

EVALUACIÓN DE ALGORITMOS DE OPTIMIZACIÓN DE PRIMER ORDEN PARA LA OPTIMIZACIÓN TOPOLÓGICA ROBUSTA DE DISPOSITIVOS NANOFOTÓNICOS

José Leonidas García Gonzales
jose.garcia@utec.edu.pe

Asesor: Jorge Gonzalez
Dic/2022



Agenda

Introducción

Marco teórico

Trabajos relacionados

Metodología

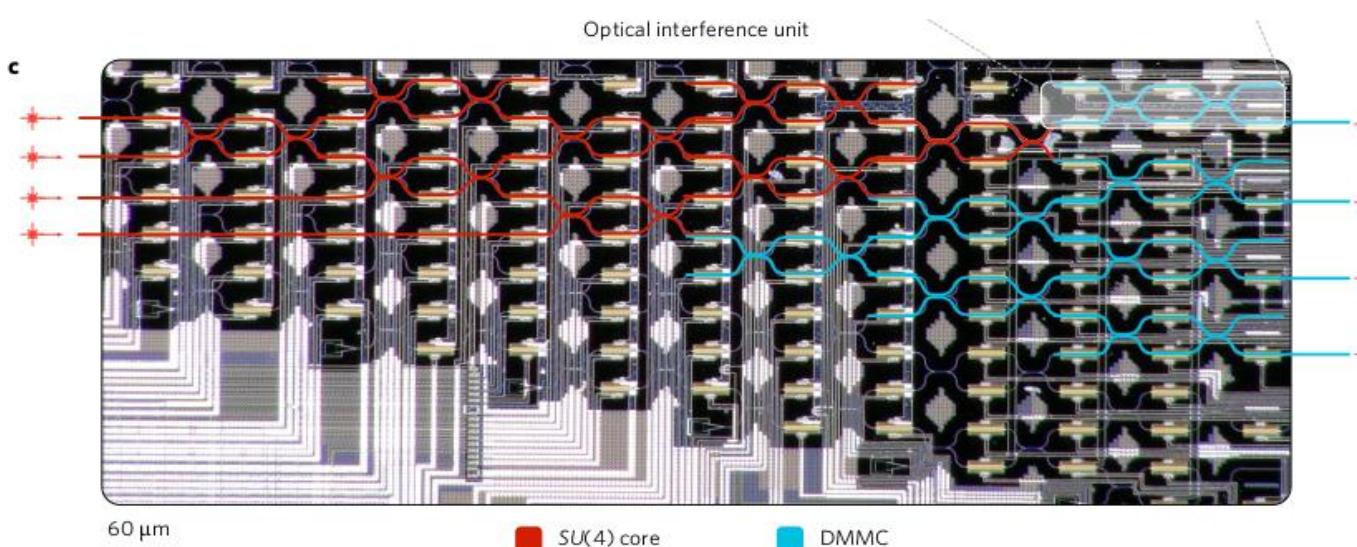
Resultados

Conclusiones

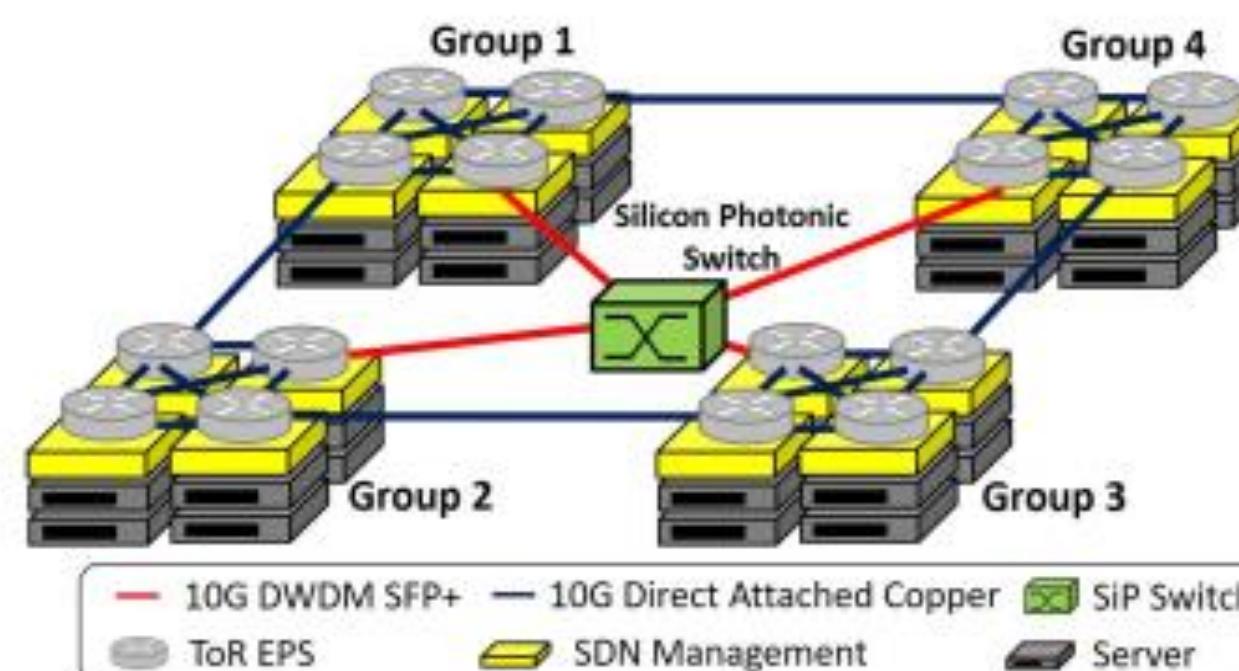
Introducción

La fotónica integrada son circuitos que usan fotones para funcionar. En este trabajo nos centramos en circuitos fotónicos fabricados en silicio (Si) (índice de refracción, precio, compatibilidad) [Lipson+, 2003]

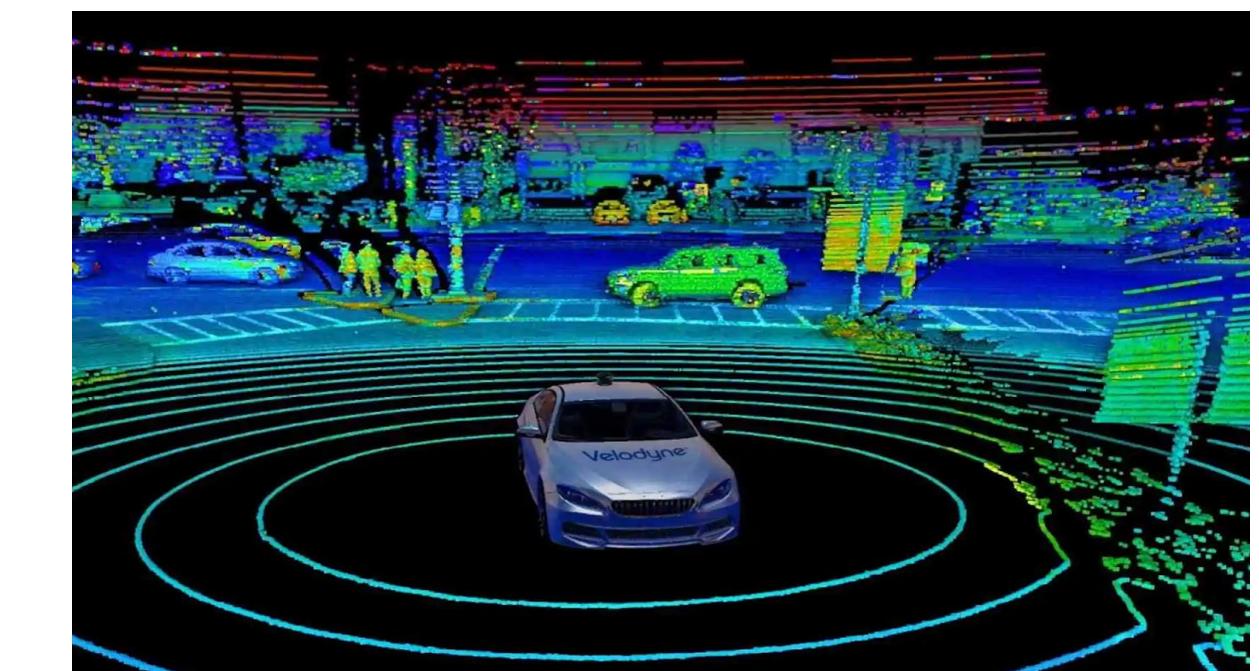
Características	Aplicaciones
Bajo consumo energético	Redes neuronales ópticas
Alto ancho de banda	Interconexiones reconfigurables en centros de datos
Interconexiones independientes de la distancia	IoT (e.g., LIDAR)



[Shen+, 2017]



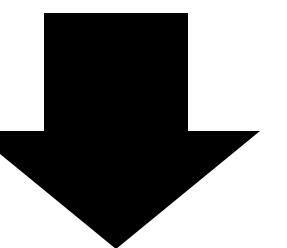
[Shen+, 2019]



[Vučković, 2019]

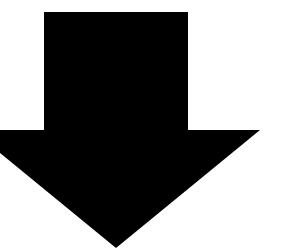
Fotónica integrada en silicio

Deseadas características-aplicaciones



dispositivos x chip

Es un problema complicado



Dispositivos fundamentales

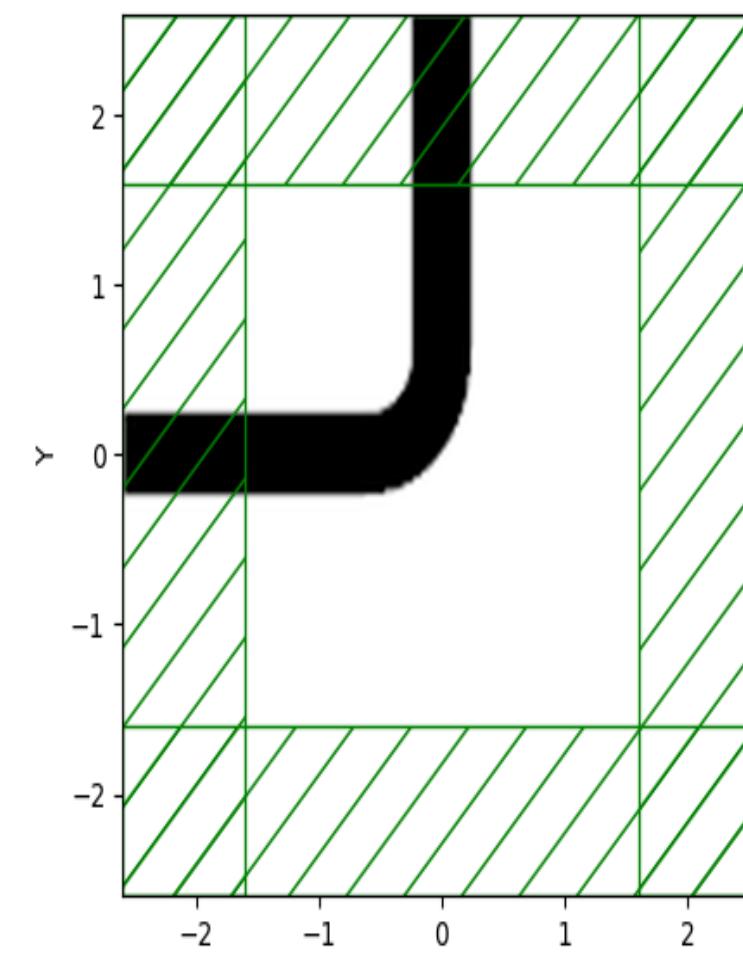
Más sencillo de estudiar

I) *Bend*

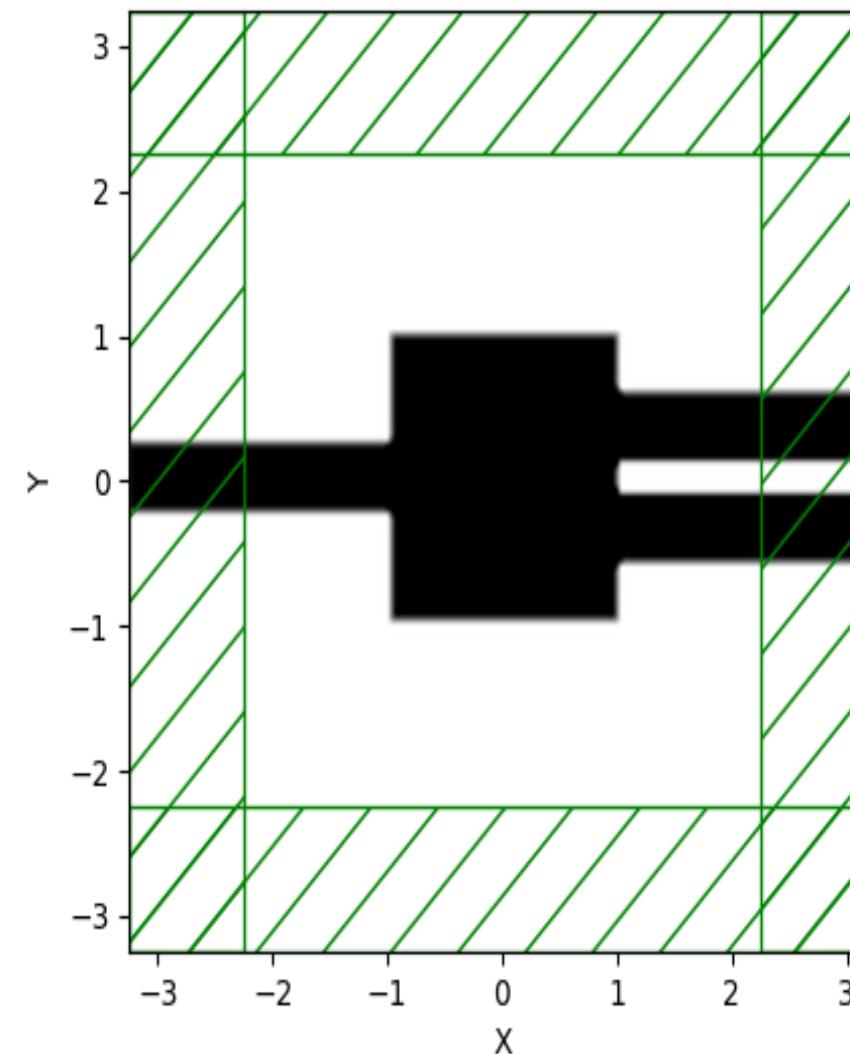
II) WDM

Diseño intuitivo

I) Bend

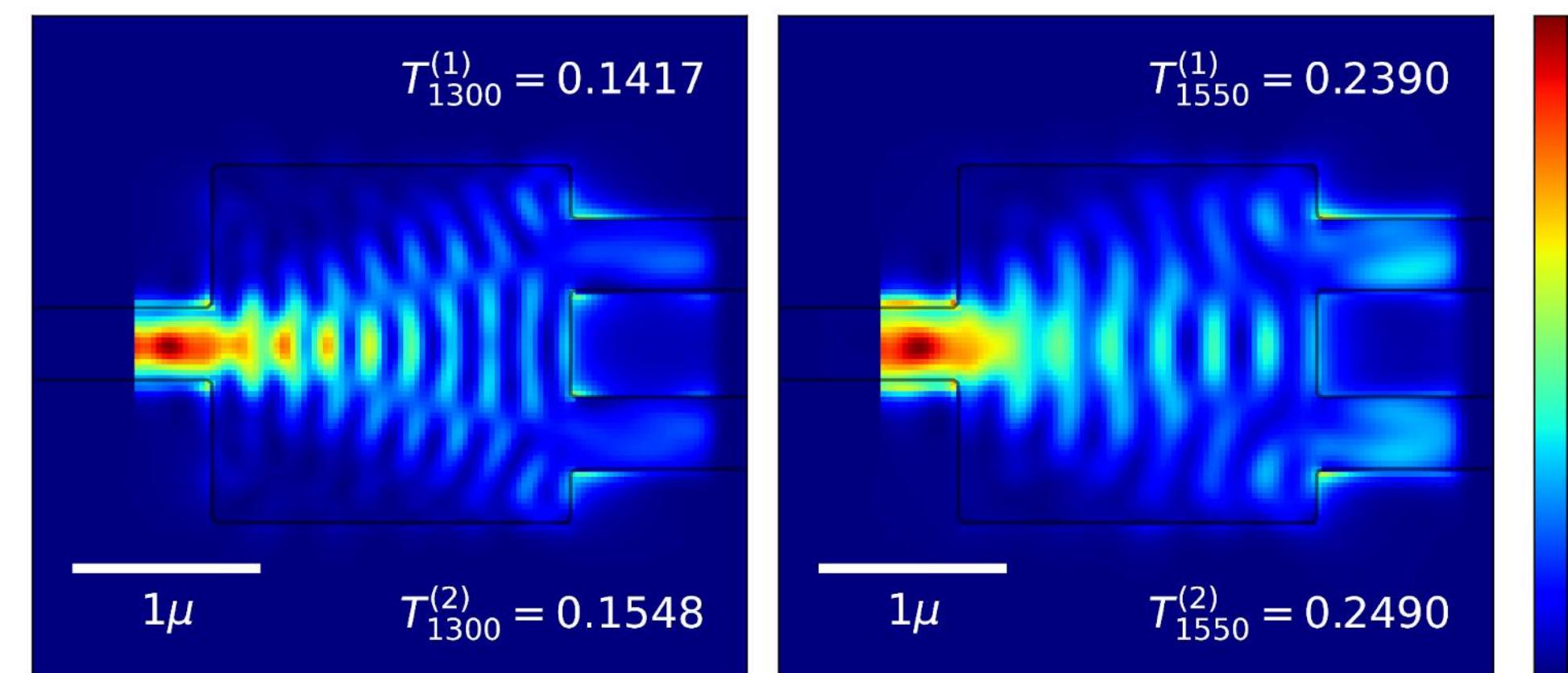
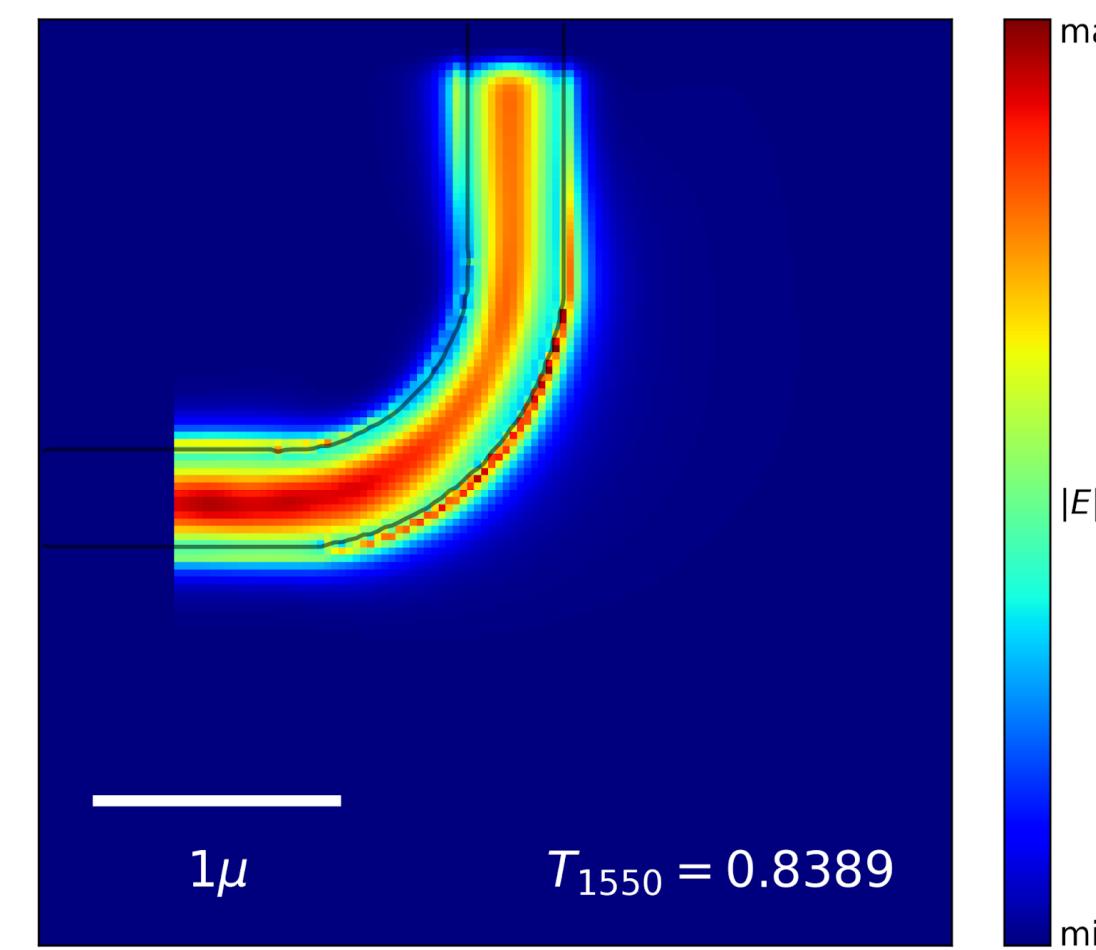


II) WDM



No son eficientes en esta escala
($2\mu\text{m} \times 2\mu\text{m}$)

Simulaciones



Bend: ↑ transmitancia a 1550nm ☺

WDM(1300nm): ↑ transmitancia en el brazo superior ☺

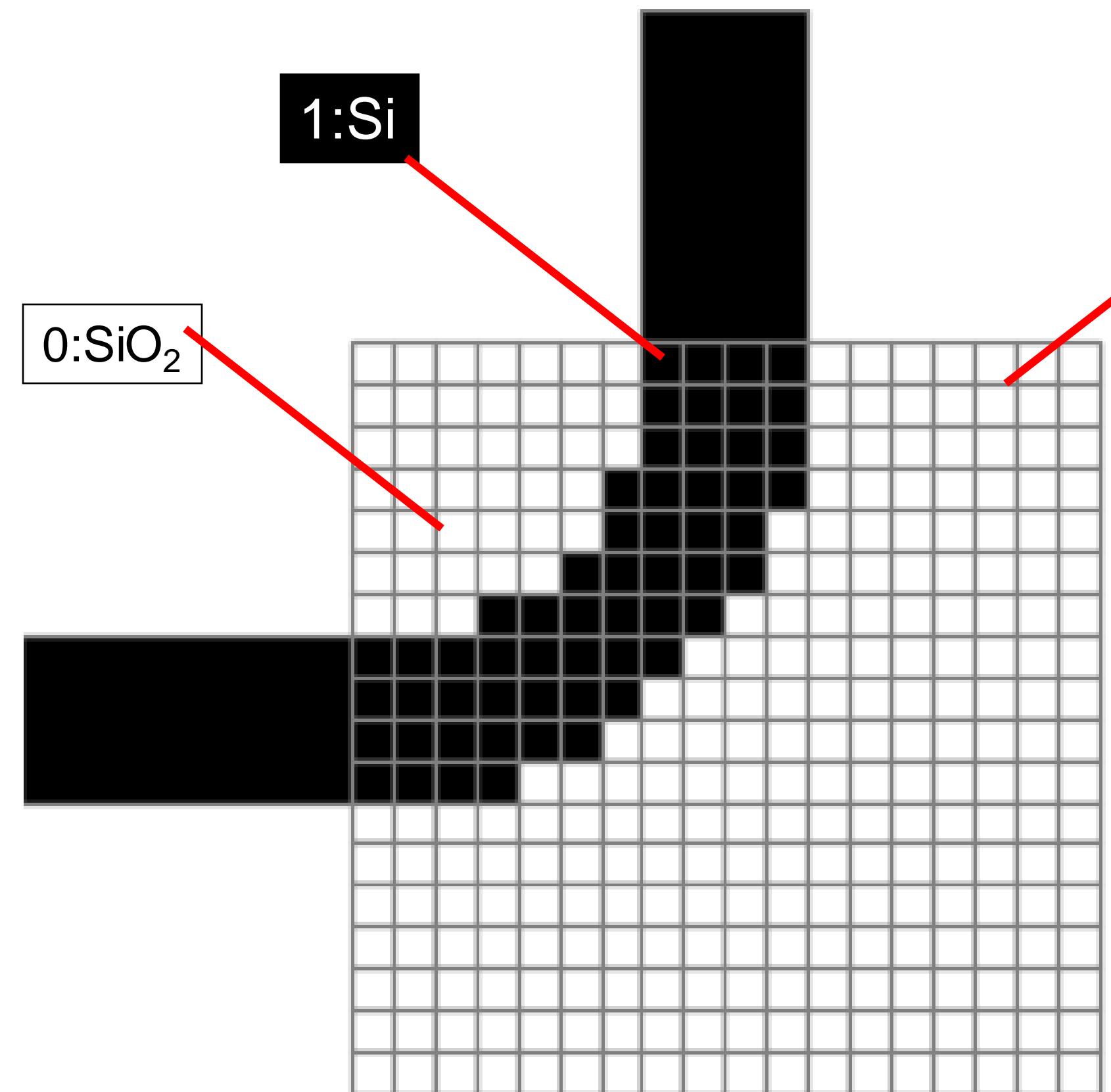
WDM(1550nm): ↑ transmitancia en el brazo inferior ☺

Diseño inverso



Diseños eficientes [Su+, 2020]

Discretización de un *bend*



Transmitancia: Relación entre la potencia del flujo que sale del dispositivo con la potencia del flujo que ingresa.

Usamos una matriz **P** de 0s y 1s para representar los diseños (**parametrización**)

Dificultades computacionales

1. Imposible evaluar todos los diseños
2. **Simulaciones** computacionales costosas
3. Espacio de búsqueda no convexo
4. **No** todos los dispositivos son **fabricables**
5. Distintos dispositivos → distintos problemas

Dificultades experimentales

1. **Error de precisión** en los instrumentos de fabricación
2. **Sensibilidad** ante cambios de temperatura

Problema

Encontrar una parametrización que genere un dispositivo que optimice alguna propiedad deseada,

calculado mediante **simulaciones computacionales**, que mantenga un **óptimo funcionamiento** al ser **fabricado**.

Objetivos

1. Diseñar un bend y WDM con eficiencias mayores al 90 % y robustos ante errores de fabricación en un área de diseño de $2\mu\text{m} \times 2\mu\text{m}$.
 - a. Seleccionar estrategia de optimización
 - b. Definir función objetivo
 - c. Encontrar parametrizaciones óptimas
 - d. Encontrar parametrizaciones robustas
2. Comparar el desempeño y la convergencia de cinco algoritmos de optimización populares usados para optimizar dispositivos nanofotónicos.

Agenda

Introducción

Marco Teórico

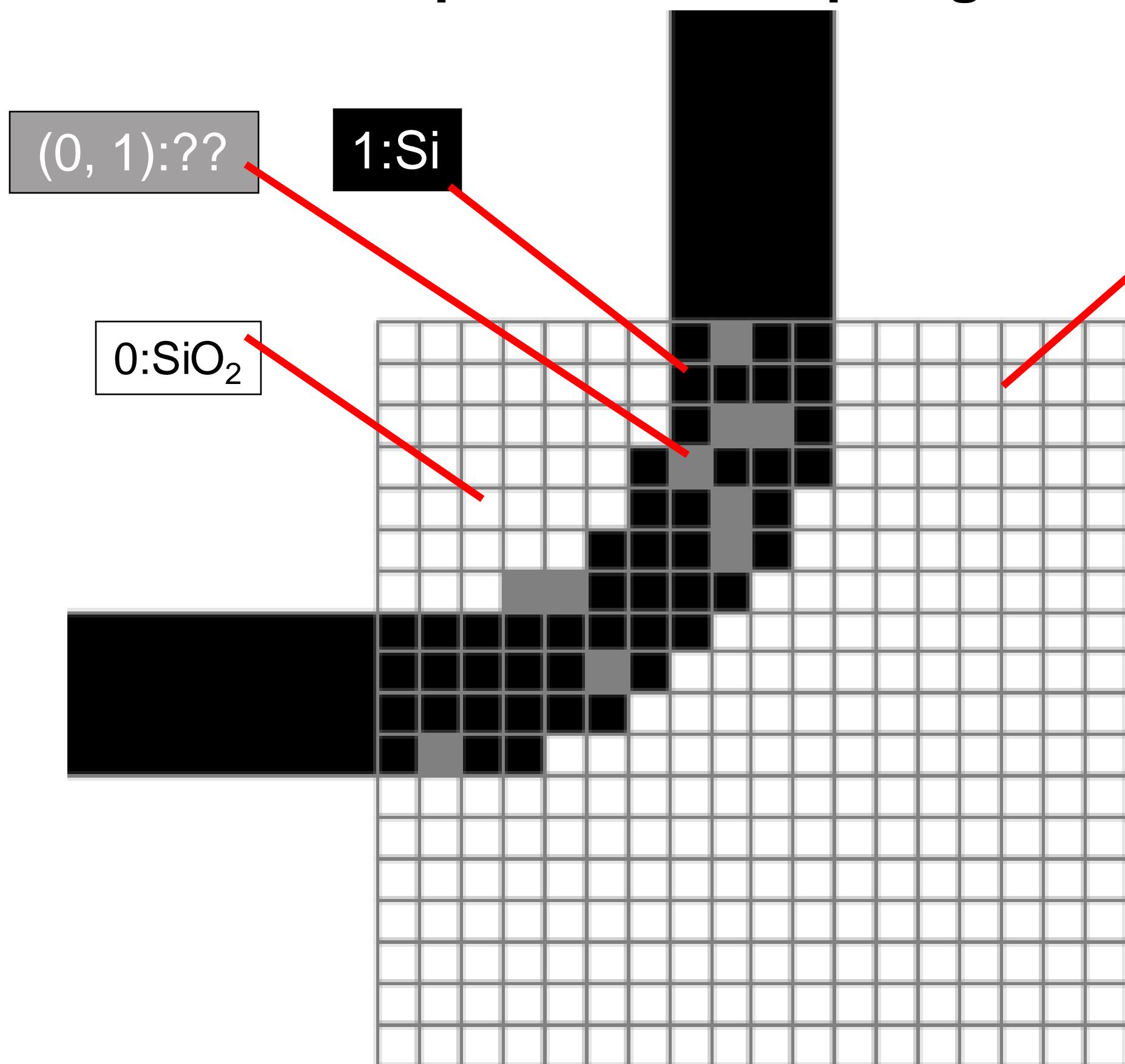
Revisión de la literatura

Metodología

Resultados

Conclusiones

Optimización topológica



Ahora la matriz **P** (parametrización) toma valores en [0, 1]

Función objetivo [Su+, 2020]

Bend $f_{obj}(\mathbf{P}) = \max \{T_{1550}(\mathbf{P})\}$

Maximizar transmitancia a 1550nm

WDM $f_{obj}(\mathbf{P}) = \max \left\{ \frac{\left(T_{1300}^{(1)}(\mathbf{P})\right)^2 + \left(1 - T_{1300}^{(2)}(\mathbf{P})\right)^2 + \left(1 - T_{1550}^{(1)}(\mathbf{P})\right)^2 + \left(T_{1550}^{(2)}(\mathbf{P})\right)^2}{4} \right\}$

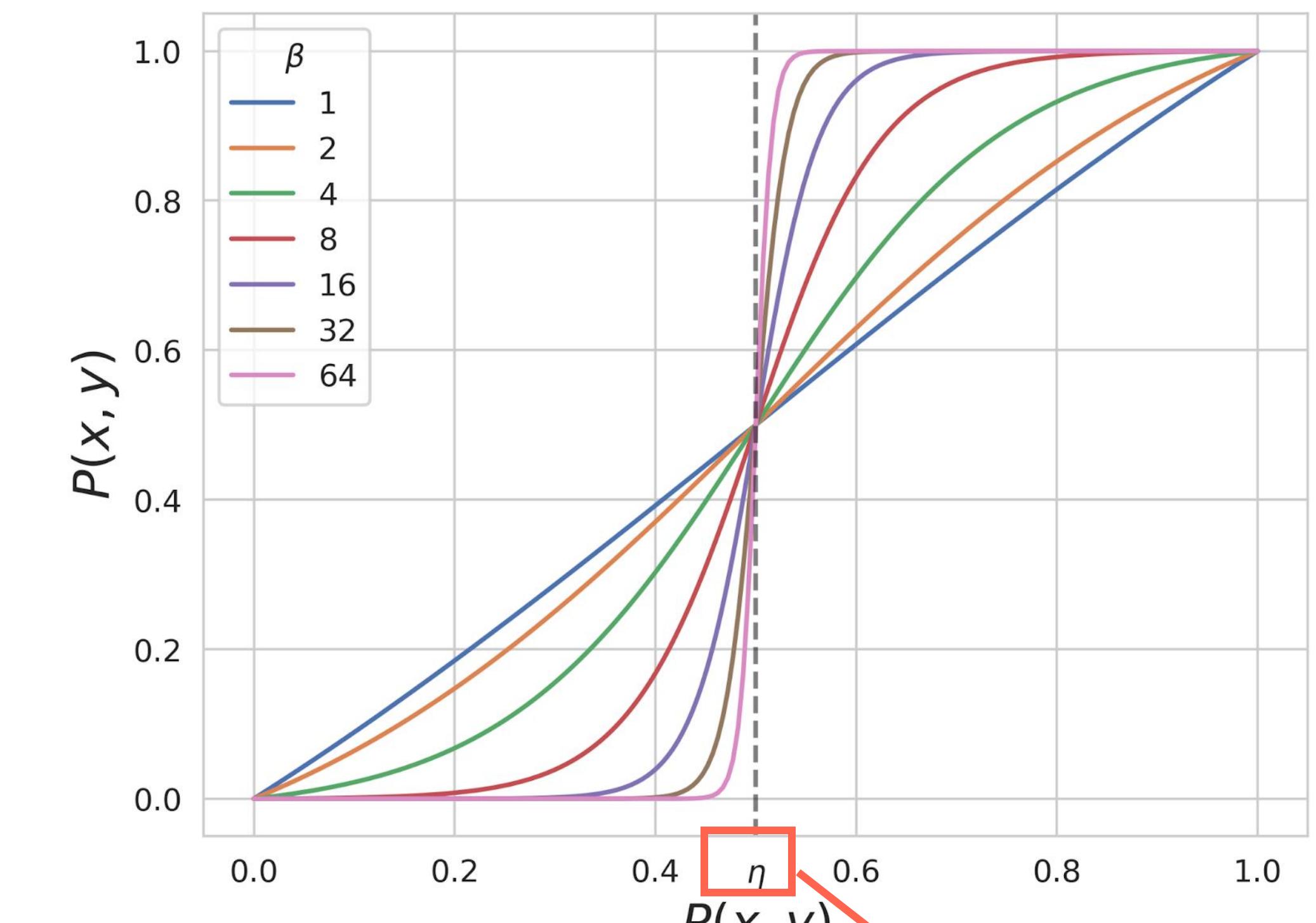
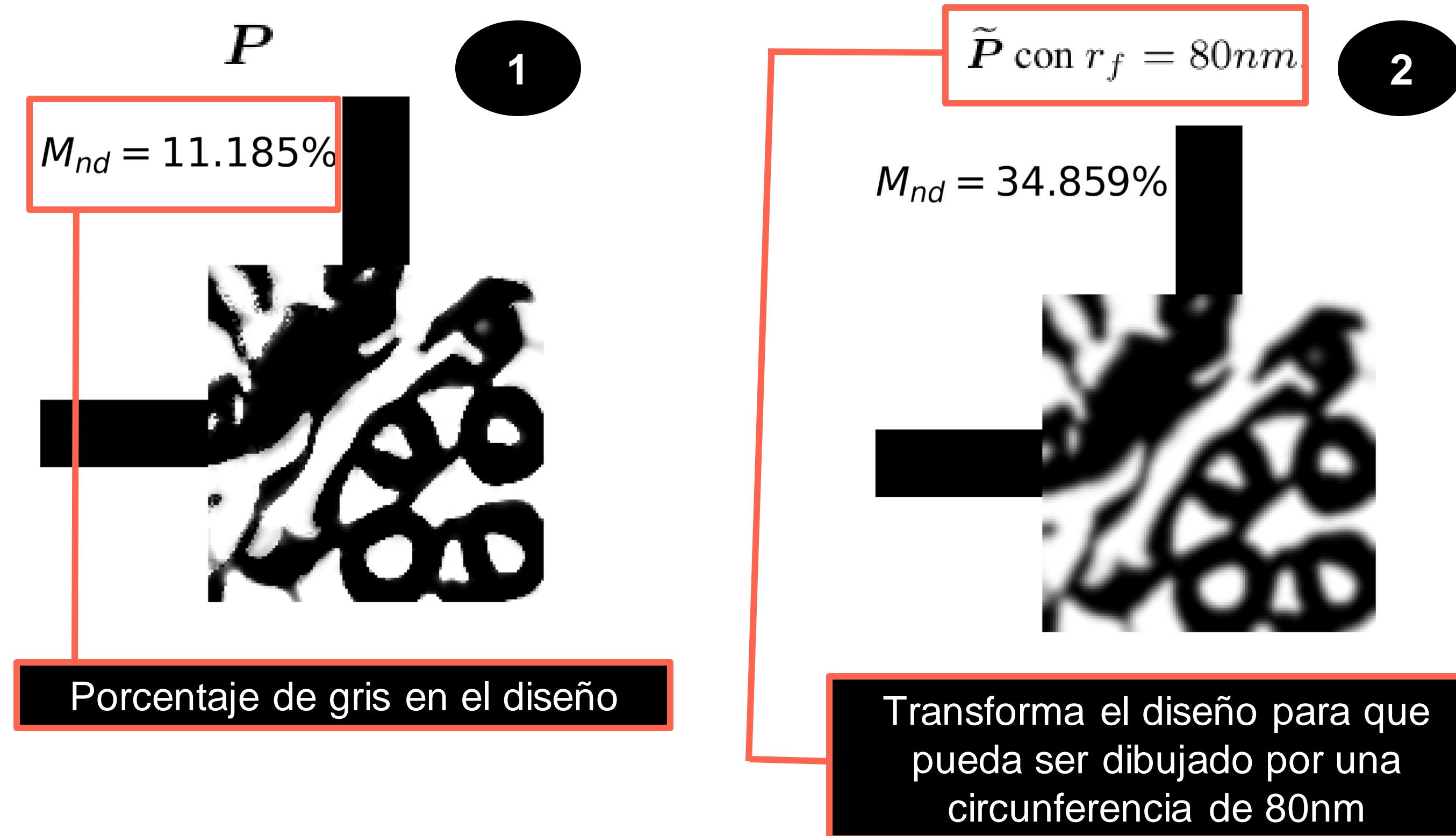
$$\varepsilon(x, y) = \varepsilon_{Si} * \mathbf{P}(x, y) + (1 - \mathbf{P}(x, y))\varepsilon_{SiO_2} \mid x \in [1, n] \wedge y \in [1, m],$$

$$\varepsilon_{SiO_2} = 1.44^2$$

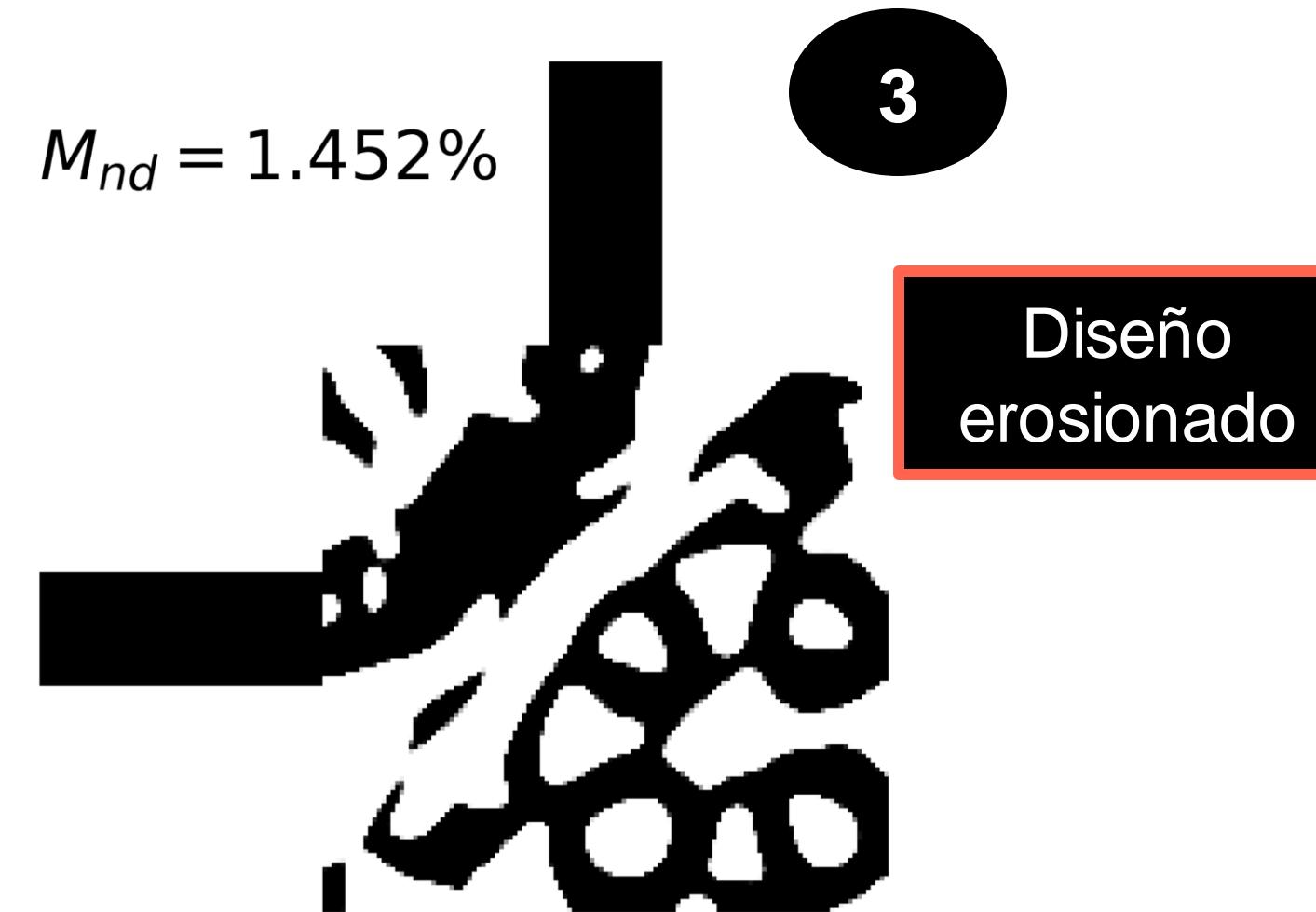
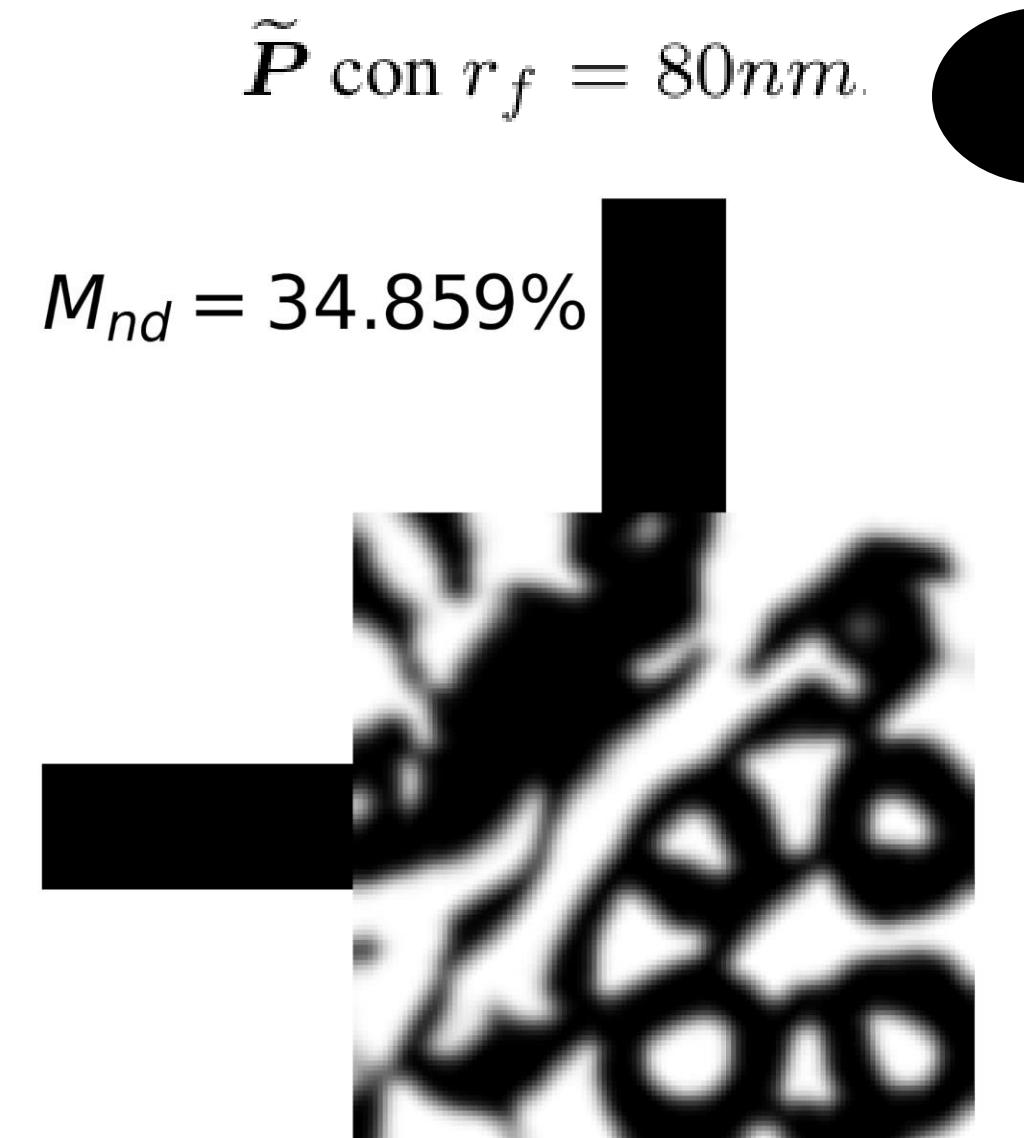
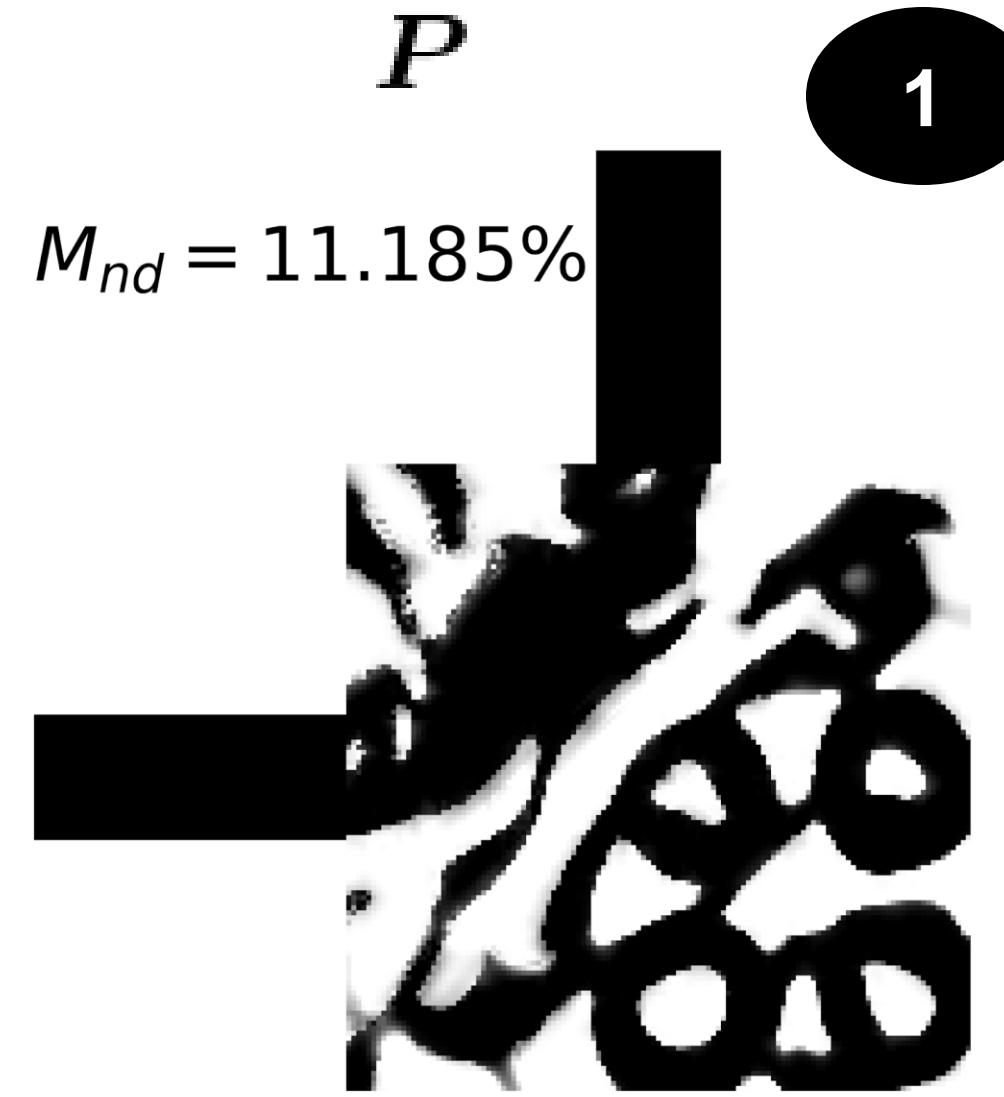
$$\varepsilon_{Si} = 3.48^2$$

- 1300nm: **maximizar** transmitancia en el brazo **superior** y **minimizar** en el **inferior**
- 1550nm: **minimizar** transmitancia en el brazo **superior** y **maximizar** en el **inferior**

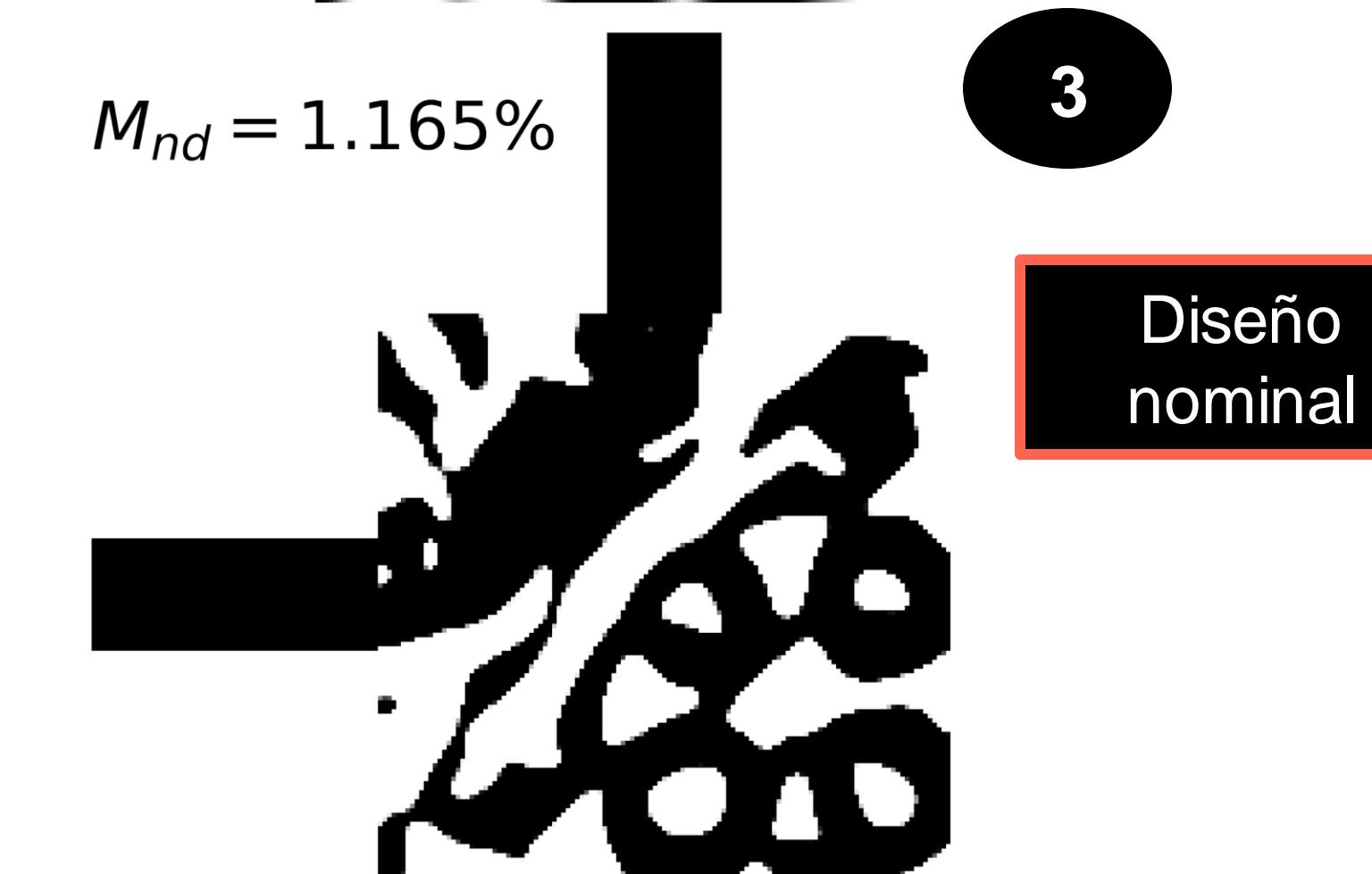
Transformaciones



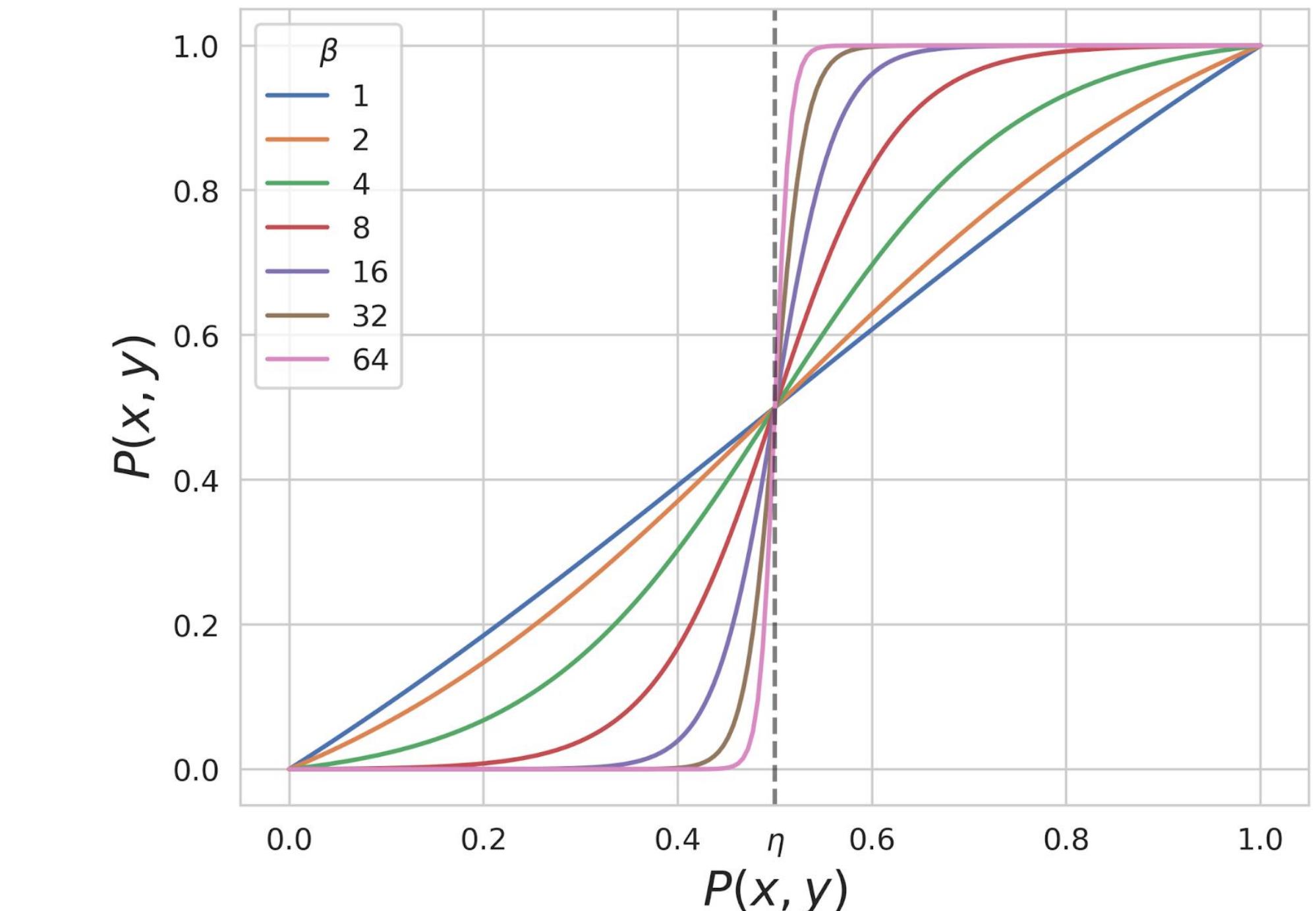
Transformaciones



\tilde{P} con $\eta_e = 0.7$ y $\beta = 2^6$



\tilde{P} con $\eta_i = 0.5$ y $\beta = 2^6$

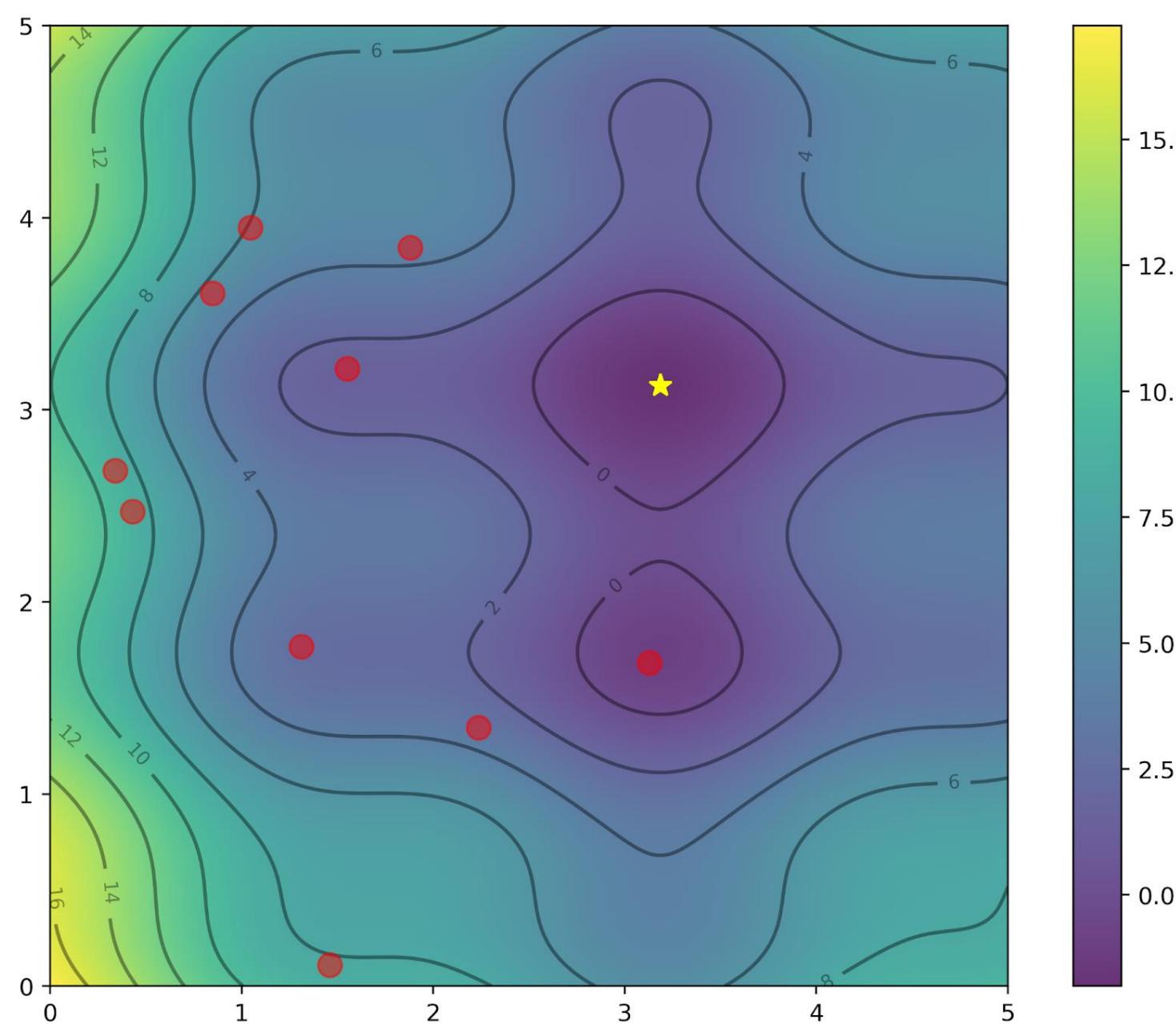


\tilde{P} con $\eta_d = 0.3$ y $\beta = 2^6$

Algoritmos gradient-free (1/3)

$$g(x, y) = (x - 3.14)^2 + (y - 2.72)^2 + \sin(3x + 1.41) + \sin(4y - 1.73).$$

GA



Algorithm 1: Genetic Algorithms (GA)

Data: $P, population_size, GA_range, n_selected_parents, prob_mutation$

Result: $\min(population)$

- 1 $population = generate_population()$
- 2 **for** $t = 0; t < k; t++$ **do**
- 3 $parents = select(population)$
- 4 $children = crossover(parents)$
- 5 $population = mutation(children)$

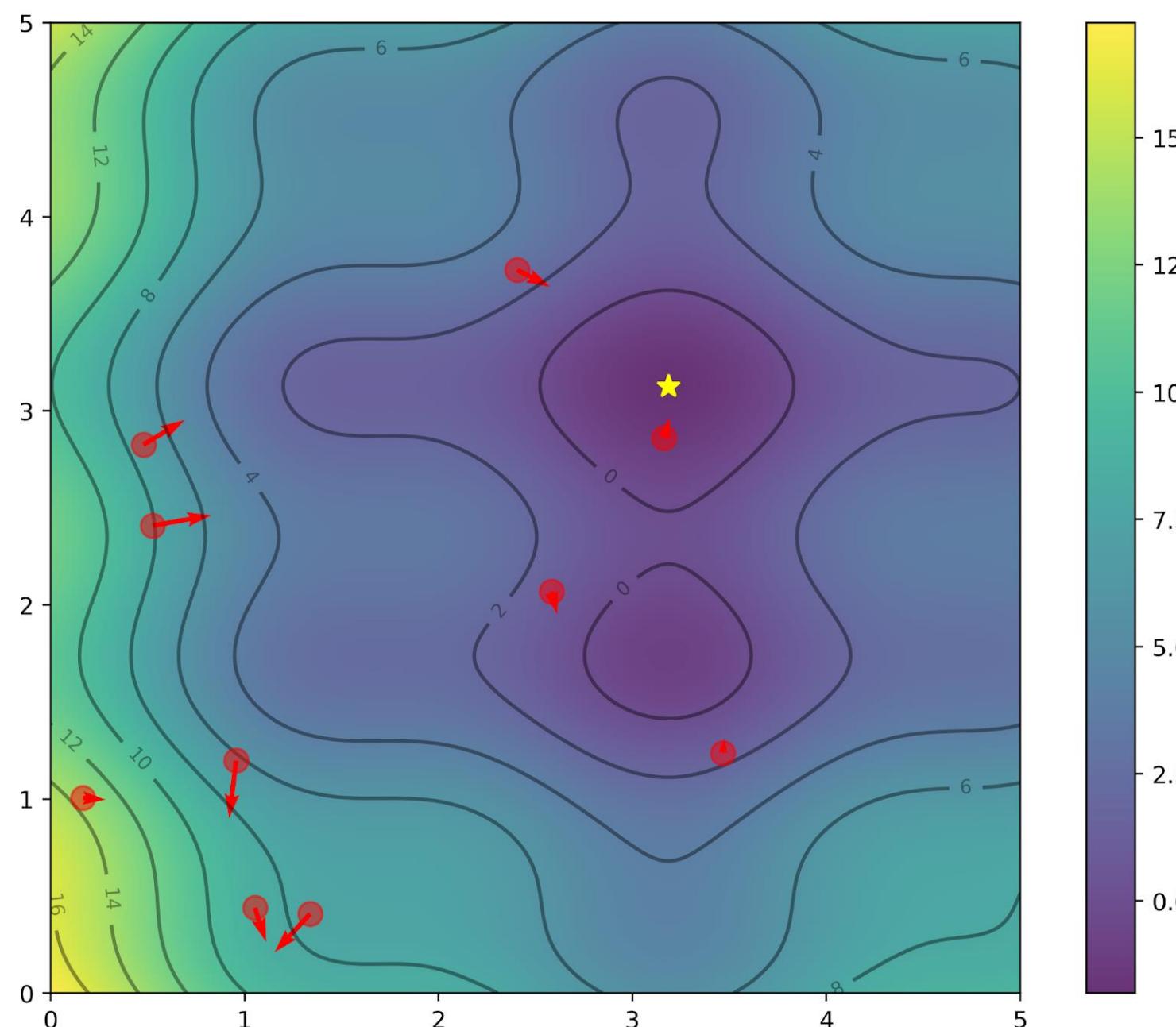
$$prob_i = \frac{\max(f) - f^{(i)}}{\sum_j \max(f) - f^{(j)}},$$

En las líneas 4 y 5 generamos nuevas parametrizaciones P

Algoritmos gradient-free (2/3)

$$g(x, y) = (x - 3.14)^2 + (y - 2.72)^2 + \sin(3x + 1.41) + \sin(4y - 1.73).$$

PSO



$$\mathbf{P}^{(i)} \leftarrow \mathbf{P}^{(i)} + \mathbf{V}^{(i)}$$

$$\mathbf{V}^{(i)} \leftarrow \omega \mathbf{V}^{(i)} + c_1 r_1 (\mathbf{P}_b^{(i)} - \mathbf{P}^{(i)}) + c_2 r_2 (\mathbf{P}_b - \mathbf{P}^{(i)})$$

Algorithm 2: Particle Swarm Optimization (PSO)

Data: \mathbf{P} , $population_size$, PSO_range , ω , c_1 , c_2

Result: \mathbf{P}_b

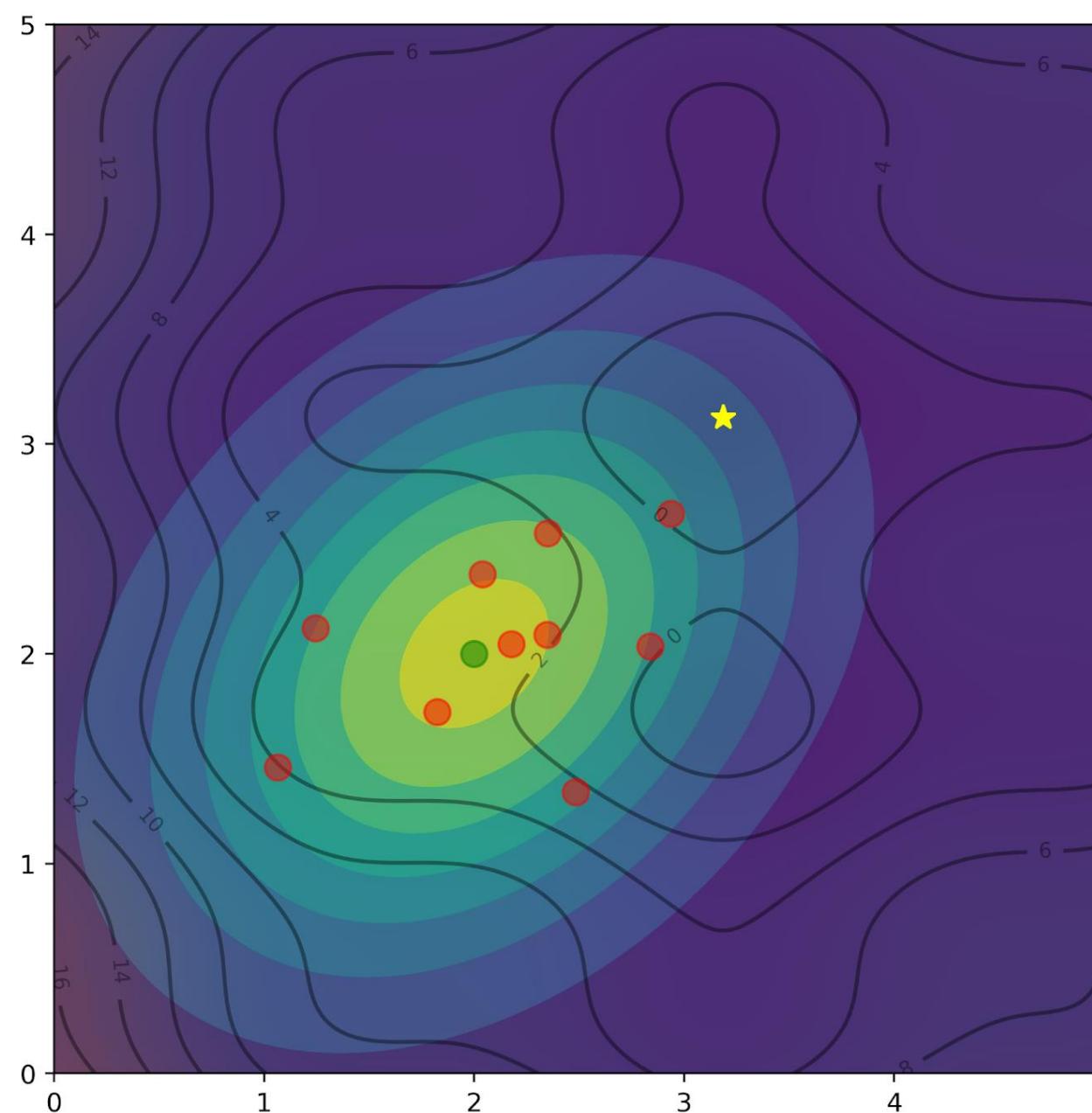
- 1 $population = generate_population()$
 - 2 **for** $t = 0$; $t < k$; $t++$ **do**
 - 3 $\mathbf{P}_b = select(population)$
 - 4 $population = mutation(population, \mathbf{P}_b)$
-

En la línea 4 generamos nuevas parametrizaciones \mathbf{P}

Algoritmos gradient-free (3/3)

$$g(x, y) = (x - 3.14)^2 + (y - 2.72)^2 + \sin(3x + 1.41) + \sin(4y - 1.73).$$

CMA-ES



$$\mathcal{N}(\mu, \sigma^2 C)$$

Algorithm 3: CMA-ES

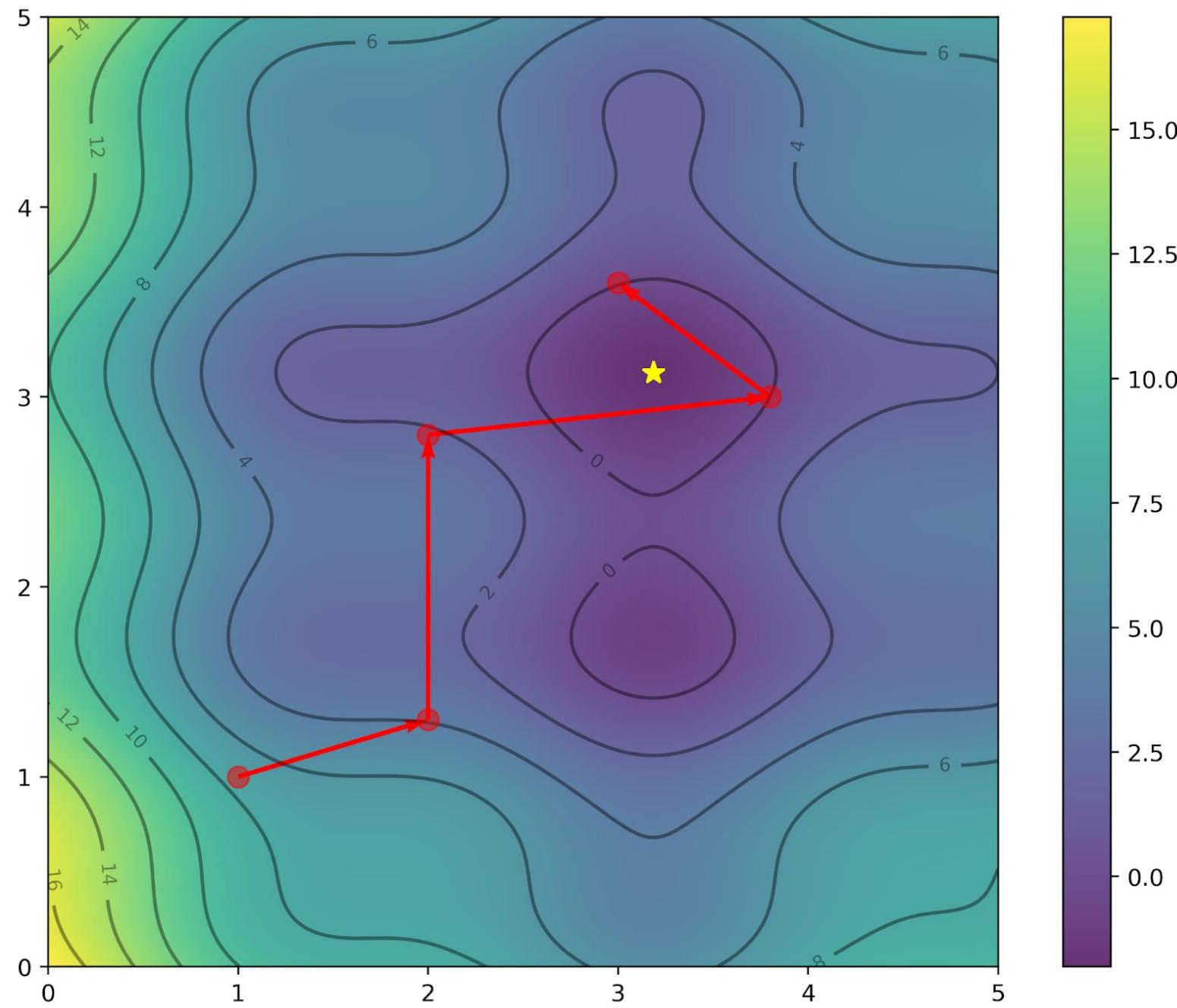
Data: P , $population_size$, σ
Result: μ

- 1 $\mu = flatten(P)$
- 2 **for** $t = 0$; $t < k$; $t++$ **do**
- 3 sample() // Obtener $population_size$ puntos de $\mathcal{N}(\mu, \sigma^2 C)$
- 4 update() // Ecuación 2.18
- 5 control() // Ecuación 2.19
- 6 adapt() // Ecuación 2.21

En la línea 3 generamos nuevas parametrizaciones P

Algoritmos gradient-based

GD



$$\boxed{\mathbf{P}^{(t+1)} \leftarrow \mathbf{P}^{(t)} - \gamma \nabla \mathbf{P}^{(t)}}$$

$$\gamma^{(t+1)} = \frac{|(\mathbf{P}^{(t)} - \mathbf{P}^{(t-1)})^T (\nabla f(\mathbf{P}^{(t)}) - \nabla f(\mathbf{P}^{(t-1)}))|}{\|\nabla f(\mathbf{P}^{(t)}) - \nabla f(\mathbf{P}^{(t-1)})\|^2}$$

MMA

Se calcula $f_j(\mathbf{x}^{(i)})$ y $\nabla f_j(\mathbf{x}^{(i)})$ para $j = 0, 1, \dots, m^+$.

$$f_j^{(i)}(\mathbf{x}) = r_j^{(k)} + \sum_{t=1}^n \left(\frac{p_{jt}^{(i)}}{U_t^{(i)} - x_t} + \frac{q_{jt}^{(i)}}{x_t - L_t^{(i)}} \right)$$

$$p_{jt}^{(i)} = \begin{cases} (U_t^{(i)} - x_t^{(i)})^2 \partial f_j / \partial x_t, & \text{si } 0 < \partial f_j / \partial x_t \\ 0, & \text{en otro caso} \end{cases},$$

$$q_{jt}^{(i)} = \begin{cases} -(x_t^{(u)} - L_t^{(i)})^2 \partial f_j / \partial x_t, & \text{si } \partial f_j / \partial x_t < 0 \\ 0, & \text{en otro caso} \end{cases},$$

$$r_j^{(i)} = f_j(\mathbf{x}^{(i)}) + \sum_{t=1}^n \left(\frac{p_{jt}^{(i)}}{U_t^{(i)} - x_t^{(i)}} + \frac{q_{jt}^{(i)}}{x_t^{(i)} - L_t^{(i)}} \right)$$

L-BFGS-B

En esencia, busca optimizar este modelo cuadrático:

$$m_i(\mathbf{x}) = f(\mathbf{x}_i) + (\nabla f(\mathbf{x}_i))^T (\mathbf{x}_i - \mathbf{x}) + \frac{1}{2} (\mathbf{x}_i - \mathbf{x})^T \mathbf{B}_i (\mathbf{x}_i - \mathbf{x})$$

donde la matriz \mathbf{B} es una aproximación de la matriz Hessiana.

Estos son llamados algoritmos de optimización de primer orden porque usan la gradiente para guiar su búsqueda.

Agenda

Introducción

Marco Teórico

Revisión de la literatura

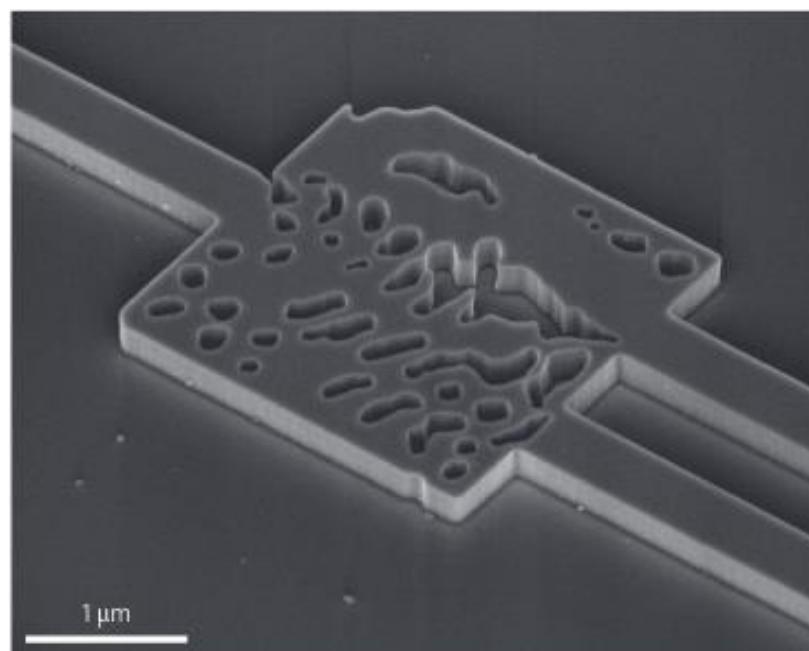
Metodología

Resultados

Conclusiones

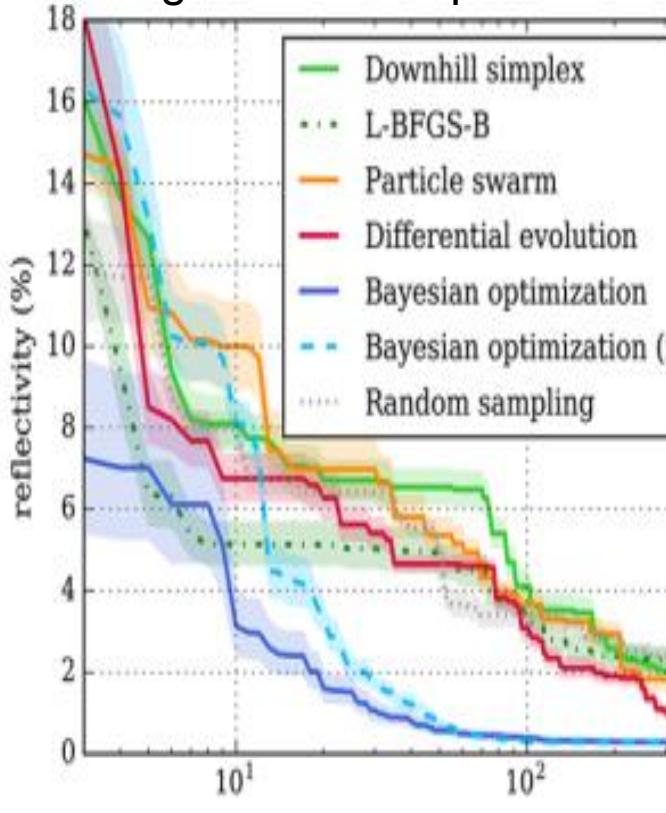
Revisión de la Literatura

Imagen SEM de un WDM



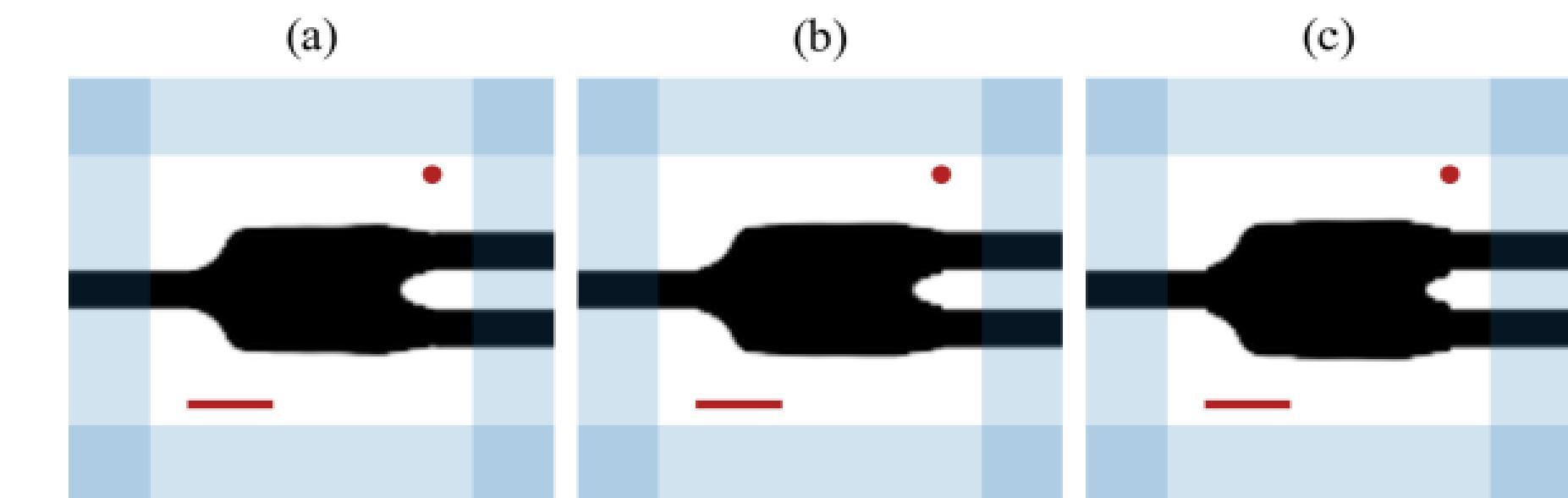
[Lazarov +, 2016]

Comparación de desempeño de algoritmos de optimización



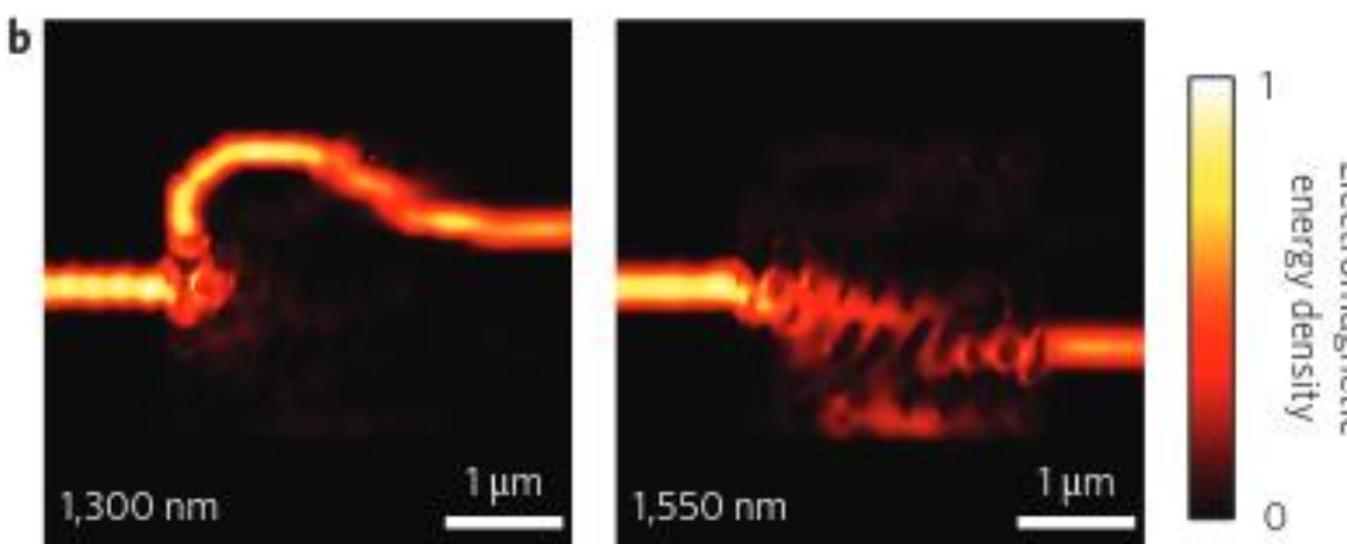
[Malheiros-Silveira y Delalibera, 2020]

Simulación del diseño erosionado, nominal y dilatado de un *splitter*



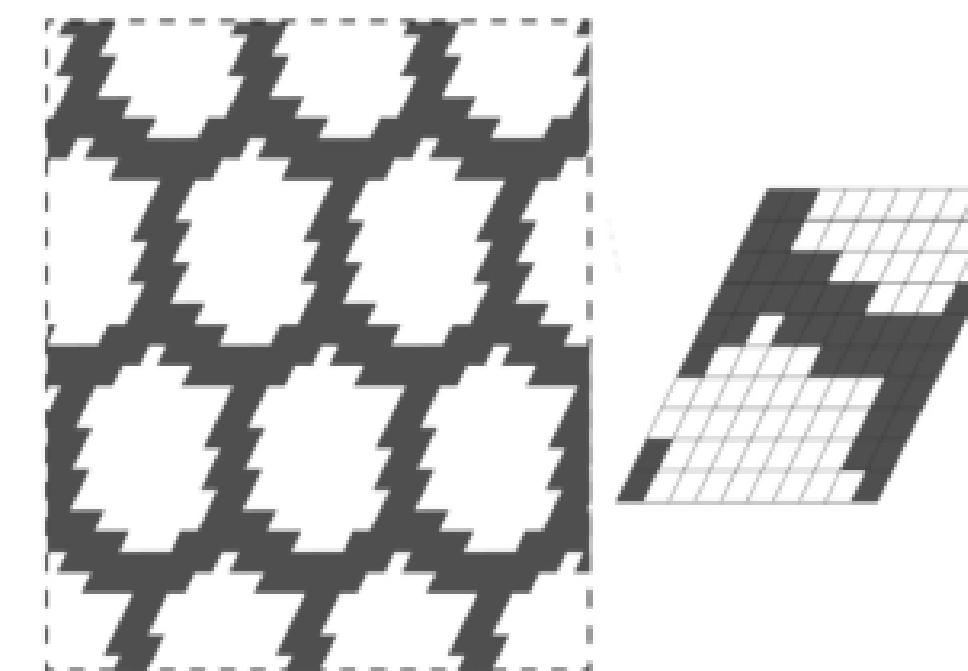
[Hammond +, 2021]
(Grupo Ralph)

[Piggott +, 2015]
(Grupo Vučković)



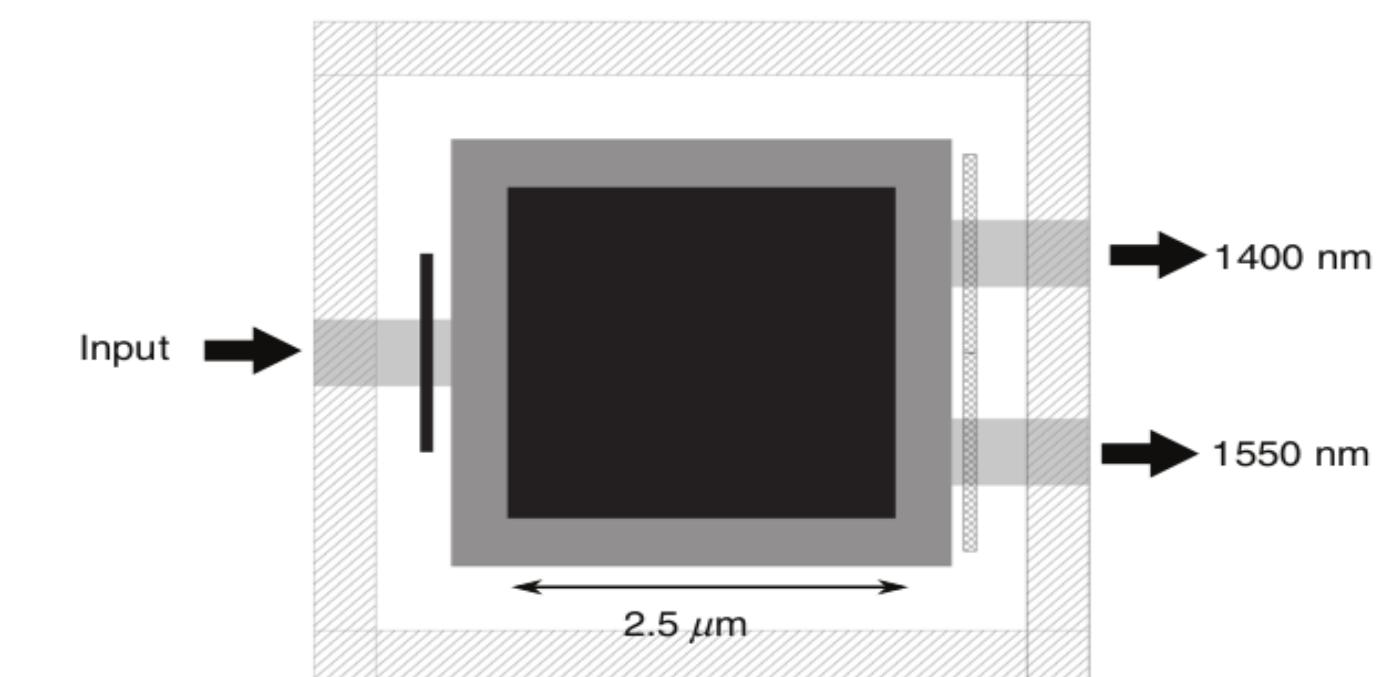
Densidad de energía en el diseño de un WDM

[Schneider +, 2019]



Estructura optimizada usando programación entera

[Su +, 2020]
(Grupo Vučković)



Diseño a optimizar de un *bend*

Agenda

Introducción

Marco Teórico

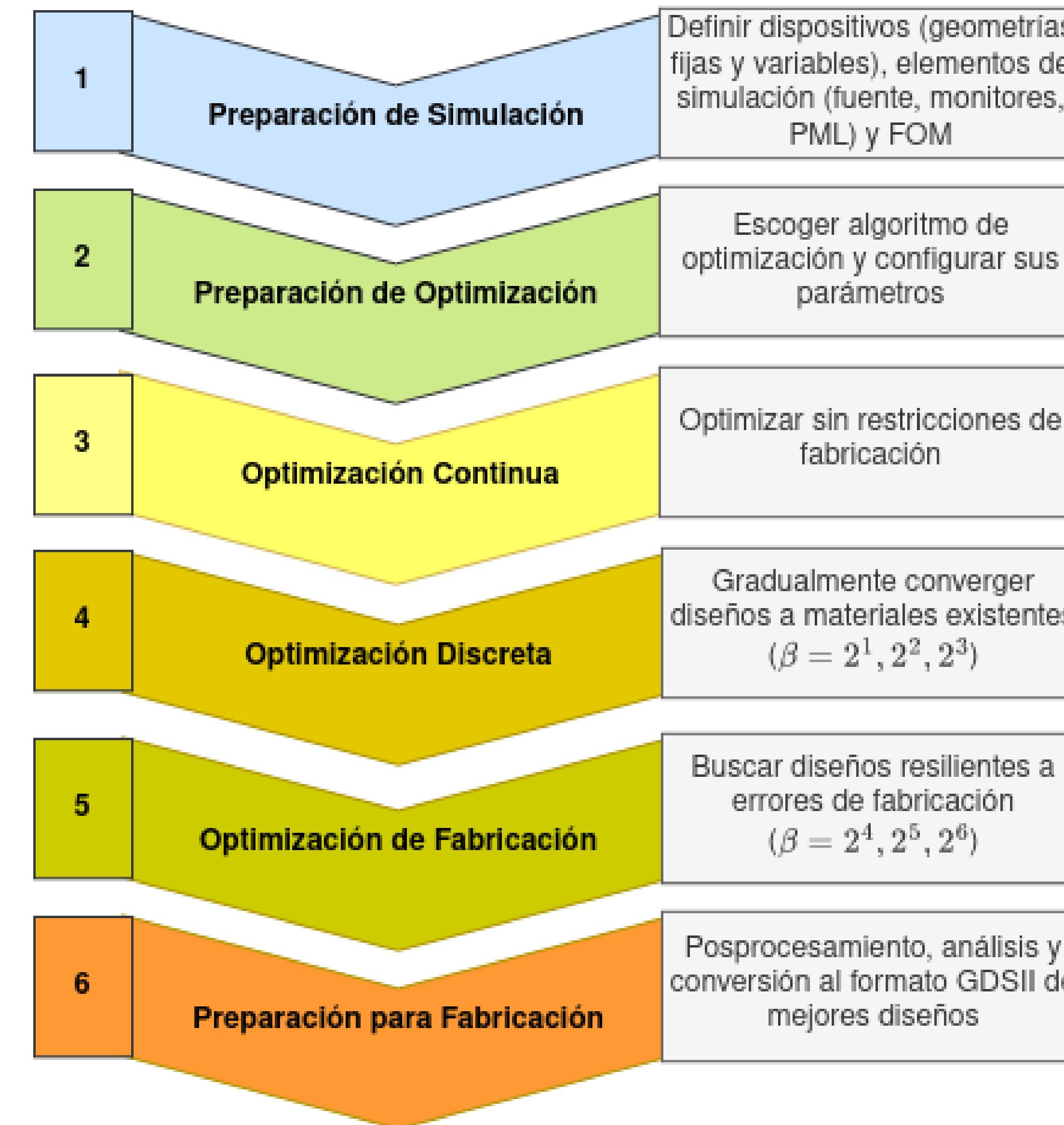
Revisión de la literatura

Metodología

Resultados

Conclusiones

Metodología

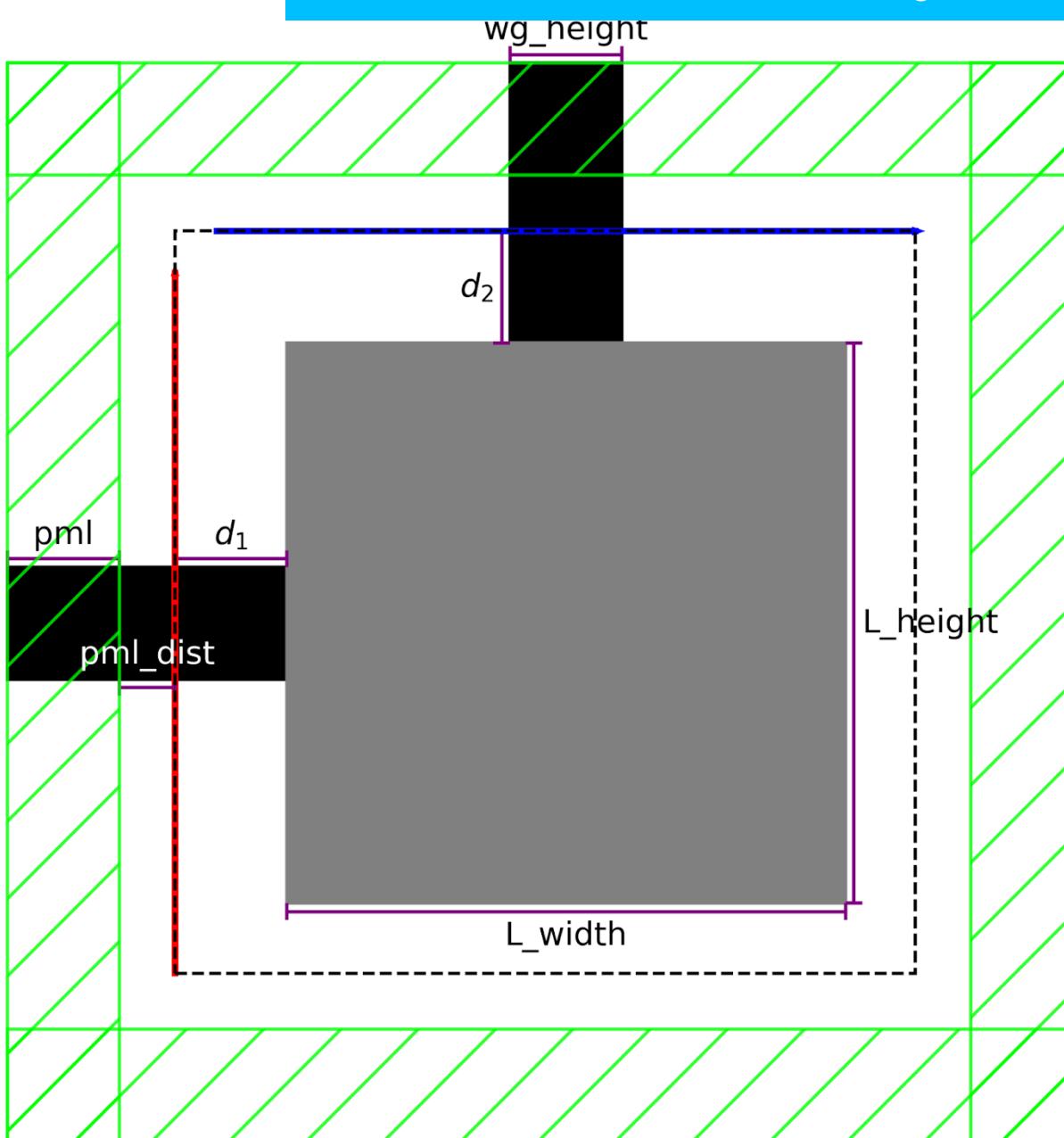


1. Preparación de simulación

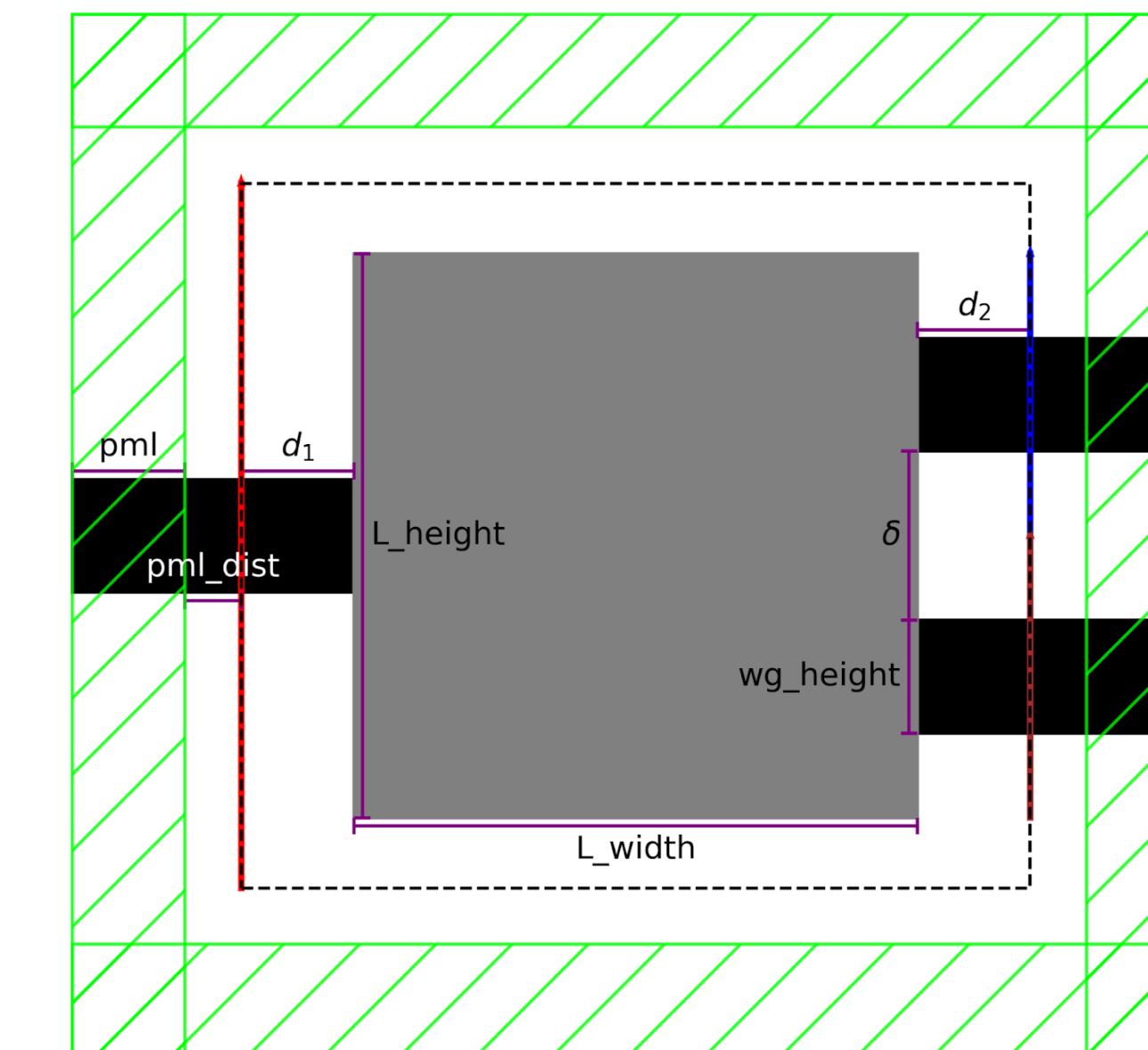
Definimos los dispositivos (*bend* y WDM) en SPINS-B

- Region gris: región de optimización
 - Región negra: región fija
 - Recta roja: fuente
- Recta azul y marrón: monitores
 - Región verde: PML

Se fijó la resolución a 40nm [Su+, 2020]

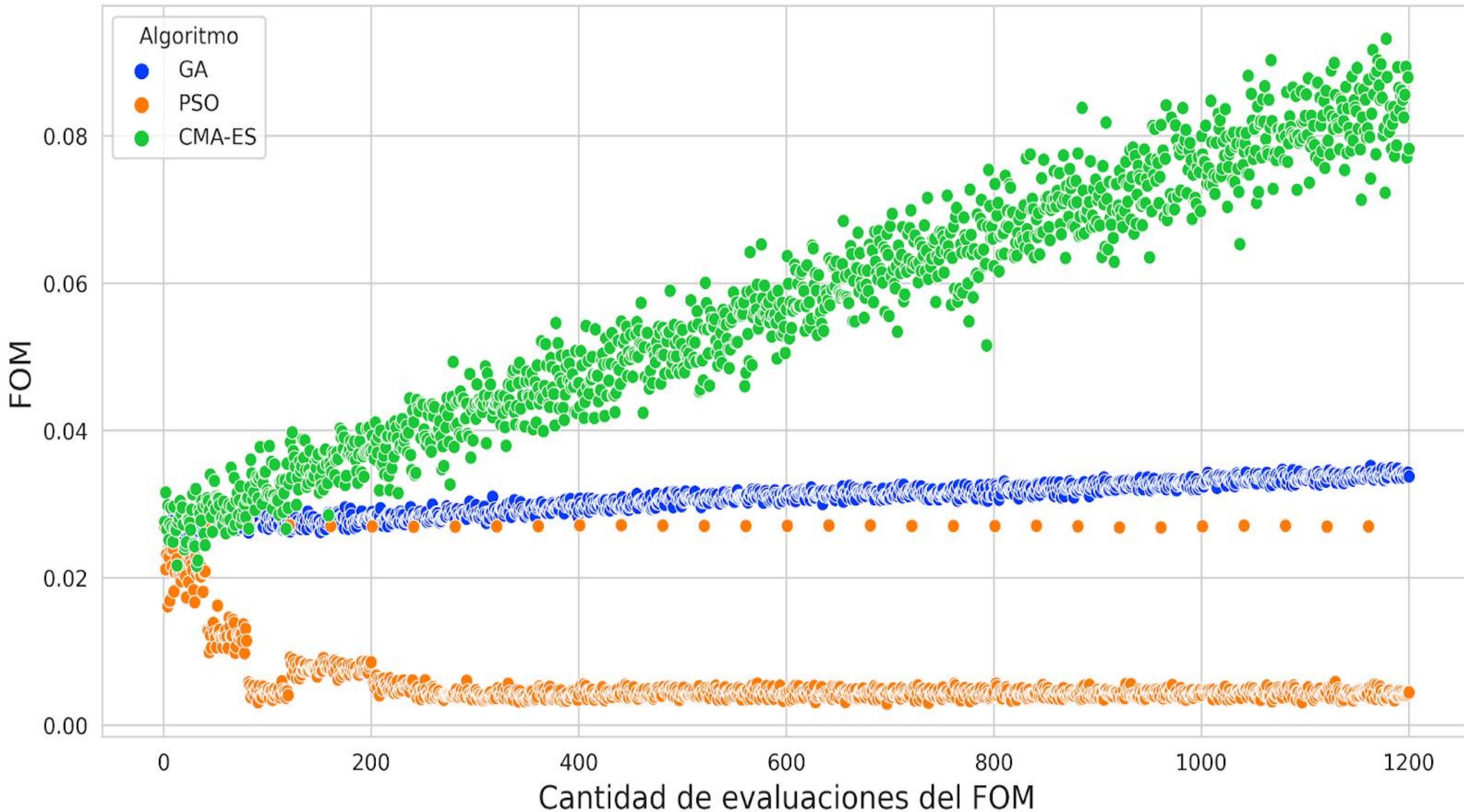


I) *Bend*



II) WDM

2. Preparación de optimización



Los algoritmos *gradient-free* mostraron valores de FOM bajos (<0.1).

Idea: Usar la información de la gradiente para guiar sus búsquedas

Algoritmos a evaluar

1. G-PSO
2. G-GA
3. G-CMA-ES
4. L-BFGS-B
5. MMA

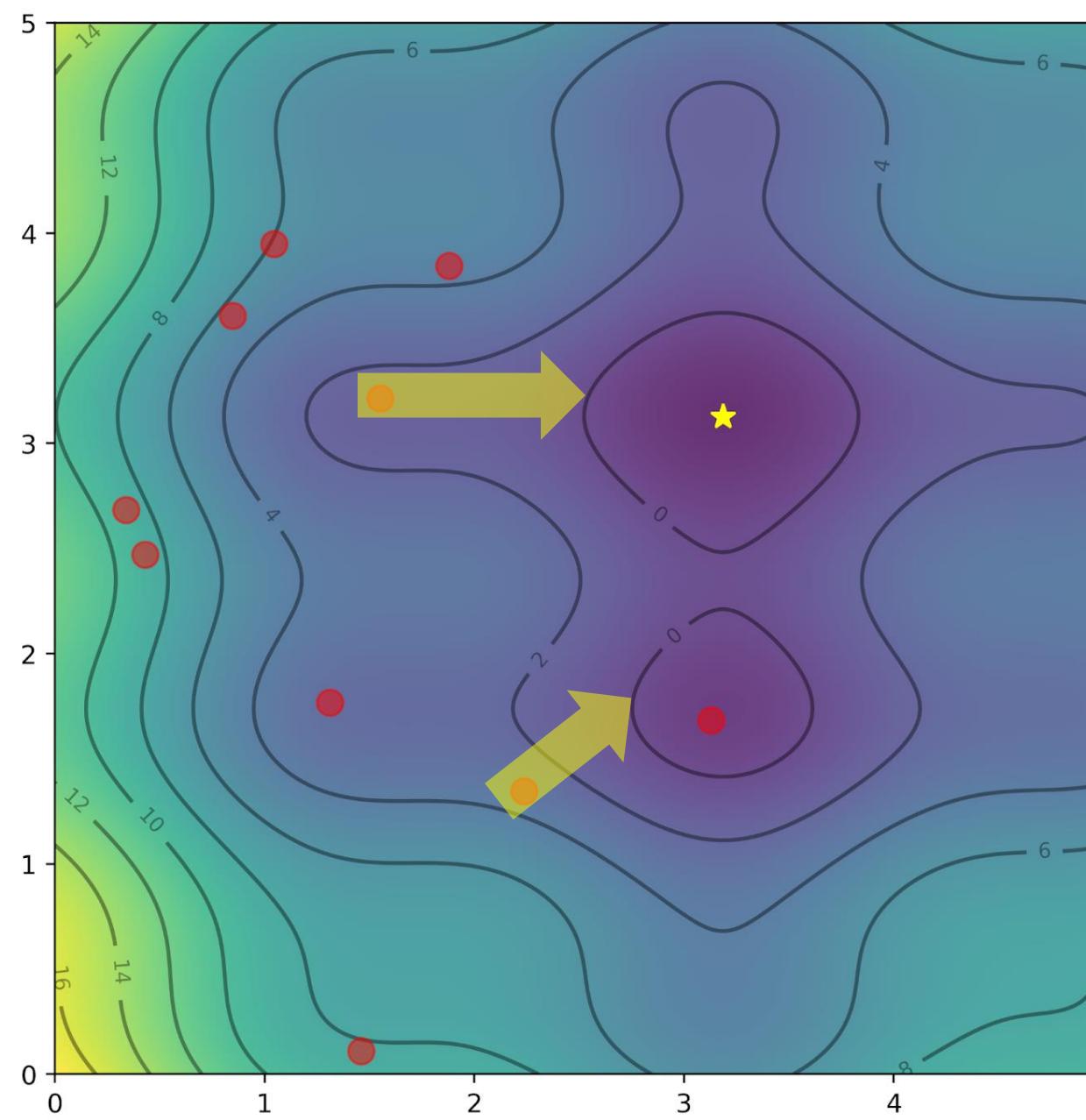
Reproducibilidad

1. seed = 128
2. seed = 256
3. Seed = 512

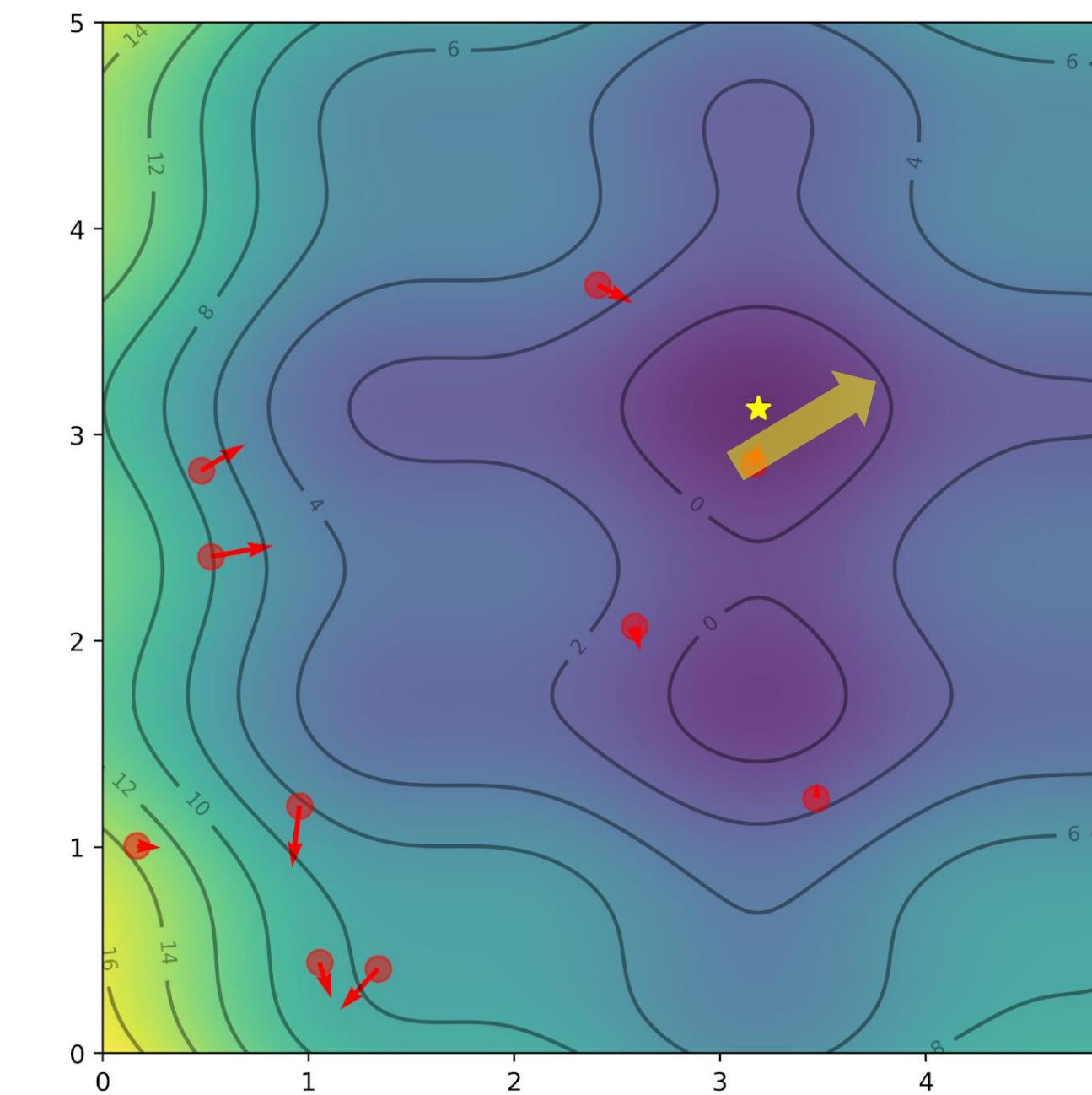
Algoritmos free-gradient-based

$$g(x, y) = (x - 3.14)^2 + (y - 2.72)^2 + \sin(3x + 1.41) + \sin(4y - 1.73).$$

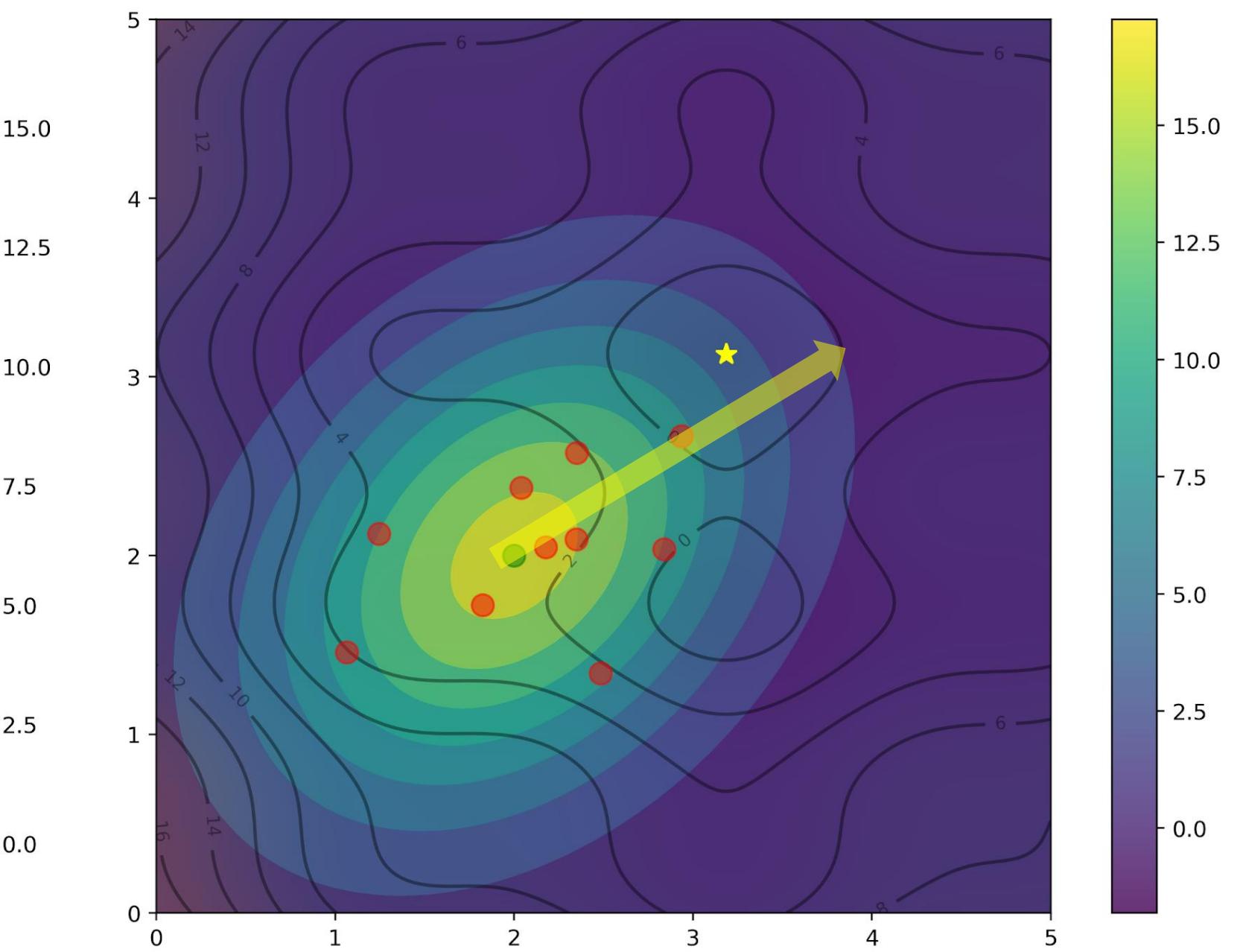
G-GA



G-PSO



G-CMA-ES



$$prob_i = \frac{\max(f) - f^{(i)}}{\sum_j \max(f) - f^{(j)}},$$

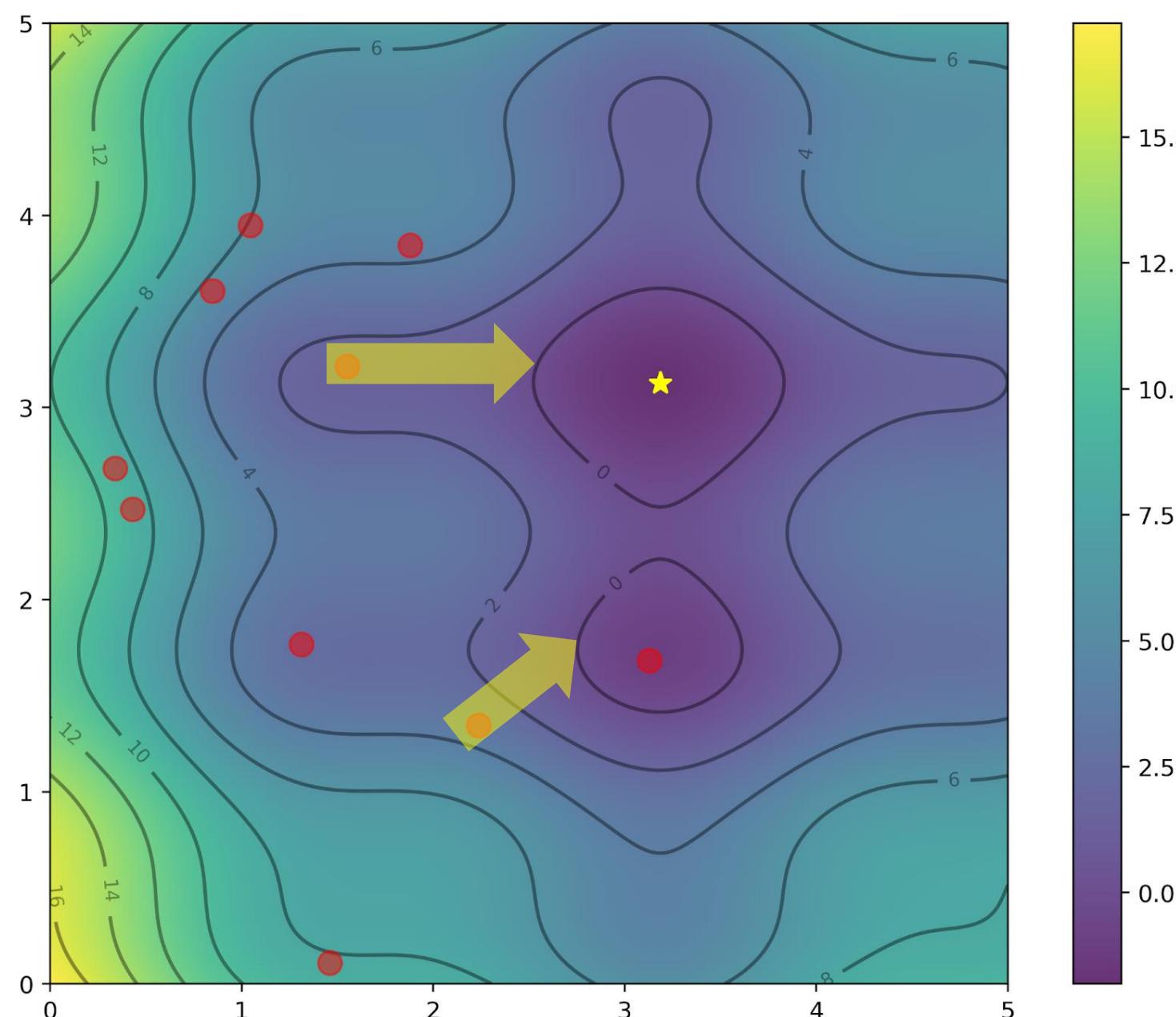
$$\begin{aligned} \mathbf{P}^{(i)} &\leftarrow \mathbf{P}^{(i)} + \mathbf{V}^{(i)}, \\ \mathbf{V}^{(i)} &\leftarrow \omega \mathbf{V}^{(i)} + c_1 r_1 (\mathbf{P}_b^{(i)} - \mathbf{P}^{(i)}) + c_2 r_2 (\mathbf{P}_b - \mathbf{P}^{(i)}) \end{aligned}$$

$$\mathcal{N}(\mu, \sigma^2 \mathbf{C})$$

Algoritmos free-gradient-based

$$g(x, y) = (x - 3.14)^2 + (y - 2.72)^2 + \sin(3x + 1.41) + \sin(4y - 1.73).$$

G-GA



Algorithm 1: Genetic Algorithms (GA)

Data: $P, population_size, GA_range, n_selected_parents, prob_mutation$

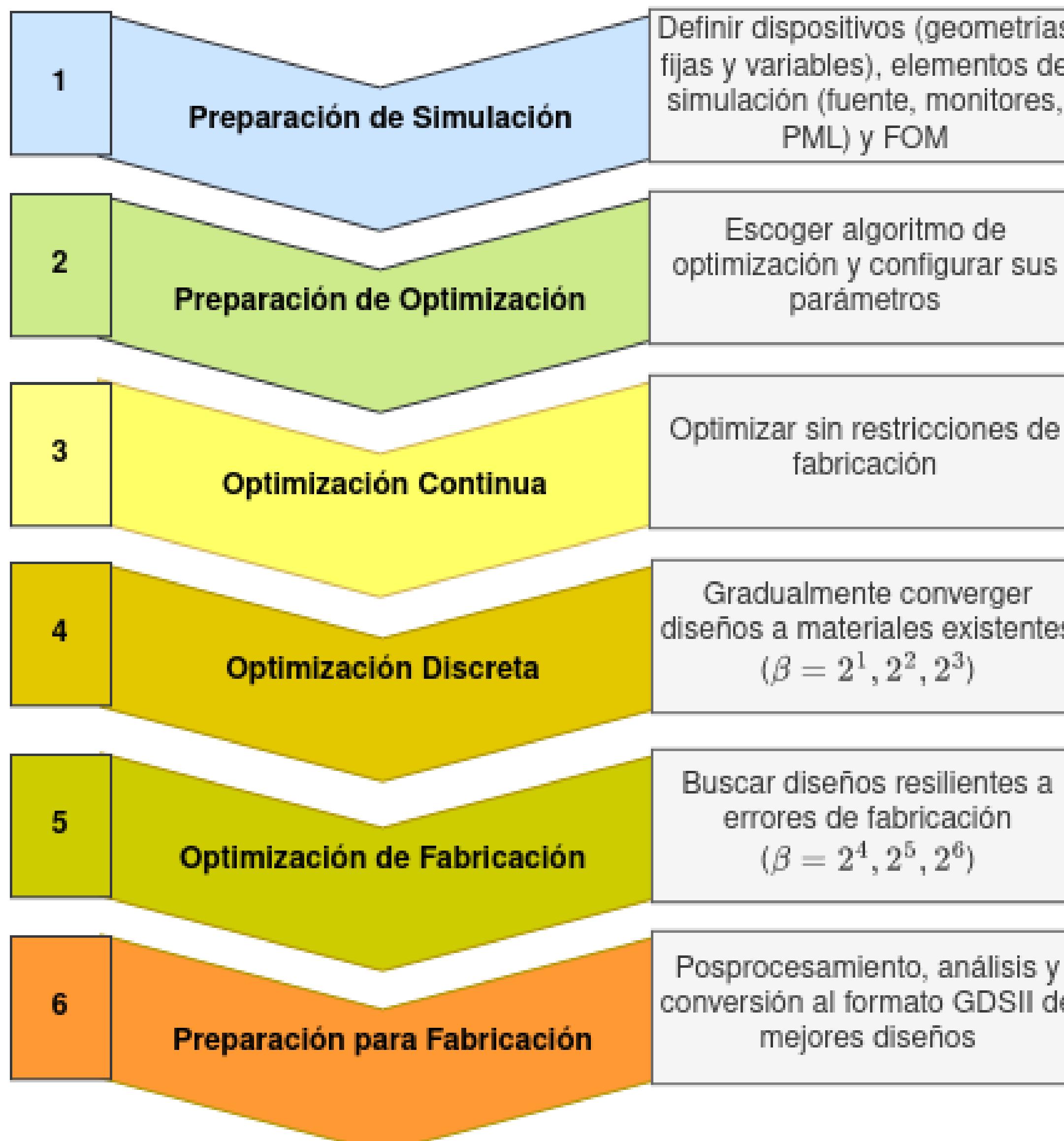
Result: $\min(population)$

- 1 $population = generate_population()$
- 2 **for** $t = 0; t < k; t++$ **do**
- 3 $parents = select(population)$
- 4 $children = crossover(parents)$
- 5 $population = mutation(children)$

Aquí se realiza la ejecución de GD para dos agentes aleatorios

$$prob_i = \frac{\max(f) - f^{(i)}}{\sum_j \max(f) - f^{(j)}},$$

3,4,5. Optimización



$$\frac{\partial f_{obj}}{\partial \mathbf{P}(x,y)} = \sum_{(x',y') \in \bar{B}_{r_f}(x,y)} \frac{\partial f_{obj}}{\partial \tilde{\mathbf{P}}(x',y')} \frac{\partial \tilde{\mathbf{P}}(x',y')}{\partial \tilde{\mathbf{P}}(x',y')} \frac{\partial \tilde{\mathbf{P}}(x',y')}{\partial \mathbf{P}(x,y)}. \quad (2.12)$$

Nº evaluaciones

2000

$$\mathbf{P} \rightsquigarrow \varepsilon$$

$$\mathbf{P} \rightsquigarrow \tilde{\mathbf{P}} \rightsquigarrow \tilde{\tilde{\mathbf{P}}} \rightsquigarrow \varepsilon \quad r_f = 80nm$$

3x800

$$F_{obj} = \max(\min(f_{obj}(\tilde{\mathbf{P}}_d), f_{obj}(\tilde{\mathbf{P}}_i), f_{obj}(\tilde{\mathbf{P}}_e)))$$

3x400

Contribuciones

Se propuso una **variante de GA y PSO**

Se **adaptó** el paquete de python **SPINS-B** para que funcione correctamente con SLURM
(cluster Khipu)

Se **extendieron funcionalidades** de SPINS-B.

- Algoritmos de optimización
 - Visualización
 - Generación de GDS

Se generaron **comparativas de cinco algoritmos de optimización** para dispositivos nanofotónicos

Agenda

Introducción

Marco Teórico

Revisión de la literatura

Metodología

Resultados

Conclusiones

Configuración de los experimentos

Maxwell-B (servidor)
[GPU]

SPINS-B (cliente)
[CPU]

Cluster Khipu

- 1. Nodo GPU g001: Tesla T4, 16 GB
- 2. Nodo GPU ag001: A100, 40 GB

Laboratorio deñ CEG
Quadro RTX, 8GB

Tiempo de ejecución

Tiempo promedio (s)	Quadro RTX	Tesla T4	Ampere A100
Optimización continua (<i>bend</i>)	14.261	15.432	-
Optimización discreta (<i>bend</i>)	15.961	18.718	-
Optimización de fabricación (<i>bend</i>)	47.084	50.639	-
Optimización continua (WDM)	16.876	-	17.479
Optimización discreta (WDM)	18.431	-	19.780
Optimización de fabricación (WDM)	53.941	-	55.406

Tiempo promedio de simulación en cada etapa de optimización para el *bend* y WDM con los tres sistemas de cómputo usados

- Usando el Quadro RTX optimizar el *bend* podía tomar hasta 34 horas
- Usando A100 optimizar un WDM podía tomar hasta 41 horas.
- Solo una simulación a la vez podía usar los recursos GPU.

Relación de orden

I) Bend

Desempeño	Convergencia	M_{nd}
L-BFGS-B (0.9895)	MMA (165)	L-BFGS-B (1.157 %)
G-PSO (0.9830)	L-BFGS-B (566)	G-PSO (1.519 %)
G-GA (0.9744)	G-PSO (3094)	G-GA (2.062 %)
G-CMA-ES (0.9155)	G-GA (3117)	G-CMA-ES (5.459 %)
MMA (0.0383)	G-CMA-ES (3320)	MMA (23.958 %)

Se desea:

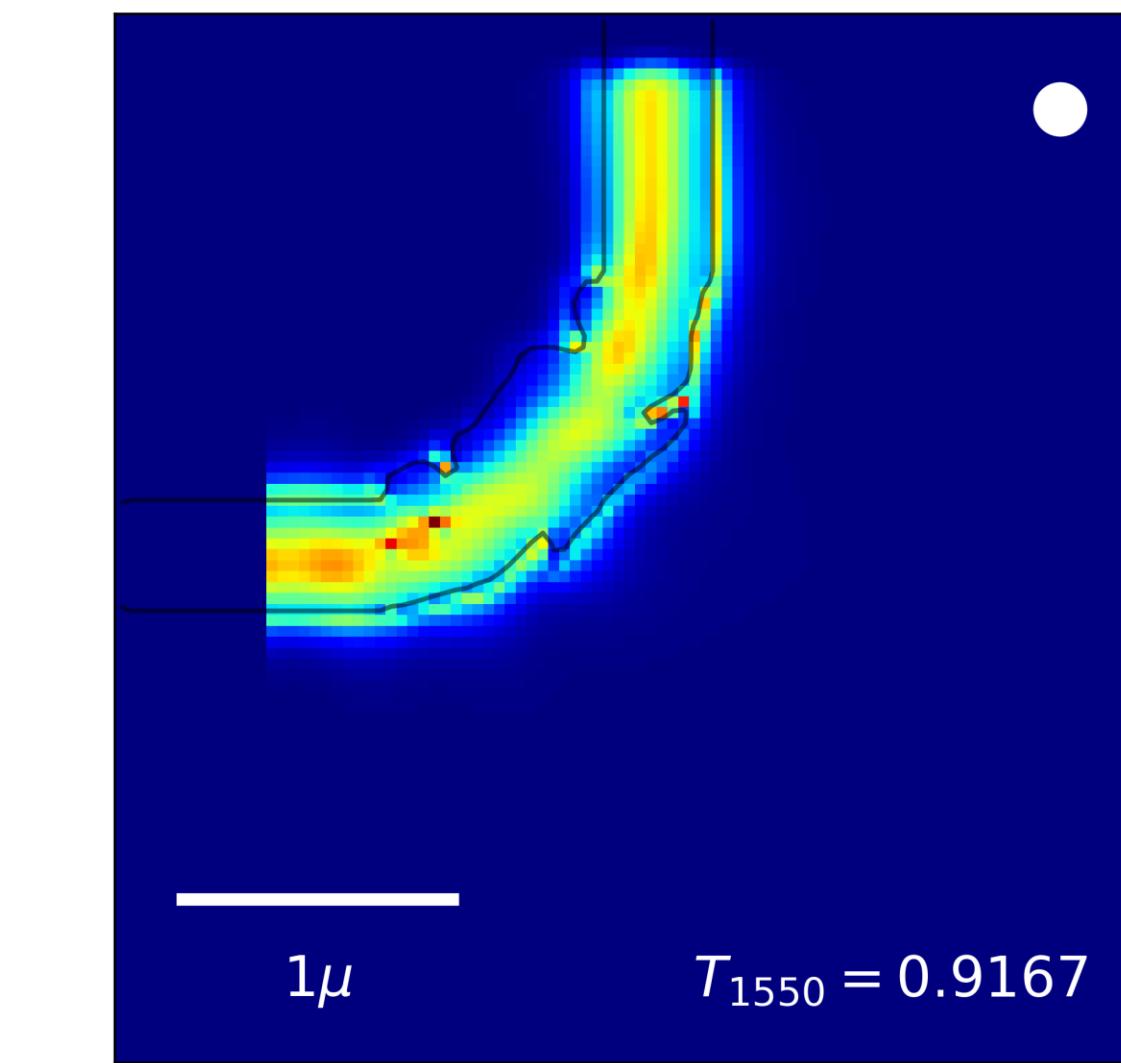
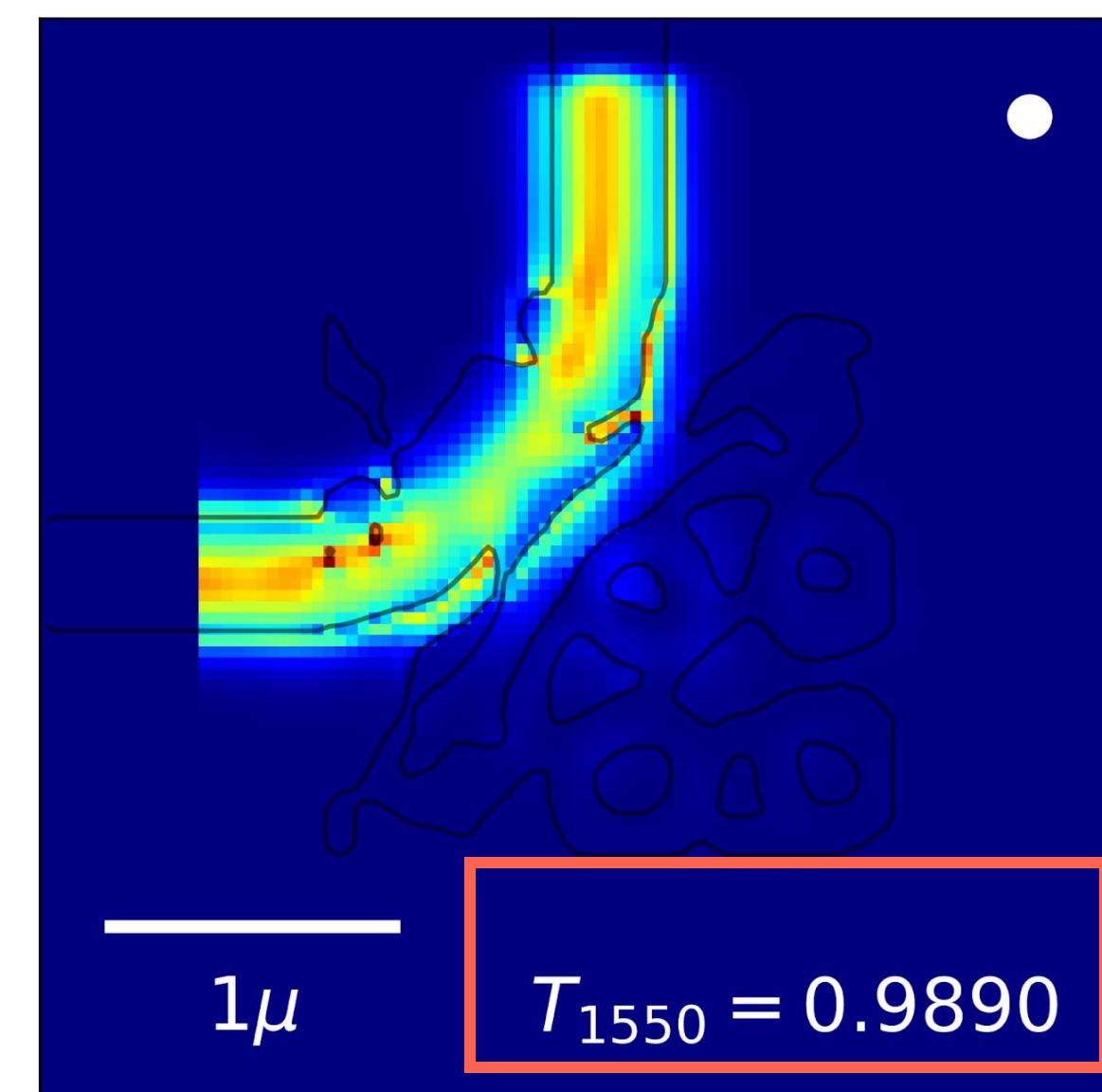
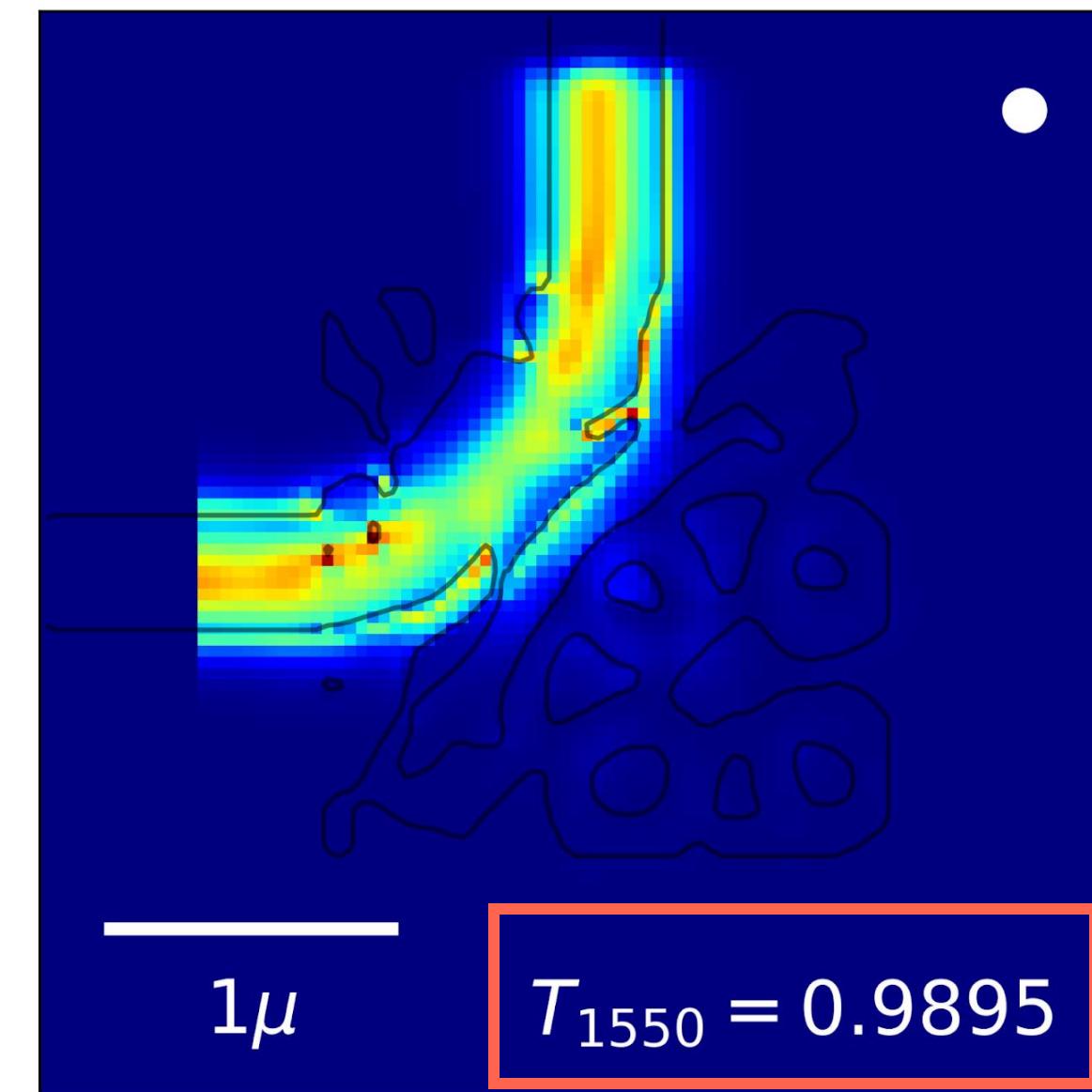
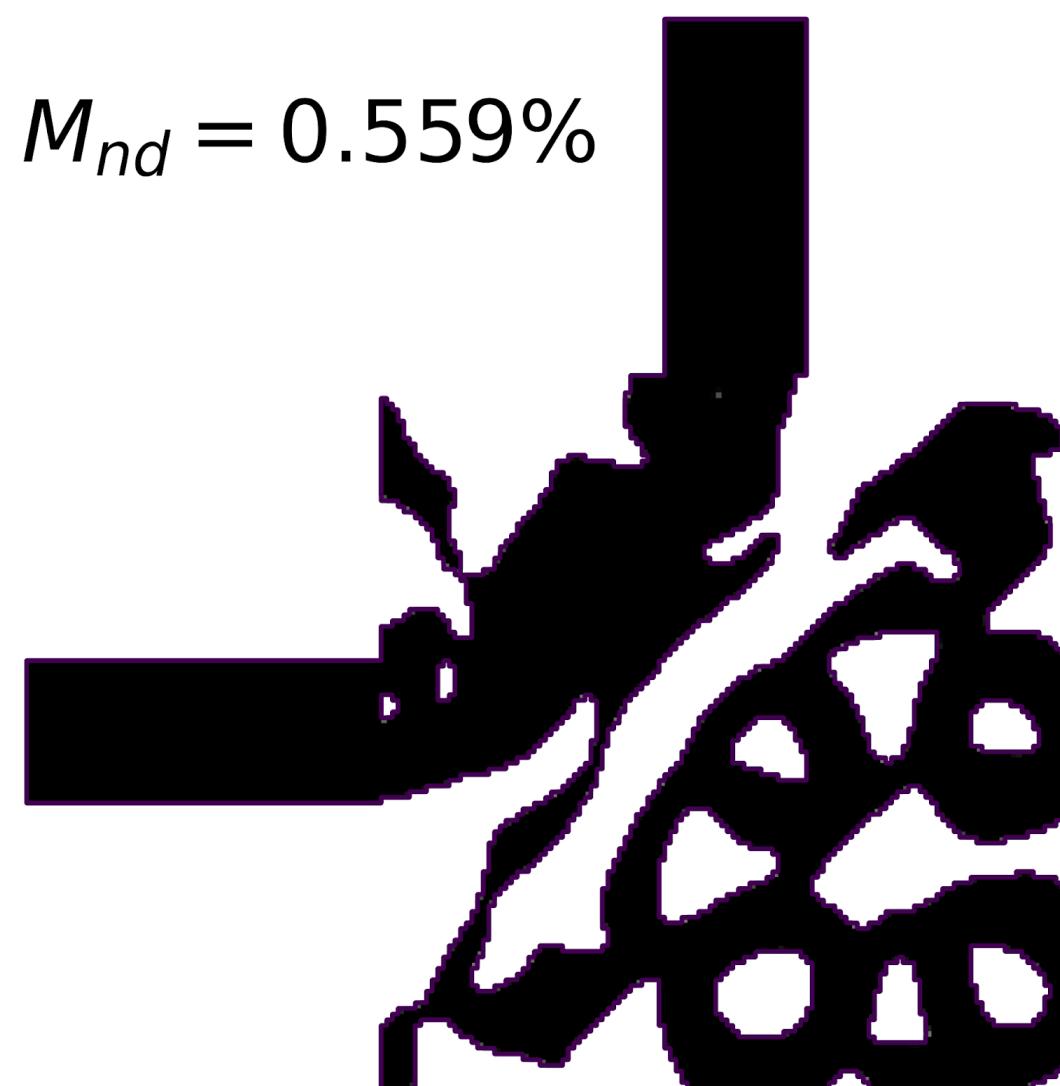
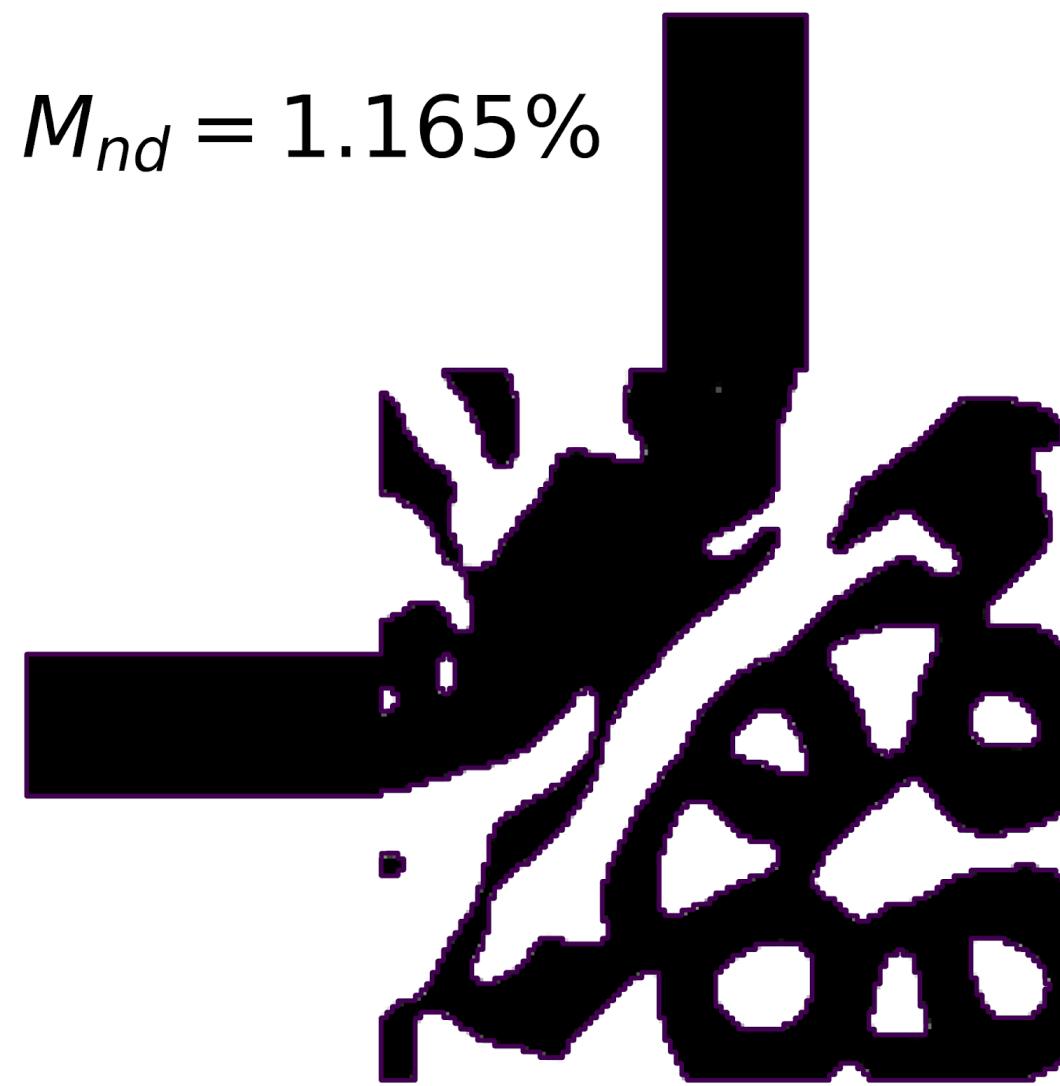
- ↑ desempeño
- ↓ convergencia
 - ↓ M_{nd}

II) WDM

Desempeño	Convergencia	M_{nd}
L-BFGS-B (0.9465)	MMA (168)	L-BFGS-B (1.237 %)
G-PSO (0.8005)	L-BFGS-B (946)	G-PSO (2.224 %)
G-GA (0.7176)	G-GA (2137)	G-GA (4.300 %)
G-CMA-ES (0.6427)	G-PSO (3103)	G-CMA-ES (7.977 %)
MMA (0.4594)	G-CMA-ES (3320)	MMA (23.898 %)

Para el *bend* y WDM se sigue la misma relación de orden en el criterio de desempeño, convergencia y M_{nd} . La única diferencia está encerrada en el recuadro rojo.

Bend mejor optimizado (L-BFGS-B, seed=256)

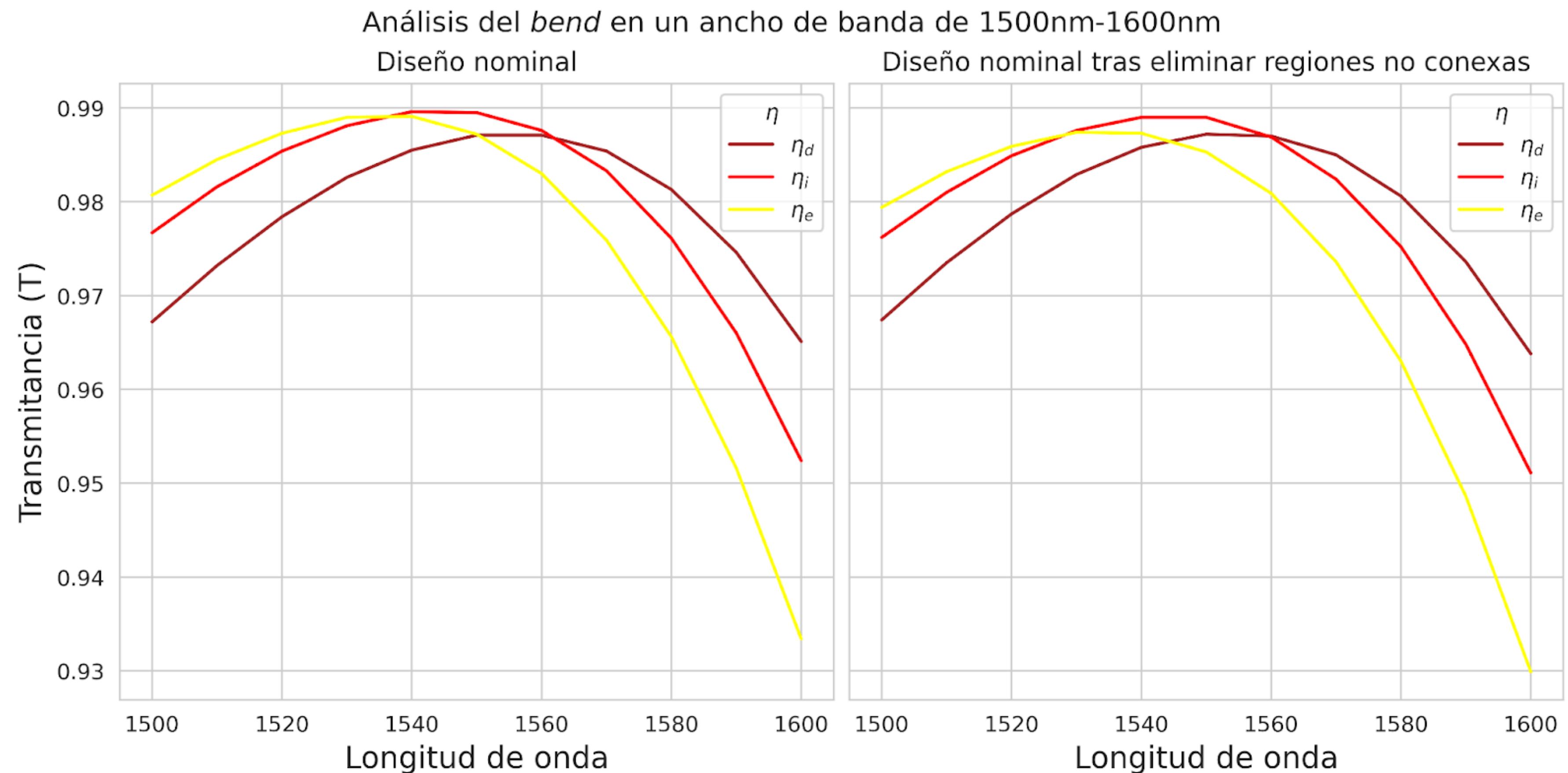


El diseño intuitivo en un área de diseño de $2\mu \times 2\mu$ posee un valor de transmitancia de 0.8399.

Bend mejor optimizado (L-BFGS-B, seed=256)

Máxima diferencia de transmitancia

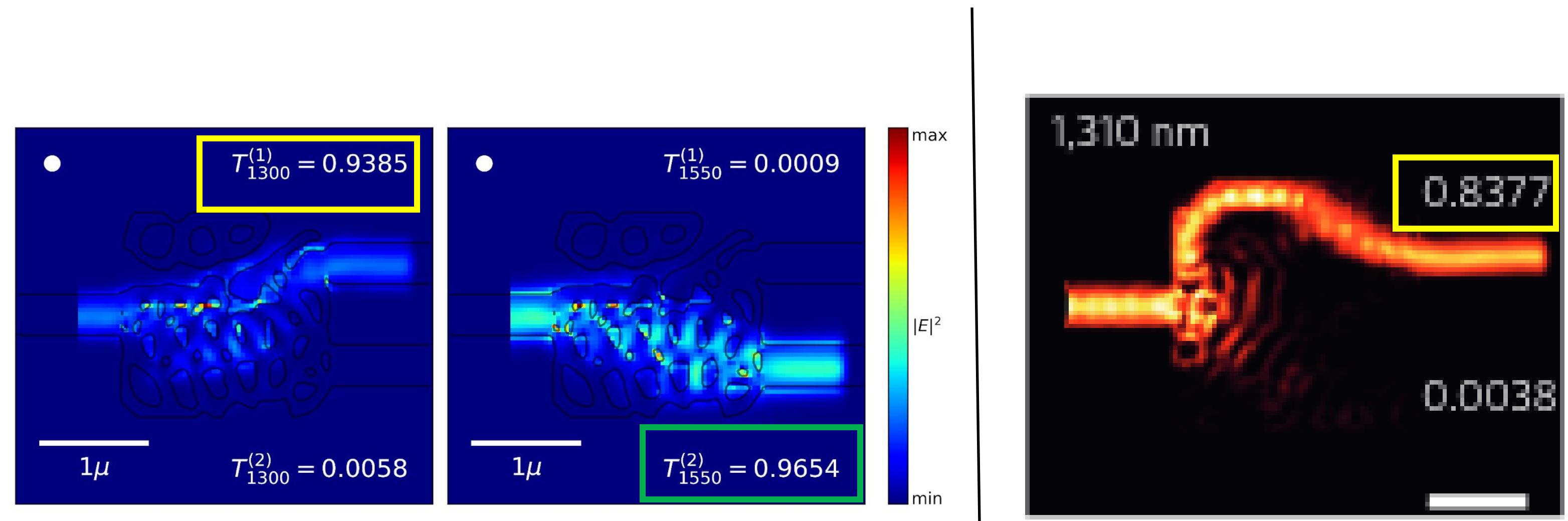
- Diseño nominal: 0.032
- Posprocesamiento: 0.034



WDM mejor optimizado (L-BFGS-B, seed=128)

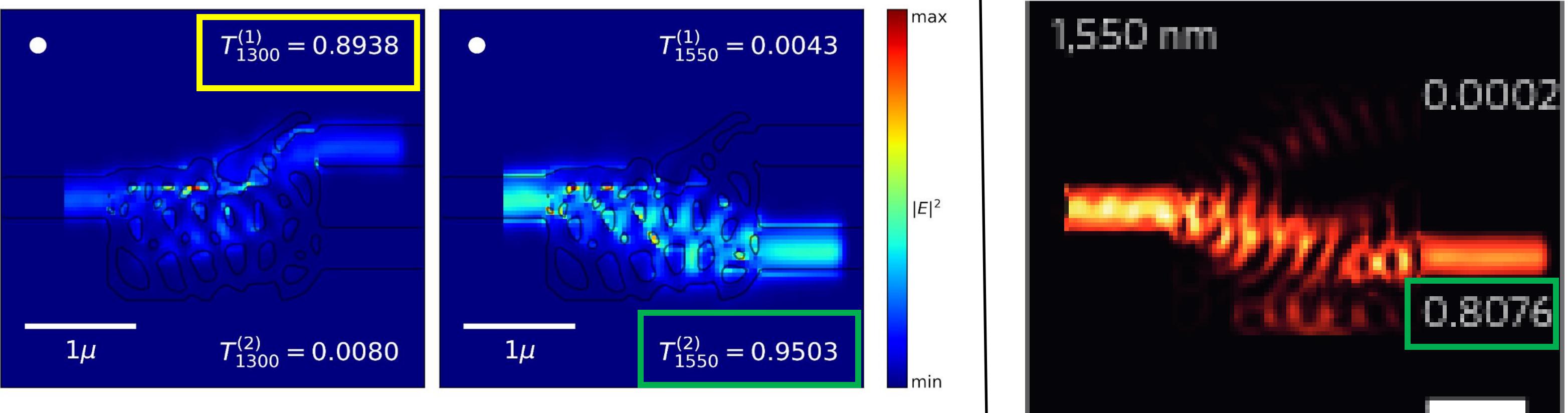
Diseño nominal

$$M_{nd} = 1.233\%$$



$$M_{nd} = 0.469\%$$

Post.



El diseño de Piggott+ use un área de diseño de $2.8\mu \times 2.8\mu$ y obtiene un dispositivo menos eficiente que el conseguido en este trabajo en $2.0\mu \times 2.0\mu$.

[Piggott+, 2015]

WDM mejor optimizado (L-BFGS-B, seed=128)

Máxima diferencia de transmitancia

- Diseño nominal

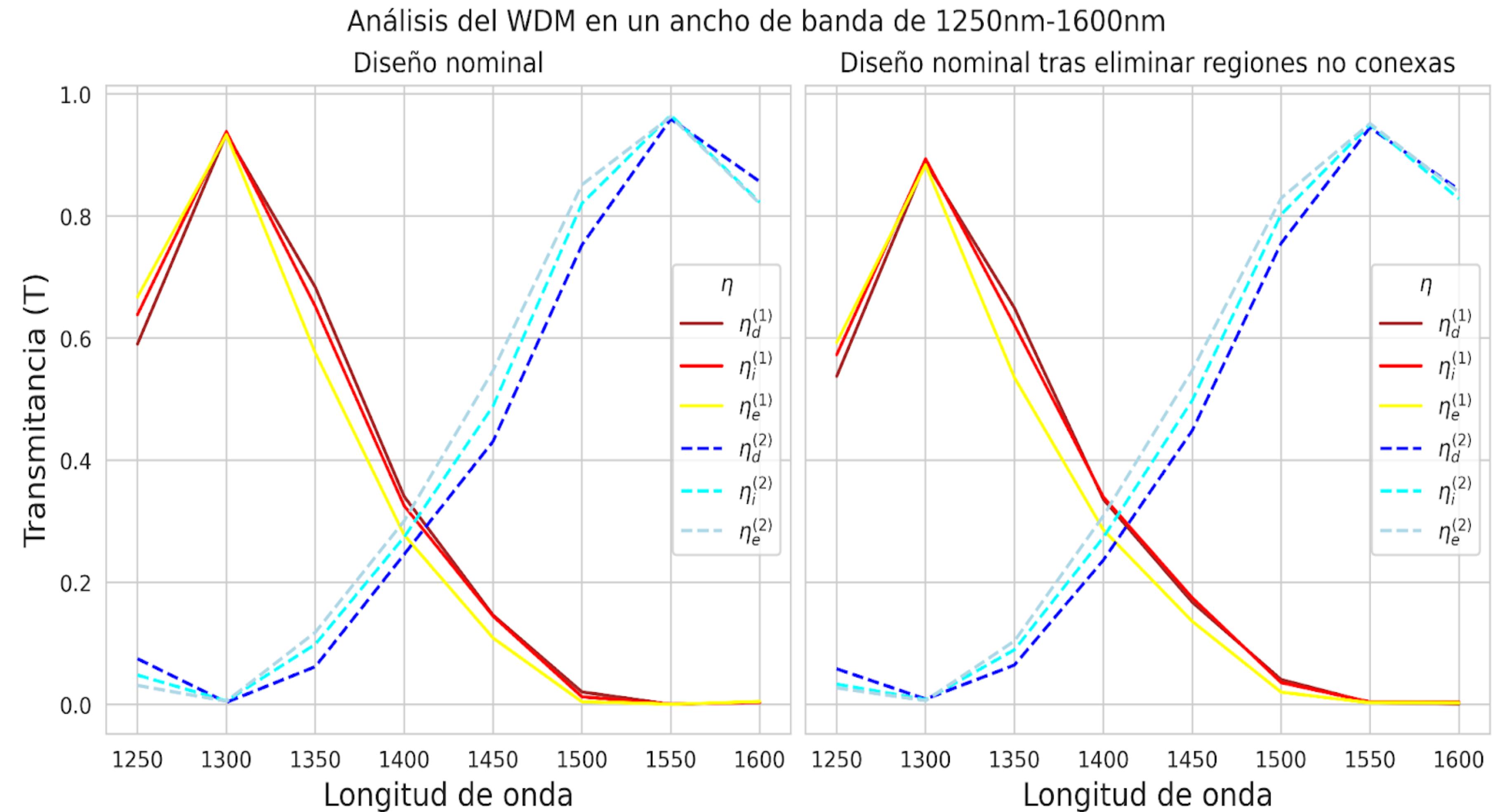
- (1) 0.108

- (2) 0.116

- Posprocesamiento:

- (1) 0.114

- (2) 0.098



Agenda

Introducción

Marco Teórico

Revisión de la literatura

Metodología

Resultados

Conclusiones

Conclusiones

- La estrategia de optimización muestra resultados promisorios para la optimización del *bend* y WDM.
- Usar algoritmos de primer orden para la optimización topológica robusta:
 $\text{MMA} < \text{G-CMA-ES} < \text{G-GA}$, $\text{G-PSO} < \text{L-BFGS-B}$.
- Función objetivo: *bend* (>90%), WDM (>90%).
- Se requiere un mayor estudio del MMA para este caso (*bend* y WDM).
- Comenzar futuras investigaciones usando L-BFGS-B.

Futuros trabajos

- Fabricación de los diseños.
- Imponer restricciones de conectividad. Posibles opciones:
 - Restricciones de energía [Zhang+, 2021].
 - Restricciones de temperatura [Li+, 2016].
- Entrenar una red para predecir los resultados de simulación o incluso usar la red para predecir diseños óptimos [Liu+, 2018] [Peurifoy+, 2018].

Gracias

¿Preguntas?