

**DESARROLLO DE UN ALGORITMO DE OPTIMIZACIÓN PARA ESTIMAR EL
ESTADO DE CARGAS DE UN COMPONENTE ESTRUCTURAL BASADO EN
SU FORMA ÓPTIMA**

CHRISTIAN FLOREZ GALVIS

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICOMECÁNICAS
ESCUELA DE INGENIERÍA MECÁNICA
BUCARAMANGA**

2017

**DESARROLLO DE UN ALGORITMO DE OPTIMIZACIÓN PARA ESTIMAR EL
ESTADO DE CARGAS DE UN COMPONENTE ESTRUCTURAL BASADO EN
SU FORMA ÓPTIMA**

CHRISTIAN FLOREZ GALVIS

Trabajo de grado para optar por el título de Ingeniero Mecánico

Director

DIEGO F. VILLEGAS BERMUDEZ

Ingeniero Mecánico, MSc., PhD.

Co-director

Octavio Andrés González Estrada

Ingeniero Mecánico, MSc., PhD.

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICOMECÁNICAS
ESCUELA DE INGENIERÍA MECÁNICA
BUCARAMANGA**

2017

TABLA DE CONTENIDO

	Pág.
INTRODUCCIÓN	13
1. DESCRIPCIÓN DEL PROBLEMA	14
2. JUSTIFICACIÓN DE LA SOLUCIÓN	18
3. OBJETIVOS	20
3.1. OBJETIVO GENERAL	20
3.2. OBJETIVOS ESPECÍFICOS	20
4. FUNDAMENTOS TEÓRICOS	21
4.1. MÉTODO DE ELEMENTOS FINITOS (MEF)	21
4.2. OPTIMIZACIÓN TOPOLÓGICA POR DISTRIBUCIÓN DE MATERIAL ISOTRÓPICO	29
5. METODOLOGÍA	34
5.1. ANÁLISIS DE ALTERNATIVAS DE SOLUCIÓN	34
5.2. IMPLEMENTACIÓN	36
5.3. VALIDACIÓN DE RESULTADOS	37
6. DESARROLLO DEL ALGORITMO	38
6.1. VARIABLES DE ENTRADA Y SALIDA	38
6.2. RESTRICCIONES Y CONDICIONES	39
6.3. EL ALGORITMO	40
7. RESULTADOS	46
8. ANÁLISIS DE RESULTADOS	52
9. CONCLUSIÓN	55
BIBLIOGRAFÍA	58

ANEXOS.....60

LISTA DE ILUSTRACIONES

Fig. 1 Optimización de forma de un hueso. Fuentes: (izq) Quora. (der) AGHILI, A. Latif, et al. Editado por autor.	16
Fig. 2 Categorías de optimización estructural. a) Optimización de dimensión de una armadura, b) optimización de forma y c) optimización topológica. Fuente: M.P. Bendsoe (2004).	30
Fig. 3 Condiciones iniciales para las vigas en voladizo de los ejemplos de validación; 1.) Ejemplo 1, 2.) Ejemplo 2, 3.) Ejemplo 3. Fuente: Autor.	40
Fig. 4 Movimiento de la carga para una viga empotrada en voladizo. Fuente: Sigmund, Ole., 2001. Editado por el autor.	41
Fig. 5 Descripción gráfica del proceso de optimización topológica en su fase inicial, intermedia y su resultado final. Fuente: Autor.	43
Fig. 6 La topología de referencia (izquierda) y la construcción de la matriz de densidades (derecha). Fuente: Autor.	44
Fig. 7 Factor de similitud vs. Iteración cuando la magnitud de la carga es conocida. Fuente: Autor.	46
Fig. 8 Gráficas de comportamiento para magnitudes de referencia de 1 (izquierda) y 500 (derecha) unidades de fuerza. Fuente: Autor.	47
Fig. 9 Topologías óptimas de una viga empotrada a cargada a diferentes magnitudes y en la misma posición. Fuente: Autor.	48

Fig. 10 El factor de similitud para varia magnitudes y posiciones con un punto de referencia de magnitud =1 [uf]. Fuente: Autor.....49

Fig. 11 Dominios de diseño y topologías finales de los modelos estimados; 1.) Ejemplo 1, 2.) Ejemplo 2, 3.) Ejemplo 3. Fuente: Autor.50

LISTA DE TABLAS

Tabla 1 Definición de ejemplos de validación	39
Tabla 2 Resultados de los ejemplos de validación definidos en el capítulo 6. L es la longitud total del dominio de diseño; x es la posición estimada la carga; m es la magnitud de la fuerza; uf representa unidades de fuerza.	49
Tabla 3 Ponderación de Criterios.....	61
Tabla 4 Matriz de Decisión de Alternativas	61

LISTA DE ANEXOS

ANEXO A: QFD Y EVALUACIÓN PUGH.....	61
ANEXO B: CÓDIGO EDITADO DE OPTIMIZACIÓN TOPOLÓGICA	62
ANEXO C: CÓDIGO DE MEF CON INTERACCIÓN ENTRE MATLAB Y ANSYS	65
ANEXO D: CÓDIGO DE ESTIMACIÓN DE CARGAS.....	70
ANEXO E: DIAGRAMA DE FLUJO DEL ALGORITMO	72

RESUMEN

TITULO: DESARROLLO DE UN ALGORITMO DE OPTIMIZACIÓN PARA ESTIMAR EL ESTADO DE CARGAS DE UN COMPONENTE ESTRUCTURAL BASADO EN SU FORMA ÓPTIMA *.

AUTOR: CHRISTIAN FLOREZ GALVIS**

PALABRAS CLAVE: Métodos numéricos, optimización topológica, estimación de carga, biomecánica del hueso.

DESCRIPCIÓN

Las estructuras biológicas adaptan su forma a su entorno mecánico, con el fin de optimizar la energía que necesita para funcionar. En este sentido, un hueso cambiará su topología de acuerdo con su estado de carga, generando hueso más denso donde mayor esfuerzo sea soportado. Partiendo del hecho que las formas de la naturaleza son óptimas y que los estados de carga en el cuerpo humano no son completamente conocidos, este estudio apunta a encontrar el estado de carga basado en la forma óptima, en contraparte a la metodología de diseño mecánico.

Debido a que el dominio de diseño del componente y su forma real son conocidos, este algoritmo es enfocado de manera iterativa, tomando como parámetro primario la construcción de formas óptimas a través del método de los elementos finitos para cargas asumidas, seguido por un módulo que compara el modelo computacional obtenido con la topología real de la estructura.

Inicialmente, el problema es resuelto para una viga empotrada, soportando una carga estática, puntual y vertical. Así, el algoritmo es probado para un caso simple, buscando aplicarlo en estructuras de mayor complejidad y poder acercarse a piezas biológicas reales.

Los autores creen que este estudio permitirá a los especialistas en las ciencias médicas hacer análisis más completos con respecto a enfermedades y peculiaridades de los huesos y de elementos estructurales de los seres vivos.

* Trabajo de grado

** Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería Mecánica. Director Diego Fernando Villegas Bermúdez. Codirector Octavio Andrés González Estrada

ABSTRACT

TITLE: DEVELOPMENT OF AN OPTIMIZATION ALGORITHM FOR ESTIMATING THE STATE OF LOADS OF A STRUCTURAL COMPONENT BASED ON ITS OPTIMAL SHAPE *

AUTHOR: CHRISTIAN FLOREZ GALVIS**

KEY WORDS: Numerical methods, topology optimization, load estimation, biomechanics of bone.

DESCRIPTION

Biological structures adapt their shapes to their mechanical environment with the aim of optimizing the energy they need for functioning. On that way, a bone will change its shape according to its state of loading making denser bone where more load is supported. Starting from the fact that shapes on nature are optimal and that loading states in human body parts are not completely known, this study targets to find a precise state of load based on shape, and not the way that mechanical designing does which seeks an optimal shape based on loads.

Since the component's design domain and actual shape are known, the algorithm described is approached iteratively, taking the first parameter to build optimal shapes through the Finite Elements Method for assumed loads, followed by a module that compares the obtained computational model with the actual shape of the part.

As a starting point, the problem is solved for a cantilever beam, loaded by vertical, static, concentrated forces. In this way, the algorithm is proved for a simple case, in order to be applied for more complex components and come closer to real biological structures.

The authors believe that this study will make specialists in medical sciences go further and do more complete analysis around illnesses and peculiarities of bones and structural components of living beings.

* Degree Thesis.

** Faculty of Phisico-Mechanical Engineering. Department of Mechanical Engineering. Director: Diego Fernando Villegas Bermúdez. Co-director: Octavio Andrés González Estrada

INTRODUCCIÓN

Partiendo del conocimiento que las morfologías propias de la naturaleza son óptimas (Frost, 1990) y que no se conocen completamente los estados de carga de las partes del cuerpo humano, este proyecto apunta a encontrar un estado de carga específico para un componente basado en su forma.

De esta manera, se desarrolla un método computacional eficaz para estimar con precisión y exactitud los estados de carga de elementos estructurales, con base en los métodos de optimización topológica utilizados comúnmente, además de conceptos concernientes a métodos heurísticos para la solución de problemas de optimización.

La finalidad de este proyecto es facilitar el análisis mecánico de las formas de la naturaleza utilizando herramientas computacionales, evitando sobrecostos económicos y de tiempo. Sin embargo, y debido a que este proyecto se desarrolla a nivel de pregrado, se deben hacer suposiciones y restricciones que permitan la solución del problema desde los conocimientos adquiridos, alejando los resultados de un análisis biomecánico completo.

Por lo tanto, este proyecto debe ser percibido desde el punto de vista de la concepción de una metodología para solucionar problemas como el descrito anteriormente.

1. DESCRIPCIÓN DEL PROBLEMA

Actualmente, los métodos computacionales se han posicionado como una fuerte alternativa para visualizar y analizar los diferentes fenómenos de la naturaleza. Por un lado, esto se debe al crecimiento exponencial de la capacidad de procesamiento de las computadoras, con lo cual el ser humano se ha podido librar de cálculos mecánicos y tediosos, y así llevar su potencial hacia áreas que requieren mayor creatividad e ingenio. Por otro lado, el avance en los lenguajes de programación ha promovido el aumento en la cantidad de aplicaciones computacionales al servicio de la ciencia. Esto se puede ver en los lenguajes de alto nivel, los cuales son más parecidos al lenguaje natural; por lo cual no requieren gastos excesivos de tiempo en su aprendizaje y permiten la producción de programas cada vez más complejos y eficaces.

Sin embargo, aún existe un amplio margen de escenarios sin estudiar, con lo cual el trabajo ingenieril se hace importante al amalgamarse con el campo computacional; sobrepasando límites, con el fin de encontrar maneras nuevas de entender y transformar el universo de la ingeniería en sus diferentes modalidades y aplicaciones.

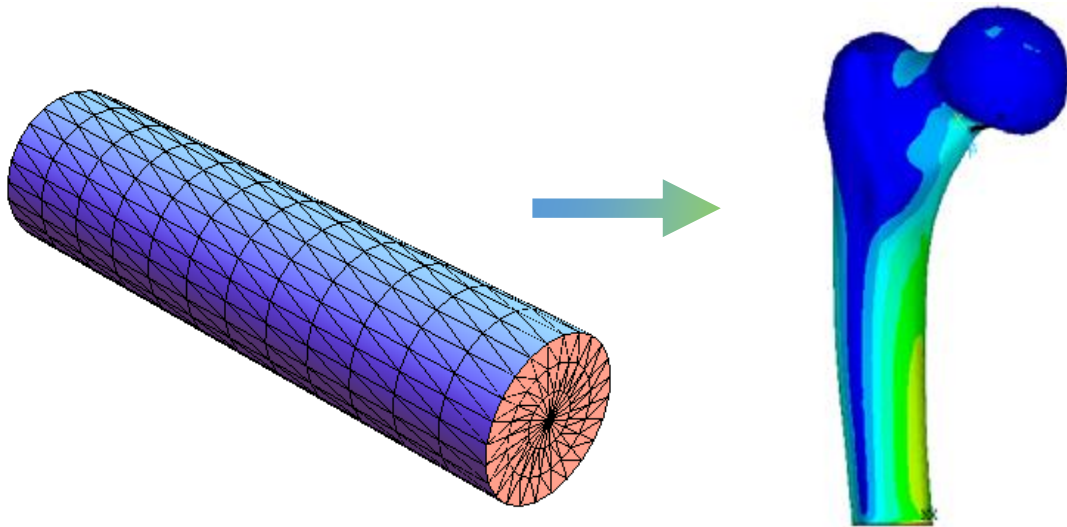
Entre las tantas preguntas sin responder, algunas de ellas se ciernen a las causas que han llevado a los seres de la naturaleza a tener su forma actual. Tales son los ejemplos de la forma de las hojas de ciertos árboles, adaptadas específicamente para la mayor o menor captación de energía solar; y el aspecto que adopta la cabeza del fémur de un ser humano al sufrir de enfermedades como la osteoartritis (OA).

Con el conocimiento que se adquiriera a través de la estimación de las cargas que han provocado su deformación, se ayudará a entender la relación que existe entre la biomecánica de los órganos y la forma tienen, y cómo se afectan mutuamente.

En este sentido, es importante reconocer la validez de este conocimiento en el área de la rehabilitación médica, siendo importante en la optimización de métodos terapéuticos para tratar diversas enfermedades y traumas. Aún más, el desarrollo de un método para saber la razón por la cual un cuerpo tenga su forma puede impactar directamente en la técnica médica, convirtiéndose en una herramienta de diagnóstico de enfermedades del sistema osteomuscular de manera temprana; ya que, a partir del análisis de las deformaciones sobre un órgano determinado, se podrá estimar qué actividades hechas por el paciente las han provocado, contribuyendo así, a la correcta prescripción de tratamientos.

Más allá de aquello, se abre una puerta al análisis de cargas en componentes biológicos sin incidir en su entorno natural, ya que la única interacción será en la medición geométrica de éste, apartando la necesidad de invadir al organismo para obtener información sobre su comportamiento mecánico. Así, por ejemplo, puede estimarse el estado de cargas de un hueso sin insertar sensores para medirlas, partiendo de su forma, entendida como óptima (Frost, 1990), y de una geometría básica de aproximación.

Fig. 1 Optimización de forma de un hueso. Fuentes: (izq) Quora¹. (der) AGHILI, A. Latif, et al². Editado por autor.



Ya desde otro punto de vista, conocer por qué un cuerpo tiene cierta forma concierne al área de la detección de fallas en máquinas, debido a que es de gran ayuda en la evaluación de maquinaria estimar las causas que han provocado deformaciones impidiendo su trabajo óptimo.

En muchos casos es difícil realizar exámenes poco invasivos que permitan dar información relevante acerca del estado de las máquinas, por lo cual se decide esperar a su falla arriesgando la continuidad de la producción y la plenitud de los trabajadores; llevar a cabo actividades de mantenimiento innecesarias, perdiendo tiempo y dinero; o implementar los diversos métodos de mantenimiento predictivo,

¹ Quora. [En línea]. Disponible en: <https://www.quora.com/I-am-using-Hypermesh-as-a-preprocessor-for-finite-element-analysis-How-do-I-decide-which-element-to-use-and-on-what-basis-is-this-decided>

² AGHILI, A. Latif, et al. Stress analysis of human femur under different loadings. En ASLAN, Muhammed Hasan, et al. (ed.). AIP Conference Proceedings. AIP, 2011. p. 518-522.

óptimos en sentido ingenieril, aunque a costa de grandes inversiones monetarias, a las cuales aún muchos empresarios se encuentran renuentes.

Por estas razones, se pretende la búsqueda de metodologías económicamente viables que permitan responder y controlar una gama de situaciones a través de las cuales sea posible contribuir a la preservación y el enriquecimiento de la integridad del ser humano y del medio ambiente.

2. JUSTIFICACIÓN DE LA SOLUCIÓN

El punto al cual se quiere llegar con este proyecto es tener la capacidad de visualizar las causas, mecánicamente hablando, que han producido determinadas deformaciones en un componente estructural, es decir, por qué un cuerpo tiene la forma óptima que efectivamente posee. En este sentido, a través de su análisis, se busca diagnosticar los posibles escenarios que llevaron a cierto componente a deformarse de una manera específica.

En primer lugar, se hace necesaria la confirmación de los enunciados teóricos de la mecánica de sólidos básica; piénsese en vigas, ejes, y columnas de sección transversal constante, etc.

En un futuro cercano, se espera que el método impregne estructuras optimas de mayor complejidad, como elementos de la naturaleza, tales como los tallos de los árboles, las hojas y flores de algunas plantas, y los esqueletos de los seres vivos.

De manera general, ésta se puede visualizar como una herramienta para el diagnóstico de fallas; de tal suerte que se pueda intervenir directamente sobre la causa que provoque cargas perjudiciales en un equipo elemento.

Se puede pensar, como ejemplo, en el diagnóstico de enfermedades del sistema musculo-esquelético en seres humanos; ya que se ha comprobado la relación existente entre los esfuerzos que actúan sobre los huesos y la forma que desarrollan los mismos. De esta manera, se pueden tomar acciones tempranas para combatir este tipo de enfermedades, ayudando a mantener la calidad de vida de los hombres y mujeres implicados.

Sin embargo, esta herramienta puede funcionar como apoyo en los procesos de diseño; teniendo en cuenta que en muchos casos se utiliza el concepto de

biomimética, con el fin de captar ideas a partir de las apariencias adoptadas por la naturaleza a lo largo del tiempo. Así, puede serle dada una mayor validez a la recursión a estos elementos como parte de un proceso de miles de años de optimización.

Finalmente, cabe anotar que se busca la facilidad de uso de la herramienta, entregando grandes esfuerzos a la posibilidad de visualizar las direcciones y magnitudes de las cargas inherentes a un elemento estructural determinado, con el fin de agilizar los procesos investigativos e industriales.

3. OBJETIVOS

3.1. OBJETIVO GENERAL

Desarrollar un algoritmo en Matlab que interactúe con ANSYS para la estimación del estado de cargas de un componente estructural basado en su forma óptima.

3.2. OBJETIVOS ESPECÍFICOS

- Examinar los métodos y algoritmos empleados en la actualidad para llevar a cabo la optimización de forma de estructuras mecánicas.
- Identificar y establecer el algoritmo de optimización adecuado para componentes mecánicos estructurales.
- Implementar un algoritmo computacional en MATLAB para la interacción con ANSYS, con el fin de obtener el estado de cargas en un componente estructural con forma óptima.
- Validar los resultados obtenidos utilizando el algoritmo desarrollado en un problema de optimización de forma previamente solucionado.

4. FUNDAMENTOS TEÓRICOS

La base sobre la cual se desarrolla la completitud del trabajo realizado se distribuye en dos pilares primordiales.

Por un lado, el método de los elementos finitos permite llevar a cabo el análisis computacional de componentes sólidos como los ejecutados por el algoritmo desarrollado, siendo éste el punto de partida para elaborar un modelo de optimización topológica, el cual entrega la información necesaria para elaborar un modelo de estimación de las cargas, además de contener en su esencia nociones indispensables para el desarrollo conceptual del proceso final.

4.1. MÉTODO DE ELEMENTOS FINITOS (MEF)

El método de elementos finitos es un método de aproximación numérica utilizado para el análisis de problemas continuos de diseño estructural, modal y demás situaciones concernientes a la ingeniería. Éste contiene una serie de pasos a seguir, con el fin de dar solución a los problemas mencionados.

4.1.1. Planteamiento del problema: En primer lugar, debe especificarse qué significa solucionar un problema de tipo estructural, es decir, qué información se está buscando para describir la situación presentada. En este sentido, el ingeniero requiere conocer tres características en cualquier punto del dominio del problema. Éstos son:

- El campo de desplazamientos: $u(x, y)$.
- El campo de deformaciones: $\varepsilon(x, y)$.
- El campo de tensiones: $\sigma(x, y)$.

Aunque los problemas estructurales son sustancialmente tridimensionales, en algunos casos se incurre en errores despreciables al simplificarlos en términos de dos dimensiones, es decir, en tensión o deformación plana. Entonces,

- En problemas de deformación plana: $\varepsilon_z = 0$.
- En problemas de tensión plana: $\sigma_z = 0$.

Para la formulación continua del problema se necesitan las ecuaciones diferenciales que describen el comportamiento de un elemento diferencial genérico de la estructura. En su análisis éstas deben ser integradas para dar con la solución.

A pesar que con el MEF se busca dar con los campos detallados con anterioridad, esto se hace a partir de los valores que se encuentran en ciertos puntos del dominio. Por lo tanto, éste último se divide en subdominios formando una malla. Así, al plantear las ecuaciones que se obtiene la formulación discreta del problema.

4.1.2. Ecuaciones del problema continuo: Matemáticamente, el problema se define en términos de relaciones cinemáticas, relaciones constitutivas y las ecuaciones de equilibrio.

Primeramente, se debe definir la relación existente entre desplazamiento y deformación, siendo ésta la primera derivada de aquella. De esta manera,

- Campo vectorial de desplazamientos: $d(x, y) = \begin{Bmatrix} u(x, y) \\ v(x, y) \end{Bmatrix}$.
- Campo tensorial de deformaciones:
$$\begin{aligned} \varepsilon_x &= \frac{\partial u}{\partial x} & \varepsilon_y &= \frac{\partial v}{\partial y} \\ \gamma_{xy} &= \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} & \gamma_{xz} &= \gamma_{yz} = 0 \end{aligned}$$

Posteriormente, se define el comportamiento del material del problema especificado, el cual relaciona los esfuerzos y las deformaciones a partir de la hipótesis de elasticidad lineal. En este sentido, se define la matriz D como aquella que contiene las constantes elásticas del material. En tensión plana esto es:

$$D = \frac{E}{1 - \nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1 - \nu}{2} \end{bmatrix}.$$

Por otro lado, para deformación plana esta matriz se expresa como:

$$D = \frac{E(1 - \nu)}{(1 + \nu)(1 - 2\nu)} \begin{bmatrix} 1 & \frac{\nu}{1 - \nu} & 0 \\ \frac{\nu}{1 - \nu} & 1 & 0 \\ 0 & 0 & \frac{1 - 2\nu}{2(1 - \nu)} \end{bmatrix}.$$

Finalmente, las relaciones antes mencionadas se formulan matemáticamente a través del equilibrio entre las fuerzas externas y las fuerzas internas generadas debido a la deformación elástica del componente.

Usualmente, se utiliza para este fin el principio de los trabajos virtuales, descrito en la siguiente igualdad:

$$\iint_A \delta \varepsilon \sigma \, dA = \iint_A \delta u \, b \, dA + \int_l \delta u \, t \, ds + \sum_i \delta u_i q_i$$

Como adquiere una gran complejidad la resolución de esta expresión de manera analítica, es necesario hacer una aproximación discreta utilizando elementos finitos.

4.1.3. El problema discreto: En términos generales, resolver el problema continuo se hace muy tedioso e incluso imposible después de cierto grado de complejidad. Por lo tanto, se resuelve el problema en algunos puntos del dominio y se extrapolan a cualquier otro dentro del mismo.

Plantear el problema de esta forma sigue ciertos pasos. Éstos son:

- Dividir el dominio en un número finito de elementos, formando una malla.
- Aplicar el principio de los trabajos virtuales sobre cada uno de los elementos, obteniendo un sistema de ecuaciones.
- Resolver aquel sistema y hallar los campos de desplazamiento, deformación y tensiones, de acuerdo a las condiciones de contorno consideradas.

De esto se desprende la discretización de los campos incógnita.

Por un lado, el campo de desplazamientos se puede escribir como:

$$u(x, y) = N(x, y)a^{(e)}$$

Siendo N la función de forma del elemento y $a^{(e)}$ el desplazamiento de cada nodo. Ellos se describen de la siguiente forma:

$$a_i^{(e)} = [u_i, v_i]^T$$

$$N = [N_1 \quad \dots \quad N_i \quad \dots \quad N_n]$$

$$N_i = \begin{bmatrix} N_i(x, y) & 0 \\ 0 & N_i(x, y) \end{bmatrix}$$

Los elementos utilizados en este método se formulan a partir del concepto de elementos isoparamétricos. Con ellos se puede definir la geometría de un elemento finito desde sus coordenadas naturales, es decir, ξ y η . Entonces,

$$x = \sum_{i=1}^n N_i(\xi, \eta) x_i \qquad y = \sum_{i=1}^n N_i(\xi, \eta) y_i.$$

De allí, se relaciona el espacio real (cartesiano) con espacios unitarios imaginarios (natural). Esta transformación se realiza a través de la matriz Jacobiano $J^{(e)}$ aplicada a las funciones de forma.

Para cada tipo de elemento, las funciones de forma pueden ser formuladas, ya sean triangulares, cuadriláteros lagrangianos o cuadriláteros serendípitos.

El primero de ellos se forma a partir de tres planos que pasan por dos nodos nulos y uno unitario. Además, se puede formular un elemento triangular cuadrático, el cual se forma por seis nodos; tres de estos en los puntos medios de los lados del triángulo.

Los elementos lagrangianos se forman de acuerdo a interpolaciones polinómicas de Lagrange en dos dimensiones en las coordenadas naturales (ξ y η). El número de términos en aquellos polinomios se puede conseguir desde el triángulo de Pascal a partir del grado polinómico en ambas direcciones. Usualmente, se utiliza el elemento rectangular de 4 nodos -uno en cada esquina del cuadrilátero- y el rectangular de 9 nodos -además de en las esquinas, con nodos en puntos entre las esquinas y uno central-.

En tercer lugar, los elementos serendípitos son aquellos obtenidos seleccionando el número de nodos a cada lado para definir el tipo de variación deseada. Posteriormente, se debe escoger el número mínimo de nodos de modo que la variación polinómica sea del mismo grado que la variación en los lados. El elemento

más sencillo de este grupo es el cuadrilátero de 4 nodos, perteneciendo éste también a los elementos lagrangianos.

Por otro lado, la discretización del campo de deformaciones se desprende de su definición en elasticidad. Por lo tanto, se puede escribir:

$$\varepsilon = B\alpha^{(e)}$$

Siendo $B = LN$. Interpretando esta relación como la matriz gradiente de deformación del elemento. Por su parte, L es la matriz de diferenciales.

Por su parte, el campo de tensiones se logra a través la relación entre tensión y deformación de la siguiente manera:

$$\sigma = DB\alpha$$

Donde D es la matriz de elasticidad del material.

Finalmente, se obtiene la matriz de rigidez de cada elemento con el fin de edificar la matriz global del cuerpo. Para esto, se debe solucionar la siguiente integral para cada elemento dando forma a aquella.

$$k_{ij}^{(e)} = \iint_{A^{(e)}} B_i^t D B_j t dA$$

Esta integral puede ser resuelta por vía numérica o analítica, según la facilidad del problema.

El ensamblaje de la matriz de rigidez global se forma ubicando el coeficiente k_{ij} de la matriz del elemento en la posición ij de la matriz de rigidez global.

4.1.4. Discretización de las fuerzas externas: Primero que nada, se debe decir que estas fuerzas son todas aquellas contribuciones que actúan sobre el cuerpo. Éstas pueden ser distribuidas, concentradas, o gravitatorias. Al menos una de ellas debe ser considerada para que el problema tenga sentido.

- **Distribuidas:** A partir de la ecuación de equilibrio, el término correspondiente a las fuerzas de este tipo es:

$$\int_l \delta u \ t \ ds$$

De éste se reemplazan los términos ya discretizados, cambiando como sigue:

$$f_p^{(e)} = \int_{l^{(e)}} N^t \ t \ ds$$

Esta ecuación se interpreta como la repartición de la fuerza distribuida sobre cada uno de los nodos del espacio que la contiene como fuerzas concentradas.

Se debe solucionar esta integral para cada elemento para luego ensamblarse en el vector global de acciones sobre el componente.

- **Concentradas:** Estas fuerzas se conciben como ubicadas sobre un nodo, por lo cual solo deben ser ensambladas en el vector global de fuerzas aplicadas.
- **Gravitatorias:** Este tipo de carga es debido a la acción gravitatoria sobre el volumen del cuerpo, por lo cual se reemplazan los términos continuos de la integral en la ecuación de equilibrio por términos discretos.

$$\iint_A \delta u \ b \ dA$$

$$f_g^{(e)} = \iint_{A^{(e)}} N^t \ b \ dA$$

De esta manera, se entiende esta discretización como la distribución en los nodos del cuerpo de la fuerza gravitatoria ejercida sobre él.

Hecha esta clasificación, se avanza hacia el ensamblaje de las fuerzas sobre cada elemento en un vector global de cargas aplicadas, en el cual se hace una sumatoria de las fuerzas que actúan sobre cada nodo y se ubican en la posición correspondiente, según la numeración hecha sobre ellos. Así, se le da forma al vector de fuerzas mostrado a continuación.

$$f = \begin{Bmatrix} f_i \\ \vdots \\ f_n \end{Bmatrix}$$

4.1.5. Condiciones de contorno: Para poder resolver un problema estructural se deben incluir restricciones que eviten la aparición de movimiento de sólido rígido, fijando el componente al espacio.

En el plano cartesiano, estas restricciones pueden hacer en dos direcciones, es decir, en x e y ; definiéndose ellas en el campo de los desplazamientos. En consecuencia, se impone una o todas las condiciones siguientes.

$$u_i = 0;$$

$$v_i = 0$$

De esta manera, se simplifican las matrices de la ecuación general $K a = f$.

4.1.6. Cálculo de la solución: Como se dijo inicialmente, para analizar el problema estructural se deben conocer los campos de desplazamientos, deformaciones y esfuerzos.

En primer lugar, se obtienen los desplazamientos a través de la solución a la ecuación $K a = f$. De aquí, se obtiene igualmente la estructura deformada del cuerpo.

Posteriormente, se halla el campo de deformaciones con la ayuda de la relación existente entre éste y los desplazamientos, es decir, $\varepsilon = B a$.

Finalmente, el campo de tensiones se consigue a partir del cálculo del esfuerzo en el punto de integración y su consecutiva extrapolación a cualquier otro punto.

Usualmente, se calculan también las reacciones en el cuerpo. Éstas se encuentran a través de la solución de equilibrio del sistema $R = K a - f$.

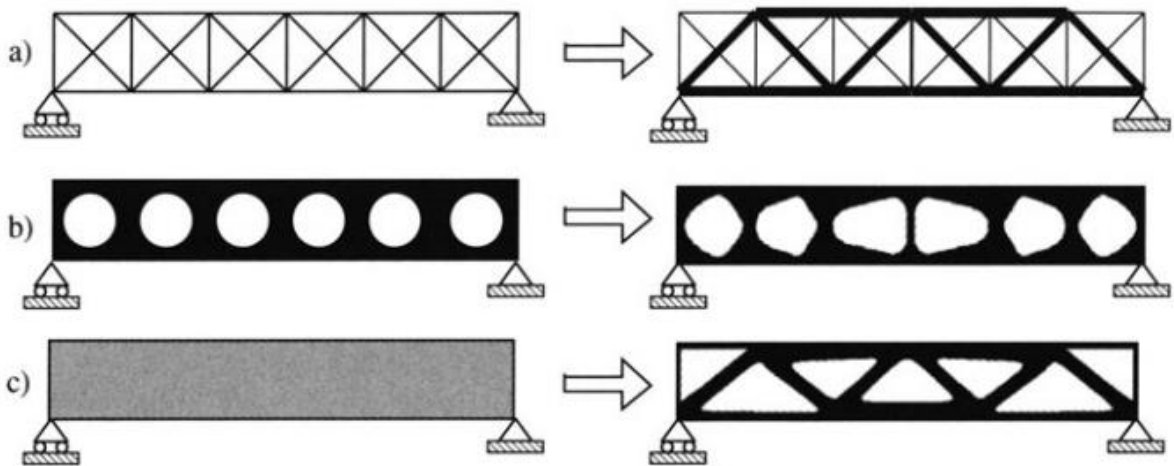
4.2. OPTIMIZACIÓN TOPOLÓGICA POR DISTRIBUCIÓN DE MATERIAL ISOTRÓPICO

Los problemas de optimización estructural se encuentran enmarcados por la disposición de las estructuras, en términos de su topología, su forma y su tamaño. De esta manera, un problema estructural puede ser resuelto siguiendo tres caminos: optimización de dimensión, de forma, o topológica. Cada uno atiende diferentes aspectos del diseño.

Por un lado, en un problema tradicional de dimensionamiento se buscaría el área óptima de los elementos de una armadura. En este caso, se minimiza una cantidad mientras se satisfagan las restricciones impuestas; de tal manera que se conocen el dominio de diseño y las variables de estado. Por otro lado, en una optimización de forma se averigua la forma óptima de aquel dominio, es decir, el problema de

forma se encuentra en un dominio que es ahora una variable de diseño. Finalmente, la optimización topológica determina el número, ubicación y forma de orificios, y la conectividad del dominio. A continuación, se describe gráficamente la diferencia entre los enfoques mencionados.

Fig. 2 Categorías de optimización estructural. a) Optimización de dimensión de una armadura, b) optimización de forma y c) optimización topológica. Fuente: M.P. Bendsoe (2004).



Siguiendo esta línea, a través del método de distribución de material, se satisfacen estos tres problemas simultáneamente.

4.2.1. Formulación del problema: Un problema de optimización topológica basado en el enfoque de distribución de material donde se busca la máxima rigidez global bajo ciertas restricciones de recursos se puede escribir como:

$$\min_x c(x) = U^T K U = \sum_{e=1}^N (X_e)^p u_e^T k_0 u_e$$

Sujeto a:

$$\frac{V(x)}{V_0} = f$$

$$KU = F$$

$$0 < x_{min} \leq x \leq 1$$

Donde U y F son los vectores globales de desplazamiento y fuerza, K es la matriz global de rigidez, u_e y k_e son el vector de desplazamiento y la matriz de rigidez de los elementos. Mientras tanto, x es el vector de variables de diseño, x_{min} es un vector de densidades relativas mínimas, N es el número de elementos del dominio de diseño, p es la potencia de penalización (usualmente $p=3$), $V(x)$ y V_0 es el volumen de material y el volumen del dominio de diseño, y f es la fracción de volumen definida.

4.2.2. Método de solución: El problema de optimización descrito anteriormente puede ser solucionado desde diversos enfoques matemáticos. Sin embargo, como escribe O. Sigmund en su trabajo sobre este tema; por simplicidad, se usará un método estándar de criterio óptimo (Optimality Criteria method)³.

En este sentido, Bendsoe ha formulado un esquema para las variables de diseño, como se muestra:

$$x_e^{nuevo} = \begin{cases} \max(x_{min}, x_e - m) & \text{si } x_e B_e^\eta \leq \max(x_{min}, x_e - m), \\ x_e B_e^\eta & \text{si } \max(x_{min}, x_e - m) < x_e B_e^\eta < \min(1, x_e + m), \\ \min(1, x_e + m) & \text{si } \min(1, x_e + m) \leq x_e B_e^\eta \end{cases}$$

³ O. Sigmund, "A 99-line topology optimization code written in Matlab", Springer-Verlag, Struct Multidisc Optim, 21, 120-127 (2001). Página 121.

Siendo m (movimiento) un límite positivo de movimiento, $\eta \left(= \frac{1}{2} \right)$ un coeficiente de amortiguamiento numérico y B_e se halla de la siguiente manera:

$$B_e = \frac{-\frac{\partial c}{\partial x_e}}{\lambda \frac{\partial V}{\partial x_e}}$$

En este caso, λ es un multiplicador de Lagrange que puede ser encontrado por medio de un algoritmo de bisección.

Por su parte, se puede averiguar la sensibilidad de la función objetivo como sigue:

$$\frac{\partial c}{\partial x_e} = -p(x_e)^{p-1} u_e^T k_0 u_e$$

4.2.3. Existencia de soluciones: Con el fin de garantizar la existencia de soluciones para un problema de optimización topológica, se debe agregar una restricción al diseño resultante; utilizando una técnica de filtrado⁴.

Dicho filtro llamado de “independencia de malla”⁵ funciona a través de la modificación de la sensibilidad de los elementos, definida como:

⁴ Para estudios más profundos sobre este tema, ver: Sigmund, O. Design of material structures using topology optimization. Ph.D. Thesis, Department of Solid Mechanics, Technical University of Denmark. 1994.

⁵ O. Sigmund, “A 99-line topology optimization code written in Matlab”, Springer-Verlag, Struct Multidisc Optim, 21, 120-127 (2001). Página 122.

$$a_q \frac{\partial c}{\partial x_e} = \frac{1}{x_e \sum_{f=1}^N \widehat{H}_f} \sum_{f=1}^N \widehat{H}_f x_f \frac{\partial c}{\partial x_f}$$

Aquí, \widehat{H}_f es el operador de convolución escrito como sigue:

$$\widehat{H}_f = r_{min} - dist(e, f) \quad \{f \in N \mid dist(e, f) \leq r_{min}\}, \quad e = 1, \dots, N.$$

Donde el operador $dist(e, f)$ es la distancia entre el centro del elemento e y el centro del elemento f . Mientras tanto, el operador de convolución \widehat{H}_f es cero (0) fuera del área de filtrado.

5. METODOLOGÍA

El problema tratado en este trabajo se ha enfocado desde la perspectiva del diseño tradicional. En este sentido, se presenta el proceso de análisis y decisión de alternativas, llevando a cabo una evaluación QFD para dar validez al dictamen hecho. Posteriormente, se describe la forma en la que se implementa la alternativa escogida y finalmente el tipo de validación de resultados a ejecutar.

5.1. ANÁLISIS DE ALTERNATIVAS DE SOLUCIÓN

5.1.1. Alternativa 1: Algoritmo basado en el manejo de las funciones matemáticas para la optimización estructural.

De acuerdo con R.A. MERIC en *material and load optimization of thermoelastic solids*, es posible estimar el estado de cargas sobre el contorno de un cuerpo, a través del manejo de las funciones objetivo definidas por el diseñador.

En otras palabras, de la misma forma en la que es llevada a cabo la optimización de forma de un cuerpo; en la cual, la variable a optimizar es su contorno, teniendo en cuenta restricciones que comprenden parámetros como un valor de esfuerzo admisible y las propiedades de un material; es posible también tomar como variable de diseño el estado de cargas sobre él, enviando al terreno de los parámetros constantes su contorno.

Sin embargo, la mayor dificultad para su realización reside en la complejidad de la manipulación de dichas funciones matemáticas, ya que requieren de un manejo a cabalidad de los conceptos de la mecánica computacional e incluso implica conocimientos significativos de la mecánica del medio continuo. Cabe resaltar que estas temáticas se encuentran por fuera del alcance de este trabajo.

5.1.2. Alternativa 2: Algoritmo iterativo con base en métodos de optimización de forma.

El problema a resolver implica conocer inicialmente ciertos parámetros del cuerpo a examinar, tales como su geometría y el material del cual se compone. De esta manera, se propone llevar a cabo un proceso iterativo, en el cual sean reproducidas varias fases de optimización de forma, cada una relacionada con la anterior, hasta conseguir la geometría que se ajuste, dentro de un rango de exactitud, al modelo mecánico conocido.

La mayor bondad de esta alternativa se encuentra en que está basada en las metodologías ampliamente investigadas por diversos autores a lo largo de al menos treinta años, razón por la cual existe una inmensa cantidad de ejemplos sobre los cuales escribir un código de optimización de forma, el cual, por su parte, debe acoplarse de la mejor manera al desarrollo iterativo llevado a cabo para solucionar el problema.

En contraparte, es posible que los requerimientos computacionales para la implementación sean altos con respecto a otra alternativa, esto debido a la iteración de un procedimiento que en sí mismo, requiere de una medianamente alta capacidad. En este caso, es necesario en una escala superior, un proceso de optimización de código, con el objetivo de minimizar la demanda computacional y aumentar la eficiencia.

5.1.3. Elección de Alternativa:

Con el fin de escoger la alternativa que más convenga para la consecución del proyecto, se ha llevado a cabo una evaluación basada en el QFD, acompañado de una evaluación Pugh. El procedimiento llevado a cabo se puede examinar en el anexo A.

De acuerdo al proceso de selección mencionado anteriormente, la alternativa 2 es la más apropiada para ser desarrollada en el proyecto.

5.2. IMPLEMENTACIÓN

Para la implementación del algoritmo, se va a realizar un código en Matlab® que interactúe con ANSYS®, de tal manera que tomando inicialmente la forma óptima del componente y estableciendo su dominio de diseño, se realice al análisis de elementos finitos a través del segundo, alimentando al primero para llevar a cabo la solución del problema de estimación de cargas.

Con este fin, se tomará como base un algoritmo de código abierto de optimización topológica⁶, el cual es manipulado para introducir en él los resultados necesarios obtenidos vía MEF. A su vez, éste último proceso es llevado a cabo por medio de un código de interacción entre Matlab® y ANSYS®, como se mencionó con anterioridad; recibiendo los datos topológicos del problema y entregando los resultados pertinentes a la solución del problema de optimización. El código editado de optimización topológica y el código del proceso por MEF son escritos en Matlab® y adjuntados en los anexos B y C, respectivamente.

Además de esto, vale destacar que ambos softwares son ampliamente utilizados académica e industrialmente, aspecto que facilita su implementación en diferentes máquinas y eventualmente permitirá extrapolar con mayor sencillez el algoritmo elaborado.

⁶ SIGMUND, Ole. A 99-line topology optimization code written in Matlab. Structural and multidisciplinary optimization, 2001, vol. 21, no 2, p. 120-127.

5.3. VALIDACIÓN DE RESULTADOS

La validación de los resultados obtenidos requiere del conocimiento previo de los mismos para el problema enunciado.

Para este caso, se ha propuesto la validación a través de la aplicación del algoritmo sobre varios problemas de optimización de forma de una viga, con diferentes valores de magnitud y posición de la carga. En este sentido, se espera que los resultados sean muy similares con respecto a los encontrados y en caso contrario, se hace posible reconocer los errores en el algoritmo.

Finalmente, la expectativa en una escala próxima es traer a colación el algoritmo en cuerpos de mayor complejidad, entregando resultados con mayor validez en situaciones de la realidad.

6. DESARROLLO DEL ALGORITMO

Para introducirse de mejor manera al algoritmo desarrollado, se debe definir el proceso de estimación de la carga como un conjunto de subprocesos que incluye la optimización de forma del componente (en este caso tomado desde el enfoque “Power-law” o SIMP) a partir de una carga puntual aleatoriamente escogida, y la comparación que debe hacerse después de la forma que se ha extraído del proceso anterior, con respecto a la forma real -o modelo experimental- del componente.

6.1. VARIABLES DE ENTRADA Y SALIDA

Antes de abordar en detalle las secciones que componen el algoritmo desarrollado, hace falta definir las variables de entrada y salida del mismo.

Por un lado, aunque anteriormente se ha hecho notar que el trabajo consiste en estimar la carga de un componente basado en su forma óptima, es decir, es necesario conocer la geometría real de éste; también lo es definir el dominio de diseño dentro del cual se encuentra aquel componente.

El dominio de diseño se precisa a partir de las dimensiones máximas del elemento a analizar. De esta manera, si se está trabajando sobre un componente definido en el plano en coordenadas cartesianas, su dominio de diseño será el área determinada por la magnitud máxima del mismo en cada una de las direcciones del plano. Esto se puede visualizar de mejor manera con la siguiente ilustración.

La importancia de delimitar correctamente el dominio de diseño reside en que el enfoque de optimización topológica utilizado para este trabajo necesita de esta información para funcionar. Por su parte, conocer la geometría real o efectiva del componente es de vital importancia para llevar a cabo el módulo de comparación, descrito con posterioridad.

Por otro lado, al término del algoritmo se halla la posición y magnitud estimada de la carga que debió actuar sobre el elemento para causar su forma efectiva. Vale aclarar que, para facilidad de la solución y demostración del algoritmo, se han tomado restricciones de varios tipos.

6.2. RESTRICCIONES Y CONDICIONES

Las restricciones deben ser listadas, de tal manera que se puedan describir fácilmente y reconocer su aparición en la solución del problema conjeturado.

- El componente se encuentra definido en coordenadas cartesianas.
- El componente es una viga empotrada en voladizo de espesor constante.
- Sobre el componente se aplican cargas estáticas concentradas en dirección perpendicular al eje axial del mismo.

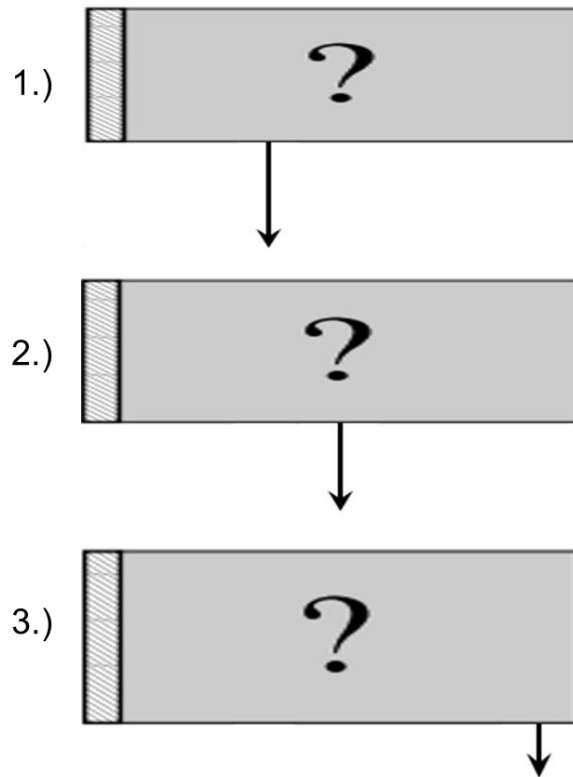
Para los ejemplos de validación, una carga es ubicada a $(x)L$ medida desde el lado fijo, donde L es la longitud de la viga; con (m) unidades de fuerza de magnitud, siendo arriba positivo. En la tabla 1 se presenta la configuración de cargas de referencia a utilizar.

Tabla 1 Definición de ejemplos de validación

Ejemplo	Posición $(x)L$	Magnitud $(m)[uf]$
1	0,35	-355
2	0,5	-200
3	0,92	-50

Las topologías óptimas para estas configuraciones son presentadas en la Fig. 3.

Fig. 3 Condiciones iniciales para las vigas en voladizo de los ejemplos de validación; 1.) Ejemplo 1, 2.) Ejemplo 2, 3.) Ejemplo 3. Fuente: Autor.



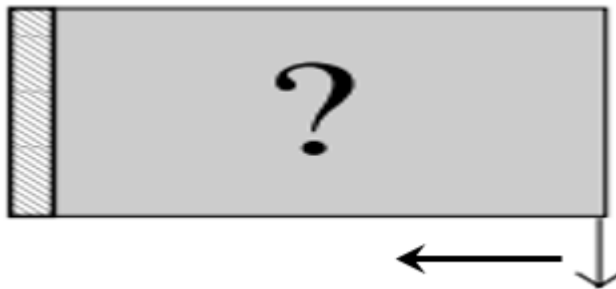
6.3. EL ALGORITMO

En el algoritmo se pueden diferenciar tres secciones o módulos principales, cada uno de los cuales realiza una sucesión de operaciones que establecen los parámetros para el funcionamiento del siguiente. Ellos son reconocibles a pesar de saber que el proceso es, en realidad, un bucle que requiere presuntamente de varias iteraciones para conseguir un resultado satisfactorio. Por esta razón, las secciones descritas se deben entender como un proceso lineal enmarcado dentro del proceso iterativo global.

6.3.1. Ubicación de carga sobre el campo geométrico del componente: Este módulo se puede describir inicialmente como la cabeza del algoritmo de estimación como tal, ya que en él se ubica la carga en un punto del dominio de diseño del componente.

Inicialmente, la carga es ubicada en el punto de la dirección axial igual a cero (0), siendo éste el paso cero del bucle. Si a lo largo del proceso se encuentra que debe realizarse una iteración más, la carga se ubicará en el siguiente nodo del modelo computacional en la misma dirección y llevando a cabo todo el proceso de la misma manera.

Fig. 4 Movimiento de la carga para una viga empotrada en voladizo. Fuente: Sigmund, Ole., 2001. Editado por el autor.



El movimiento que se ha establecido para la carga en cada paso se explica por su facilidad en la implementación para un caso sencillo como éste. Vale la pena manifestar que un estudio más profundo, fuera del alcance de este trabajo, debe ser llevado a cabo para optimizar este proceso, con el fin de evitar gastos computacionales excesivos.

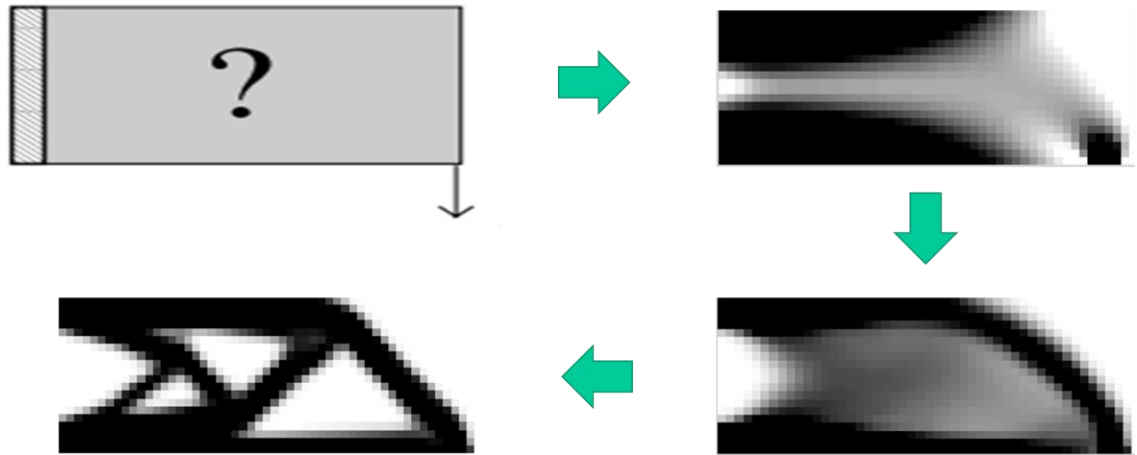
6.3.2. Optimización topológica: Como se ha dicho con anterioridad, se ha escogido para la implementación de este algoritmo el enfoque de optimización “Power-law” o SIMP, debido a su capacidad de ejecución en problemas de este nivel.

Al abordar el problema de optimización desde este punto de vista, se deben ingresar como datos la carga impuesta, definida en el módulo anterior; el dominio de diseño del componente, determinado en los pasos preliminares del proceso; y las restricciones de minimización, valores que son recomendados por el autor del enfoque seleccionado.

A lo largo de este proceso, como se ha mencionado antes, se concibe la idea de densidad relativa, concerniente al material dedicado a cada elemento del modelo de elementos finitos de acuerdo a los esfuerzos que recibe. De esta manera, al final de la optimización topológica habrá elementos que contengan material, así como otros que no lo tengan. Sin embargo, para esta aplicación aquellos elementos no dejan de existir, sino que simplemente pasan a llamarse elementos vacíos. De esta manera, se mantiene el dominio de diseño original del componente y se puede elaborar un modelo de comparación que permita dar con la solución del problema.

Debido a la interacción requerida entre este algoritmo de optimización escrito en *Matlab*, y *Ansys*; éste último realiza el análisis de elementos finitos sobre el dominio de diseño inicialmente definido, para posteriormente entregar resultados de desplazamientos en cada nodales al proceso global de optimización. Éste se realiza iterativamente hasta encontrar la mejor distribución de material sobre el dominio de diseño establecido. Por este motivo, *Ansys* actúa como una función en la optimización topológica escrita en *Matlab* y permite la capacidad de trabajar sobre un rango más amplio de geometrías y condiciones. Los anexos B y C dan cuenta clara del procedimiento llevado a cabo. Allí, el análisis de métodos finitos por medio de *Ansys* es una subrutina de la optimización topológica.

Fig. 5 Descripción gráfica del proceso de optimización topológica en su fase inicial, intermedia y su resultado final. Fuente: Autor.

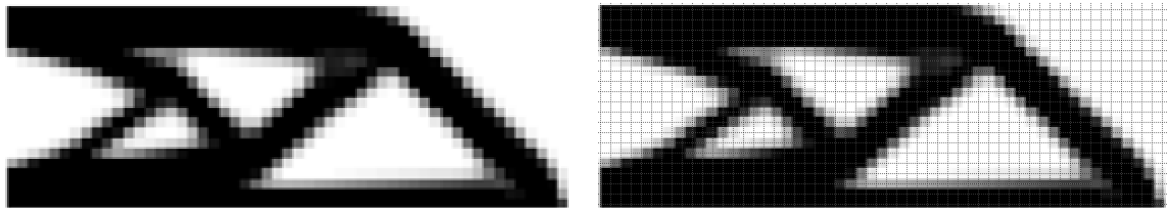


Como se puede observar en el anexo C, en primera instancia se escribe un archivo de texto desde *Matlab* para ser leído y ejecutado por *Ansys*. En este archivo se define el dominio de diseño del problema de acuerdo a las coordenadas de cada nodo y sus respectivos elementos. Posteriormente, se especifican las propiedades de material, es decir, módulo de elasticidad y coeficiente de Poisson, dependiendo de la distribución de material en cada uno de los elementos; esto puntualizado previamente por el algoritmo de optimización topológica. Acto seguido, a cada elemento se le precisa un material de manera específica, de acuerdo a dicha distribución de material sobre el dominio de diseño. A partir de allí, se solicita el tipo de resultados a exportar, en este caso los desplazamientos en cada nodo. Finalmente, se ejecuta *Ansys* en su modo *batch* y se exportan los datos obtenidos hacia *Matlab* ya como arreglos numéricos.

6.3.3. Comparación de modelos: A partir de la geometría óptima, en sentido topológico, entregada por el módulo anterior; se puede modelar una metodología de comparación con respecto al componente real, con la cual se defina en qué momento se ha conseguido una solución aceptable.

Como se ha dicho anteriormente, el dominio de diseño del componente se mantiene constante, por lo cual se deben definir elementos que contengan material, mientras otros serán vacíos. En este sentido, es posible realizar una comparación elemento a elemento entre el modelo generado y el modelo real del componente, a través de la discretización de la estructura de referencia.

Fig. 6 La topología de referencia (izquierda) y la construcción de la matriz de densidades (derecha). Fuente: Autor.



Esta comparación se hace creando una matriz de tamaño igual al número de elementos del dominio de diseño del componente, la cual se rellena con valores uno (1) y cero (0) en la posición de cada elemento, dependiendo de un criterio de similitud como se muestra a continuación.

$$R_e = \begin{cases} 1 & \text{if } A_e = B_e \\ 0 & \text{if } A_e \neq B_e \end{cases}$$

En estos términos, el proceso de estimación terminará cuando esta matriz, llamada matriz de comparación, se componga exclusivamente de unos (1) en todas sus posiciones. Sin embargo, debe agregarse un porcentaje de error máximo de conformidad para la solución del problema de estimación en caso tal que no se

encuentre una solución exacta, de acuerdo a la matriz de comparación; ya sea por falta de finura de la malla modelada o cualquier otro factor.

$$SIMF = \frac{\#unos}{\#elementos}$$

En el anexo D se encuentra el código de Matlab para la estimación de cargas.

Además de esto y con el fin de sintetizar la información expuesta, se presenta un diagrama de flujo del mismo en el anexo E.

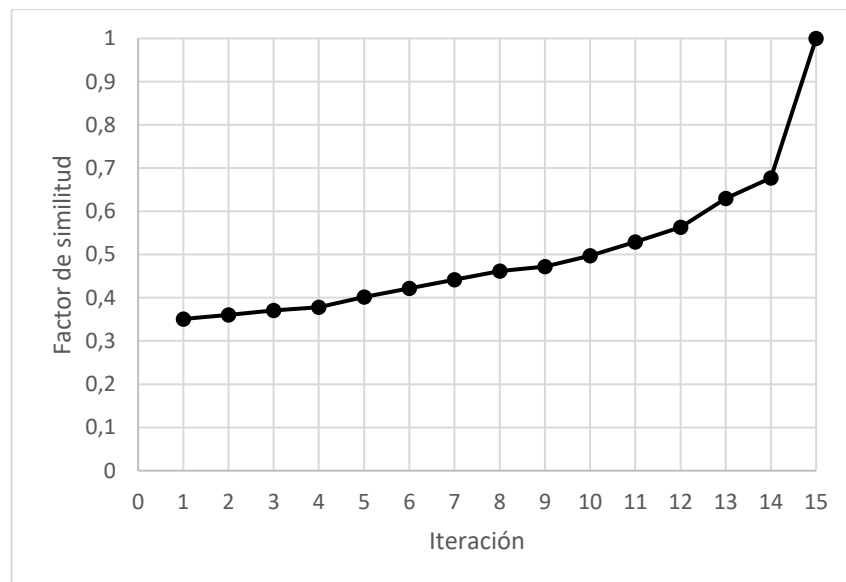
7. RESULTADOS

Como fue expuesto con anterioridad, el algoritmo es probado para un componente estructural simple; una viga en voladizo soportando una carga concentrada. La información de este proceso ha sido recolectada y graficada para este ejemplo, incluyendo la topología de referencia de la estructura, el dominio de diseño y datos mostrando la localización estimada de la carga.

En primer lugar, el algoritmo es comprobado en piezas, buscando en el primer intento solo la ubicación de la carga, suponiendo una carga inicial. En este caso, una carga de 1 unidad de fuerza es supuesta, estando localizada a $3L/4$ del punto fijo. De este modo, la localización estimada de la carga es $0,767L$.

En la Fig. 7, un gráfico manifiesta la evolución del factor de similitud con respecto a las iteraciones. Una curva creciente suave es detallada.

Fig. 7 Factor de similitud vs. Iteración cuando la magnitud de la carga es conocida. Fuente: Autor.

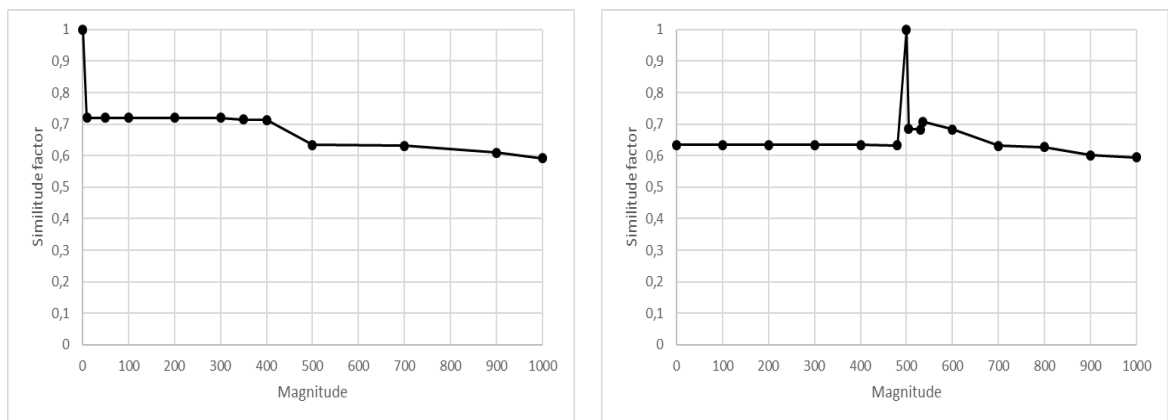


Por otro lado, se hace un análisis para diferentes magnitudes de carga de referencia, todas ellas localizadas en el mismo punto sobre la estructura. De la

misma forma, se muestra el comportamiento para magnitudes de referencia de 1 y 500 unidades de fuerza, ubicada a $0,92L$ del lado fijo.

Una conducta interesante es encontrada en la variación de la topología óptima de la estructura a medida que la magnitud de la carga soportada se altera en un rango relativamente amplio.

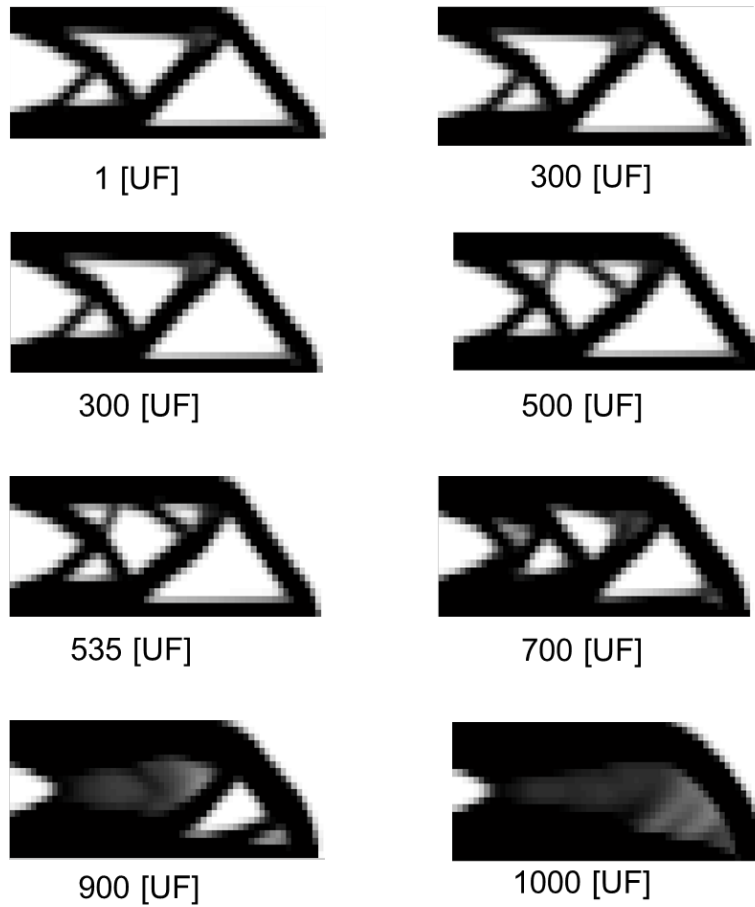
Fig. 8 Gráficas de comportamiento para magnitudes de referencia de 1 (izquierda) y 500 (derecha) unidades de fuerza. Fuente: Autor.



La Fig. 8 representa la variación del factor de similitud de acuerdo a los cambios en la magnitud, los cuales son comparados con un valor de referencia. Estos datos muestran un comportamiento llamativo, debido a que se encuentra un rango con valores casi constantes en el factor de similitud, donde incluso con modificaciones significativas en la magnitud, éste se mantiene en un valor promedio de 65% de similitud.

Mientras tanto, cuando la magnitud converge hacia el punto de referencia, se encuentra un pico que inmediatamente después es seguido por una meseta. No obstante, empiezan a aparecer diferencias sustanciales al traspasar el rango de las 500 unidades de fuerza en ambos límites de la referencia.

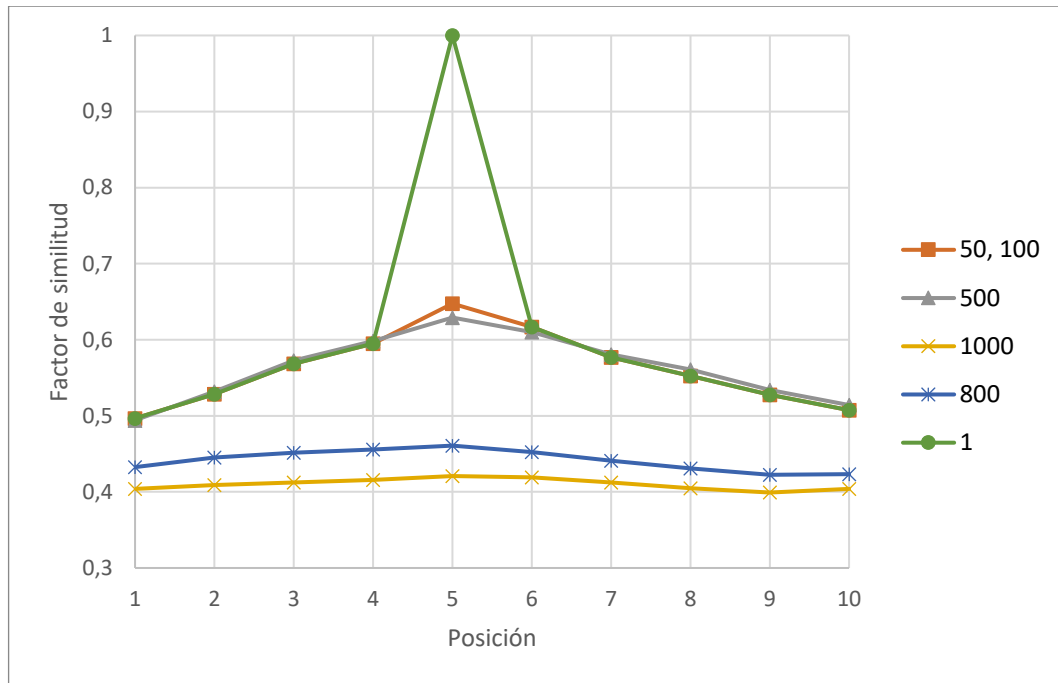
Fig. 9 Topologías óptimas de una viga empotrada a cargada a diferentes magnitudes y en la misma posición. Fuente: Autor.



Dado el comportamiento mostrado por la variable de la ubicación de la carga, el algoritmo realiza una búsqueda de la magnitud de ésta después de encontrar su posición. De esta manera, se hizo necesario clarificar la noción de la variación del factor de similitud con respecto a la magnitud y la posición de manera simultánea, sintetizando la dinámica de la metodología.

Así, la Fig. 10 representa aquella conducta para 1 unidad de fuerza como referencia de magnitud, estando a $0,92L$ del lado fijo, es decir, el quinto nodo desde el lado libre.

Fig. 10 El factor de similitud para varia magnitudes y posiciones con un punto de referencia de magnitud =1 [uf]. Fuente: Autor.



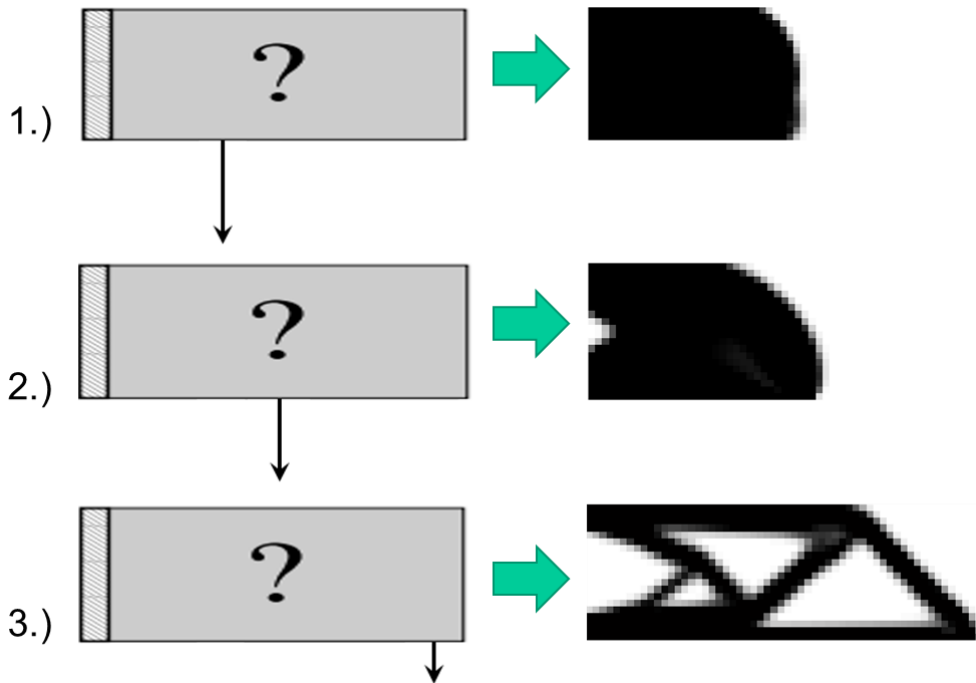
Finalmente, el proceso en su conjunto es ejecutado para los problemas de validación especificados en el capítulo anterior. A continuación, se muestra una tabla con los resultados obtenidos, además del tiempo de ejecución para cada uno de ellos.

Tabla 2 Resultados de los ejemplos de validación definidos en el capítulo 6. L es la longitud total del dominio de diseño; x es la posición estimada la carga; m es la magnitud de la fuerza; uf representa unidades de fuerza.

Ejemplo	Posición (x)	Magnitud (m)[uf]	Tiempo de ejecución [s]
1	0,35L	-355	337,85
2	0,5L	-200	230,49
3	0,92L	-50	254,59

Las topologías finales de los modelos estimados son presentadas en la Fig. 11.

Fig. 11 Dominios de diseño y topologías finales de los modelos estimados; 1.) Ejemplo 1, 2.) Ejemplo 2, 3.) Ejemplo 3. Fuente: Autor.



Cabe resaltar que los resultados obtenidos aportan, además, un proceso de interpretación complementario que permite aclarar y corroborar la metodología desarrollada.

Aunque esta metodología puede ser implementada individualmente sobre cualquier eje, se hace notar la necesidad posterior de avanzar hacia la estimación de cargas en las 3 direcciones coordenadas en conjunto.

De esta manera, se sientan las bases metodológicas y numéricas para la elaboración de modelos más completos, acercando la perspectiva a la estimación de cargas de componentes estructurales biológicos, accediendo a mayor cantidad de información relativa al comportamiento mecánico de los mismos. Por consiguiente, los análisis posteriores deben considerar las propiedades propias de

los materiales orgánicos, aquí sólidos, con el fin de asegurar mayor cercanía a los datos obtenidos experimentalmente.

8. ANÁLISIS DE RESULTADOS

Los resultados más tempranos, presentados en la Fig. 7, muestran que existe un crecimiento suave en el factor de similitud mientras los pasos iterativos se acercan a las soluciones. Este comportamiento significa que, para variaciones en la ubicación de la carga, la topología sigue una tendencia hacia la convergencia, la cual podría estar limitada por la resolución de la malla utilizada para hacer discreto el modelo. Como puede observarse en la Fig. 10, esta tendencia sigue una curva cercana a la campana de Gauss, anticipando que esta conducta puede ser mejor representada por medio de esta herramienta matemática; sin embargo, complementada con el comportamiento seguido por la variación de la magnitud de la carga en sí misma.

Avanzando hacia el análisis de la magnitud de la carga, la estabilidad en valores constantes del factor de similitud puede representar la resiliencia del material con respecto a las variaciones de magnitud, ya que mientras éste último se aleja del punto de referencia, hay una desviación importante, marcando el final del rango de tipo meseta, lo cual expresa que existe una necesidad casi espontánea de mayor cantidad de material para manejar los esfuerzos generados por el estado de cargas. Esto puede ser advertido en la Fig. 8 (izquierda) al llegar a las 500 unidades de fuerza.

En este sentido, la magnitud varía de una manera considerablemente diferente a la posición de la carga, manifestando una casi discontinuidad al tocar los valores de referencia, mientras que la variable posición se modela como una curva más suave para conseguir una solución.

Cuando ambas variables se conjugan en un algoritmo, se encuentra un proceso de doble iteración; el primero, como se puede observar en cada línea de la Fig. 10, donde el mayor factor de similitud es buscado; y el segundo, acercando la vista a

los puntos in la posición 5 de esta figura, donde el máximo factor de similitud global es obtenido. Siguiendo esta línea, aquel gráfico expresa cómo los valores dentro de un rango de magnitud establecido anteriormente se logran confirmar, e incluso fuera de él, el comportamiento de topologías soportando otras magnitudes es el mismo. No obstante, aflora una tendencia hacia el aplanamiento en aquellas magnitudes más lejanas, exponiendo el hecho que los valores alejándose del rango de referencia pierde influencia en la búsqueda de semejanza.

Finalmente, los resultados obtenidos del problema de validación no son más que una garantía de la utilidad del algoritmo y una confirmación de la sinergia envuelta en el proceso de doble iteración antes mencionado.

Es además importante observar que el algoritmo ofrece una precisión suficientemente buena para una investigación inicial. A pesar de ello y como se esperó desde el principio, el tiempo de ejecución puede ser un tema relevante in estructuras de mayor complejidad, tales como componentes 3D y topologías 2D de alta complejidad.

Esta carencia de velocidad en el proceso puede ser una respuesta al enfoque de búsqueda utilizado, debido a que éste es un método que puede promover el aumento o disminución del tiempo de ejecución dependiendo de la intuición humana, en términos de la colocación de los primeros puntos de iteración, influenciando directamente la cantidad de pasos necesarios para encontrar una solución óptima. Sin embargo, en problemas sencillos como el estudiado anteriormente, el tiempo no representa una dificultad significativa.

Más allá de esto, la velocidad puede estar también comprometida, debido a la variabilidad en el poder de cálculo in los distintos dispositivos computacionales.

Dado que no se han hallado estudios numéricos previos enfocando el tema tratado en este trabajo, éste se encuentra abierto a optimizaciones en todos los módulos tal

como en la metodología en sí, buscando un mejor sistema para optimizar el tiempo de trabajo.

9. CONCLUSIÓN

Las estructuras biológicas se caracterizan por su complejidad y dificultad para comprenderlas a cabalidad, ya que existe un gran número de variables actuando en ellas y se hace problemática la medición de éstas últimas. Por medio de la hipótesis que expresa que aquellas estructuras son respuestas a problemas de optimización, hay una ventana hacia la obtención de variables tales como los estados de cargas, estimándolas. Mediante el uso de un enfoque de optimización topológica, un algoritmo iterativo es construido para resolver este problema.

Como puede ser deducido, la dificultad de actuar directamente en una estructura biológica, como el hueso, es formidable. En este sentido, se apunta inicialmente a resolver aquella pregunta para un componente estructural simple, una viga en voladizo, con una carga concentrada actuando sobre ella.

No obstante, los resultados obtenidos han entregado información significativa concerniente al comportamiento de la posición y magnitud de una carga soportada, cada una aparte y finalmente su trabajo conjunto en el proceso de solución; concluyendo que estas variables tienen conductas disímiles, la primera aproximándose a una curva gaussiana, mientras que la última se acerca a una función constante hasta encontrar un pico en los alrededores inmediatos a la solución.

El entendimiento de estas variables hizo posible la concreción del proceso de doble iteración; encontrando, primeramente, la ubicación de la carga para posteriormente proceder a la búsqueda de la magnitud que se asemejara de mejor manera a una topología de referencia.

Empero, una inquietud se mantiene viva con respecto al tiempo necesitado para obtener resultados óptimos, el cual en dominios de mayor complejidad puede

aumentar puede representar una cuestión a tener en cuenta. Por esta razón, se pueden aplicar enfoques más eficientes de búsqueda para resolver estos problemas, reduciendo la cantidad de pasos para conseguir soluciones u optimizando la velocidad de cálculo en sí misma.

Por otro lado, se hace necesario aclarar las limitaciones que han restringido el desarrollo de este trabajo; entre las cuales se encuentra la capacidad computacional con la cual se contó, debido a que el uso de *Ansys* para realizar análisis de elementos finitos requiere de un alto uso de recursos y debido a que el enfoque de optimización de forma utilizado itera sobre un análisis nuevo en cada paso, este se hace muy elevado. No obstante, esta situación puede mejorarse optimizando el uso de *Ansys*, por medio de la omisión de pasos en el código de interacción con *Matlab* (Anexo C), los cuales podría no haber necesidad de definir en cada iteración del proceso. Vale la pena agregar que el uso de *Ansys* implica cierto grado de error en su seno, de acuerdo a la finura de la malla construida. Por tal motivo, debe indagarse sobre la exactitud de los resultados a obtener en casos de mayor complejidad, logrando para cada uno de ellos la mejor malla posible.

Asimismo, se debe resaltar la prioridad en la elaboración de una metodología que sirviera como base para las fases posteriores de un proyecto de mayor envergadura, por lo cual la implementación ha sido limitada a la estimación sobre cargas en un plano, aunque puede ser extrapolado a las demás dimensiones. Esto puede realizarse fácilmente, cambiando los nodos sobre los cuales se itera la posición de la carga en el anexo D; en otras palabras, tomando el rango de nodos que estén localizados sobre la dirección de estimación deseada. Sin embargo, este proceso excede el alcance del presente trabajo.

Por otro lado, debe puntualizarse la capacidad limitada del enfoque de optimización topológica implementada a lo largo del trabajo, ya que éste sólo trabaja adecuadamente para materiales isotrópicos. Debido a esto, si en trabajos

posteriores se busca la implementación en especímenes anisótropos, se debe considerar la posibilidad de editar el proceso de optimización o de cambiarlo por alguno más efectivo en estos casos; trabajo que debe ser analizado profundamente en dicho escenario para obtener los resultados más certeros.

Más allá de esto, la ventana permanece abierta para investigaciones más profundas, teniendo en cuenta que aún hay muchos casos por cubrir, tales como cargas distribuidas, cargas de torsión y demás. Acercamientos tridimensionales representarían también grandes avances, recordando la meta de modelar las estructuras biológicas en su completitud.

BIBLIOGRAFÍA

ANSYS. Ansys mechanical APDL [Software para análisis de método de elementos finitos]. Última versión: 18. Requerimientos del sistema: Windows 7/8.1/10. (64 bits).

BENDSOE, Martin P. Optimization of structural topology, shape, and material. Berlin etc: Springer, 1995.

BENDSOE, Martin Philip; SIGMUND, Ole. Topology optimization: theory, methods, and applications. Springer Science & Business Media, 2013.

CHRISTENSEN, Peter W.; KLARBRING, Anders. An introduction to structural optimization. Springer Science & Business Media, 2008.

CURREY, J. D. The many adaptations of bone. Journal of biomechanics, 2003, vol. 36, no 10, p. 1487-1495.

CURREY, J. D. The adaptation of bones to stress. Journal of theoretical biology, 1968, vol. 20, no 1, p. 91-106.

FRATZL, Peter. Biomimetic materials research: what can we learn from nature's structural materials? Journal of the Royal Society Interface, 2007, vol. 4, no 15, p. 637-642.

FROST, H. M. Skeletal structural adaptations to mechanical usage (SATMU): 1. Redefining Wolff's law: the bone modeling problem. The Anatomical Record, 1990, vol. 226, no 4, p. 403-413.

HAFTKA, Raphael T.; GRANDHI, Ramana V. Structural shape optimization—a survey. Computer methods in applied mechanics and engineering, 1986, vol. 57, no 1, p. 91-106.

HAFTKA, Raphael T.; GÜRDAL, Zafer. Elements of structural optimization. Springer Science & Business Media, 2012.

HERNANDEZ, C. J., et al. The influence of bone volume fraction and ash fraction on bone strength and modulus. *Bone*, 2001, vol. 29, no 1, p. 74-78.

LÓPEZ, Santos. Optimización de componentes mecánicos mediante algoritmo genético y software comercial de elementos finitos. 2008. Tesis Doctoral. Tesis de Máster, Universitat Politècnica de València.

MATHWORKS. Matlab R2016b [Software matemático que ofrece un entorno de desarrollo integrado]. Última versión: R2017a. Requerimientos del sistema: Windows 7/8/8.1/10.

MERIC, R. A. Material and load optimization of thermoelastic solids. Part I: sensitivity analysis. *Journal of Thermal Stresses*, 1986, vol. 9, no 4, p. 359-372. Part II: Numerical results. p. 373-388.

O.C. Zienkiewicz and R.L. Taylor. *The finite element method*, McGraw Hill, Vol. I., 1989, Vol. II, (1991).

PEDERSEN, Pauli. Optimal Designs-Structures and Materials-Problems and Tools. 2003.

PIRES, Joao Pedro. Structural Optimization using the Finite Element Method. Trabajo de Grado Magister en Ingeniería Civil. Lisboa: Universidade Católica Portuguesa. Facultad de Ingeniería. 2014.

SIGMUND, Ole. A 99-line topology optimization code written in Matlab. *Structural and multidisciplinary optimization*, 2001, vol. 21, no 2, p. 120-127.

ANEXOS

ANEXO A: QFD Y EVALUACIÓN PUGH

Tabla 3 Ponderación de Criterios

Ponderación de Criterios											
Requisito	Valoración	Eficacia		Rapidez		Intuitivo		Facilidad de Implementación		Costo Computacional	
Costo Computacional	5	1	5	5	25	3	15	1	5	5	25
Confiabilidad	4	5	20	2	8	1	4	1	4	1	4
Rapidez	3	1	3	5	15	3	9	1	3	5	15
Interactividad	2	2	4	1	2	5	10	1	2	3	6
Aspecto	1	1	1	1	1	1	1	1	1	3	3
Totales			33		51		39		15		53

Tabla 4 Matriz de Decisión de Alternativas

Puntuación	Criterio	Porcentaje	Alternativa 1		Alternativa 2	
53	Costo Computacional	0,277	3	0,8325	2	0,555
51	Rapidez	0,267	4	1,0681	3	0,801
39	Intuitivo	0,204	2	0,4084	4	0,8168
32	Eficacia	0,168	5	0,8377	5	0,8377
15	Facilidad de Implementación	0,079	2,5	0,1963	5	0,3927
190	Totales	1,0		3,34		3,40

ANEXO B: CÓDIGO EDITADO DE OPTIMIZACIÓN TOPOLÓGICA

```

##### A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLE SIGMUND, JANUARY 2000
#####
##### CODE MODIFIED FOR ANSYS INTERACTION, April 2017, BY CHRISTIAN
FLOREZ#####
function [x]=top(nelx,nely,volfrac,penal,rmin,pos,mag)
% INITIALIZE
x(1:nely,1:nelx) = volfrac;
loop = 0;
change = 1.;
% START ITERATION
while change > 0.02 %Edited by author to increase speed
    loop = loop + 1;
    xold = x;
% FE-ANALYSIS EDITED BY AUTHOR TO INTERACT WITH ANSYS
[U,x]=ANSYS1(nelx,nely,x,pos,mag);
% OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
[KE] = lk;
c = 0.;
for ely = 1:nely
    for elx = 1:nelx
        n1 = (nely+1)*(elx-1)+ely;
        n2 = (nely+1)* elx +ely;
        Ue = U([2*n1-1;2*n1; 2*n2-1;2*n2; 2*n2+1;2*n2+2; 2*n1+1;2*n1+2],1);
        c = c + x(ely,elx)^penal*Ue'*KE*Ue;
        dc(ely,elx) = -penal*x(ely,elx)^(penal-1)*Ue'*KE*Ue;
    end
end
% FILTERING OF SENSITIVITIES
[dc] = check(nelx,nely,rmin,x,dc);
% DESIGN UPDATE BY THE OPTIMALITY CRITERIA METHOD
[x] = OC(nelx,nely,x,volfrac,dc);
% PRINT RESULTS
change = max(max(abs(x-xold)));
disp([' It.: ' sprintf('%4i',loop) ' Obj.: ' sprintf('%10.4f',c) ...
      ' Vol.: ' sprintf('%6.3f',sum(sum(x))/(nelx*nely)) ...
      ' ch.: ' sprintf('%6.3f',change )])
% PLOT DENSITIES
colormap(gray); imagesc(-x); axis equal; axis tight; axis off;pause(1e-
6);
end
##### OPTIMALITY CRITERIA UPDATE
#####
function [xnew]=OC(nelx,nely,x,volfrac,dc)
l1 = 0; l2 = 100000; move = 0.2;
while (l2-l1 > 1e-4)
    lmid = 0.5*(l2+l1);
    xnew = max(0.001,max(x-move,min(1.,min(x+move,x.*sqrt(-dc./lmid)))));
    if sum(sum(xnew)) - volfrac*nelx*nely > 0;

```

```

        l1 = lmid;
    else
        l2 = lmid;
    end
end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% MESH-INDEPENDENCY FILTER
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [dcn]=check(nelx,nely,rmin,x,dc)
dcn=zeros(nely,nelx);
for i = 1:nelx
    for j = 1:nely
        sum=0.0;
        for k = max(i-floor(rmin),1):min(i+floor(rmin),nelx)
            for l = max(j-floor(rmin),1):min(j+floor(rmin),nely)
                fac = rmin-sqrt((i-k)^2+(j-l)^2);
                sum = sum+max(0,fac);
                dcn(j,i) = dcn(j,i) + max(0,fac)*x(l,k)*dc(l,k);
            end
        end
        dcn(j,i) = dcn(j,i)/(x(j,i)*sum);
    end
end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% FE-ANALYSIS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [U]=FE(nelx,nely,x,penal,pos,mag)
[KE] = lk;
K = sparse(2*(nelx+1)*(nely+1), 2*(nelx+1)*(nely+1));
F = sparse(2*(nely+1)*(nelx+1),1); U = zeros(2*(nely+1)*(nelx+1),1);
for elx = 1:nelx
    for ely = 1:nely
        n1 = (nely+1)*(elx-1)+ely;
        n2 = (nely+1)* elx +ely;          %Nodes & elements are defined column
wise from left to right%
        edof = [2*n1-1; 2*n1; 2*n2-1; 2*n2; 2*n2+1; 2*n2+2; 2*n1+1; 2*n1+2];
        K(edof,edof) = K(edof,edof) + x(ely,elx)^penal*KE;
    end
end
end
% DEFINE LOADS AND SUPPORTS (CANTILEVER BEAM)
F(pos,1) = mag;          %Applied force in the force vector%%%%%%%%%
fixeddofs = [1:2*(nely+1)];          %Constraints%%
alldofs = [1:2*(nely+1)*(nelx+1)];
freedofs = setdiff(alldofs,fixeddofs);
% SOLVING
U(freedofs,:) = K(freedofs,freedofs) \ F(freedofs,:);
U(fixeddofs,:)= 0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% ELEMENT STIFFNESS MATRIX
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [KE]=lk
E = 1.;
nu = 0.3;
k=[ 1/2-nu/6    1/8+nu/8 -1/4-nu/12 -1/8+3*nu/8 ...
    -1/4+nu/12 -1/8-nu/8  nu/6      1/8-3*nu/8];
KE = E/(1-nu^2)*[ k(1) k(2) k(3) k(4) k(5) k(6) k(7) k(8)
                  k(2) k(1) k(8) k(7) k(6) k(5) k(4) k(3)

```

k(3) k(8) k(1) k(6) k(7) k(4) k(5) k(2)
k(4) k(7) k(6) k(1) k(8) k(3) k(2) k(5)
k(5) k(6) k(7) k(8) k(1) k(2) k(3) k(4)
k(6) k(5) k(4) k(3) k(2) k(1) k(8) k(7)
k(7) k(4) k(5) k(2) k(3) k(8) k(1) k(6)
k(8) k(3) k(2) k(5) k(4) k(7) k(6) k(1)];

ANEXO C: CÓDIGO DE MEF CON INTERACCIÓN ENTRE MATLAB Y ANSYS

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%GET FEM ANALYSIS FROM ANSYS%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [U,x]=ANSYS1(nelx,nely,x,pos,mag)
eldim=0.01; %Element dimension
%Change working directory to desired place
cd 'd:\universidad\semestre 10\trabajo de grado\ejecución\ansys batch'
fileid = fopen('truss1.txt','wt'); %Open input text file
fprintf(fileid, '/prep7\n');
%Element types & material properties
fprintf(fileid, 'et, 1, plane182\n'); %Element type: 4 node el.
%Defining element types
penal=0.1:.1:1; %Defined for 10 material
steps
rr=size(penal,2);
for ii=1:rr
    fprintf(fileid, '%s %i %s %f\n','mp,ex',ii,',',penal(ii)^3);
    fprintf(fileid, '%s %i %s\n','mp,prxy',ii,',',0.3');
end
%Meshing
%Read nodes
fprintf(fileid,'nrrang,1,,1\n');
fnodes = fopen('nodes.txt','wt');
nnode=0;
coordx=0;
for elx=1:(nelx+1)
    coordy=0;
    for ely=1:(nely+1)
        nnode=nnode+1;
        fprintf(fnodes, '%8i %.13f %3s %.13f\n',nnode,...
            coordx, ' ', coordy);
        coordy=coordy-eldim;
    end
    coordx=coordx+eldim;
end
fprintf(fileid, 'nread,nodes,txt,\n');
%Read elements
fprintf(fileid,'errang,1,,1\n');
felm = fopen('elements.txt','wt');
el=0;
for eelx=1:nelx
    for eely=1:nely
        el=el+1;
        n1=(nely+1)*(eelx-1)+eely;
        n2=(nely+1)*eelx +eely;
        fprintf(felm, '%6i %5i %5i %5i %s %5i\n', (n1+1), (n2+1), n2, n1, ...
            ' 0 0 0 0 1 1 1 1 0',el);
    end
end
fprintf(fileid, 'eread,elements,txt,\n');

```

```

%Loads & constraints
%Constraints definition
fixednodes=1:(nely+1);
for fn=1:(nely+1);
    fprintf(fileid, '%1s %1.3f
%1s\n', 'd,', fixednodes(fn), ',all,0,,,,,,,,');
end
%Load positioning
fprintf(fileid, '%1s %1i %1s %1.3f %1s\n', 'f,', pos, ', fy,', mag, ',,,'');
fprintf(fileid, 'sbctran\n');
%Write geometry info
fprintf(fileid, 'cdwrite,geom,geometry,txt,,,,\n');
%Density matrix parallelism
elnum=0;
for i=1:nelx
    for j=1:nely
        elnum=elnum+1;
        if x(j,i)==1
            fprintf(fileid, '%1s %1i %1s %1i\n', 'emodif,', ...
                elnum, ',mat,', length(penal));
        elseif x(j,i)>=0.9
            if x(j,i)<1
                fprintf(fileid, '%1s %1i %1s
%1i\n', 'emodif,', elnum, ',mat,', 9);
            end
        elseif x(j,i)>=0.8
            if x(j,i)<0.9
                fprintf(fileid, '%1s %1i %1s
%1i\n', 'emodif,', elnum, ',mat,', 8);
            end
        elseif x(j,i)>=0.7
            if x(j,i)<0.8
                fprintf(fileid, '%1s %1i %1s
%1i\n', 'emodif,', elnum, ',mat,', 7);
            end
        elseif x(j,i)>=0.6
            if x(j,i)<0.7
                fprintf(fileid, '%1s %1i %1s
%1i\n', 'emodif,', elnum, ',mat,', 6);
            end
        elseif x(j,i)>=0.5
            if x(j,i)<0.6
                fprintf(fileid, '%1s %1i %1s
%1i\n', 'emodif,', elnum, ',mat,', 5);
            end
        elseif x(j,i)>=0.4
            if x(j,i)<0.5
                fprintf(fileid, '%1s %1i %1s
%1i\n', 'emodif,', elnum, ',mat,', 4);
            end
        elseif x(j,i)>=0.3
            if x(j,i)<0.4
                fprintf(fileid, '%1s %1i %1s
%1i\n', 'emodif,', elnum, ',mat,', 3);

```

```

        end
        elseif x(j,i)>=0.2
            if x(j,i)<0.3
                fprintf(fileid, '%1s %1i %1s
%1i\n', 'emodif', 'elnum', 'mat', '2');
            end
            elseif x(j,i)>=0.1
                if x(j,i)<0.2
                    fprintf(fileid, '%1s %1i %1s
%1i\n', 'emodif', 'elnum', 'mat', '1');
                end
            elseif x(j,i)<0.1
                fprintf(fileid, '%1s %1i %1s
%1i\n', 'emodif', 'elnum', 'mat', '1');
            end
        end
    end
end
fprintf(fileid, '/output,elprops,txt\n');           %List element attributes
fprintf(fileid, 'elist,all,,,0,0\n');
fprintf(fileid, '/out\n');
fprintf(fileid, 'finish\n');
%Solving
fprintf(fileid, '/solu\n');
fprintf(fileid, 'antype, static\n');
fprintf(fileid, 'ncnv,0,1.0e10,,, \n');
fprintf(fileid, 'solve\n');
fprintf(fileid, 'save\n');
fprintf(fileid, 'finish\n');
fprintf(fileid, '/post1\n');
fprintf(fileid, '/output,comp1_output,txt\n'); %List nodal x-
displacement
fprintf(fileid, 'prnsol,u,x\n');
fprintf(fileid, '/out\n');
fprintf(fileid, '/output,comp2_output,txt\n'); %List nodal y-
displacement
fprintf(fileid, 'prnsol,u,y\n');
fprintf(fileid, '/out\n');
fclose(fileid);
!"C:\Program Files\ANSYS Inc\v162\ansys\bin\winx64\ANSYS162.exe" -b -i
truss1.txt -o trusso.txt
%%%%%%%%Reading data%%%%%%%%
%Read x-displacements
[a]=readnfo('comp1_output.txt');
ux=a(:,2);
% %Read y-displacements
[b]=readnfo('comp2_output.txt');
uy=b(:,2);
% %Get material info
[c]=readnfo('elprops.txt');
mat=c(:,2);
%Generate a combined displacement vector
U=zeros;
for i=1:(nelx+1)*(nely+1)
    p=(2*(i-1))+1;

```

```

        U(p,1)=ux(i);
    end
    for j=1:(nelx+1)*(nely+1)
        q=2*(j);
        U(q,1)=uy(j);
    end
    elnum=0;
    for i=1:nelx
        for j=1:nely
            elnum=elnum+1;
            if mat(elnum)==1;
                x(j,i)=penal(1);
            elseif mat(elnum)==2;
                x(j,i)=penal(2);
            elseif mat(elnum)==3;
                x(j,i)=penal(3);
            elseif mat(elnum)==4;
                x(j,i)=penal(4);
            elseif mat(elnum)==5;
                x(j,i)=penal(5);
            elseif mat(elnum)==6;
                x(j,i)=penal(6);
            elseif mat(elnum)==7;
                x(j,i)=penal(7);
            elseif mat(elnum)==8;
                x(j,i)=penal(8);
            elseif mat(elnum)==9;
                x(j,i)=penal(9);
            elseif mat(elnum)==10;
                x(j,i)=penal(10);
            end
        end
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%CODE TO SCAN NUMERICAL ROWS AND BUILD MATRIXES FROM TEXT
FILES%%%%%%%%
function [a] = readnfo(filename)
    fid = fopen(filename,'rt');           %Open the text file to
    read.
    tmp = textscan(fid,'%s','Delimiter','\n'); %Scan the whole file to
    Matlab.
    fclose(fid);
    tmpcell=tmp{1,1};                    %Turn the scan into cell
    array.
    sizetmp=size(tmpcell,1);
    j=0;
    %Turn string from cell array to a numerical array.
    for i=1:sizetmp
        sec=tmpcell{i,1};
        third=str2num(sec);
        if isempty(third)==0
            j=j+1;
            a(j,:)=third;
        end
    end
end

```

end

ANEXO D: CÓDIGO DE ESTIMACIÓN DE CARGAS

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%CODE FOR LOAD ESTIMATION%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tic
%Initial parameters
nelx=60;
nely=20;
allelm=nelx*nely;          %total number of elements
%%Looking for the position of the force%%
loop=0;
lpm=0;
itfrac=int16(2*nelx/3);   %Maximum iterations
simf=zeros;
for i=1:itfrac
    loop=loop+1;
    pos=(nelx+1)*(nely+1)-(i-1)*(nely+1);   %Load position
    %Initial magnitude to start iterating
    imag=-10;
    %Topological optimization function
    [dc]=top(nelx,nely,0.5,3.0,1.5,pos,imag);
    %Comparison matrix
    comp=rc==dc;          %Logical array
    compnum=double(comp); %Numeric array
    count=histc(compnum(:,1),1); %Count of true values
    simf(i+1)=count/allelm; %Similitude factor
    disp(['sim: ' sprintf('%6.3f',simf(i+1)) ...
          ' #it: ' sprintf('%6.3f',loop) ])
    %Loop to find simf's local maximum values
    if simf(i+1) < simf(i)
        lpm=lpm+1;
        lmax(lpm)=simf(i);
        it(lpm)=i;
    end
end
%Finding simf's global maximum for position
gmax=max(lmax);
pgmax=find(lmax==gmax);
itp=it(pgmax);
place=1-((itp-1)/nelx);
%%Looking for the magnitude of the load in the position found%%
msimf=zeros;          %Magnitude similitude
factor
mloop=0;
posm=(nelx+1)*(nely+1)-(itp-2)*(nely+1); %Load position
[magm]=(-1)*(10:5:500); %Magnitude search
range
simcom=0.99;          %Similarity
compliance factor
sz=size(magm,2);
for j=1:sz

```

```

mloop=mloop+1;
%Running topological optimization function
[mrc]=top(nelx,nely,0.5,3.0,1.5,posm,magm(j));
compm=rc==mrc; %Logical array of
comparison
compnumm=double(compm); %Numerical array
countm=histc(compnumm(:),1); %Count of true values
msimf(j+1)=countm/allelm;
disp(['mag.sim: ' sprintf('%6.3f',msimf(j+1)) ...
      ' #it: ' sprintf('%6.3f',mloop) ])
%Estimating magnitude for a turning point meeting compliance
if msimf(j) > simcom
    if msimf(j+1) < msimf(j)
        magnitude=magm(j-1);
        break;
    end
end
end
disp(['Position: ' sprintf('%f',place) ...
      ' Magnitude: ' sprintf('%f',magnitude) ])
toc

```

ANEXO E: DIAGRAMA DE FLUJO DEL ALGORITMO

