



**HAL**  
open science

## Detección de daños en una armadura unidimensional por medio del algoritmo de optimización de la luciérnaga y elementos finitos

Javier Mauricio Pachón Ballesteros, Octavio Andrés González-Estrada, Heller Guillermo Sánchez Acevedo

### ► To cite this version:

Javier Mauricio Pachón Ballesteros, Octavio Andrés González-Estrada, Heller Guillermo Sánchez Acevedo. Detección de daños en una armadura unidimensional por medio del algoritmo de optimización de la luciérnaga y elementos finitos . Revista Avances, 2017, 13 (1), pp.1-7. hal-01534688

**HAL Id: hal-01534688**

**<https://hal.science/hal-01534688>**

Submitted on 8 Jun 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NoDerivatives 4.0 International License

# Detección de daños en una armadura unidimensional por medio del algoritmo de optimización de la luciérnaga y elementos finitos

## Damage detection in a unidimensional truss using the firefly optimization algorithm and finite elements

Javier Mauricio Pachón Ballesteros<sup>1</sup>, Octavio Andrés González-Estrada<sup>2</sup>, Heller Guillermo Sánchez Acevedo<sup>3</sup>

<sup>1</sup> Ingeniero Mecánico, Universidad Industrial de Santander, javier03\_01@hotmail.com

<sup>2</sup> Ingeniero Mecánico, M.Sc., Ph.D., Universidad Industrial de Santander, agonzale@uis.edu.co

<sup>3</sup> Ingeniero Mecánico, Ph.D., Universidad Industrial de Santander, hsanchez@uis.edu.co

Fecha de recepción: 24/05/2016 Fecha de aceptación del artículo: 06/12/2016

### Resumen

El presente trabajo tiene por objeto investigar la detección de daños en estructuras como un problema de optimización, utilizando la caracterización modal para evaluar la respuesta dinámica de la estructura ante un modelo de daño. Se llevó a cabo la implementación del algoritmo de optimización de la luciérnaga, tomando como caso de estudio un modelo numérico de daño sencillo, para ver los alcances del método propuesto y sus ventajas dentro del campo de monitoreo estructural (SHM). Se indican detalles de la implementación y los resultados obtenidos con este planteamiento.

### Palabras clave

Algoritmo de la luciérnaga, optimización, elementos finitos, análisis modal, monitoreo estructural.

### Abstract

In this paper, we investigate the damage detection of structures seen as an optimization problem, using modal characterization to evaluate the dynamic response of the structure given a damage model. We implemented the firefly optimization algorithm with a simple numerical damage model to assess the performance of the method and its advantages for structural health monitoring (SHM). We show some implementation details and discuss the obtained results.

### Keywords

Firefly algorithm, optimization, finite element method, modal analysis, structural health monitoring.

### 1. Introducción

La importancia de la detección temprana de daños ha aumentado en las últimas décadas en la industria y la academia, pasando de la inspección visual hasta el análisis de vibraciones [1]. La detección de daños en estructuras por medio del análisis modal se fundamenta en que el deterioro de la condición de una estructura o elemento, se ve reflejado en la pérdida de rigidez y amortiguamiento, afectando las propiedades dinámicas del sistema. La variación que presentan los parámetros modales es un indicador de la magnitud y localización del daño [2]. El monitoreo de la integridad estructural (SHM) utilizando la respuesta dinámica se puede clasificar en dos grupos de acuerdo a implementación [3]: los métodos experimentales basados en técnicas no destructivas y los métodos numéricos basados en FEA (*Finite Element Analysis*). Estos últimos particularmente han llevado al desarrollo de estructuras o sistemas inteligentes, las cuales son capaces de detectar el daño online y cuantificar su grado de severidad.

En la literatura se encuentran diferentes métodos para la detección de daños basados en el análisis a los parámetros dinámicos debidos a cambios en la rigidez del sistema [2], [4], [5]. En [4] se consideran algoritmos genéticos para resolver el problema de la detección de

daños en estructuras y elementos de máquina, el cual se aborda como un problema de optimización.

En los primeros trabajos se utilizó un algoritmo genético con codificación binaria y su función objetivo fue basada en el vector de fuerza residual [5]. Dicho enfoque tiene la desventaja de requerir formas modales completas, lo cual en la práctica actualmente se hace inviable debido a razones técnicas y económicas.

Posteriores trabajos propusieron detectar el daño dividiéndolo en dos etapas para definir su localización y magnitud [6]. En la primera etapa se determina un conjunto de elementos posiblemente dañados a partir de una metodología de localización de elementos con daño basada en energía. En la segunda etapa, se cuantifica el daño utilizando un micro-algoritmo genético, el cual realiza un proceso de optimización donde se busca, por la combinación óptima de elementos dañados y extensiones de daño, minimizar una función objetivo basada en frecuencias naturales y formas modales.

En [7] se utiliza un algoritmo genético modificado para la detección de daño estructural. El algoritmo considera una representación de cromosomas definida con números reales y una función objetivo basada en cambios en las frecuencias naturales y las formas modales. El algoritmo reinicia los individuos que presentan una mínima diferencia en la función objetivo para que conformen la nueva población. Este tipo de codificación resulta altamente aplicable para resolver el problema de detección de daño en vista de que el número y la posición de los elementos dañados no son conocidos *a priori*.

En este trabajo, se plantea la utilización de un algoritmo de optimización de tipo metaheurístico, llamado *Firefly Algorithm* (FA), para la detección de daños en estructuras mediante caracterización modal. El algoritmo tipo enjambre considera las variaciones en la respuesta dinámica de la estructura, dado un modelo de daño sencillo, el cual se caracteriza por una disminución de las propiedades elásticas de los elementos dañados. La caracterización modal se realiza por medio de un modelo numérico de elementos finitos (FE).

## 2. Metodología

### 2.1. Planteamiento del problema

Consideremos el problema de vibración libre no amortiguado. El modelo matemático que define la ecuación del movimiento para el sistema de un grado de libertad se puede escribir como:

$$m\ddot{u}(t) + ku(t) = 0, \quad (1)$$

donde  $m$  es la masa,  $k$  la constante elástica y  $u$  es el desplazamiento solución del sistema, que depende del tiempo. La solución general de (1) es

$$x = A \sin(\omega t + B), \omega = \sqrt{k/m}, \quad (2)$$

donde  $A$  y  $B$  son constantes reales y  $\omega$  representa la frecuencia natural del sistema. La ec. (2) se puede generalizar para sistemas de varios grados de libertad como

$$[M][\ddot{\mathbf{u}}(t)] + [K][\mathbf{u}(t)] = 0, \quad (3)$$

donde  $\mathbf{M}$  es la matriz de masa,  $\mathbf{K}$  la matriz de rigidez del sistema y  $\mathbf{u}$  el campo de desplazamientos. La solución de (3) no es única y para un sistema de  $n$  grados de libertad tiene hasta  $n$  soluciones  $\mathbf{u}_n$  o modos de vibración, cada uno asociado a una frecuencia natural del sistema.

### 2.2. Formulación de elementos finitos y análisis modal

Para un sistema continuo es posible plantear la solución mediante una discretización de elementos finitos, donde en la ec. (3)  $\mathbf{u}$  representa los desplazamientos nodales y las matrices  $\mathbf{M}$  y  $\mathbf{K}$  se construyen en el dominio de cada elemento como:

$$\mathbf{M} = \int_{\Omega_e} \mathbf{N}^T \rho \mathbf{N} d\Omega, \quad \mathbf{K} = \int_{\Omega_e} \mathbf{B}^T \mathbf{D} \mathbf{B} d\Omega, \quad (4)$$

donde  $\rho$  es la masa por unidad de volumen,  $\mathbf{N}$  la matriz de funciones de forma utilizadas para la discretización de elementos finitos y  $\mathbf{B}$  sus derivadas.

## 2.3. Algoritmo luciérnaga

El algoritmo de la luciérnaga fue planteado por Yang [8], inspirado en el comportamiento de las luciérnagas. Es un algoritmo de optimización metaheurístico de inteligencia de enjambre. Este algoritmo brinda ventajas de funcionamiento cuando busca en espacios de solución extensos, ya que al no tener un punto de partida evita caer en óptimos locales, mejorando su rendimiento en el espacio global [9]. En la Figura 1 se presenta el pseudocódigo propuesto por Yang.

Algoritmo luciérnaga

```

Función objetivo  $f(x)$ ,  $x = (x_1, \dots, x_d)^T$ 
Generación de la población inicial de luciérnagas  $x_i (i = 1, 2, \dots, n)$ 
Intensidad de luz  $I_i$  de  $x_i$  determinada por  $f(x_i)$ 
Definición de coeficiente de absorción  $\gamma$ 
while ( $t < MaxGeneration$ )
  for  $i = 1:n$  todas las  $n$  luciérnagas
    for  $j = 1:n$  todas las  $n$  luciérnagas
      if ( $I_i < I_j$ ), la luciérnaga  $i$  se mueve hacia  $j$ 
      end if
      Variación del atractivo respecto a la distancia  $r, \exp[-\gamma r]$ 
      Evaluar las nuevas posiciones y actualizar la intensidad de luz
    end for  $j$ 
  end for  $i$ 
  Ordenar las luciérnagas según intensidad buscando el óptimo global
end while
Procesar los resultados

```

Figura 1. Pseudocódigo del algoritmo FA.

Las luciérnagas utilizan la luz para atraer otras luciérnagas durante el apareamiento. En el algoritmo la intensidad de la luz se puede formular de forma tal que se asocia con la función objetivo a optimizar. El algoritmo genera una cantidad determinada de posibles soluciones dentro el campo de búsqueda, en el cual las configuraciones (luciérnagas) que den la mejor respuesta de la función objetivo serán las que atraerán a las demás configuraciones que estén cercanas. De esta manera se examinan puntos clave del campo solución de una forma más eficiente. Los parámetros de control del algoritmo FA se definen como:

$$f(\mathbf{x}) = \sum_{j=1}^{n_{\omega}} W_{\omega j} \left[ 1 - \left( \frac{\omega_{mj}}{\omega_{aj}} \right) \right]^2 + \sum_{j=1}^{n_{\omega}} \sum_{i=1}^s W_{\phi ji} (\phi_{mji} - \phi_{aij}), \quad (5)$$

$x$ : Población de luciérnagas.

$n$ : Tamaño de la población, número total de luciérnagas.

$I_i$ : Intensidad de luz de la luciérnaga  $i$ .

$MaxGeneration$ : Número máximo de generaciones para las luciérnagas.

$\alpha$ : Establece la aleatoriedad del proceso. Define el paso en el movimiento de las luciérnagas.

$\gamma$ : Coeficiente de absorción de luz.

$\Delta$ : Establece la reducción de la aleatoriedad cada vez que se origina una nueva generación.

$\beta$ : Atractivo. Coeficiente de atracción entre luciérnagas, que varía respecto al coeficiente de absorción de luz y la distancia entre ellas, dado por

$$\beta(r) = \beta_0 e^{-\gamma r^m}, m > 1 \quad (6)$$

donde  $\beta_0$  es el atractivo a  $r = 0$ .

## 2.4. Reglas ideales del algoritmo

Yang [9] estableció tres reglas fundamentales que rigen el algoritmo, las cuales determinan el patrón de comportamiento de las luciérnagas. Estas reglas son:

- Las luciérnagas son unisex, es decir, una luciérnaga se verá atraída por otra luciérnaga sin importar su sexo.
- El atractivo de una luciérnaga es proporcional a su brillo, y estos dos (atractivo y brillo) se ven obligados a disminuir cuando la distancia respecto a otra luciérnaga aumenta. Las luciérnagas con menor brillo se moverán hacia una con mayor brillo. Si una luciérnaga no es atraída por ninguna otra, debido a que no existe alguna con un brillo mayor que el suyo, esta se moverá de forma aleatoria.
- El brillo de una luciérnaga está determinado por la evaluación de la función objetivo.

## 2.5. Función objetivo

El problema de optimización para la detección del daño estructural, se puede definir a través de un funcional que expresa la diferencia cuadrática ponderada entre la respuesta del modelo sano y el modelo dañado. La función objetivo a optimizar, propuesta por Moslem y Nafaspour [1], es: donde se minimiza la diferencia cuadrática en función de las

eigenvalores  $\omega$  y los eigenvectores  $\phi$  del problema en consideración.  $W_{\omega j}$  y  $W_{\phi ji}$  son los factores de ponderación para  $\omega$  y  $\phi$  respectivamente,  $n_{\omega}$  representa el número de eigenvalores a considerar y  $s$  define el tamaño del eigenvector.

En este sentido, la función objetivo se encargará de la comparación de las características dinámicas del modelo de prueba y los modelos de la base de datos que tiene preconfiguraciones de posibles daños.

## 2.6. Ruido

Con el fin de simular distorsión en las señales del modelo de prueba se introdujo ruido en los valores de las frecuencias naturales  $\omega$  y formas modales  $\phi$ . El ruido se expresa en función de pequeñas sumas o restas dadas por

$$\omega_r = \omega * (1 + Rand(-1,1) * Ruido_{\omega}), \quad (7)$$

$$\phi_r = \phi * (1 + Rand(-1,1) * Ruido_{\phi}), \quad (8)$$

donde  $Ruido_i$  representan los porcentajes de ruido a utilizar en los eigenvalores y los eigenvectores, distribuidos de manera aleatoria.

## 2.7. Metodología de solución

Para la solución del problema se utilizan dos programas: ANSYS y MATLAB. En primer lugar, se requiere evaluar la respuesta dinámica para diferentes escenarios de daño. Se construye en Matlab un arreglo de celdas o base de datos que contiene las respuestas dinámicas para diferentes configuraciones de daño (barras afectadas y porcentaje de reducción de la elasticidad). Se utiliza código APDL de ANSYS para simular las pérdidas de rigidez en las barras y obtener la caracterización modal de forma automática.

Con la base de datos ya construida se procede a la implementación del algoritmo luciérnaga. El algoritmo genera un número determinado de luciérnagas (posibles soluciones dentro de la base de datos) que exploran el campo solución. Las luciérnagas van evaluando la función objetivo y, por medio de la intensidad, se van agrupando en los óptimos locales del espacio solución. El problema de optimización planteado a partir de la función objetivo en (5), compara el modelo de prueba con diferentes modelos

de daño y encuentra el que mejor se ajusta a los datos de entrada.

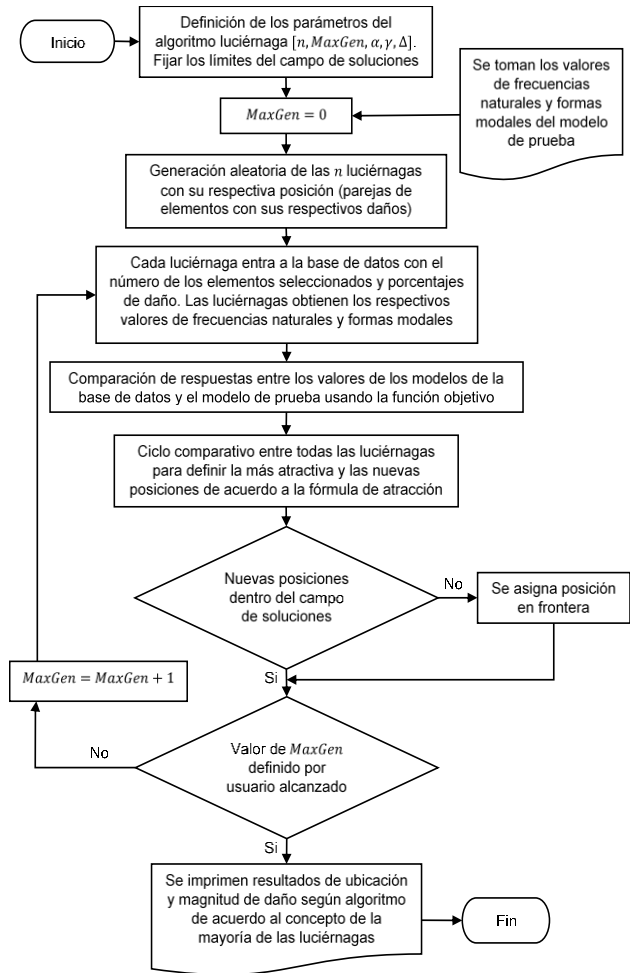


Figura 2. Diagrama de flujo del algoritmo propuesto.

El diagrama general del funcionamiento del código completo se muestra en la

Figura 2.

## 3. Análisis y resultados

### 3.1. Armadura

La armadura de la Figura 3, está compuesta por 13 barras de acero A-36, con módulo de Young  $E = 2 \times 10^{11} Pa$ , coeficiente de Poisson  $\nu = 0.3$  y densidad  $\rho = 7850 Kg/m^3$ . La estructura está simplemente apoyada en los nodos 1 y 5. El área de la sección transversal de las barras es de  $A = 4 \times 10^{-4} m^2$ . La altura total de la estructura es de 2.4284 m y el largo 7.3152 m. El problema consiste en determinar si alguna de las

barras de la estructura presenta algún fallo que pueda poner en riesgo su operación. Para el modelo numérico de la estructura implementado en Ansys se consideraron elementos LINK1. Se resuelve el problema de vibración libre sin amortiguamiento planteado en la ec. (3). Inicialmente, para la definición del problema de optimización definido por la función objetivo en (5), la ponderación se define con  $W_{\omega_j} = 1$  y  $W_{\phi_{ji}} = 1$ . Los parámetros de configuración del algoritmo FA, según recomienda [10], se toman como  $\alpha = 0.2$ ,  $\beta_0 = 1$ ,  $\gamma = 1$ , y  $\Delta = 0.97$ .

En primer lugar, se genera *offline* la base de datos que contiene diferentes modelos de daño para ayudar en la búsqueda del daño en el modelo de entrada. La base de datos contiene las respuestas para daños en una o más barras dentro de un rango de [0, 40] en porcentaje de daño, discretizados cada 5%. Para la solución de (3), y con el fin de acotar el problema, solo se consideró la extracción de las primeras 8 formas modales.

El algoritmo de optimización FA busca dentro de una base de datos precalculada, que contiene la respuesta dinámica de la estructura para diferentes fallos. Al hacer la respectiva comparación por medio de la función objetivo, se determinan no solo los elementos dañados, sino también la pérdida de rigidez de cada uno de ellos.

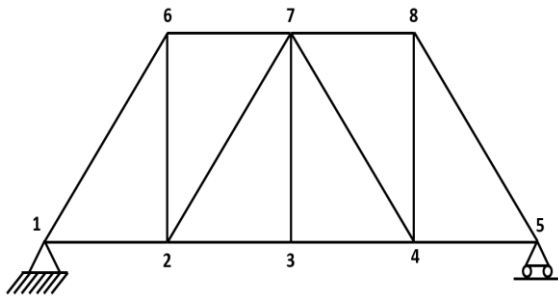


Figura 3. Armadura simplemente apoyada

### 3.2. Detección de daños

En el trabajo propuesto, para hacer las respectivas pruebas se introdujo ruido en las mediciones según (7) y (8). Los valores de ruido son del 1% al 3% en las frecuencias naturales y del 2% al 5% en los respectivos modos. De este modo se realizaron pruebas de asertividad en el algoritmo propuesto con el objeto de

verificar como respondía al ruido, tratando de emular la respuesta ante una señal de entrada con distorsiones. Se realizaron tres intentos para cada combinación de configuración de daño y número de modos extraídos y se obtuvo el porcentaje de efectividad.

En la Figura 4 se observa la efectividad del algoritmo frente al ruido para distinto número de formas modales extraídas. Los resultados obtenidos muestran que cuando consideramos solamente 2 formas modales el algoritmo de optimización es muy sensible al ruido y no es capaz de encontrar en algunos casos los elementos dañados. Al aumentar el número de modos de vibración entre 4 y 6, aumenta la asertividad del método al tener un espacio de búsqueda enriquecido. Para 8 modos el porcentaje de efectividad del algoritmo buscando elementos dañados fue del 100%. Los tiempos de cálculo fueron estables, empleando tiempos entre 40 s y 60 s por cada ejecución del algoritmo, independiente del número de modos considerados.

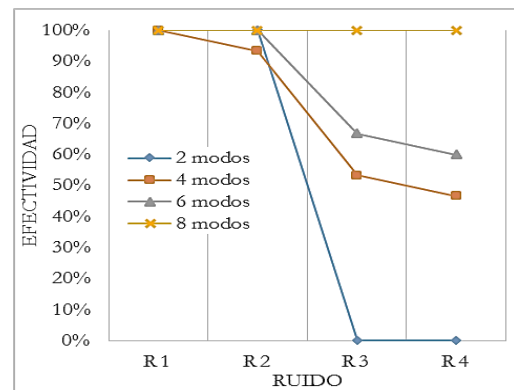


Figura 4. Efectividad del algoritmo frente a diferentes porcentajes de ruido en el modelo de prueba  $R=[\text{Ruido}_{\omega}, \text{Ruido}_{\phi}]$ : R1= [0%, 0%], R2= [0.5%, 1%], R3= [1%, 3%], R4= [2%, 5%].

### 3.3. Desempeño del algoritmo

Se evaluó el desempeño del algoritmo variando los parámetros de número de luciérnagas  $n$  y número máximo de iteraciones  $\text{MaxGen}$ , para verificar los aciertos del algoritmo, convergencia de las luciérnagas y el tiempo de cálculo respecto a dos diferentes configuraciones de daño. Para este análisis se simuló con un porcentaje de ruido de  $\text{Ruido}_{\omega} = 1\%$  y  $\text{Ruido}_{\phi} = 3\%$ , tal como se indica en [11].

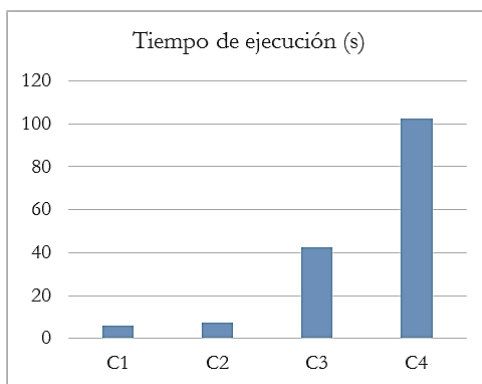
Según Yang [9], el número recomendado de luciérnagas oscila entre 25 y 45, mientras que el número de generaciones varía respecto a la complejidad del campo solución. Se estudiaron 4 configuraciones, según se muestra en la Tabla 1.

**Tabla 1.** Configuraciones para desempeño.

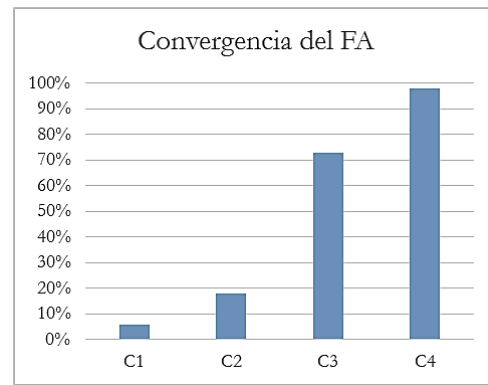
<i>n</i>	<i>MaxGen</i>	Notación
25	500	C1
30	1000	C2
40	5000	C3
60	8000	C4

Para las configuraciones mostradas se analizó tiempo de ejecución y porcentaje de convergencia de las luciérnagas (número de luciérnagas que coinciden en la identificación del daño). En la

Figura 5 se muestran los tiempos en segundos para las cuatro configuraciones preestablecidas. Se observa que el número de luciérnagas o puntos de evaluación de la función objetivo aumenta considerablemente el tiempo de cálculo. Así mismo, en la Figura 6 se observa que a mayor número de luciérnagas, también aumenta el porcentaje de luciérnagas que se agrupan alrededor de la solución final encontrada [12]. En las figuras comparativas se evidencia que las recomendaciones hechas por [9], dan un desempeño satisfactorio en cuanto a gasto computacional y tiempo de cálculo.



**Figura 5.** Desempeño en tiempo.



**Figura 6.** Desempeño convergencia.

## Conclusiones

En el presente trabajo se desarrolló una metodología que aborda el problema de la detección de daños en estructuras como un problema de optimización. Para la caracterización dinámica de las estructuras se utilizó el software de elementos finitos ANSYS. Para la solución del problema de optimización se escogió el algoritmo luciérnaga como la herramienta para determinar la condición de daño en los modelos.

La metodología se fundamentó en el hecho de que un daño afecta las propiedades de rigidez del sistema lo que repercute en un cambio en su respuesta dinámica. Se construyó una base de datos con los eigenvalores y eigenvectores precalculados para diferentes tipos de daño. El algoritmo evalúa la respuesta dinámica del modelo de prueba y la compara con la base de datos para determinar la magnitud y localización de daño. Los resultados muestran que, para diversas configuraciones de daño, y con un porcentaje de ruido moderado (casos R1, R2, R3), el algoritmo consigue detectar los elementos o regiones damnificadas y cuantificar el porcentaje de daño. Cuando existe un porcentaje de ruido severo (caso R4), es necesario tener en cuenta un número de modos de vibración mayor (6 u 8), para obtener una respuesta confiable por parte del algoritmo. Este comportamiento se debe a que al trabajar con pocos modos de vibración el algoritmo no tiene información suficiente en los desplazamientos nodales para determinar la ubicación y magnitud del daño.

Se observó que una mala elección de los parámetros principales del FA puede llevar a una identificación errónea de los elementos o regiones damnificadas o

generar un recalcu­lo innecesario. Esto ocasiona una pérdida de tiempo y gasto computacional, siendo acertada la configuración recomendada por el autor del algoritmo.

## Referencias

1. K. Moslem and R. Nafaspour, “Structural Damage Detection by Genetic Algorithms,” *AIAA J.*, vol. 40, no. 7, pp. 1395–1401, May 2012.
2. H. G. A. Sánchez, D. M. C. Marulanda, and E. G. Florez, “Application of Vibration Based Damage Identification Techniques on Metallic Structures,” *Adv. Mater. Res.*, vol. 875–877, pp. 875–879, 2014.
3. Y. Zou, L. Tong, and G. P. Steven, “Vibration-Based Model-Dependent Damage (Delamination) Identification and Health Monitoring for Composite Structures — a Review,” *J. Sound & Vibration*, vol. 230, no. 2, pp. 357–378, 2000.
4. A. Rytter, “Vibration Based Inspection of Civil Engineering,” Aalborg University, 1993.
5. J. F. Schutte and A. A. Groenwold, “Sizing design of truss structures using particle swarms,” *Struct. Multidiscip. Optim.*, vol. 25, no. 4, pp. 261–269, Oct. 2003.
6. S. M. Bland and R. K. Kapania, “Damage Identification of Plate Structures Using a Hybrid Genetic-Sensitivity Approach,” *AIAA J.*, vol. 43, no. 2, pp. 439–442, May 2005.
7. J. E. Laier and J. D. V Morales, “Computational Structural Engineering: Proceedings of the International Symposium on Computational Structural Engineering, held in Shanghai, China, June 22–24, 2009,” Y. Yuan, J. Cui, and H. A. Mang, Eds. Dordrecht: Springer Netherlands, 2009, pp. 833–839.
8. X.-S. Yang, *Nature-Inspired Metaheuristic Algorithms*. Luniver Press, 2008.
9. X.-S. S. Yang and X. He, “Firefly algorithm: recent advances and applications,” *Int. J. Swarm Intell.*, vol. 1, no. 1, p. 36, Aug. 2013.
10. X.-S. Yang, “Nature-Inspired Metaheuristic Algorithms,” Feb. 2008.
11. B. Chen and S. Nagarajaiah, “Flexibility-based structural damage identification using Gauss-Newton method,” in *Proc. SPIE 6529, Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems*, 2007.
12. J. Pachón and D. Silva, “Caracterización de daño en estructuras mediante ANSYS y la técnica de optimización firefly,” Universidad Industrial de Santander, 2015.