

# Capítulo 5

## Optimización Topológica de Estructuras Usando un Algoritmo de Estimación de Distribución Paralelo, Basado en el Muestreador de Gibbs

Noé Faurrieta Ortíz<sup>1</sup>  
Sergio Ivvan Valdez Peña<sup>2</sup>  
Salvador Botello Rionda<sup>3</sup>

---

**Abstract:** We introduce a proposal to solve the structural optimization problem. The problem is to find the structure design that minimizes a cost function that depends on weight, displacements in some points and stress level. The global optimization is performed via an Estimation of Distribution Algorithm (EDA), while a local optimization procedure is applied to each of the solutions proposed by the EDA. The EDA requires to learn and sample from a probability distribution. In this vein, the Gibbs Sampler is used to generate samples of the joint distribution of the design variables. The evaluation of each individual in the population of the EDA involves performing a von Mises stress analysis, which is calculated by the Finite Element Method (FEM), considering the weight of the structure. Due to the computational cost and the characteristics of the algorithm, the evaluation of the function is parallelized in distributed memory by using Message Passing Interface (MPI). Finally, some examples of literature problems in structural optimization are presented. The results show a considerable weight reduction.

---

<sup>1</sup>noe.faurrieta@cimat.mx. Centro de Investigación en Matemáticas (CIMAT).

<sup>2</sup>ivvan@cimat.mx. Centro de Investigación en Matemáticas (CIMAT), Apdo. Postal 402, 36000 Guanajuato, Gto. México

<sup>3</sup>botello@cimat.mx. Centro de Investigación en Matemáticas (CIMAT), Apdo. Postal 402, 36000 Guanajuato, Gto. México

**Keywords:** Von Mises, Gibbs Sampler.

**Resumen:** El presente trabajo es una propuesta para resolver el problema de optimización estructural, que consiste en crear el diseño de una estructura que minimice una función de costo que depende del peso, desplazamientos y nivel de estrés de la estructura. La propuesta combina la optimización local y global utilizando un Algoritmo de Estimación de Distribución (EDA), se utiliza el muestreador de Gibbs para obtener muestras de la distribución conjunta de las variables de diseño. La evaluación de cada individuo de la población del EDA implica realizar un análisis de esfuerzos von Mises, que es calculado por el Método de Elementos Finitos (FEM) considerando el peso propio de las estructuras. Debido al costo computacional y a las características del algoritmo, se paraleliza la evaluación de la función utilizando Message Passing Interface (MPI). Por último, se presentan resultados con problemas de optimización estructural en la literatura actual, donde los resultados obtenidos muestran una reducción considerable del peso.

*Palabras clave:* Von Mises, Gibbs Sampler.

## 5.1 Introducción

La optimización de estructuras hace referencia al problema de encontrar el diseño de una estructura óptima que minimice cierta función objetivo; además, la estructura debe de satisfacer un conjunto de restricciones, tales como: no sobrepasar cierto desplazamiento, el máximo esfuerzo permitido por el material, etc.

Existen diferentes técnicas para obtener una posible solución de un problema de diseño, los cuales pueden ser métodos analíticos, numéricos y empíricos [Martin2004]. Los métodos numéricos poseen la mayor relevancia para este trabajo y se pueden dividir en tres categorías: optimización de dimensiones, optimización de forma y optimización topológica. La optimización topológica es uno de los métodos numéricos más utilizados para obtener estructuras con una distribución de material óptima. En este tipo de optimización, el objetivo es distribuir el material en el dominio para encontrar la estructura. Se pueden introducir huecos o cavidades que en un principio no estaban presentes. Inicialmente sólo se cuenta con la información de las condiciones de carga y las restricciones, así como el dominio inicial donde se desarrollará la estructura.

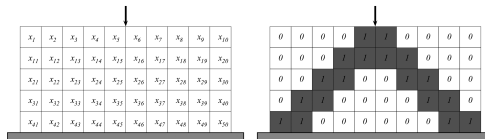
Algunos de los métodos que se han desarrollado realizan una minimización del peso con restricciones en la tensión de los elementos. Este tipo de algoritmos se basa en la utilización de diferentes mallas con los que se hace un refinado

de la solución que se aproxima [Yiqiang2014]. Otros métodos realizan una optimización en la que sólo se modifican los contornos de la geometría. Una vez realizado esto, se procede a introducir un agujero, de tal modo que las tensiones sean mínimas y se repite este proceso hasta haber introducido un número máximo de agujeros o el radio de los agujeros sean mínimos [Ullah2013]. También existen los algoritmos en los que se va eliminando material de manera progresiva, se eliminan los elementos en los cuales no se tiene cierto nivel de esfuerzo, el cual generalmente es el esfuerzo von Mises [Querin1998, SAVAS]. Otros algoritmos que se han utilizado son los algoritmos genéticos, en los que se realiza una optimización en base a ciertas heurísticas y representaciones [Ruiyi2009]. En nuestro caso, realizamos optimización topológica, se utilizan los esfuerzos von Mises como medida de factibilidad, además también los utilizamos en un proceso de optimización local, y para la creación del diseño de las estructuras utilizamos algoritmos de estimación de distribución (EDA).

## 5.2 Definición del problema

### 5.2.1 Descripción del problema

En la Figura 5.1 se muestra un ejemplo de discretización de una placa y la representación binaria de una estructura. La placa es discretizada por FEM [Oñate1992], cada elemento en la discretización es representado en un arreglo  $x$  y cada una de sus componentes  $x_i \in \{0, 1\}$  representan la ausencia o presencia del elemento en la estructura. Por tanto, el problema de optimización que se intenta resolver es de orden  $2^n$ , donde  $n$  es el número de variables de diseño o elementos en la discretización.



**Figura 5.1:** Ejemplo de problema de optimización estructural discretizado y representación de estructura en binario.

La forma en que se mide el desempeño de las estructuras de este tipo se describe en la siguiente sección.

### 5.2.2 Función objetivo

El objetivo principal es encontrar estructuras con un peso mínimo. La estructura no debe de sobrepasar cierto desplazamiento en los nodos que se le aplican cargas y tampoco se debe de sobrepasar el máximo esfuerzo permitido por el material,

ya que de ser así la estructura podría llegar a romperse. Estas características se consideran por medio de la ecuación (5.2.1).

$$f(x) = \begin{cases} (VM(x) + 1)(D(x) + 1) & \text{si } VM(x) > 0 \\ & \text{o } D(x) > \epsilon \\ \frac{W(x)}{W_{max}} + C(x) & \text{e.o.c.} \end{cases} \quad (5.2.1)$$

donde  $W(x)$  indica el peso de la estructura dado por:

$$W(x) = \sum_{i=1}^n w_i x_i, \quad (5.2.2)$$

siendo  $w_i$  el peso de cada uno de los elementos y  $W_{max}$  el peso máximo de la estructura.  $VM(x)$  indica el exceso del esfuerzo von Mises [Akin2005] en la estructura dado por la ecuación (5.2.3), donde  $\hat{\gamma}_i$  es el exceso de esfuerzo presente en el elemento  $i$ -ésimo.

$$VM(x) = \sum_{i=1}^n x_i \hat{\gamma}_i, \quad (5.2.3)$$

Otra de las consideraciones a tomar en cuenta, es que el desplazamiento sea mínimo en los nodos en los cuales se aplican cargas. Estos excesos se calculan por medio de la ecuación (5.2.4).

$$D(x) = \sum_{i=1}^m |\delta_i|, \quad (5.2.4)$$

donde  $m$  pertenece al conjunto de nodos con cargas,  $\delta_i$  es el exceso de desplazamiento en el nodo y  $\epsilon$  es el valor de desplazamiento máximo permitido en la estructura.

Por último  $C(x)$  representa una penalización en la configuración de la estructura dada por la ecuación (5.2.5), que se describirá más a detalle en las Sección 5.3.1.

$$C(x) = \frac{\beta}{4n} \sum_{C:\epsilon(x)} V_C, \quad (5.2.5)$$

siendo  $\beta$  constante.

## 5.3 EDA

Los Algoritmos de Estimación de Distribución (EDA), son métodos de optimización estocásticos [Larrañaga2001], los cuales realizan la búsqueda del óptimo mediante la construcción y el muestreo de modelos probabilísticos. Utilizan la distribución de probabilidad para la generación de la nueva población, que puede ser continua, discreta, univariada o multivariada [Larrañaga2001] y se ha aplicado tanto a problemas de optimización discreta como continua [Larrañaga1999].

Los pasos de un EDA se muestran en el algoritmo siguiente.

---

### Algoritmo 1 EDA estándar

---

**Require:**  $t = 0$ .

- 1: Generar una muestra  $X_t$  de  $p(x, t)$ .
  - 2: Evaluar población  $X_t$ .
  - 3: Seleccionar un subconjunto  $S_t$  de  $X_t$ .
  - 4: Calcular modelo  $p(x, t + 1)$  mediante la aproximación de la distribución subyacente  $\hat{p}^S(x, t)$  de  $S_t$ .
  - 5: Si no se alcanza el criterio de paro, hacer  $t = t + 1$  y regresar a paso 1, en caso contrario terminar.
- 

Entre el paso 4 y 5 comúnmente se selecciona un subconjunto del conjunto seleccionado al que se le denomina conjunto *Elite*, esto se realiza con la finalidad de garantizar que los mejores individuos no desaparecen con la evolución de la distribución.

Uno de los criterios de paro más comunes en estos algoritmos es haber realizado un número máximo de generaciones. Otro criterio es la medida de variabilidad de la distribución calculada por la ecuación (5.3.1) (que llamaremos residuo) sea mínima, esta medida se deduce del llamado *ordering parameter*, usado algunas veces en EDAs para medir convergencia [Kvasnicka1996, Pelikan1999].

$$\chi(p) = \frac{n}{4} - \sum_{i=1}^n \left(p_i - \frac{1}{2}\right)^2, \quad (5.3.1)$$

donde  $n$  es el número de variables de diseño y  $p$  la distribución de probabilidad aproximada por el modelo.

### 5.3.1 Gibbs Sampler

El muestreador de Gibbs pertenece los algoritmos markovianos conocidos hoy en día como MCMC ó Monte Carlo Markov Chain [Brémuaud1999]. Geman y Geman introducen el algoritmo de Gibbs Sampler para la reconstrucción de imágenes

[Geman1984] y se convirtió en una herramienta popular en la solución a problemas bayesianos cuando Gelfand y Smith lo utilizaban para obtener densidades marginales [Gelfand1990], las cuales no se podían obtener de forma analítica, esto se logra a partir de muestras aleatorias de distribuciones conocidas.

La descripción del algoritmo de Gibbs se considera una distribución de probabilidad de la forma:

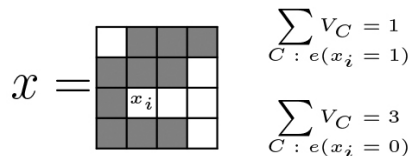
$$p(x, t) = \frac{1}{Z} e^{-\frac{1}{T}U(x)}, \quad (5.3.2)$$

donde a  $T > 0$  se le conoce como factor de temperatura o enfriamiento,  $U(x)$  es la energía de la configuración y  $Z$  es una constante de normalización.  $p(x, t)$  toma valores en  $[0, 1]$  y  $U(x) \in (-\infty, \infty)$ , de modo que a mayor nivel de energía se tiene menor probabilidad de ocurrencia.

El nivel de energía o potencial  $U(x)$  penaliza las configuraciones de cada variable de diseño de forma local con base a la configuración de sus vecinos. La función de potencial que se propone es la descrita en la ecuación (5.3.3).

$$U(e(x)) = -(1 - \lambda) \log \left( \frac{p(x) + \varepsilon}{1 - p(x) + 2\varepsilon} \right) e(x) + \lambda \sum_{C:e(x)} V_C. \quad (5.3.3)$$

Donde  $p(x)$  es la distribución marginal a priori y se mantiene fija durante el muestreo. Se agrega el valor de  $\varepsilon = 1 \times 10^6$  para evitar errores numéricos.  $e(x) \in \{0, 1\}$ .  $C : e(x)$  son las vecindades de tamaño 2 por arista de cada variable de diseño,  $V = 1$  si  $x_i \neq x_j$  y  $V = 0$  si  $x_i = x_j$ , es decir, se penaliza cuando no se tiene una conexión por arista como se muestra en la Figura 5.2. El valor de  $\lambda \in (0, 1)$  permite controlar el peso que aportan la distribución  $p(x)$  y la configuración de la estructura durante el muestreo. Si se sustituye la ecuación (5.3.3) en la ecuación (5.3.2) para  $\lambda = T = 1$  y  $\varepsilon = 0$  se obtiene la distribución marginal.



**Figura 5.2:** Ejemplo de penalización del potencial de una variable de diseño.

El algoritmo de Gibbs se muestra a continuación.

La topología de paralelización en MPI que se utiliza es en estrella. Donde uno de los procesos (el nodo maestro) controla el intercambio de información

---

**Algoritmo 2** Gibbs Sampler

---

```

1: for  $j = 1, \dots, J$  do
2:   for  $i = 1, 2, \dots, |x|$  do
3:     Calcular  $U(x_i^j = 1)$  y  $U(x_i^j = 0)$ .
4:      $p(x_i^j = 1) = \frac{1}{Z} e^{-\frac{1}{T} U(x_i^j = 1)}$ 
5:      $r \sim \text{Unif}(0, 1)$ 
6:     if  $r < p(x_i^j = 1)$  then
7:        $x_i^j = 1$ 
8:     else
9:        $x_i^j = 0$ 
10:    end if
11:  end for
12: end for

```

---

con el resto de los procesos (o nodos esclavos) [Pacheco1997]. La estructura del EDA paralelo se muestra en la Figura 5.3. En la primer iteración del EDA se realiza la evaluación del doble de la población para tener una mayor variabilidad. La evaluación de cada individuo involucra aplicar el muestreador de Gibbs y realizar un análisis estructural por FEM. Una vez aproximado el modelo en cada generación, se selecciona de forma aleatoria un individuo del conjunto Elite y se le intentan remover los elementos menos esforzados, si esta nueva estructura tiene un mejor valor de aptitud, la estructura seleccionada es reemplazada, en caso contrario permanece la estructura seleccionada.

## 5.4 Resultados

En esta sección se presentan varios ejemplos de optimización utilizando el algoritmo de optimización basado en el muestreador Gibbs y esfuerzos. Para todos los ejemplos las propiedades del material son: módulo de Young  $E = 2.1 \times 10^{11}$  Pa, coeficiente de Poisson  $\nu = 0.3$ , espesor de 0.01 m y un módulo máximo de resistencia de  $2.8 \times 10^8$  Pa. Utilizando  $\beta = \lambda = 0.1$ ,  $T = 1$  y un desplazamiento máximo de  $\varepsilon = 1 \times 10^{-4}$ . El número de elementos utilizados en la discretización es mayor a 10000 elementos, con el fin de tener un buen análisis estructural. La contribución de la matriz de rigidez de los elementos ausentes en el diseño es escalado por  $1 \times 10^{-4}$ , esto con el fin de no generar errores numéricos al hacer el análisis estructural. Las características de las computadoras utilizadas para las simulaciones son: Hardware con CPUs de  $24 \times 2.1$  GHz (2 físicos con 12 threads cada uno), 31.39 GB de memoria RAM. Y se utilizó el CAD de GiD [GiD] para visualizar los resultados obtenidos.

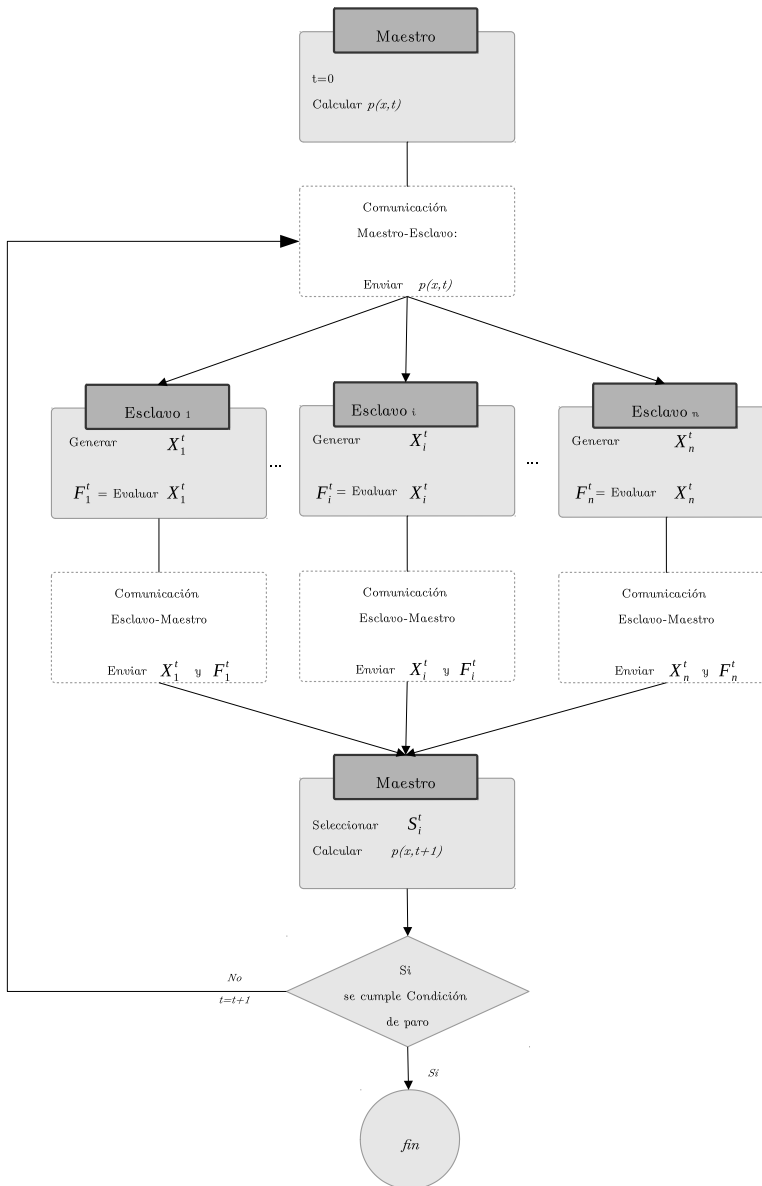


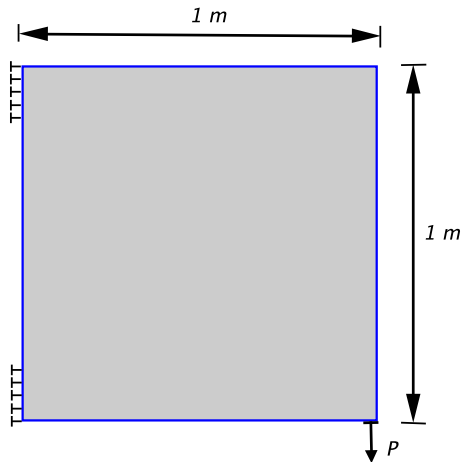
Figura 5.3: EDA paralelo.

#### 5.4.1 Viga en voladizo corto

El primer ejemplo es una viga en voladizo corto. En la Figura 5.4 se muestra el dominio de diseño, que se encuentra sujeto en la parte lateral izquierda superior e inferior. Se aplica una carga  $\mathbf{P} = 5000 \text{ N}$  en dirección hacia abajo del lado derecho

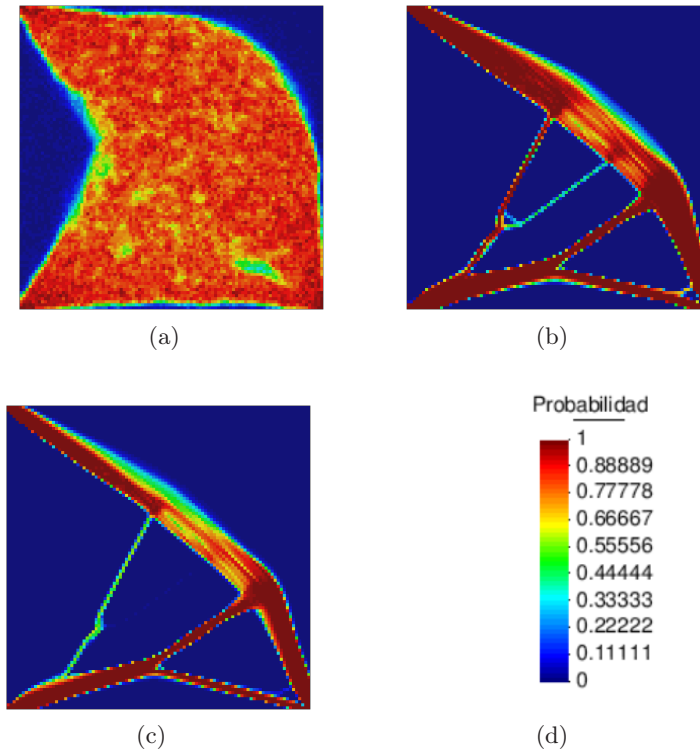


inferior de la viga. Se utilizó una malla de 10000 elementos, población de 1400 individuos, conjunto élite y conjunto seleccionado de la mitad de la población y un máximo de 8000 generaciones.



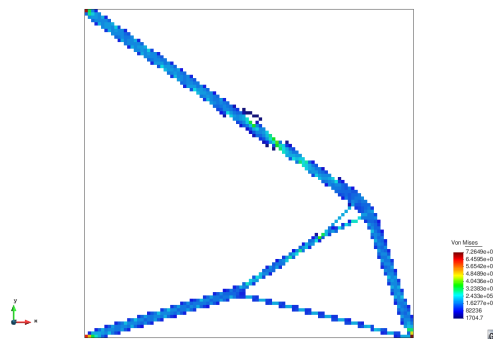
**Figura 5.4:** Dominio de diseño, carga y las condiciones de contorno para viga en voladizo corto.

En la Figura 5.5 se muestra la estimación de la distribución de probabilidad en diferentes generaciones, así como un mapa de falsos colores para el valor de probabilidad de cada elemento. En la secuencia se puede apreciar como el algoritmo va aproximando la distribución del diseño de la estructura.



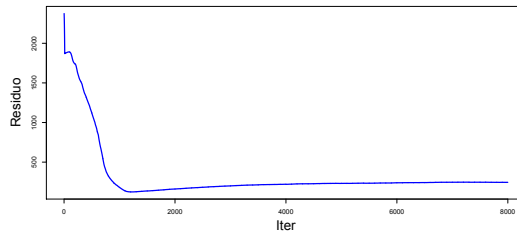
**Figura 5.5:** Distribución de probabilidad en las generaciones a) 500, b) 4000 y c) 8000. El mapa de falsos colores muestra la probabilidad de que exista material en cada región.

En la Figura 5.6 se muestra el diseño de la estructura del mejor individuo de la población que se logró aproximar, donde se puede apreciar que las zonas más esforzadas son las partes apoyadas y donde se aplica la carga. La estructura cuenta un 8.6% del volumen total.



**Figura 5.6:** Diseño final de estructura para el problema de la viga en voladizo corto tras aplicar el algoritmo de optimización. Se muestra en colores el valor de esfuerzo von Mises del diseño.

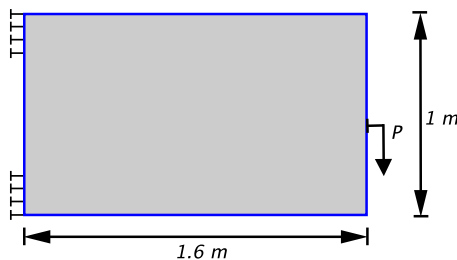
Por último en la Figura 5.7 se muestra el residuo calculado en cada una de las generaciones. Este valor decrece rápidamente en las primeras generaciones correspondientes a las generaciones donde la distribución tiene un cambio mayor, pero después el residuo tiende a crecer lentamente, que es cuando el algoritmo comienza a definir los bordes de las estructuras.



**Figura 5.7:** Convergencia del algoritmo para cada una de las generaciones del problema de la viga en voladizo corto.

### 5.4.2 Viga en voladizo

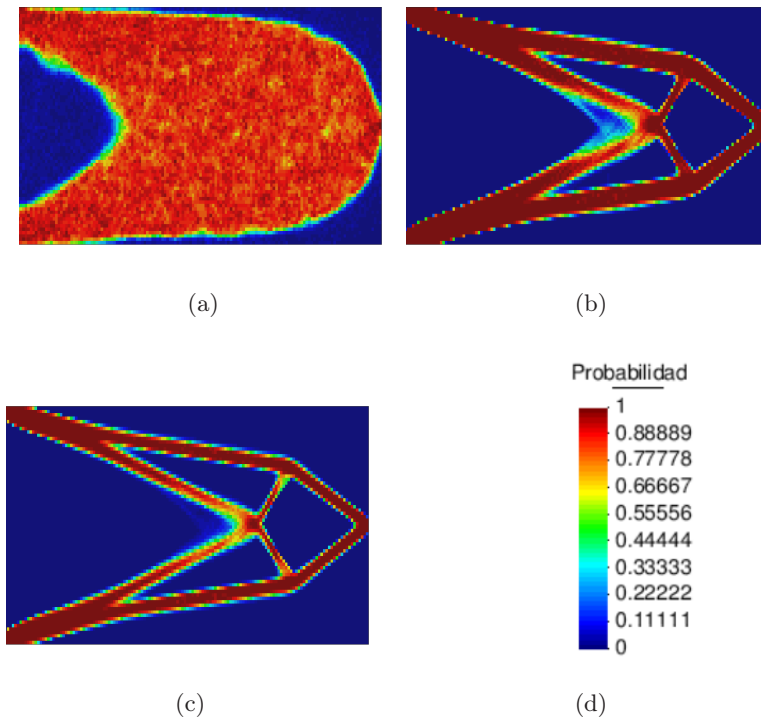
El segundo ejemplo es una viga en voladizo. En la Figura 5.8 se muestra el dominio de diseño, que se ve sujeto en la parte superior e inferior del lado izquierdo. Se aplica una carga  $\mathbf{P} = 3000$  N en dirección hacia abajo en la parte central derecha de la viga. Se utilizó una malla de 10500 elementos, población de 1400 individuos y un máximo de 8000 generaciones.



**Figura 5.8:** Dominio de diseño, carga y las condiciones de contorno para voladizo.

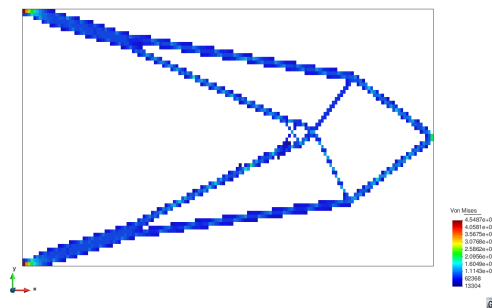
Al igual que en el ejemplo anterior, en la Figura 5.9 se muestra la estimación de la distribución de probabilidad en diferentes generaciones, así como un mapa de falsos colores para el valor de probabilidad de cada elemento. En este ejemplo se puede apreciar un eje de simetría en la distribución, también se puede ver que la zona con mayor variabilidad es la parte donde se forma una especie de equis en la parte central del diseño.

En la Figura 5.10 se muestra el diseño de la estructura del mejor individuo de la población que se logró crear al término de las generaciones. Las zonas



**Figura 5.9:** Distribución de probabilidad en las generaciones a) 500, b) 4000 y c) 8000. Se puede apreciar la presencia un eje de simetría. El mapa de falsos colores muestra la probabilidad de que exista material en cada región.

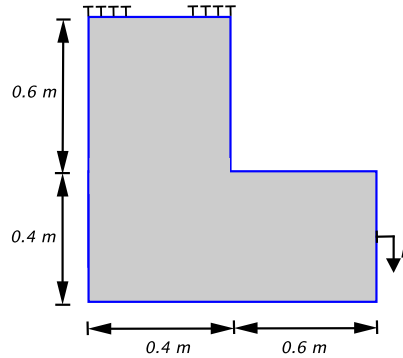
más esforzadas son las partes apoyadas localizadas en el costado izquierdo. La estructura cuenta un 11.65% del volumen total.



**Figura 5.10:** Diseño final de estructura para el problema de la viga en voladizo tras aplicar el algoritmo de optimización. Se muestra en colores el valor de esfuerzo von Mises del diseño.

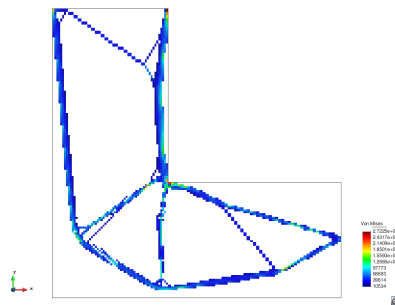
### 5.4.3 Viga en L

Se le aplica el algoritmo de optimización a una viga en L. En la Figura 5.11 se muestra el dominio de diseño, que está sujeto en la parte superior del borde izquierdo y derecho. Se aplica una carga  $\mathbf{P} = 600 \text{ N}$  en dirección hacia abajo en la parte central derecha de la viga. Se utilizó una malla de 10200 elementos para la simulación.

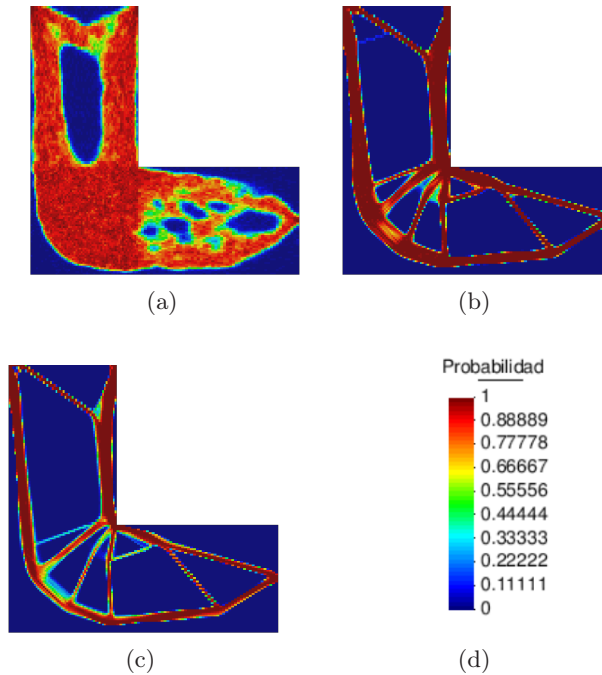


**Figura 5.11:** Dominio de diseño, carga y las condiciones de contorno para una viga en L.

En la Figura 5.12 se muestra la estimación de la distribución de probabilidad en diferentes generaciones, así como un mapa de colores para el valor de probabilidad de cada elemento. Podemos notar que la zona donde se presenta mayor variabilidad es en la parte central de la viga. En la Figura 5.13 se muestran el diseño de la estructura con mejor valor de aptitud, donde se puede apreciar que las zonas más esforzadas son las partes apoyadas. La estructura cuenta un 12.82% del volumen total.



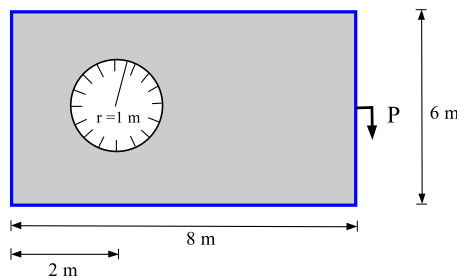
**Figura 5.13:** Diseño final de estructura para el problema de la viga en L tras aplicar el algoritmo de optimización. Se muestra en colores el valor de esfuerzo von Mises del diseño.



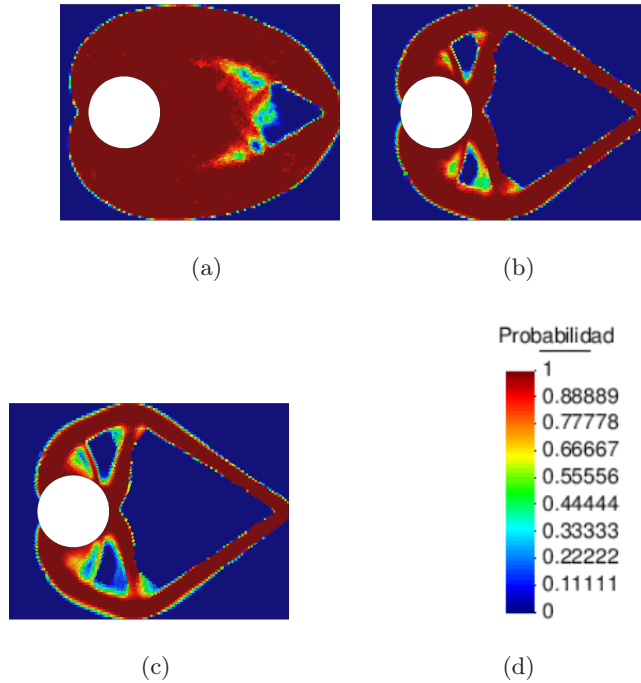
**Figura 5.12:** Distribución de probabilidad en las generaciones a) 500, b) 4000 y c) 6950. Se puede apreciar la generación de agujeros durante el proceso de optimización. El mapa de falsos colores muestra la probabilidad de que exista material en cada región.

#### 5.4.4 Viga de Michael

Este ejemplo es el problema de una viga de Michael. En la Figura 5.14 se muestra el dominio de diseño, se encuentra sujeta en el círculo interior. Se aplica una carga  $\mathbf{P} = 600 \text{ N}$  en dirección hacia abajo en la parte central derecha de la viga. Se utilizó una malla de 10740 elementos para la simulación, con población de 1000 individuos y un máximo de 2000 generaciones.

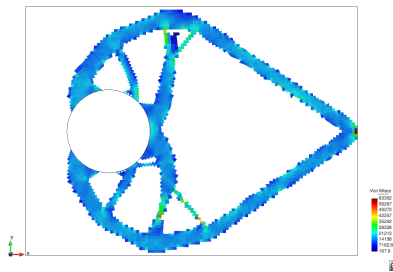


**Figura 5.14:** Dominio de diseño, carga y las condiciones de contorno de una viga de Michael.



**Figura 5.15:** Distribución de probabilidad en las generaciones a) 250, b) 1000 y c) 2000. El mapa de falsos colores muestra la probabilidad de que exista material en cada región.

En la Figura 5.15 se muestra la estimación de la distribución de probabilidad en diferentes generaciones, así como un mapa de colores para el valor de probabilidad de cada elemento y en la Figura 5.16 se muestran el diseño de la estructura con mejor valor de aptitud, donde se puede apreciar que las zonas más esforzadas son las partes apoyadas. La estructura cuenta un 20.0621% del volumen total.



**Figura 5.16:** Diseño final de estructura para el problema de la viga de Michael tras aplicar el algoritmo de optimización. Se muestra en colores el valor de esfuerzo von Mises del diseño.

## 5.5 Conclusiones

En el presente trabajo se muestra una propuesta para resolver el problema de optimización topológica de estructuras con una formulación discreta. Este es un problema habitual en ingeniería, que consiste en crear el diseño óptimo de una estructura con un mínimo peso que sea capaz de soportar las restricciones estructurales impuestas como no sobrepasar el máximo esfuerzo permitido por el material, así como desplazamientos en ciertas zonas. El problema de optimización se llevó a un problema de minimización, que se resolvió utilizando algoritmos de estimación de distribución, en el que se utilizó el muestreador de Gibbs.

Algunas de las características del algoritmo presentado es que se requiere de alto poder de cómputo. No se requiere de tener conocimiento a priori del diseño de la estructura. Tampoco se requiere de la definición de parámetros como la restricción en volumen, que en algunos métodos es fundamental para la solución final del diseño, lo que les impide llegar a realmente aproximar un óptimo global del problema. Al utilizar una formulación discreta del problema y considerar el peso propio de la estructura, es posible llegar a construir las estructuras encontradas por el algoritmo y éste logra reducir de forma considerable el volumen de las estructuras.



# Bibliografía

- [Martin2004] Martin Phili Bendse and Ole Sigmund et al. 2004. **Topology Optimization: Theory, Methods and Applications**. Springer.
- [Yiqiang2014] Yiqiang Wang, Zhan Kang and Qizhi He. et al 2014. **Adaptive Topology Optimization with Independent Error Control for Separated Displacement and Density Fields**. Computers & Structures, 135:50–61
- [Ullah2013] B. Ullah and J. Trevelyan et al. 2013. **Correlation Between Hole Insertion Criteria in a Boundary Element and Level Set Based Topology Optimisation Method**. Engineering Analysis with Boundary Elements, 37: 1457–1470.
- [Querín1998] OM Querín, GP Steven and YM Xie et al. 1998. **Evolutionary Structural Optimisation (ESO) Using a Bidirectional Algorithm**. Engineering Computations, 15(8):1031–1048.
- [SAVAŞ] SAVAŞ, Sedat et al 2012. **Evolutionary Topological Design of Two Dimensional Composite Structures**. American International Journal of Contemporary Research. 2(3):76–88.
- [Ruiyi2009] Su Ruiyi, Gui Liangjin, Fan Zijie et al 2009. **Truss Topology Optimization Using Genetic Algorithm with Individual Identification Technique**. Proceedings of the World Congress on Engineering.
- [Oñate1992] Eugenio Oñate et al. 1992. **Cálculo de Estructuras por el Método de Elementos Finitos - Análisis Estático Lineal**. Artes Gráficas Torres, S.A.
- [Akin2005] J. E. Akin et al. 2005. **Finite Element Analysis with Error Estimators. An Introduction to the FEM and Adaptive Error Analysis for Engineering Students**.
- [Larrañaga2001] P. Larrañaga and J. A. Lozano et al 2001. **Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation**. Kluwer Academic Publishers.

- [Larrañaga1999] P. Larrañaga, R. Etxeberria, J. Lozano, and J. Peña et al. 1999. **Optimization by Learning and Simulation of Bayesian and Gaussian Networks**. Department of Computer Science and Artificial Intelligence, University of the Basque Country.
- [Kvasnicka1996] Vladimir Kvasnicka, Martin Pelikan, and Jiri Pospichal et al. 1996. **Hill Climbing with Learning (An Abstraction of Genetic Algorithm)**. *Neural Networks World*, 5:773–796.
- [Pelikan1999] Pelikan, Martin and Mühlenbein, Heinz et al. 1999. **The Bivariate Marginal Distribution Algorithm**. *Advances in Soft Computing*, pages 521–535, Springer.
- [Brémuaud1999] Pierre Brémuaud et al. 1999. **Markov Chains - Gibbs Fields, Monte Carlo Simulations and Queues**. Springer.
- [Geman1984] S. Geman and D. Geman et al. 1984. **Stochastic Relaxation, Gibbs Distributions and the Bayesian Restoration of Images**. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741.
- [Gelfand1990] Gelfand, A.E. and Smith, A.F.M. et al. 1990. **Sampling-Based Approaches to Calculating Marginal Densities**. *Journal of the American Statistical Association*, 85:398–409.
- [Pacheco1997] Peter S. Pacheco et al. 1997. **Parallel programming with MPI**. Morgan Kaufmann.
- [GiD] <http://gid.cimne.upc.es/>