Modal Measurements

The splitter is excited through a guided optical mode launched at the input waveguide. The source is implemented as a Gaussian excitation, providing a broadband spectrum that allows the performance of the splitter to be evaluated over the entire wavelength range considered in the optimization.

It is configured to excite only the fundamental mode of the structure (see the class `EigenModeSource`), which is the mode of practical interest for device operation.

The source is defined to excite the fundamental transverse-electric (TE) mode supported by the waveguide, ensuring that the optimization is carried out for the polarization of interest.

Before analyzing the splitter itself, a preliminary simulation is performed on a straight waveguide. The purpose of this simulation is to determine the input power associated with the injected mode.

This reference is subsequently used to normalize all power measurements, ensuring that the device performance can be evaluated independently of the absolute source amplitude.

A first monitoring section is placed near the input waveguide and is used to quantify the power reflected back toward the source. Two additional monitoring sections are positioned at the upper and lower output branches of the splitter and are used to measure the transmitted power in each output waveguide.

Rather than measuring the total electromagnetic field, the adopted modal analysis extracts only the contribution associated with the fundamental guided mode. This distinction is important because it allows the optimization to evaluate the useful power effectively coupled into the desired propagation mode, while excluding radiated fields and possible higher-order modes.

The excitation source and and output modal monitors described above are employed during both the forward and adjoint simulations.

During the forward simulation, the modal coefficients measured at the monitoring sections are used to evaluate the objective functions and quantify the optical performance of the current design. Subsequently, the adjoint simulation uses the information provided by these objective functions to determine how small variations of the material distribution affect the device response.

By combining the results of the forward and adjoint simulations, the optimization framework is able to efficiently compute the gradients of the objective functions with respect to all design variables. These gradients are then provided to the optimization algorithm, which updates the material distribution in order to progressively improve the splitter performance.

Objective Functions

The optical performance of the splitter is evaluated through two objective functions. These functions are computed from the modal powers measured at the reflection monitor and at the two output waveguides. Their purpose is to quantify the quality of the splitting operation and provide the optimization algorithm with a numerical measure of the device performance.

The first objective function is associated with the reflected power and is defined as

$$J_1 = R,$$

where

$$R = \frac{|S_{11}|^2}{P_{\text{in}}}$$

represents the normalized reflectance of the device. Since the desired behavior is to transfer as much power as possible toward the output waveguides, the reflected power should be minimized. Therefore, the optimal value of J_1 is zero.

The normalization factor P_{in} corresponds to the optical power carried by the fundamental mode of a straight waveguide. This quantity is obtained through a preliminary normalization simulation performed before the optimization begins, as described in the previous section. By dividing all reflected and transmitted powers by P_{in}, the objective functions become independent of the absolute amplitude of the excitation source and depend only on the actual performance of the splitter.

The second objective function is designed to evaluate the quality of the power splitting. It simultaneously accounts for three different requirements: equal power distribution between the two output branches, low insertion loss, and low reflection. It is defined as

$$J_2 = 2\left(T_1 - \frac{1}{2}\right)^2 + 2\left(T_2 - \frac{1}{2}\right)^2 + R + [1 - (T_1 + T_2)]^2 ,$$

where

$$T_1 = \frac{|S_{21,\mathrm{top}}|^2}{P_{\mathrm{in}}}$$

and

$$T_2 = \frac{|S_{31,\mathrm{bottom}}|^2}{P_{\mathrm{in}}}$$

are the normalized transmitted powers at the upper and lower output waveguides, respectively.

The first two terms penalize deviations from the ideal 50:50 power split. The factors of 2 multiplying these terms increase the importance of the balancing condition within the optimization process, encouraging the algorithm to achieve equal power distribution between the two outputs. The third term penalizes reflected power, while the last term penalizes insertion losses by measuring the deviation from the ideal condition $T_1+T_2 = 1$, corresponding to complete transmission of the input power into the output waveguides.

The minimum value of J_2 is obtained when

$$T_1 = T_2 = 0.5,$$

$$R = 0,$$

and

$$T_1 + T_2 = 1.$$

Under these conditions all terms vanish and the objective function becomes equal to zero.

The quantities S_{11}, $S_{21,\mathrm{top}}$, and $S_{21,\mathrm{bottom}}$ are complex modal coefficients obtained from the modal monitors. Since optical power is proportional to the squared magnitude of the electromagnetic field, the reflected and transmitted powers are computed using the squared modulus of the corresponding scattering parameters. Consequently, the terms $|S_{11}|^2$, $|S_{21,\mathrm{top}}|^2$, and $|S_{31,\mathrm{bottom}}|^2$ represent the optical power carried by the fundamental mode at the reflection monitor and at the two output waveguides, respectively.

Both objective functions are evaluated at each of the six design wavelengths. Consequently, the optimization process produces a total of twelve objective values at every iteration.

These values are not minimized independently. Instead, they are passed to the epigraph formulation described previously, where they define the *minimax* constraints of the optimization problem.

Through the epigraph formulation, the optimization algorithm seeks to minimize the largest objective value among all wavelengths and all objective functions.

As a result, the final design is not optimized for a single wavelength or a single performance metric, but rather for the worst-case condition over the entire operating bandwidth. This approach promotes a more uniform and robust splitter response across the considered wavelength range.

OptimizationProblem

The `mpa.OptimizationProblem` object represents the core component of the adjoint optimization framework. Its purpose is to establish the connection between the electromagnetic simulation, the design variables, the objective functions, and the optimization algorithm. Once the simulation geometry, the design region, the modal monitors, the objective functions, and the operating wavelengths have been defined, these elements are collected into a single `OptimizationProblem` object.

During the optimization process, the object is called every time the current performance of the device must be evaluated. Given a set of design weights, it first updates the material distribution inside the design region and performs a forward electromagnetic simulation of the current structure. The modal coefficients measured at the input and output monitors are then used to evaluate the objective functions at all design wavelengths.

After the objective functions have been computed, the same object automatically performs the corresponding adjoint simulations. By combining the information obtained from the forward and adjoint runs, it evaluates the sensitivity of the objective functions with respect to the design variables. In other words, it determines how the objective functions would change if the material distribution were modified locally within the design region.

As a result, the `OptimizationProblem` object returns two types of information. The first is the numerical value of the objective functions evaluated at all wavelengths, which quantifies the performance of the current design. The second is the corresponding gradient information, which indicates the direction in which the design variables should be modified in order to improve the device performance.

These outputs are subsequently used by the optimization framework. The objective function values are processed by the epigraph formulation to construct the *minimax* constraints, while the gradients are supplied to the numerical optimizer. In the present work, the optimizer is implemented through the `NLopt` library using the CCSA algorithm, which exploits the gradient information provided by the adjoint method to iteratively update the design variables and improve the splitter performance.

The `OptimizationProblem` object, therefore, acts as the interface between the electromagnetic solver and the optimization algorithm. It receives the current device geometry as input and provides the performance metrics and sensitivities required by the optimizer to generate the next design iteration.

NLopt and the CCSA Algorithm

While the `OptimizationProblem` object is responsible for evaluating the performance of a given design and computing the corresponding gradients, the actual optimization process is carried out by the `NLopt` library. `NLopt` is an open-source numerical optimization package that provides a large collection of optimization algorithms for constrained and unconstrained problems. Its role within the present framework is to determine how the design variables should be updated in order to improve the splitter performance while satisfying all imposed constraints.

At each optimization iteration, `NLopt` receives the current values of the objective functions and the corresponding gradients computed through the adjoint method. Using this information, it generates a new set of design variables and repeats the evaluation process. This iterative procedure continues until the prescribed number of iterations is reached or until no significant improvement can be achieved.

In this work, the selected optimization algorithm is the Conservative Convex Separable Approximation with quadratic terms (`CCSAQ`).

At each iteration, `CCSAQ` receives the values of the objective functions and their corresponding gradients computed through the adjoint method. Using this information, the algorithm determines how the design variables should be modified in order to improve device performance while satisfying all imposed constraints.

Rather than attempting to solve the complete optimization problem directly, `CCSAQ` constructs a simplified local approximation around the current design and uses it to compute an updated set of design variables. This process is repeated iteratively until the optimization converges. The conservative nature of the algorithm prevents excessively large design updates, improving the stability and robustness of the optimization procedure.

`NLopt` and `CCSAQ` do not directly perform any electromagnetic computation. Their role is to use the information generated by the adjoint framework to determine the most favorable modification of the design variables at each optimization step.

Gradient Computation and Backpropagation

The optimization process relies on the availability of gradient information describing how the objective functions vary with respect to the design variables. In general, the gradient represents the sensitivity of a given objective function to small modifications of the material distribution within the design region. Each component of the gradient indicates whether increasing or decreasing a specific design variable would improve or worsen the current device performance.

The gradients are computed automatically through the adjoint method implemented within the `OptimizationProblem` object. For each design iteration, a forward simulation is first performed to evaluate the objective functions. Subsequently, the corresponding adjoint simulations are executed. By combining the electromagnetic fields obtained from the forward and adjoint runs, the adjoint framework evaluates the derivatives of the objective functions with respect to all design variables simultaneously.

The gradients computed by the adjoint solver are not directly expressed with respect to the design weights manipulated by the optimizer. Indeed, before entering the electromagnetic simulation, the design weights are transformed through the filtering and projection operations described previously. As a consequence, the gradients initially obtained from the adjoint method correspond to the filtered and projected structure rather than to the original optimization variables.

To overcome this issue, a backpropagation step is performed. Backpropagation is the process that transforms the gradients computed with respect to the filtered and projected structure into gradients with respect to the original design weights used by the optimizer. In practice, it transfers the sensitivity information obtained from the electromagnetic simulation back to the variables that `NLopt` can actually modify.

This operation is based on the chain rule of differential calculus. Since the objective functions J_1 and J_2 depend on the filtered and projected weights $\tilde{w}$, and these in turn depend on the original weights w, the derivative of the objective function with respect to the optimization variables is obtained as

$$\frac{\partial J}{\partial w} = \frac{\partial J}{\partial \tilde{w}} \frac{\partial \tilde{w}}{\partial w} \tag{6.3}$$

The first term is provided by the adjoint simulation, while the second term describes the effect of the filtering and projection operations. By combining these two contributions, it is possible to determine how the objective functions vary with respect to the original design variables.

In the implemented code, this operation is performed automatically through the `tensor_jacobian_product` function. The resulting gradients are then passed to the `CCSAQ` optimizer, which uses them to update the design weights and generate the next iteration of the optimization process.

6.2.2 Optimization Workflow

The overall optimization procedure implemented in the main program can be summarized through the following sequence of operations:

1. **Normalization simulation** A preliminary simulation of a straight waveguide is performed in order to compute the reference power associated with the fundamental guided mode. This quantity is later used to normalize the reflected and transmitted powers appearing in the objective functions.

2. **Generation of the design masks** The silicon and air border masks are generated. These masks enforce the presence of the input and output waveguides and ensure that the boundaries of the design region remain fixed throughout the optimization.

3. **Initialization of the design variables** The design weights are initialized to a uniform intermediate value of 0.5 across the entire design region, corresponding to a gray material distribution between silicon and air, where a value of 1 represents silicon and a value of 0 represents air, while the regions corresponding to the waveguide ports are fixed to silicon or air according to the previously defined masks. The

epigraph variable is also initialized to a value of 1, providing a feasible starting point for the *minimax* optimization problem.

4. **Definition of the continuation strategy** The continuation strategy is defined through a sequence of increasing projection parameters,

$$\beta = [8, 16, 32, 64, 90, 128, 160, 208, 240]$$

with the corresponding maximum number of optimization iterations per epoch

$$max\_evals = [50, 80, 80, 100, 100, 120, 120, 120, 100]$$

As the value of β increases, the projection function becomes sharper and the design progressively evolves toward a binary material distribution. For this reason, the number of optimization iterations is also increased at larger β values, allowing the optimizer more time to adapt to the increasingly constrained design space. This sequence was chosen to ensure a gradual and stable optimization process rather than an excessively rapid transition, giving the objective functions and the design variables sufficient time to converge before moving to the next continuation stage. The optimization is therefore executed through successive epochs characterized by increasing values of β.

5. **Optimization epoch for each value of** β For each value of the projection parameter β, a complete optimization epoch is executed. At the beginning of each epoch, a new `NLopt` solver is created using the `CCSAQ` algorithm. The number of optimization variables is equal to the number of design weights plus one additional variable corresponding to the epigraph variable.

   The lower and upper bounds of the optimization variables are then assigned. The design weights are constrained between 0 and 1, while the regions fixed by the border masks remain constrained to air or silicon. The epigraph variable is instead allowed to vary without fixed finite bounds.

   The scalar objective function passed to `NLopt` is the epigraph variable itself. Therefore, during each epoch, the optimizer attempts to minimize t. The *minimax* behavior is obtained by adding the epigraph constraints, which enforce the condition that all objective-function values must remain lower than or equal to t.

6. **Construction of the electromagnetic optimization problem** For the current value of β, the electromagnetic problem is built by defining the `MaterialGrid`, the design region, the splitter geometry, the excitation source, the modal monitors, and the objective functions. These elements are collected into an `OptimizationProblem` object, which represents the interface between the electromagnetic simulations and the numerical optimizer.

   At this stage, no optimization step has yet been performed. The `OptimizationProblem` object simply contains all the information required to evaluate the optical response of a given design and, when requested, to compute the corresponding gradients through the adjoint method.

7. **Initial calibration of the epigraph variable** Before starting the actual optimization epoch, the current design is evaluated once without computing gradients. In the code, this is done by calling the `OptimizationProblem` object with `need_gradient=False`. This evaluation performs only the forward simulation of the current structure and returns the values of the objective functions at all design wavelengths.

   The maximum value among these objective functions is then used to initialize the epigraph variable. In this way, the initial value of t is chosen consistently with the current optical performance of the device. This provides a feasible and stable starting point for the *minimax* optimization, since the epigraph variable is initialized close to the largest objective value.

   When the linewidth and spacing constraints are active, the initial value of the epigraph variable is also compared with the value required by these fabrication constraints. The largest between the optical objective values and the constraint-related value is used, with a small additional margin, to ensure that the optimizer starts from a feasible point.

8. **Execution of the NLopt optimization** After the epigraph variable has been initialized, the optimization epoch is started through the `NLopt` solver. During this stage, `NLopt` repeatedly evaluates the objective and constraint functions in order to update the design variables.

   Each time `NLopt` evaluates the epigraph constraints, the current vector of optimization variables is passed to the constraint function. This vector contains both the current epigraph value and all design weights. The design weights are first processed through the border masks, the conic filter, and the projection function. The resulting filtered and projected weights are then passed to the `OptimizationProblem` object.

   At this point, the forward and adjoint simulations are executed internally by MEEP. First, the forward simulation computes the electromagnetic response of the current splitter geometry. The modal monitors extract the reflected and transmitted modal powers, which are used to evaluate the objective functions. Then, the adjoint simulations are automatically performed by the adjoint solver in order to compute the gradients of the objective functions with respect to the design variables.

   The gradients returned by the adjoint solver are subsequently backpropagated through the filtering and projection operations. This step converts the sensitivities with respect to the filtered and projected design into sensitivities with respect to the original design weights controlled by the optimizer.

   The objective-function values are then compared with the current epigraph variable to form the *minimax* constraints. In particular, for each objective value f_i, the constraint $f_i - t \leq 0$ is evaluated. These constraint values, together with their gradients, are passed back to `NLopt`.

   If the fabrication constraints are active for the current value of β, the linewidth and spacing constraint functions are also evaluated. Their values and gradients are passed to `NLopt` in the same way as the epigraph constraints.

Using all this information, namely the epigraph objective, the *minimax* constraints, the fabrication constraints, and the corresponding gradients, the `CCSAQ` algorithm computes an updated set of design weights and an updated epigraph variable. This process is repeated until the maximum number of evaluations specified for the current epoch is reached.

9. **Post-processing and saving of the epoch result** At the end of the epoch, the optimized design weights returned by `NLopt` are filtered and projected once more in order to obtain the material distribution corresponding to the final design for the current value of β. This mapped design is reshaped into the two-dimensional design grid and saved for post-processing.

   For each epoch, the program saves the numerical optimization data, the unmapped design weights, the filtered and projected design, and a bitmap image of the optimized structure. These outputs make it possible to analyze the evolution of the design as β increases and to inspect how the structure progressively becomes more binary throughout the continuation process.

6.3 Optimization Results

The optimization procedure described in the previous section was executed using the continuation strategy defined by the sequence of projection parameters and the corresponding numbers of optimization iterations associated with each epoch. The gradual increase of β was adopted to avoid excessively abrupt transitions toward a binary design and to ensure a stable convergence of both the objective functions and the design variables throughout the optimization process.

The fabrication constraints associated with the minimum linewidth and minimum spacing were activated only in the final stages of the optimization, starting from $\beta = 224$. This choice allows the device to first approach a high-performance optical solution and only subsequently enforce manufacturability requirements, reducing the risk of prematurely restricting the design space.

The results presented in this section are intended to illustrate both the electromagnetic performance and the geometrical evolution of the splitter during the optimization process. In particular, the following aspects are analyzed:

- the evolution of the transmitted powers T_1 and T_2 and of the reflectance R as a function of the projection parameter β, evaluated over the six design frequencies;
- the evolution of the epigraph variable during the optimization process and the evolution of the fabrication constraint metrics M_1 and M_2 during the final optimization stages;
- the progressive evolution of the design region for increasing values of β, highlighting the transition toward a binary structure.

Together, these results provide a comprehensive overview of the convergence behavior of the optimization procedure and of the final performance achieved by the optimized splitter.

Optical Performance Evolution

This subsection analyzes the evolution of the optical performance throughout the optimization process. In particular, the transmitted powers at the two output ports, denoted as T_1 and T_2, and the reflectance R are monitored both during all optimization iterations and at the end of each optimization epoch corresponding to a specific value of the projection parameter β (Fig. 6.2). These results provide insight into the convergence behavior of the optimization algorithm, the effect of the continuation strategy, and the impact of the fabrication constraints introduced during the final stage of the optimization.

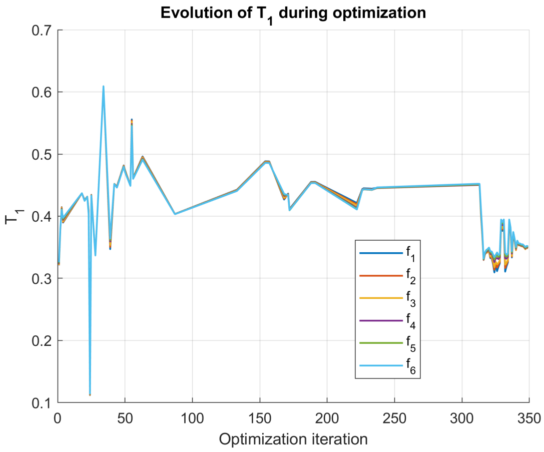
It should be noted that the iteration numbers reported in the following plots do not correspond exactly to the optimization iterations. During the optimization, the objective functions may be evaluated multiple times within a single optimization iteration. Moreover, in order to avoid storing an excessive amount of data, the values of T_1, T_2, and R are recorded only once every ten objective function evaluations, while the epigraph variable is stored once every twenty updates. Consequently, it is not possible to associate each reported point with a specific optimization iteration. Nevertheless, this sampling strategy does not affect the evolution of the curves, which accurately represent the convergence behavior of the optimization process.

The evolution of the transmission coefficients T_1 and T_2 during the optimization process is examined. Initially, both coefficients exhibit noticeable fluctuations while the optimizer explores different device configurations. Starting from approximately $\beta = 32$, both T_1 and T_2 become considerably more stable, converging toward values between approximately 0.45 and 0.50. As the projection parameter further increases, the curves become almost flat, indicating that the optimization has reached a stable solution and that only minor refinements of the geometry are performed.

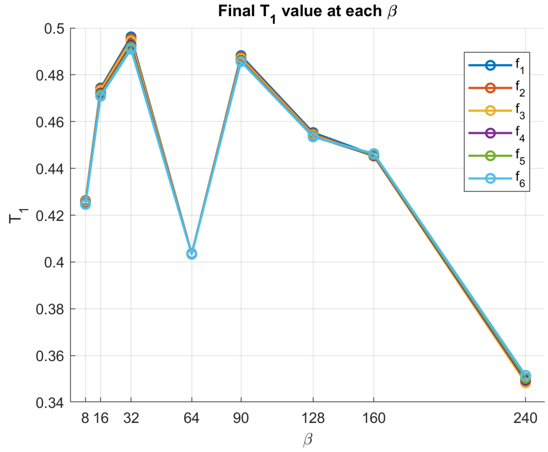
A particularly interesting result is that this behavior is observed for all six operating frequencies. The corresponding curves remain almost completely overlapped throughout the optimization, demonstrating that the optimized splitter exhibits a broadband response rather than being optimized for a single wavelength.

A slight change can be observed during the final optimization stage, when the fabrication constraints are introduced. At this point, both transmission coefficients start to oscillate slightly again, since the optimizer must satisfy the manufacturability constraints. As a consequence, the final value of T_2 remains close to 0.46/0.47, whereas T_1 decreases to approximately 0.35. This behavior represents the expected trade-off between maximizing the optical performance and obtaining a device that can be fabricated.

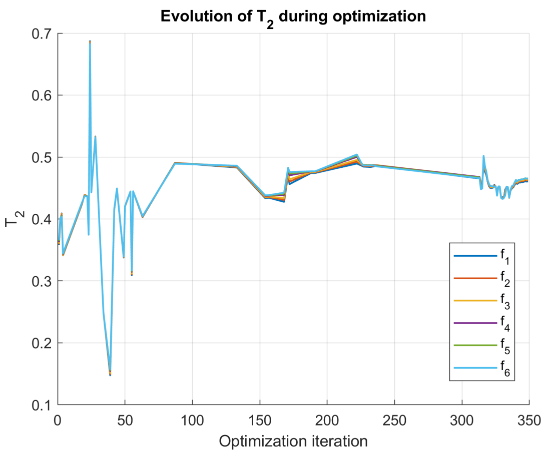
The behavior of the reflection coefficient R during the optimization process is examined. Similarly to the transmission coefficients, the reflection initially exhibits relatively large fluctuations while the optimizer explores the design space. As the optimization progresses and the projection parameter increases, the reflection rapidly decreases, reaching



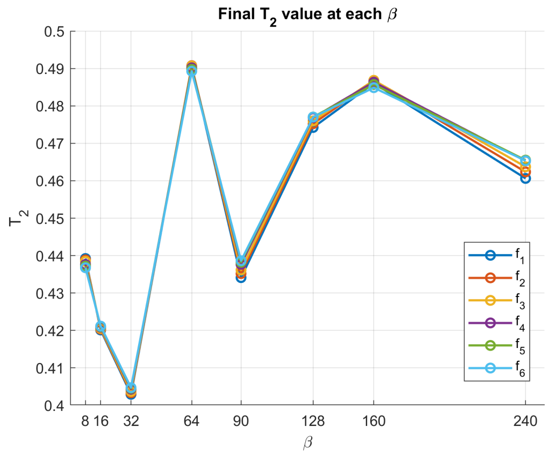
(a) T_1 during all optimization iterations



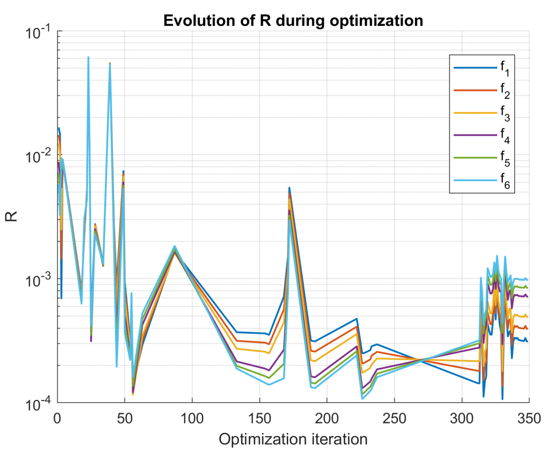
(b) Final T_1 value at each β



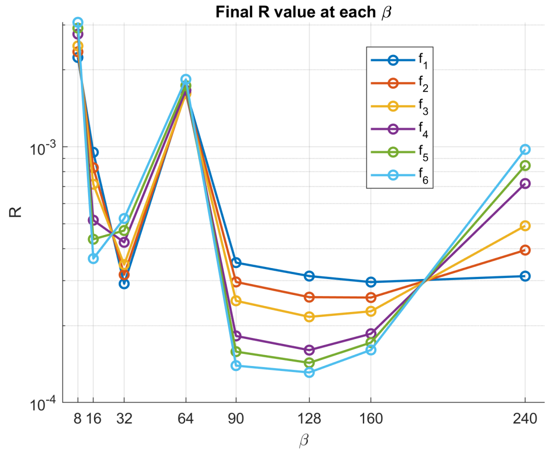
(c) T_2 during all optimization iterations



(d) Final T_2 value at each β



(e) R during all optimization iterations



(f) Final R value at each β

Figure 6.2: Evolution of the optical performance during the optimization process. The left column shows the evolution of the transmitted powers and reflectance throughout all optimization iterations, while the right column reports the final values obtained at the end of each optimization epoch corresponding to a specific value of the projection parameter β.

values on the order of $1 \times 10^{-4} - 1 \times 10^{-5}$. These very low values indicate that only a negligible fraction of the optical power is reflected back toward the input waveguides.

During the final optimization stage, when the fabrication constraints are activated, the reflection slightly increases, reaching values on the order of 1×10^{-3}. Although this corresponds to approximately a one order of magnitude increase, the reflected power remains sufficiently low for practical operation. It can also be observed that the final reflection values become more dependent on the operating wavelength, as the six curves no longer completely overlap. This suggests that the fabrication constraints affect the different wavelengths in a slightly different way while still maintaining an overall low reflected power.

Table 6.2 compares the transmission coefficients T_1 and T_2, together with the reflection coefficient R, immediately before the activation of the fabrication constraints and at the end of the optimization process:

Table 6.2: Transmission and reflection coefficients before and after the activation of the fabrication constraints for the six operating frequencies.

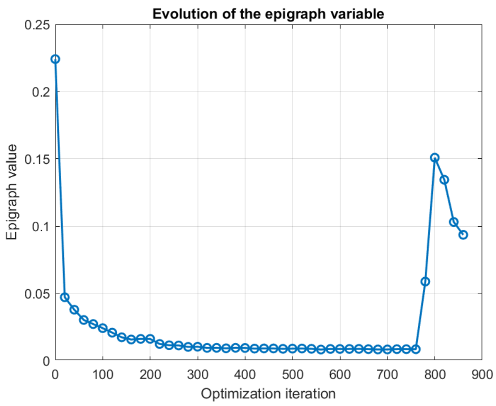
Freq	**Wavelength (nm)**	**Before constraints**			**After constraints**		
		T_1	T_2	R	T_1	T_2	R
f_1	1265	0.46538	0.47102	2.87×10^{-5}	0.34883	0.46062	3.12×10^{-3}
f_2	1270	0.46522	0.47111	1.70×10^{-5}	0.34835	0.46235	3.95×10^{-3}
f_3	1275	0.46530	0.47087	1.03×10^{-5}	0.34832	0.46372	4.91×10^{-3}
f_4	1285	0.46603	0.46948	5.69×10^{-6}	0.34937	0.46530	7.19×10^{-3}
f_5	1290	0.46662	0.46842	5.79×10^{-6}	0.35035	0.46550	8.45×10^{-3}
f_6	1295	0.46730	0.46719	6.62×10^{-6}	0.35154	0.46533	9.77×10^{-3}

Overall, the introduction of fabrication constraints leads to a moderate degradation of all performance metrics. However, these constraints do not significantly compromise the overall device performance: the reflection remains below 1 %, while the transmission exhibits only a limited reduction, with losses of a few percent in one output branch and up to approximately 15 % in the other. This behavior highlights the expected trade-off between ideal device performance and practical fabrication feasibility, while confirming that the optimized structure still maintains acceptable performance levels.

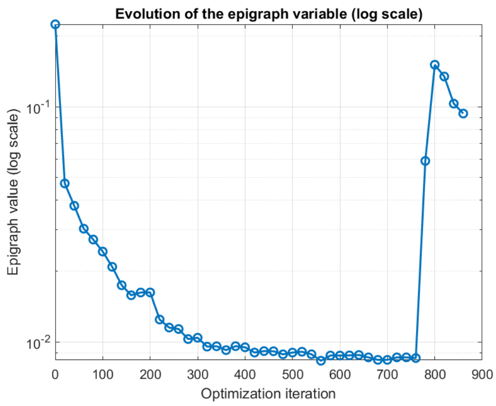
Epigraph and Fabrication Constraint Analysis

The evolution of the epigraph variable throughout the optimization process (Fig. 6.3) provides a direct indication of the convergence behavior of the *minimax* formulation. Since the optimization objective is the minimization of the epigraph variable t, lower values correspond to a reduction in the worst-case objective function among all considered wavelengths.

The optimization starts from an initial epigraph value of approximately 0.22. During the first iterations, the value rapidly decreases and then continues to progressively reduce, eventually reaching values on the order of 8×10^{-3}. After several optimization iterations, the epigraph becomes almost constant, confirming the convergence of the optimization



(a) Evolution of the epigraph variable during the optimization process.



(b) Evolution of the epigraph variable in logarithmic scale

Figure 6.3: Evolution of the epigraph variable throughout the optimization process. The left plot shows the epigraph value as a function of the optimization iteration, while the right plot show the same results in logarithmic scale

process and the stabilization of the solution, in agreement with the behavior previously observed for T_1, T_2, and R.

The logarithmic representation highlights this convergence more clearly, showing a gradual reduction over more than one order of magnitude before reaching a stable plateau. The final value obtained before the introduction of fabrication constraints is therefore very low, confirming the effectiveness of the optimization procedure.

A clear change in behavior is observed when the fabrication constraints are introduced during the last optimization stage. The epigraph value suddenly increases up to approximately 0.15, reflecting the additional difficulty of simultaneously satisfying both the optical objective and the manufacturability requirements. After this sudden increase, the optimizer progressively reduces the epigraph again, reaching a final value close to 0.10.

Although this value is considerably higher than that obtained before the fabrication constraints, it represents the expected degradation associated with applying a fabricable device geometry. This behavior is consistent with what has been observed in the transmission and reflection coefficients, where a moderate degradation of the optical performance is also evident after the introduction of fabrication constraints.

The evolution of the epigraph variable confirms the effectiveness of the adopted continuation strategy. The gradual increase of β allows the optimization to first explore the design space efficiently and subsequently converge toward a binary structure. The late introduction of the fabrication constraints further prevents premature restrictions of the design space, enabling the optimizer to first identify a high-performance optical solution and only then enforce the geometrical requirements needed for fabrication.

The effect of the fabrication constraints can be further analyzed through the evolution of the linewidth and spacing metrics M_1 and M_2 (Fig. 6.4). These quantities measure the violation of the minimum feature-size requirements imposed during the final optimization stage. For both metrics, values smaller than or equal to zero indicate that the corresponding fabrication constraint is satisfied, whereas positive values indicate a violation of the required minimum linewidth or spacing.

Ideally, both quantities should be equal to zero or assume negative values, indicating that the fabrication constraints are fully satisfied. In general, the smaller their values, the closer the optimized structure is to meeting the desired manufacturability requirements.

As shown in the figure, both M_1 and M_2 already start from relatively low values, on the order of 10^{-4}. During the optimization they rapidly decrease, reaching minimum values of approximately 4×10^{-5}, after which both curves become almost constant, indicating the convergence of the optimization process. The nearly identical evolution of the two curves also demonstrates that the fabrication constraints are enforced in a balanced manner, without favoring one metric over the other.

Although the constraints are not completely satisfied, since both values remain slightly positive, the final values are very close to zero. This indicates that the optimized design approaches the desired fabrication requirements while preserving the optical performance obtained during the unconstrained optimization. As observed for the transmission coefficients, the reflection coefficient, and the epigraph variable, the optimizer successfully achieves a compromise between device performance and manufacturability, producing a

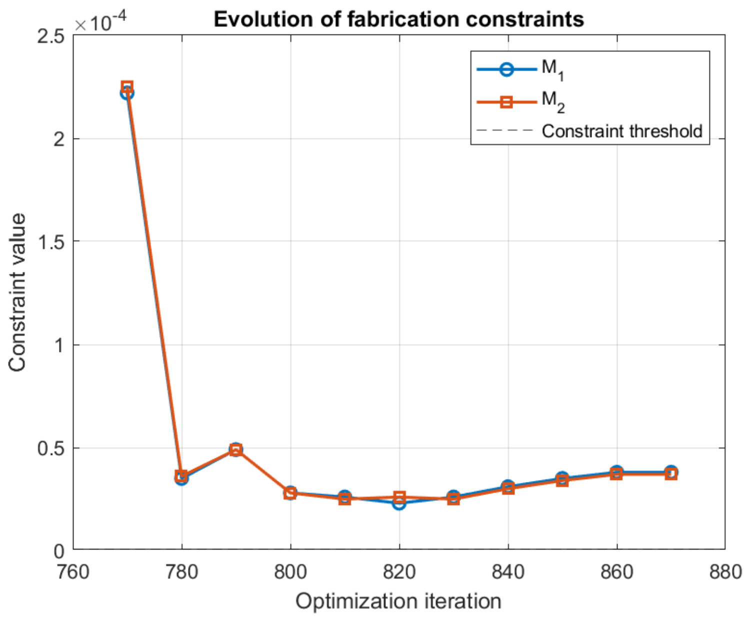
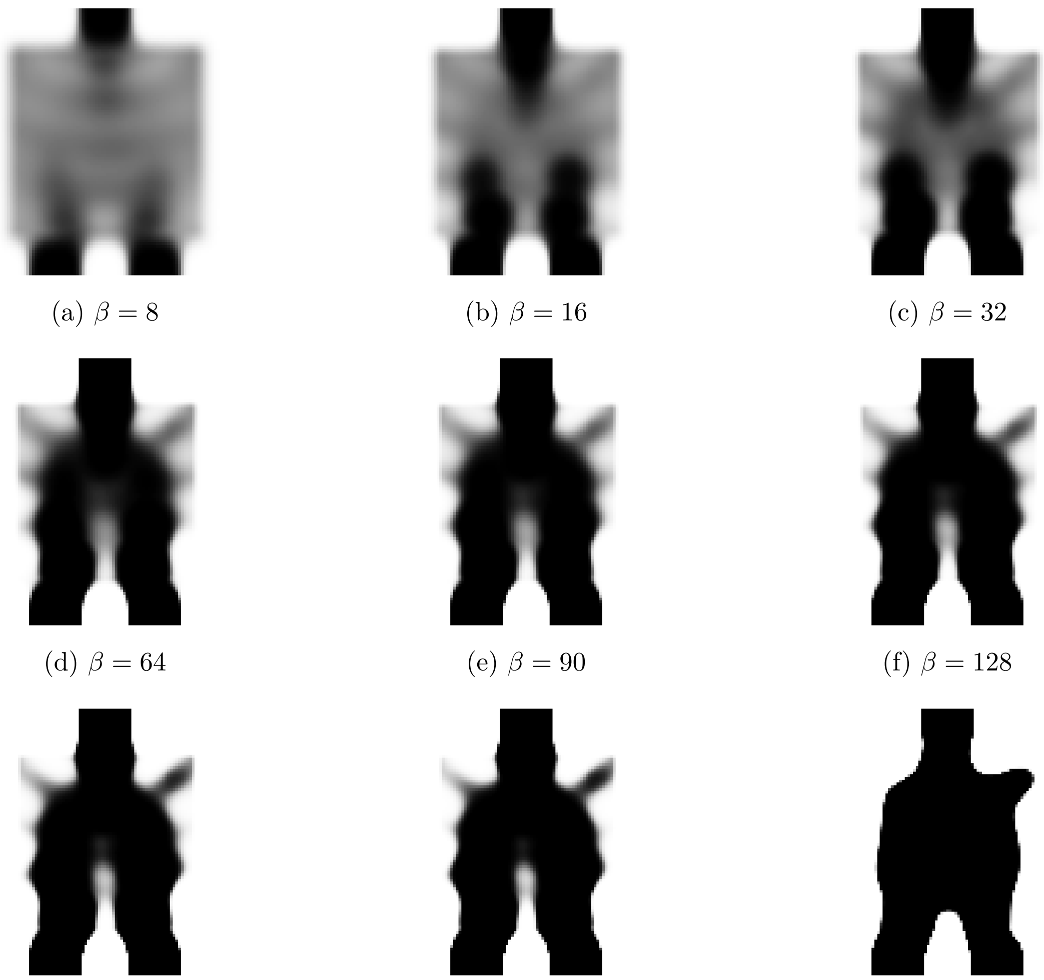


Figure 6.4: Evolution of the fabrication constraint metrics M_1 and M_2 during the final optimization stage with $\beta = 240$.

geometry that is both optically efficient and compatible with practical fabrication processes.

Design Evolution with Increasing β



(a) $\beta = 8$ (b) $\beta = 16$ (c) $\beta = 32$

(d) $\beta = 64$ (e) $\beta = 90$ (f) $\beta = 128$

(g) $\beta = 160$ (h) $\beta = 208$ (i) $\beta = 240$

Figure 6.5: Evolution of the optimized splitter geometry for increasing values of the projection parameter β. After the activation of the fabrication constraints, the geometry is significantly modified, leading to the final binary and manufacturable design obtained at $\beta = 240$.

The evolution of the design region for increasing values of the projection parameter β is illustrated in Fig. 6.5. The sequence clearly shows the effect of the continuation strategy on the material distribution and highlights the progressive transition from a continuous design representation toward a binary and manufacturable structure.

For low values of β, the projection function is relatively smooth and allows a large number of intermediate material values within the design region. As a result, the optimized structures appear blurred, with extensive gray areas corresponding to pixels whose material properties lie between those of silicon and air. In this phase, the optimizer benefits from a highly flexible design space and is therefore able to rapidly improve the optical performance of the splitter.

As the value of β increases, the two output arms of the splitter gradually become more defined. At the same time, the gray regions progressively disappear, indicating that the material distribution is moving toward a binary configuration. This behavior suggests that the optimization process is evolving as expected, progressively transforming the initially smooth density distribution into a physically meaningful device geometry while preserving the optical performance achieved during the early optimization stages.

The design obtained at $\beta = 208$ represents the best unconstrained optical solution produced during the optimization process. At this stage, the objective functions, the epigraph variable, and the transmission and reflection metrics T_1, T_2, and R reach their most favorable values, indicating excellent splitter performance. However, this design is not yet directly manufacturable, since a few gray regions corresponding to intermediate material values are still visible.

For this reason, the minimum linewidth and spacing constraints are activated in the final optimization stage. The introduction of these constraints forces the optimizer to eliminate sub-resolution features and to enlarge narrow regions that would otherwise violate the fabrication specifications. As a consequence, the geometry undergoes a moderate modification between the unconstrained design and the final optimized structure.

The final design obtained at $\beta = 240$ exhibits an almost fully binary material distribution and satisfies the prescribed fabrication constraints. The resulting geometry clearly resembles a 1×2 splitter, with two well-defined output branches and only a slight asymmetry between the two sides. This asymmetry may also explain the small imbalance observed in the final values of T_1 and T_2 after the fabrication constraints are introduced, as discussed in the previous section.

Overall, the final result can be considered satisfactory. Besides meeting the fabrication requirements, the optimization successfully produces a geometry that is visually recognizable as a photonic splitter while maintaining good optical performance. This confirms that the adjoint optimization procedure is capable of finding practical and physically meaningful solutions even after the introduction of realistic manufacturing constraints.

Possible Improvements

Further improvements could likely be achieved through a more careful tuning of the optimization hyperparameters, including the dilation and erosion thresholds, as well as by increasing the number of optimization iterations performed after the activation of the fabrication constraints. Such adjustments may allow the optimizer to better recover optical performance while still satisfying the manufacturability requirements.

In particular, extending the constrained optimization stage could provide the optimizer with additional time to adapt the geometry to the fabrication requirements while recovering part of the optical performance lost when the constraints are first introduced, potentially leading to a better final solution.

In addition, a different formulation of the second objective function could potentially improve the balance between the two output ports. The final device exhibits a small asymmetry both in its optical response and in its geometry.

6.4 Single-Wavelength Optimization

After analyzing the broadband splitter optimized over six design frequencies, the same optimization procedure was repeated, considering a single operating wavelength. The purpose of this additional study was to investigate how the optimization behaves when the device is required to satisfy the design objectives at only one wavelength, rather than over an extended wavelength range.

Below, a summary table reports all the optimization specifications adopted for the single-wavelength design 6.3. Since the optimization framework was intentionally kept identical to that used in the six-wavelength case, most of the parameters remain unchanged. The only differences with respect to the broadband optimization are highlighted below the table, allowing the reader to immediately identify the modifications introduced for the single-wavelength experiments.

Table 6.3: Main parameters of the splitter optimization problem.

Parameter	**Value**
Simulation resolution	30 pixels/µm
Design region dimensions	1.6 µm× 1.6 µm
Center-to-center separation of output waveguides	0.6 µm
PML thickness	1.0 µm
Padding above and below the design region	0.6 µm

The same optimization procedure was repeated, considering a single operating wavelength. The purpose of this additional study was to investigate how the optimization behaves when the device is required to satisfy the design objectives at only one wavelength, rather than over an extended wavelength range.

To ensure a fair comparison, the overall optimization framework was kept unchanged. The design region, material platform, objective functions, continuation strategy, fabrication constraints, and optimization algorithm remained the same as those used in the broadband case. The only modifications concerned the wavelength range considered during the simulations, the width of the Gaussian source, which was set to $0.05 \cdot f_c$, and the continuation schedule, which was adapted by modifying the values of β and the number of iterations performed in each optimization epoch:

$$\beta = [8, 16, 32, 64, 128, 256]$$

$$max\_evals = [40, 50, 60, 80, 100, 100]$$

The evolution of the projection parameter β is noticeably faster than in the six-wavelength case, since the optimization problem is less complex and the device only needs to satisfy the design objectives at a single wavelength. As a result, the optimizer reaches a binary design more rapidly, and the convergence process requires fewer iterations within each continuation stage.

Table 6.4: Difference between parameters adopted for the single-wavelength optimization runs.

Parameter	**1300 nm**	**1550 nm**
Optimization wavelength	1.30 µm	1.55 µm
Waveguide dimensions	3.0 µm×0.32 µm	3.0 µm×0.45 µm
Waveguide and design region refractive index	2.96	2.86
Background refractive index	1.0	1.0

As can be seen in Table 6.4, the waveguide dimensions and the refractive index values were kept unchanged for the 1300 nm optimization since this wavelength lies within the operating range considered in the broadband design. Conversely, for the 1550 nm case, both the waveguide height and the refractive index were adjusted to account for the different operating wavelength and to ensure a physically consistent device model.

Two independent optimization runs were performed, targeting wavelengths of 1300 nm and 1550 nm, respectively. By reducing the number of frequencies considered during the optimization, the electromagnetic problem becomes less demanding, since the device no longer needs to satisfy the performance requirements simultaneously across a broad spectral range. As a result, it is expected that the optimizer may achieve improved performance at the target wavelength, potentially leading to higher transmission and lower reflection than those obtained in the broadband design.

6.5 Optimization Results

The optical performance of the splitter optimized at 1300 nm and 1550 nm is analyzed following the same methodology adopted for the broadband design. This allows for a direct comparison between the single-wavelength and six-wavelength optimization approaches, highlighting the impact of the reduced design bandwidth on both the optimization process and the final device performance.

As before, the following aspects for the two cases are investigated:

- the evolution of the transmitted powers T_1 and T_2 and of the reflectance R as a function of the projection parameter β, evaluated over the six design frequencies;
- the evolution of the epigraph variable during the optimization process, and the evolution of the fabrication constraint metrics M_1 and M_2 during the final optimization stages;
- the progressive evolution of the design region for increasing values of β, highlighting the transition toward a binary structure.

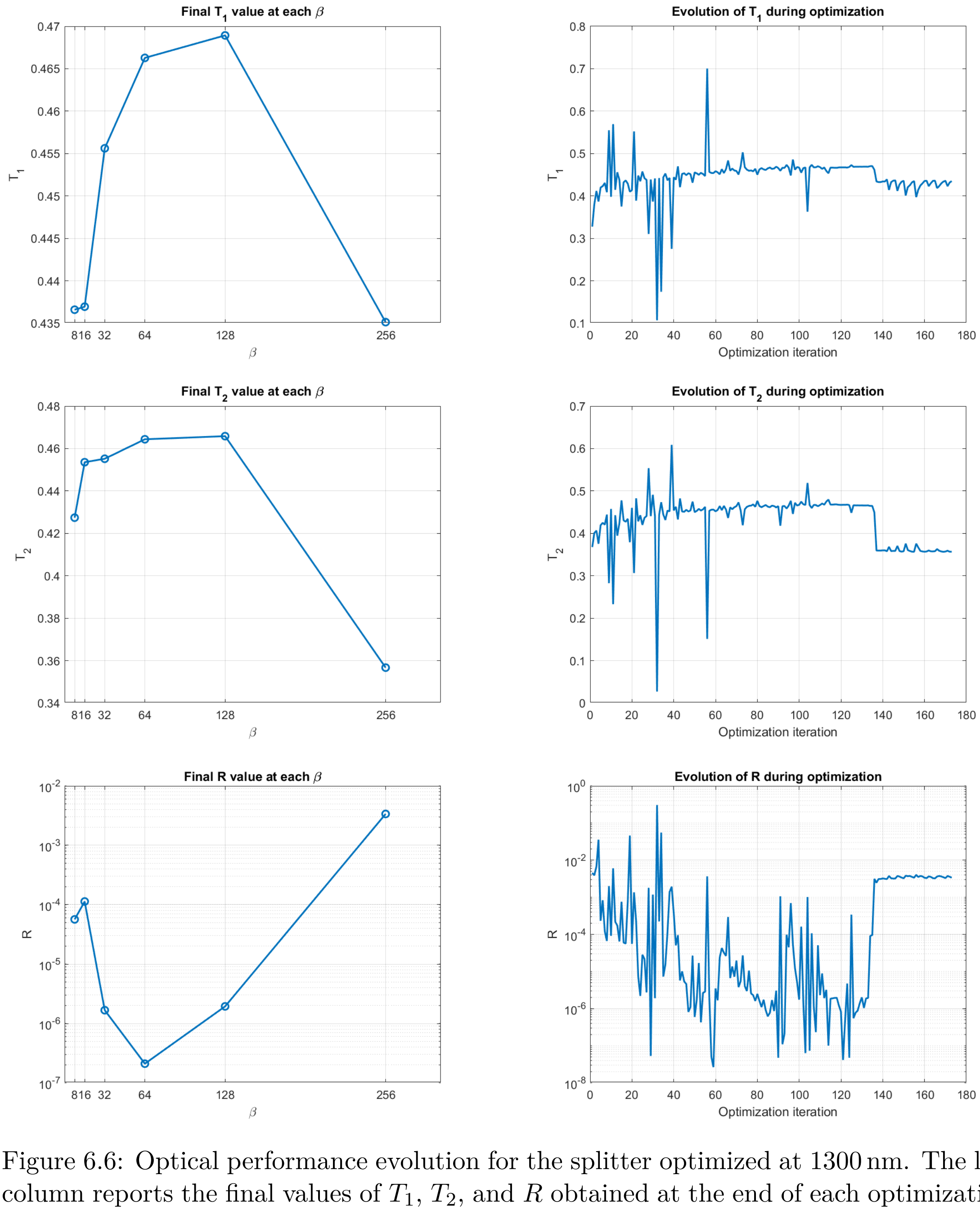


Figure 6.6: Optical performance evolution for the splitter optimized at 1300 nm. The left column reports the final values of T_1, T_2, and R obtained at the end of each optimization epoch associated with a specific value of β, while the right column shows the evolution of the same quantities throughout the optimization process.

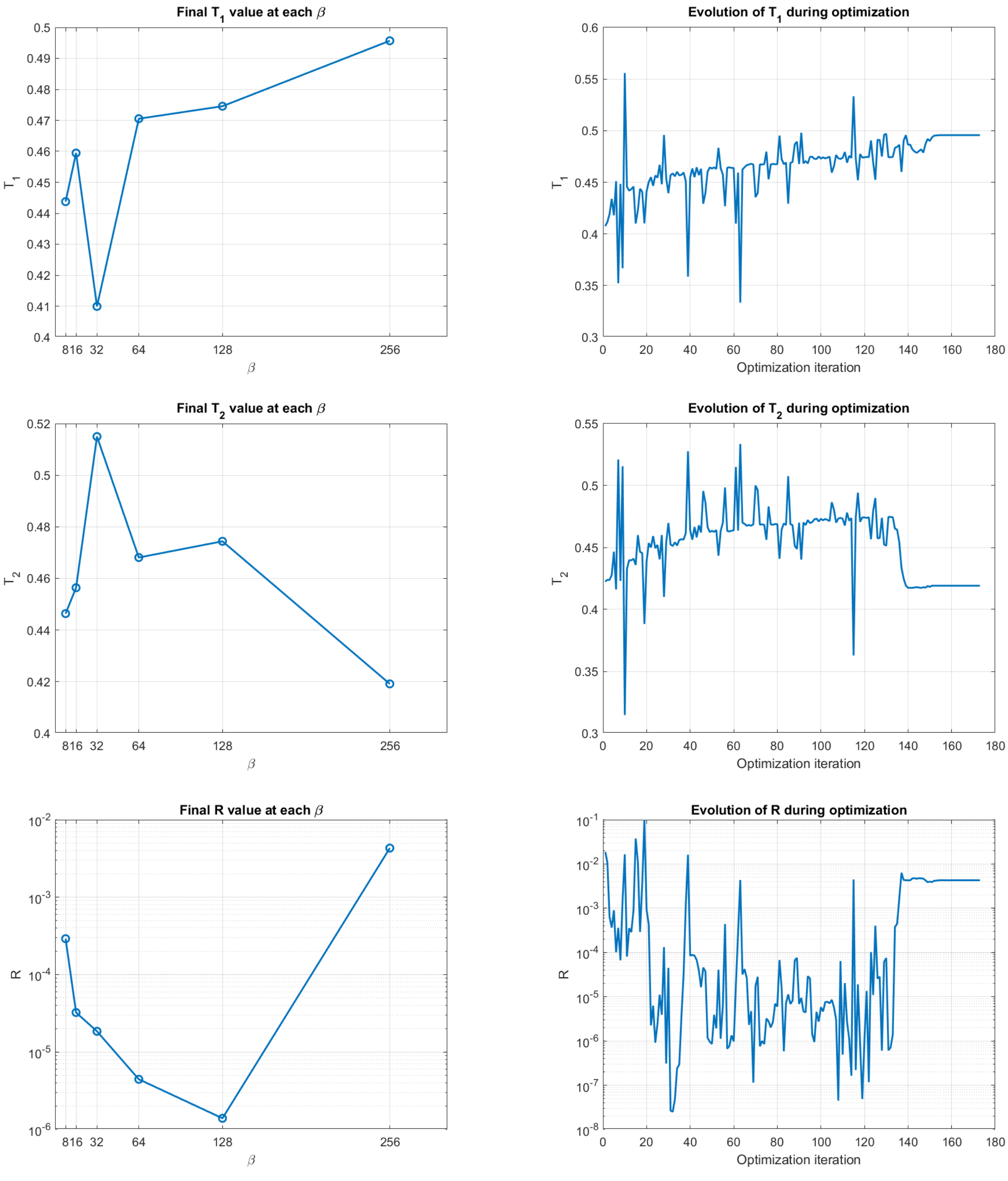


Figure 6.7: Optical performance evolution for the splitter optimized at 1550 nm. The left column reports the final values of T_1, T_2, and R obtained at the end of each optimization epoch associated with a specific value of β, while the right column shows the evolution of the same quantities throughout the optimization process.

Optical Performance Evolution

The optical performance of the single-wavelength optimizations is evaluated by monitoring the evolution of the transmitted powers at the two output ports, denoted as T_1 and T_2, together with the reflectance R. Figures 6.6 and 6.7 summarize the results obtained for the two target wavelengths considered in this study.

The left column reports the final values of T_1, T_2, and R obtained at the end of each optimization epoch associated with a specific value of β, while the right column shows the evolution of the same quantities throughout the optimization process.

For the single-wavelength optimization at 1300 nm case, it can be observed that both transmission coefficients rapidly increase during the first continuation stages. Already at $beta = 16$, both T_1 and T_2 reach values close to 0.46, while the reflection coefficient decreases to the order of 1×10^{-5}. This behavior indicates that the single-wavelength optimization problem is considerably less demanding than the broadband one, allowing the optimizer to identify an effective device topology after only a few continuation stages.

The subsequent continuation stages produce only minor variations in the transmission coefficients, suggesting that the optimal topology has already been identified at a very early stage of the optimization process. This behavior differs significantly from the broadband optimization, where more continuation stages were required before reaching comparable transmission levels. The reflection coefficient, however, follows a different trend: although it reaches very low values already in the early stages of the optimization, it continues to exhibit noticeable oscillations throughout the unconstrained optimization, remaining approximately between 1×10^{-5} and 1×10^{-6}, which is still a good result.

As expected, the activation of the fabrication constraints leads to a degradation of the optical performance. In this case, T_1 remains relatively stable, settling around 0.43, whereas T_2 experiences a more pronounced reduction, reaching approximately 0.36. Similar to the broadband optimization, the reflection coefficient increases to the order of 10^{-3}, reflecting the reduced design freedom imposed by the manufacturability constraints.

The evolution of the transmission coefficients T_1 and T_2, together with the reflection coefficient R, for the single-wavelength optimization at 1550 nm is analyzed throughout the optimization process.

Similarly to the optimization performed at 1300 nm, both transmission coefficients rapidly converge during the first continuation stages. Already at $\beta = 16$, T_1 and T_2 reach values close to 0.46/0.47.

As the continuation proceeds, only small improvements are observed in the transmission coefficients, which gradually stabilize around their final unconstrained values. The reflection coefficient exhibits a less regular behavior and continues to oscillate throughout the unconstrained optimization, remaining within the same 1×10^{-5} to 1×10^{-6} range observed in the 1300 nm case.

The activation of the fabrication constraints again produces the expected degradation of the optical response. As in the previous case, the reflection coefficient increases to the order of 1×10^{-3}, due to the reduced design flexibility imposed by the manufacturability requirements. However, the redistribution of the transmitted power differs from the optimization at 1300 nm. In this case, T_1 increases and reaches a value close to 0.50, while T_2 decreases to approximately 0.42. This indicates that the fabrication constraints introduce a slight imbalance between the two output ports, favoring transmission toward the first output waveguide.

Table 6.5 compares the transmission coefficients T_1 and T_2, along with the reflection coefficient R, before and after the activation of the fabrication constraints for the two single-wavelength optimizations

Table 6.5: Comparison of the transmission and reflection coefficients before and after the activation of the fabrication constraints for the two single-wavelength optimizations.

Wavelength	**Before constraints**			**After constraints**		
	T_1	T_2	R	T_1	T_2	R
1300 nm	0.46893	0.46583	1.94×10^{-6}	0.43513	0.35669	3.37×10^{-3}
1550 nm	0.47456	0.47439	1.39×10^{-6}	0.49569	0.41914	4.31×10^{-3}

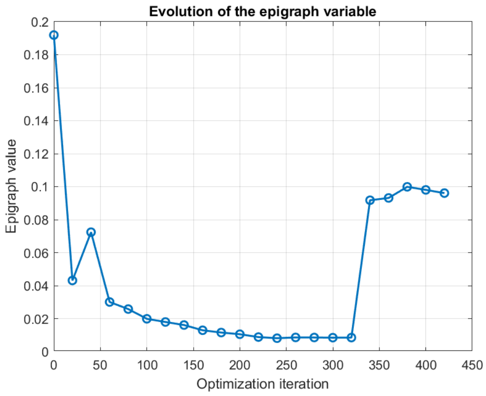
Overall, the results demonstrate that the single-wavelength optimization is able to identify an effective splitter topology after only a few continuation stages, making the optimization process significantly faster and less computationally demanding than the broadband case.

As expected, the activation of the fabrication constraints introduces a trade-off between optical performance and manufacturability. The transmission coefficients become slightly unbalanced, with the direction of the imbalance depending on the operating wavelength, while the reflection coefficient increases to the order of 10^{-3}. Nevertheless, the final constrained devices still exhibit transmission and reflection values comparable to those obtained in the broadband optimization, while requiring considerably fewer optimization iterations. These results confirm that single-wavelength optimization represents a computationally efficient approach capable of producing high-performance and manufacturable power splitters.

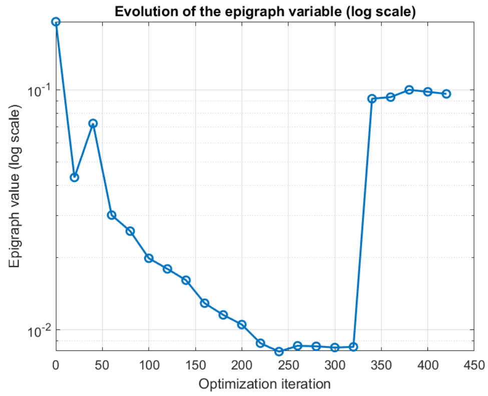
Epigraph and Fabrication Constraint Analysis

The evolution of the epigraph variable for the 1300 nm and 1550 nm optimizations is reported in Fig. 6.8 and closely follows the behavior already observed in the broadband case. In both optimizations, the epigraph starts from relatively high values, between approximately 0.1 and 0.2, and decreases rapidly during the first continuation stages. Already at $\beta = 16$, the epigraph reaches values of the order of 1×10^{-2}, consistent with the rapid improvement observed in the transmission and reflection coefficients. As the continuation progresses, the epigraph further decreases until it reaches values close to 5×10^{-3}, which are comparable to those achieved in the broadband optimization. However, these values are reached after significantly fewer optimization iterations, confirming once again that the single-wavelength problem is substantially easier to solve.

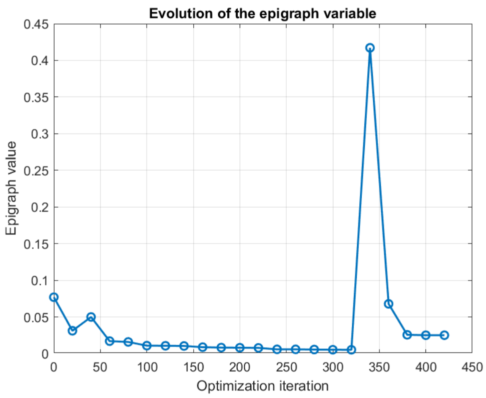
The two optimizations begin to exhibit different behaviors after the fabrication constraints are introduced. In the 1300 nm, the epigraph suddenly increases to approximately 0.1 and remains close to this value for the remainder of the optimization, indicating that the optimizer is unable to recover the optical performance lost after enforcing the manufacturability constraints. This behavior is remarkably similar to that observed in the



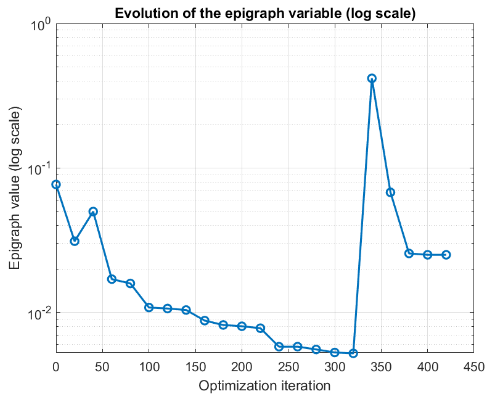
(a) 1300 nm: epigraph during optimization



(b) 1300 nm: epigraph during optimization in logarithmic scale



(c) 1550 nm: epigraph during optimization

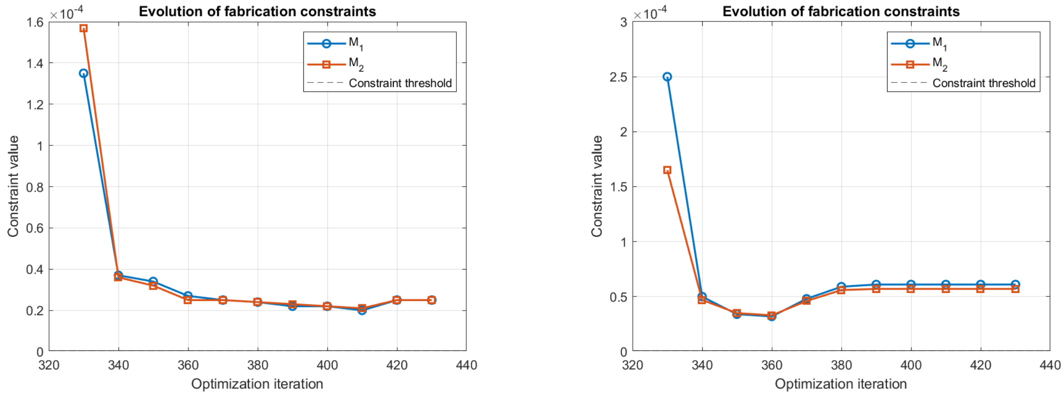


(d) 1550 nm: epigraph during optimization in logarithmic scale

Figure 6.8: Evolution of the epigraph variable for the single-wavelength optimizations at 1300 nm and 1550 nm. The left column shows the evolution during the optimization process, while the right column reports the same results in logarithmic scale.

broadband optimization and is consistent with the comparable transmission and reflection values obtained for the final constrained design.

A different trend is observed for the optimization at 1550 nm. After the activation of the fabrication constraints, the epigraph initially increases much more significantly, reaching a value close to 0.5. Unlike the previous case, however, it subsequently decreases and stabilizes at values of the order of 1×10^{-2}. This recovery is reflected in the final optical performance: although the constrained device still exhibits some degradation, the total power losses at the output ports are limited to approximately 8 %, whereas they exceed 15 % in the 1300 nm optimization. These results indicate that the optimization at 1550 nm is able to achieve a significantly better trade-off between optical performance and manufacturability.



(a) 1300 nm optimization

(b) 1550 nm optimization

Figure 6.9: Evolution of the fabrication constraint metrics M_1 and M_2 during the final optimization stage. The left panel corresponds to the optimization performed at 1300 nm, while the right panel reports the results obtained at 1550 nm.

The fabrication constraint metrics are reported in Fig. 6.9. The evolution of the fabrication constraints M_1 and M_2 is analyzed.

For the optimization performed at 1300 nm, both M_1 and M_2 decrease to values of approximately 2.2×10^{-5}–2.5×10^{-5} and remain almost constant until the end of the optimization. These values are even lower than those obtained in the broadband optimization, where the constraints converged to approximately 4×10^{-5}. This indicates that the final design satisfies the fabrication constraints more accurately than the broadband device while maintaining comparable optical performance.

A different behavior is observed for the optimization at 1550 nm. In this case, M_1 and M_2 converge to values of approximately 5.7×10^{-5}–6.0×10^{-5}, which are slightly higher than those obtained in the broadband case. Nevertheless, the final optical response is significantly better than that achieved at 1300 nm, with transmission coefficients of approximately T_1=0.496 and T_2=0.419. This suggests that the optimizer converges to a solution that satisfies the fabrication constraints slightly less strictly while preserving a considerably larger fraction of the optical performance.

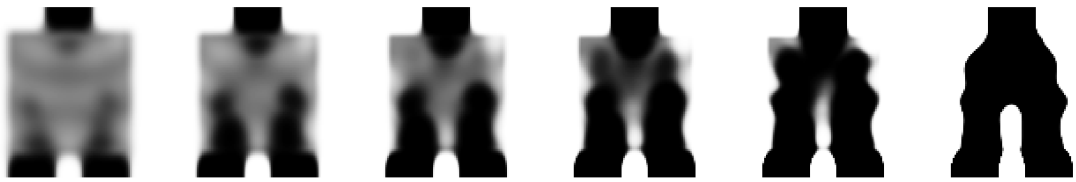
The final design results from a trade-off between optical performance and manufacturability. In particular, the optimization at 1300 nm achieves the best satisfaction of the fabrication constraints but exhibits worse optical performance compared to the 1550 nm case, which instead satisfies the constraints less strictly while preserving a better optical response. Moreover, the single-wavelength optimization at 1300 nm provides optical performance comparable to the broadband design while achieving a more accurate fulfillment of the fabrication constraints.

Design Evolution with Increasing β

1300 nm

(a) $\beta = 8$ (b) $\beta = 16$ (c) $\beta = 32$ (d) $\beta = 64$ (e) $\beta = 128$ (f) $\beta = 256$

1550 nm



(g) $\beta = 8$ (h) $\beta = 16$ (i) $\beta = 32$ (j) $\beta = 64$ (k) $\beta = 128$ (l) $\beta = 256$

Figure 6.10: Evolution of the optimized splitter geometry for the single-wavelength optimizations at 1300 nm (top row) and 1550 nm (bottom row). Increasing the projection parameter β progressively binarizes the material distribution, while the final column shows the geometry obtained after the activation of the fabrication constraints.

The evolution of the design region during the optimization process is shown in Fig. 6.10. As in the broadband optimization, increasing the projection parameter β progressively sharpens the material distribution and drives the design toward a binary structure.

At low values of β, both designs exhibit large gray regions corresponding to intermediate density values. These regions provide the optimizer with a high degree of flexibility, allowing rapid improvements in optical performance. As β increases, the interfaces between silicon and air become progressively sharper and the geometry gradually converges toward a well-defined topology. In both cases, the desired optical performance is already achieved at relatively low values of β, meaning that the subsequent continuation stages mainly contribute to the binarization of the design rather than to further performance improvements.

The evolution of the optimized geometries follows almost the same behavior already observed in the broadband optimization. Even for relatively small values of the projection parameter β, the two branches of the splitter begin to emerge clearly in both single-wavelength optimizations, indicating that the optimizer rapidly identifies the overall device topology.

The introduction of the fabrication constraints produces the expected final binarization of the structure. In both cases, the final design clearly exhibits the characteristic topology of a 1×2 splitter, demonstrating that the optimization successfully converges to a physically meaningful device. Small geometric asymmetries can be observed in the final layouts, particularly in the shape of the two output branches. These asymmetries may explain the slight imbalance in the transmitted power observed after the activation of the fabrication constraints, where the transmission coefficients T_1 and T_2 no longer exhibit the nearly ideal 50:50 splitting achieved before the constrained optimization stage.

Possible Improvements

The possible improvements for the single-wavelength optimizations are essentially the same as those discussed for the broadband case. A more careful tuning of the optimization hyperparameters, such as the dilation and erosion thresholds, together with a larger number of optimization iterations after the activation of the fabrication constraints, could allow the optimizer to recover part of the optical performance while still satisfying the manufacturability requirements.

In addition, the small asymmetries observed in the final geometries suggest that a slower continuation strategy, with more intermediate values of the projection parameter β, might promote a more gradual binarization process and potentially lead to more symmetric splitter designs. Nevertheless, this improvement cannot be guaranteed because the optimization problem is very complex, and changing the continuation strategy does not always lead to better results.

Chapter 7

Conclusions

The growing complexity of photonic integrated circuits has made inverse design one of the most promising approaches for the development of compact and high-performance optical devices. By replacing the traditional trial-and-error design process with numerical optimization techniques, inverse design enables the automatic synthesis of device geometries capable of satisfying specific optical objectives while exploring design spaces that would be impractical to investigate manually.

In this thesis, several inverse-design methodologies available in the MEEP simulation environment have been investigated and applied to the design of integrated photonic devices. Starting from the optimization of a conventional MMI splitter through parametric sweeps and `SciPy` optimization, increasingly sophisticated techniques were explored, including pixel-based optimization using Genetic Algorithms and gradient-based topology optimization using the MEEP Adjoint Solver, demonstrating the flexibility of inverse design for different photonic applications.

7.1 Discussion about the Obtained Results

The results obtained throughout this work highlight how the choice of the optimization strategy strongly depends on the complexity of the design problem and on the number of design variables involved. No single method can be considered universally optimal; instead, each approach presents specific advantages and limitations that make it more suitable for a particular class of photonic design problems.

The parametric optimization based on heatmaps and the MEEP `SciPy` library proved to be an effective solution when only a few geometrical parameters had to be optimized. The method is intuitive and provides a clear understanding of the relationship between the design parameters and the optical performance. However, its applicability rapidly decreases as the number of optimization variables increases, since the computational time grows significantly, making it difficult to explore all possible parameter combinations for more complex devices.

Introducing a pixel-based representation of the design region significantly increased the available design space. In this context, the Genetic Algorithm successfully optimized both the MMI splitter and the optical AND logic gate without requiring any gradient information. Although the computational cost is considerably high and the convergence is slow, the algorithm demonstrated its capability to explore complex design spaces and find better device configurations. It also showed that the same method can be used for different optical devices by simply changing the fitness function.

Of all the methods considered, the adjoint method emerged as the most powerful optimization strategy. By exploiting gradient information, it efficiently handled thousands of design variables while maintaining a computational cost that is almost independent of the number of optimization parameters; however, it is more complex to implement and requires a better understanding of the optimization process. It also allows fabrication constraints to be included directly in the optimization process, which is very useful for making the designed devices easier to manufacture. Although the introduction of these constraints slightly degraded the optical performance, the final structures remained compact, nearly symmetric, and characterized by low reflection and balanced power splitting, demonstrating an effective compromise between performance and manufacturability.

Overall, this work demonstrates that the different inverse-design methodologies available in MEEP are not competing alternatives but complementary tools. Together, they provide a flexible optimization framework capable of addressing photonic design problems of increasing complexity, ranging from the optimization of a few geometrical parameters to large-scale topology optimization of integrated photonic devices.

7.2 Future Work

Although the proposed optimization methodologies proved effective for the investigated photonic devices, several aspects could be further explored to improve both the optimization process and the obtained results.

For the Genetic Algorithm, the most immediate improvement would consist of increasing the optimization time by considering a larger number of generations and larger population sizes. Since all optimization runs showed a continuous improvement of the fitness function without reaching a clear convergence, longer optimization processes would likely lead to better device performances. The adoption of different evolutionary operators or alternative optimization algorithms could also be investigated and compared with the standard Genetic Algorithm implemented in this work.

In particular, for the pixel-based splitter optimized with the Genetic Algorithm, a possible improvement would be to deliberately reduce the dimensions of the MMI region in order to achieve a more compact device, even at the cost of a decrease in transmission efficiency. The pixel-based design could then be used to compensate for this loss and improve the transmission again. This approach could therefore enable the optimization process to converge towards a smaller device while still preserving acceptable transmission characteristics.

Regarding the adjoint optimization, in this work the definition of the constraint values and the implementation of the continuation strategy could have been handled more carefully, as a more refined tuning of these aspects would likely improve both the convergence behavior and the final device performance.

More generally, beyond the specific work of this thesis, the adjoint optimization could be extended to the design of more complex photonic devices, such as wavelength multiplexers, mode converters, or polarization management components. Finally, the adoption of full three-dimensional simulations would enable a more realistic representation of practical devices.

Finally, an interesting future direction is the integration of artificial intelligence techniques with inverse design algorithms. Machine-learning models could be employed to generate accurate initial geometries, accelerate the optimization process through surrogate models, or predict the performance of candidate designs, significantly reducing the computational cost associated with repeated electromagnetic simulations.

Overall, the methodologies investigated in this thesis represent only a small portion of the possibilities offered by inverse design. The continuous progress in optimization algorithms, computational resources, and artificial intelligence will further improve this approach, making it possible to design integrated photonic devices that are increasingly compact, efficient, and easier to manufacture.

7.3 Final Remarks

This thesis demonstrates that the different inverse-design methodologies available in the MEEP simulation environment should not be regarded as competing alternatives, but rather as complementary tools. The study shows that there is no universally optimal approach: the most appropriate methodology depends on the design objectives, the number of optimization variables, and the desired balance between computational efficiency and achievable performance.

This work shows how important computational optimization is becoming in integrated photonics. As photonic devices get smaller and more complex, traditional design methods are often not enough to explore all the possible geometries. Inverse design helps overcome this problem by starting from the desired optical behavior and automatically finding suitable structures, making it possible to discover solutions that would be very difficult to obtain using only intuition.

Overall, the methodologies investigated in this thesis show that inverse design is now a key tool for developing modern photonic integrated circuits. As optimization techniques and computational resources continue to improve, it will become possible to design devices that are more compact, efficient, reliable, and easier to manufacture, opening the way to many future applications.

Bibliography

[1] Ehsan Adibnia, Mohammad Ali Mansouri-Birjandi, Majid Ghadrdan, and Pouria Jafari. A deep learning method for empirical spectral prediction and inverse design of all-optical nonlinear plasmonic ring resonator switches. *Scientific Reports*, 14(1SN - 2045-2322):5787, Mar 2024.

[2] Dominic Gallagher. Leos - photonic CAD matures. *IEE LEOS News*, 22(1), 2008.

[3] MEEP Developers. Meep python tutorial: Adjoint solver. https://meep.readthedocs.io/en/latest/Python_Tutorials/Adjoint_Solver/, 2025. Accessed: May 2026.

[4] Ardavan F. Oskooi, David Roundy, Mihai Ibanescu, Peter Bermel, J.D. Joannopoulos, and Steven G. Johnson. Meep: A flexible free-software package for electromagnetic simulations by the fdtd method. *Computer Physics Communications*, 181(3):687–702, 2010.

[5] Alexander Y. Piggott, Jan Petykiewicz, Logan Su, and Jelena Vučković. Fabrication-constrained nanophotonic inverse design. *Scientific Reports*, 7(1):1786, May 2017.

[6] Mohammad H. Tahersima, Keisuke Kojima, Toshiaki Koike-Akino, Devesh Jha, Bingnan Wang, Chungwei Lin, and Kieran Parsons. Deep neural network inverse design of integrated photonic power splitters. *Scientific Reports*, 9(1):1368, Feb 2019.

[7] Zhibin Wang, Zhengyang Li, Xuwei Hou, and Jiutian Zhang. Reverse design of multifunctional demultiplexing devices. *Photonics and Nanostructures - Fundamentals and Applications*, 58:101246, 2024.

[8] Rui Wu, Xingqi Wang, Xin Qiao, Yong Wei, and Yingjie Liu. Compact and scalable mode (de)multiplexer using inverse-designed subwavelength gratings. *Opt. Express*, 33(5):11541–11554, Mar 2025.