

UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

CARRERA DE CIENCIA DE LA COMPUTACIÓN



**Evaluación de Técnicas Modernas de Optimización en
Dos Dispositivos Nanofotónicos: *bend-90°* y *2-splitter***

TESIS

Para optar el título profesional de Licenciado en Ciencia de la
Computación

AUTOR:

José Leonidas García Gonzales

ASESOR

Jorge Luis Gonzalez Reaño

Lima - Perú

11 de octubre de 2021

Índice general

	Pág.
CAPÍTULO 1 Motivación y Contexto	1
1.1 Introducción	1
1.2 Descripción del Problema	2
1.3 Justificación	3
1.4 Objetivos	3
1.5 Aportes	4
CAPÍTULO 2 Trabajos relacionados	5
2.1 Parametrización	5
2.2 Definición de la FOM	7
2.3 Optimización	7
2.4 Restricciones de fabricación	8

Capítulo 1

Motivación y Contexto

1.1 Introducción

La fotónica está atrayendo el interés de la industria debido a su potencial en términos de escalabilidad y los beneficios de costo-eficiencia. Este potencial es evidente, por ejemplo, con los siguientes tres puntos. Primero, si se quiere mantener la tendencia que cada 10 años se mejore en un factor de 1000 el *performance* de los sistemas electrónicos, entonces parece ser indispensable la convergencia de estos con sistemas fotónicos ([Glick et al., 2018](#)). Segundo, existe una inversión billonaria en la fabricación de transistores cuyos procesos se están comenzando a lograr adaptar para fabricar circuitos fotónicos ([Lukas Chrostowski, 2010](#)). Tercero, el elevado ancho de banda que ofrece en comunicaciones digitales y el éxito de los cables ópticos ([Glick et al., 2018](#), [Lukas Chrostowski, 2010](#)).

Los dispositivos fotónicos se utilizan en grandes cantidades en los circuitos fotónicos integrados ([Lukas Chrostowski, 2010](#)). Estos trabajan en la escala de nanómetros y son diseñados para funcionar bajo ciertas especificaciones. Así, para que estos dispositivos cumplan los requerimientos deseados existen dos estrategias comunes: diseño tradicional y diseño inverso ([Molesky et al., 2018](#)).

En el diseño tradicional se define el dispositivo con geometrías simples que permiten obtener funciones analíticas de sus propiedades físicas. Esto se realiza para poder optimizar la función obtenida a partir de los parámetros que la definan. Dicha optimización se suele realizar haciendo un barrido de los parámetros, con algoritmos genéticos o

usando *particle swarm optimization*. Es un enfoque simple, pero que ha obtenido buenos resultados. Sin embargo, existen dos grandes inconvenientes con este planteamiento. Primero, solo estamos explorando una pequeña fracción de todos los posibles diseños. Segundo, por lo general no es conocido el límite del *performance* del dispositivo (Molesky et al., 2018, Su et al., 2020).

En el diseño inverso se busca hacer una mayor exploración de todos los posibles diseños. Para ello, ya no nos limitamos a solo usar diseños intuitivos. Ahora, definimos geometrías arbitrarias y usamos simulaciones computacionales para determinar las propiedades físicas del dispositivo (Molesky et al., 2018, Su et al., 2020). Este enfoque ha logrado conseguir mejores resultados que los obtenidos por el diseño tradicional (Molesky et al., 2018, Su et al., 2018). Sin embargo, este planteamiento viene acompañado de nuevos desafíos.

1.2 Descripción del Problema

Una estrategia común en diseño inverso es seleccionar una región rectangular a optimizar y dividirla en $n \times m$ *pixels* como si fuera una imagen. Luego, a cada *pixel* se le asocia el número 0 o 1. Cero representa la presencia de SiO_2 en la ubicación del *pixel* y uno la presencia de Si . De esta forma tenemos $2^{n \times m}$ posibles dispositivos (Su et al., 2020). Con esta definición podemos llegar a tener, por ejemplo, $2^{10 \times 10}$ posibles diseños (Malheiros-Silveira and Delalibera, 2020). Así, surgen algunas dificultades con esta estrategia:

1. Es imposible evaluar todas los posibles diseños.
2. Las simulaciones computacionales son muy costosas (Kudyshev et al., 2020).
3. El espacio de búsqueda es altamente no convexo (Su et al., 2018).

4. No todos los diseños son fabricables (Su et al., 2020).
5. Cada dispositivo es un problema distinto (Molesky et al., 2018).

Así, existe una demanda crítica de un *framework* capaz de optimizar dispositivos con un elevado número de parámetros dentro de un espacio de búsqueda no convexo (Kudyshev et al., 2020). Este es un problema muy grande, por ello en la presente tesis nos centraremos en optimizar un *bend-90°* y un *2-splitter*.

TODO: Agregar una imagen que contenga un *bend-90°* y un *2-splitter* tradicional y uno donde se haya realizado la división en cuadrados para optimizarlos

1.3 Justificación

El *bend-90°* y *2-splitter* se han escogido debido a que han sido estudiados usando diseño inverso desde el 2004, pero aún no han encontrado aplicación industrial (Molesky et al., 2018).

Por otro lado, desde el punto de vista computacional este problema es interesante porque ya hay estrategias computacionales conocidas para resolverlo, desde algoritmos evolutivos (Hansen, 2016) hasta redes neuronales (Goodfellow et al., 2015) y *depth learning* (Malkiel et al., 2018). Además, debido al alto costo computacional de las simulaciones (Schneider et al., 2019), el trabajo requiere de computación de alto desempeño. Así, es probable que se pueda obtener buenos resultados en la investigación aplicando el conocimiento ya existente en computación.

1.4 Objetivos

- Evaluar y comparar el *performance* y la convergencia de las técnicas modernas de optimización usadas para optimizar un *bend-90°* y *2-splitter*.

- Fabricar el diseño con mejor *performance* que se obtenga del *bend-90°* y del *2-splitter* para poder comparar las simulaciones computacionales con las mediciones físicas.

1.5 Aportes

Ante la reducida cantidad de trabajos similares en el área ([Elsawy et al., 2020](#), [Schneider et al., 2019](#)), este trabajo busca aportar una comparativa de las distintas técnicas de optimización que se aplican para optimizar un *bend-90°* y un *2-splitter*.

Capítulo 2

Trabajos relacionados

Dentro de las investigaciones que buscan optimizar un dispositivo fotónico usando diseño inverso, podemos identificar cuatro grandes pasos que estos trabajos siguen: i) parametrización, ii) definición de la figura de mérito (FOM), iii) optimización, iv) restricciones de fabricación.

2.1 Parametrización

En esta etapa se busca parametrizar la región de diseño con el objetivo de conseguir distintos dispositivos conforme variamos estos parámetros. Tres estrategias conocidas son: i) división por segmentos, ii) división por *pixeles*, iii) división por conjuntos de nivel.

1. División por segmentos

[Prosopio-Galarza et al. \(2019\)](#) utiliza esta estrategia para dividir un rectángulo de $2\mu m \times 1.5\mu m$ en 13 rectángulos verticales idénticos que particionan la región de diseño. Luego, parametriza esta región con 13 variables que representan la altura de cada rectángulo. Finalmente, centra cada rectángulo verticalmente y une los extremos de rectángulos consecutivos con un recta. De esta manera, se obtienen diseños compactos y con simetría respecto al eje X . Esta región es unida a tres guías de onda fijas para formar un *2-splitter*.

2. División por *pixeles*

[Su et al. \(2020\)](#) divide una región cuadrada de $2.6\mu m \times 2.6\mu m$ en *pixeles* de $40nm \times 40nm$ para diseñar un *bend-90°*. A cada *pixel* le asocia un valor dado por la fórmula 2.1:

$$\varepsilon_i = \varepsilon_{Si} + (1 - \lambda_i)\varepsilon_{SiO_2} \quad \lambda_i \in [0, 1] \quad (2.1)$$

Donde $\varepsilon_{Si} = 3.48$ es la permitividad del *Si* y $\varepsilon_{SiO_2} = 1.44$ es la permitividad del *SiO2*. Con esta ecuación [Su et al. \(2020\)](#) mapea el intervalo $[0, 1]$ con el intervalo $[1.44, 3.48]$. Esto se realiza para determinar la permitividad que hay en la ubicación del *pixel* y poder simular las ecuaciones de Maxwell en el dispositivo. Con esta parametrización obtenemos una cantidad infinita de diseños, mas solo nos interesan aquellos donde λ_i es entero, pues en caso contrario un *pixel* se mapea a la permitividad de un material desconocido.

3. División por conjuntos de nivel

[Piggott et al. \(2017\)](#) diseña un *3-splitter* dividiendo una región de $3.8\mu m \times 2.5\mu m$ en *pixeles* de $40nm \times 40nm$. A cada *pixel* le asocia una permitividad dada por la siguiente fórmula:

$$\varepsilon(x, y) = \begin{cases} \varepsilon_{Si} & \text{si } \phi(x, y) \leq 0 \\ \varepsilon_{SiO_2} & \text{si } \phi(x, y) > 0 \end{cases}$$

Así, [Piggott et al. \(2017\)](#) puede definir ϕ como una función continua y obtener un diseño fabricable.

TODO: Insertar imágenes para que se entienda mejor.

Las distintas estrategias mostradas tienen sus propias características. [Prosopio-Galarza et al. \(2019\)](#) utiliza una parametrización que le asegura simetría y regiones compactas, pero tiene un espacio de búsqueda más reducido que las otras opciones. Por otro

lado, [Su et al. \(2020\)](#) usa *pixeles* para trabajar con un espacio de búsqueda más grande a cambio de poder parametrizar diseños no fabricables. De manera similar [Piggott et al. \(2017\)](#) usa *pixeles*, pero estos representan un dispositivo fabricable a cambio de tener que definir una nueva función.

2.2 Definición de la FOM

La FOM es la función que nos permite determinar que diseños son mejores. Existen distintas estrategias para definirla, por ejemplo:

- [Prosopio-Galarza et al. \(2019\)](#) busca optimizar la transmitancia.
- [Su et al. \(2020\)](#) define un valor deseado de transmitancia.
- [Piggott et al. \(2017\)](#) busca trabajar con 3 longitudes de onda y que con cada longitud la transmitancia se maximice en una guía de onda de salida y se minimice en las otras.

TODO: Tal vez escribirlo con fórmulas para dejarlo más claro.

2.3 Optimización

En la etapa de optimización la forma a proceder depende de la parametrización realizada. Así, tenemos:

- [Prosopio-Galarza et al. \(2019\)](#) define un rango en el que puede variar cada altura con lo cual limita el espacio de búsqueda. Luego, por separado, utiliza tres algoritmos para explorar los posibles diseños: i) *Particle Swarm Optimization*, ii) *Shrinking Box algorithm*, iii) *Steepest Ascent algorithm*. Para obtener el valor de la FOM utiliza como simulador ANSY Lumerical FDTD.

- [Su et al. \(2020\)](#) ya tiene una parametrización donde cada *pixel* tiene un rango definido de valores. Pero, para asegurar el obtener un diseño fabricable divide la optimización en dos pasos:

1. Optimización continua: Se varía el valor de λ_i en el intervalo $[0, 1]$ sin importar si se obtiene dispositivos no fabricables.
2. Optimización discreta: Se considera el resultado de la optimización continua como punto inicial, se comienza a optimizar, pero se va aplicando transformaciones que permitan que cada *pixel* vaya convergiendo a un valor entero.

Como algoritmo de optimización utiliza $L-BFGS-B$ y MMA . Las simulaciones son realizadas en SPIN.

- [Piggott et al. \(2017\)](#) trabaja sobre la función ϕ en un espacio de búsqueda no limitado. Luego, realiza la optimización con un algoritmo propio que está basado en *gradient-descent* y *line search*. De igual manera que [Su et al. \(2020\)](#), las simulaciones son realizadas en SPIN.

TODO: Agregar un párrafo sobre Scheneider y sobre Elsayy

2.4 Restricciones de fabricación

El diseño obtenido por la fabricación por lo general aún debe pasar por una etapa donde se tenga en consideración aspectos de fabricación que tenga un buen desempeño al fabricarse. Así, tenemos:

- [Prosopio-Galarza et al. \(2019\)](#) busca establecer un rango válido que deberían tener los ángulo formados entre rectángulos consecutivos.

- [Su et al. \(2020\)](#) simplemente se aseguró que el tamaño de los *pixeles* fuera más grande que el mínimo tamaño que se podía fabricar.
- [Piggott et al. \(2017\)](#) consideró que su diseño tenga un radio de curvatura mayor a un valor dado y que no haya presencia de huecos que superen un tamaño determinado.

REFERENCIAS BIBLIOGRÁFICAS

- Elsawy, M. M., Lanteri, S., Duvigneau, R., Fan, J. A., and Genevet, P. (2020). Numerical Optimization Methods for Metasurfaces. *Laser and Photonics Reviews*, 14(10):1–17.
- Glick, M., Kimmerling, L. C., and Pfahl, R. C. (2018). A Roadmap for Integrated Photonics. *Optics and Photonics News*, 29(3):36.
- Goodfellow, I. J., Vinyals, O., and Saxe, A. M. (2015). Qualitatively characterizing neural network optimization problems. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*.
- Hansen, N. (2016). The CMA Evolution Strategy: A Tutorial.
- Kudyshev, Z. A., Kildishev, A. V., Shalaev, V. M., and Boltasseva, A. (2020). Machine learning-assisted global optimization of photonic devices. *Nanophotonics*, 10(1):371–383.
- Lukas Chrostowski (2010). *Silicon Photonics Design: From Device to System*.
- Malheiros-Silveira, G. N. and Delalibera, F. G. (2020). Inverse design of photonic structures using an artificial bee colony algorithm. *Applied Optics*, 59(13):4171.
- Malkiel, I., Mrejen, M., Nagler, A., Arieli, U., Wolf, L., and Suchowski, H. (2018). Plasmonic nanostructure design and characterization via Deep Learning. *Light: Science and Applications*, 7(1).

- Molesky, S., Lin, Z., Piggott, A. Y., Jin, W., Vucković, J., and Rodriguez, A. W. (2018). Inverse design in nanophotonics. *Nature Photonics*, 12(11):659–670.
- Piggott, A. Y., Petykiewicz, J., Su, L., and Vučković, J. (2017). Fabrication-constrained nanophotonic inverse design. *Scientific Reports*, 7(1):1–7.
- Prosopio-Galarza, R., De La Cruz-Coronado, J., Hernandez-Figueroa, H. E., and Rubio-Noriega, R. (2019). Comparison between optimization techniques for Y-junction devices in SOI substrates. *Proceedings of the 2019 IEEE 26th International Conference on Electronics, Electrical Engineering and Computing, INTERCON 2019*, pages 1–4.
- Schneider, P. I., Garcia Santiago, X., Soltwisch, V., Hammerschmidt, M., Burger, S., and Rockstuhl, C. (2019). Benchmarking Five Global Optimization Approaches for Nano-optical Shape Optimization and Parameter Reconstruction. *ACS Photonics*, 6(11):2726–2733.
- Su, L., Piggott, A. Y., Sapra, N. V., Petykiewicz, J., and Vučković, J. (2018). Inverse Design and Demonstration of a Compact on-Chip Narrowband Three-Channel Wavelength Demultiplexer. *ACS Photonics*, 5(2):301–305.
- Su, L., Vercruysse, D., Skarda, J., Sapra, N. V., Petykiewicz, J. A., and Vučković, J. (2020). Nanophotonic inverse design with SPINS: Software architecture and practical considerations. *Applied Physics Reviews*, 7(1).