

IDENTIFICACION DE DAÑOS EN PORTICOS DE TRES DIMENSIONES

ALIRIO PABÓN ACOSTA

MARGARITA MARÍA REYES SUÁREZ

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENÍERIAS FISICO-MECANICAS
ESCUELA DE INGENIERIA CIVIL
BUCARAMANGA**

2013

IDENTIFICACION DE DAÑOS EN PORTICOS DE TRES DIMENSIONES

ALIRIO PABÓN ACOSTA

MARGARITA MARÍA REYES SUÁREZ

Trabajo de grado presentado como requisito para optar el título de ingeniero civil

Director:

OSCAR JAVIER BEGAMBRE CARRILLO

Ingeniero civil, MsC, PhD

Co- Director

FREDDY SOTELO MONROY

Ingeniero civil

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENÍERIAS FISICO-MECANICAS**

ESCUELA DE INGENIERIA CIVIL

BUCARAMANGA

2013

Dedicatoria

A mis padres Martin Enrique Reyes Mantilla y Claudia Azucena Suarez Villamizar por darme su apoyo incondicional y motivarme a construir mis sueños.

A mis hermanas Silvia y Daniela por ser mi compañía a lo largo de toda mi vida.

A mi novio Juan Felipe Camargo Jimenez por su amor incondicional y apoyo para seguir adelante.

A Alirio Pabón Acosta por ser mi compañero y cómplice en este logro. Sin el este logro no sería posible.

A mis amigos Jose Ricardo Prada y Sergio Botia por ser mis incondicionales confidentes.

A mi familia por hacerme el ser humano que soy y darme apoyo incondicional para lograr cada una de las metas que me propongo.

Margarita Maria Reyes Suarez

Dedicatoria

A Dios por darme salud, fortaleza y fe para cumplir satisfactoriamente una etapa más de mi vida académica y personal.

A mis padres Alirio Pabón Fletcher (Q.E.P.D.) y Esperanza Acosta Domínguez por regalarme la vida, por su apoyo incondicional y por el gran esfuerzo realizado para permitirme cumplir esta meta.

A mis hermanos Diana Carolina Pabón Acosta y Néstor Mauricio Pabón Acosta por brindarme sus consejos y apoyo incondicional en cada uno de los momentos de mi vida.

A mi novia Carolina Torres Arias quien me brindó su amor, cariño, compañía, comprensión y constante apoyo para cumplir con este logro.

A Margarita María Reyes Suárez por confiar en mí y ser el complemento perfecto para alcanzar esta meta.

A mis amigos y familiares por brindarme su amistad, compañía, confianza y alegrías en los momentos más importantes de mi vida.

Alirio Pabón Acosta

AGRADECIMIENTOS

A nuestra Alma Mater y en especial a la Escuela de Ingeniería Civil, la cual por medio de su profesorado nos formó hasta alcanzar este logro.

A nuestro director Oscar Javier Begambre Carrillo, quien siempre creyó y orientó nuestro trabajo.

A Freddy Saúl Sotelo Monroy por acompañarnos y orientarnos en la elaboración de este trabajo.

A Camilo Eduardo Celis Melo por su colaboración y constante orientación.

TABLA DE CONTENIDO

1. INTRODUCCION	19
1.1 OBJETIVOS.....	21
1.1.1. OBJETIVO GENERAL	21
1.1.2. OBJETIVOS ESPECIFICOS	21
2. DETECCION Y METODOS DE OPTIMIZACION: ANTECEDENTES	22
3. MODELADO DE LA ESTRUCTURA.....	27
3.1 PREDISEÑO.....	27
3.1.1 DIMENSIONES.....	28
3.1.1.1 PORTICO COMPLETAMENTE SIMÉTRICO.....	28
3.1.1.2 PORTICO ASIMÉTRICO	28
3.1.2 MATERIALES	28
3.1.3 SECCION TRANSVERSAL DE ELEMENTOS	29
3.2 METODO DE ELEMENTOS FINITOS	30
3.3 ANALISIS MODAL	30
4. PRINCIPIOS DE LOS METODOS DE OPTIMIZACION USADOS.....	33
4.1 OPTIMIZACION DE ENJAMBRE DE PARTICULAS (PSO)	33
4.2 ALGORITMO GENETICO (AG)	41
5. EVALUACION DE DESEMPEÑO DEL ALGORITMO GENETICO (AG) Y EL ALGORITMO DE OPTIMIZACION DE PARTICULAS (PSO) EN DETECCION 3D.	45
5.1 CASOS DE DAÑO	46
5.2 RESULTADOS OBTENIDOS USANDO PSO.....	48
5.2.1 CASO DE DAÑO ÚNICO (DATOS SIN RUIDO).....	49
5.2.2 CASO DE DAÑO MULTIPLE (DATOS SIN RUIDO).....	51

5.2.3 CASOS DE DAÑO APLICANDO RUIDO	52
5.2.3.1 CASO SIN DAÑO.....	52
5.2.3.2 CASO DE DAÑO UNICO (CON RUIDO)	54
5.2.3.3 CASO DE DAÑO MULTIPLE (CON RUIDO)	54
5.3 RESULTADOS OBTENIDOS USANDO AG	55
5.3.1 CASO DE DAÑO ÚNICO (SIN RUIDO)	56
5.3.2 CASO DE DAÑO MULTIPLE (SIN RUIDO)	57
5.3.3 CASOS DE DAÑO APLICANDO RUIDO	58
5.3.3.1 CASO DE DAÑO UNICO (CON RUIDO)	58
5.3.3.2 CASO DE DAÑO MULTIPLE (CON RUIDO)	59
5.3.4 CASO DAÑO UNICO PORTICO SIMETRICO CON VARIACION EN EL NUMERO DE ITERACIONES.....	60
5.4 ANALISIS DE RESULTADOS.....	61
6. CONCLUSIONES	69
7. RECOMENDACIONES.....	71
8. BIBLIOGRAFIA	72

LISTA DE TABLAS

Tabla 1. Sección transversal viga (20x20)	29
Tabla 2. Sección transversal viga (20x30)	29
Tabla 3. Sección transversal viga (40x40)	29
Tabla 4. Sección transversal columna (20x20)	29
Tabla 5. Sección transversal columna (40x40)	29
Tabla 6. Tabla de resultados PSO. Número máximo de iteraciones.....	48
Tabla 7. Resultados PSO – Pórtico simétrico – Simulación de daño en el elemento 1 con porcentajes de daño del 20%, 10%, 5% y1%.....	49
Tabla 8. Resultados PSO – Pórtico simétrico – Simulación de daño en el elemento 20 con porcentajes de daño del 20%, 10%, 5% y1%.....	50
Tabla 9. Resultados PSO – Pórtico asimétrico – Simulación de daño en el elemento 1 con porcentajes de daño del 20%, 10%, 5% y1%.....	50
Tabla 10. Resultados PSO – Pórtico asimétrico – Simulación de daño en el elemento 20 con porcentajes de daño del 20%, 10%, 5% y1%.	51
Tabla 11. Resultados PSO – Pórtico simétrico – Simulación de daño en los elemento 9 – 10 – 18 simultáneamente con porcentajes de daño del 20%, 10%, 5% y1 %.....	51
Tabla 12. Resultados PSO – Pórtico asimétrico – Simulación de daño en los elemento 9–13–18 simultáneamente con porcentajes de daño del 20%, 10%, 5% y 1%.....	52

Tabla 13. Resultados – PSO – Pórtico simétrico con ruido, sin daño.....	53
Tabla 14. Resultados PSO – Pórtico simétrico – Simulación de daño en los elementos 1 y 20 con porcentajes de daño del 20% y 5%.....	54
Tabla 15. Resultados PSO – Pórtico simétrico – Simulación de daño en los elementos 9-10-18 simultáneamente con porcentajes de daño del 20% y 5%.....	55
Tabla 16. Tabla de resultados AG. Número máximo de iteraciones.....	56
Tabla 17. Resultados AG – Pórtico simétrico – Simulación de daño en los elementos 1 y 20 con porcentajes de daño del 20%.....	57
Tabla 18. Resultados AG – Pórtico asimétrico – Simulación de daño en los elementos 1 y 20 con porcentajes de daño del 20%.....	57
Tabla 19. Resultados AG – Pórticos simétrico y asimétrico – Simulación de daño en los elementos 9-10-18 y 9-13-18 respectivamente y simultáneamente, con porcentaje de daño del 20%.	58
Tabla 20. Resultados AG – Pórticos simétrico con ruido – Simulación de daño en los elementos 1 y 20 respectivamente, con porcentajes de daño del 20%.....	58
Tabla 21. Resultados AG – Pórticos simétrico con ruido – Simulación de daño en los elementos 9-10-18 simultáneamente, con porcentajes de daño del 20%.	59
Tabla 22. Resultados AG – Pórticos simétrico – Simulación de daño en el elemento 1, con porcentajes de daño del 20%.	60
Tabla 23. Media y Desviación Estándar – Pórtico simétrico – PSO.....	61
Tabla 24. Media y Desviación Estándar – Pórtico asimétrico – PSO.....	62
Tabla 25. Media y Desviación Estándar – Pórtico simétrico con ruido – PSO.	64

Tabla 26. Media y Desviación Estándar – AG.....	65
Tabla 27. Evidencia al cumplimiento de objetivos específicos.....	70

LISTA DE FIGURAS

Figura 1. Diagrama de flujo del algoritmo de optimización de partículas.....	40
Figura 2. Diagrama de flujo del algoritmo genético.....	44
Figura 3. Casos de daño.....	47
Figura 4. Comparación daño único elemento 1 – PSO.....	63
Figura 5. Comparación daño único elemento 20 – PSO.....	63
Figura 6. Comparación daño múltiple – PSO.....	64
Figura 7. Comparación Valores Función Objetivo – Pórtico Simétrico sin ruido – AG y PSO.	65
Figura 8. Comparación Valores Función Objetivo – Pórtico Asimétrico – AG y PSO.	66
Figura 9. Comparación Valores Función Objetivo – Pórtico Simétrico con ruido – AG y PSO.	66
Figura 10. Comparación de tiempos promedio de computo – Pórtico Simétrico sin ruido – AG y PSO.	67
Figura 11. Comparación de tiempos promedio de computo – Pórtico Asimétrico – AG y PSO.	68
Figura 12. Comparación de tiempos promedio de computo – Pórtico Simétrico con ruido – AG y PSO.	68

LISTA DE ANEXOS

ANEXO A. Código de análisis modal de la estructura en lenguaje Tcl.	75
ANEXO B. Manual de OPENSEES (ver CD adjunto).	87
ANEXO C. Programa OPENSEES (ver CD adjunto).	87
ANEXO D. Validación de pórticos.	88
ANEXO E. Código de análisis modal de la estructura en código MATLAB.	89
ANEXO F. Código PSO.	105
ANEXO G. Código AG.	110
ANEXO H. Validación PSO y AG. Funciones de prueba.	115
ANEXO I. Código pórticos (OPENSEES)- PSO.	124
ANEXO J. Código pórticos (OPENSEES)- AG.	135
ANEXO K. Tablas de resultados pruebas algoritmos.	150

RESUMEN

TITULO:

DETECCION DE DAÑOS EN PORTICOS DE TRES DIMENSIONES

AUTORES:

ALIRIO PABON ACOSTA

MARGARITA MARIA REYES SUAREZ

PALABRAS CLAVES:

Detección de daños, pórticos 3d, algoritmo, PSO, AG.

DESCRIPCION:

Dentro del diseño de estructuras civiles, los temas de mantenimiento, control y operación han cobrado mucha importancia en los últimos años. Es por esto que existe la necesidad de generar mecanismos de monitoreo eficientes que permitan controlar las condiciones de funcionalidad para la cual la estructura fue concebida.

El presente trabajo tiene como finalidad estudiar la identificación de daños en pórticos 3D, usando algoritmos de optimización como enjambre de partículas (PSO) y algoritmo genético (AG); a partir de datos de entrada obtenidos de características modales de la estructura como la frecuencia y los modos de vibración.

Las características modales de la estructura aportada se consiguen a partir de métodos de dinámica estructural. Se realiza un modelo de elementos finitos del pórtico en 3D obteniendo frecuencias y modos de vibración, para así poder cuantificar la variación en la misma e identificar en que elementos se presenta daño y su magnitud. La cuantificación del daño se realiza mediante un estudio estadístico de los algoritmos PSO y AG, que determina su capacidad de detección.

El PSO usado fue el básico propuesto por Kennedy y Eberhart en el año 1995 y modificado por Shi y Eberhart en el año 2001 el cual se comparó con el AG básico propuesto por Holland en el año 1975.

IDENTIFICACION DE DAÑOS EN PORTICOS DE TRES DIMENSIONES

Proyecto de grado.

Facultad de ingenierías físico - mecánicas. Escuela de ingeniería civil. Director Ingeniero Civil, MsC, PhD Oscar Javier Begambre Carrillo. Codirector Ingeniero Civil Freddy Sotelo Monroy.

ABSTRACT

TITLE:

DAMAGE DETECTION IN A THREE-DIMENSIONAL FRAME

AUTHORS:

ALIRIO PABON ACOSTA

MARGARITA MARIA REYES SUAREZ

KEYWORDS:

Damage detection, 3d frames, optimization, algorithms, PSO, GA.

DESCRIPTION:

Maintenance, control and operation of buildings are a very important issue in the design of civil structures. That is why there is a need to generate efficient monitoring mechanisms that allow to controlling the conditions of functionality for which the structure was designed.

This work aims to study the damage detection problem in 3D frames using Particle Swarm Optimization, Genetic Algorithms and Modal data.

The modal characteristics of the frame are obtained from structural dynamics methods and the variation of those characteristics can show which element fails and the extension of the damage. The characteristics are obtained through structural analysis software OpenSees which requires an explicit programming of 3D frames used in the study and a coupling with the codes of the algorithms managed. Quantification of failure is performed by a statistical study of the PSO and GA algorithms, which determines its detection capability.

The PSO used in this study was basic and was proposed by Kennedy and Eberhart in 1995 and modified by shi and Eberhart in 2001. This version was compared with basic GA proposed by Holland in 1975. Those optimization methods used are stochastic and heuristic and have the characteristic of having several points of solution in a delimited space of search.

DAMAGE DETECTION IN A THREE-DIMENSIONAL FRAME

Degree Project.

Faculty of Physical-mechanical Engineerings. Schools of Civil Engineering. Director Civil Engineering, MsC, PhD Oscar Javier Begambre Carrillo. Co-Director Civil Engineering Freddy Sotelo Monroy.

1. INTRODUCCION

En Colombia los mecanismos de control legal aplicados al mantenimiento, monitoreo y operación de edificaciones son casi inexistentes, esto se debe en gran parte a la falta de información en esta área de la ingeniería y a que no existe en la actualidad un método standard para detección de daños en estructuras civiles (pese al número importante de trabajos publicados en la literatura internacional, ver por ejemplo el trabajo de Seweryn y Zbigniew (2009)). Debido a la necesidad creciente de evaluar el desempeño de métodos para identificación de daño, en este trabajo se pretende dar continuidad a los estudios y avances realizados en la escuela.

Por la razón anterior, en este estudio se tomó como punto de partida el trabajo "Identificación de daños en pórticos 2D empleando información modal y el método *Particle Swarm Optimization* (Sotelo y Acevedo, 2009)" en el cual se determinó la presencia de daño en un pórtico de acero a partir de su respuesta dinámica.

Se pretende implementar los avances ya alcanzados en el trabajo arriba referenciado y aplicarlo para pórticos en 3D teniendo en cuenta en esta ocasión la presencia de ruido en la información modal. Inicialmente se modela el pórtico con elementos finitos y se realiza un análisis modal numérico para el cálculo de las frecuencias y modos de vibración. Este proceso se realiza mediante el uso del software abierto de análisis estructural OPENSEES 2.4.0 plataforma computacional desarrollada por el *Pacific Earthquake Engineering research Center* con sede en la Universidad de California, Bekeley (www.opensees.com), los datos así obtenidos son exportados al lenguaje M de MATLAB (*MATrix LABoratory*) que es un software de alto nivel para cálculos numéricos, MATLAB fue creado por Cleve Moler en los años '70 y distribuido actualmente por MathWorks, Inc desde 1984 (<http://www.mathworks.com>), donde se realiza el proceso iterativo de optimización mediante el PSO y el AG y finalmente se obtiene un vector de valores

con la información sobre la localización y extensión aproximada de daño identificado.

Cabe resaltar que para este proceso se realizó un ensamble que permite unir el OPENSEES directamente con el código del algoritmo en MATLAB lo que genera una herramienta automática para estudiar el comportamiento de los algoritmos de optimización PSO y AG en la detección de daño.

Los algoritmos de optimización utilizados están basados en teorías de selección; el algoritmo de optimización de partículas (PSO) es un método heurístico que evoca del comportamiento de enjambres de insectos más específicamente ejemplifica con los enjambres de abejas y su comportamiento a la hora de obtener polen; mientras que el algoritmo genético (AG) se basa en la teoría de selección natural de Darwin.

Finalmente, se compara el desempeño de cada procedimiento de optimización empleado en términos de la calidad de su resultado y de su tiempo de cómputo.

El contenido de este trabajo se puede resumir de la siguiente manera. El capítulo 1 establece los objetivos del trabajo. En el capítulo 2 se describen los antecedentes del problema de detección de daño (DD) y de los métodos de optimización empleados. El capítulo 4 describe de forma detallada el modelado de la estructura y los principios de los algoritmos PSO y AG. Seguidamente, en el capítulo 5 se presenta la evaluación del desempeño del PSO y del AG en el problema DD de pórticos en 3D. Finalmente se presentan las conclusiones y recomendaciones.

1.1 OBJETIVOS

1.1.1. OBJETIVO GENERAL

- Emplear las propiedades modales de una estructura aporticada, modelada en 3D, para determinar la existencia de daño en la misma.

1.1.2. OBJETIVOS ESPECIFICOS

- Modelar con elementos finitos (o análisis matricial) un pórtico en 3 dimensiones para realizar su análisis modal numérico.
- Emplear los algoritmos PSO y AG para identificar el daño en la estructura.
- Comparar el desempeño, desde el punto de vista estadístico, de los procedimientos empleados.

2. DETECCION Y METODOS DE OPTIMIZACION: ANTECEDENTES

El monitoreo de la salud estructural aparece como resultado de los esfuerzos de los departamentos de ingeniería en los sectores aeroespaciales y petroleros por hacer sus sistemas operacionales más eficientes y seguros; y mejorar su expectativa de vida útil. (Atienza, 2004; Carden y Fanning, 2004).

Los métodos de monitoreo inicialmente consistían en una inspección visual de las estructuras civiles o mediante el uso de técnicas ultrasónicas, magnéticas y radiografías. Estos métodos requieren de un conocimiento previo de la ubicación de la falla lo que obliga a la búsqueda de estrategias globales de detección de daños. (Nagy, 2003; Riveros, 2007).

La supervisión de la estructura mediante métodos modales tienen como fin detectar cambios generados en las frecuencias y modos de vibración; estos pueden significar daño del sistema. Inicialmente los métodos basados en estas técnicas no tuvieron mucha acogida por los costos de implementación; Actualmente la detección de daños es un área de investigación en progreso la cual requiere del desarrollo de sistemas integrales de monitoreo estructural. (Carden y Fanning, 2004; Riveros, 2007).

Un sistema típico de monitoreo de salud estructural consta de tres componentes: sensores, sistema informático (datos y almacenaje) y un sistema de evaluación estructural (algoritmo de identificación de daño). (Li, Li y Song, 2004)

Hay una gran variedad de sensores utilizados para medir la respuesta de la estructura, entre ellos se encuentran el acelerómetro y los extensómetros. Estos se pueden implementar para monitoreo de diferentes daños o cambios estructurales relacionados tanto con las variaciones de temperatura como con los cambios de las cargas dinámicas y los efectos de la corrosión. (Carden y Fanning, 2004).

(Apud carden y fanning, 2004; Marulanda, Thomson y Marulanda, 2000,riveros, 2007).

Las técnicas de monitoreo de salud estructural se clasifican en cuatro niveles según su alcance:

Nivel 1: determinación de que el daño está presente en la estructura.

Nivel 2: determinación de la ubicación geométrica de los daños.

Nivel 3: cuantificación de la gravedad de los daños.

Nivel 4: predicción de la vida útil restante de la estructura.

Dentro de los métodos clasificados en el nivel 3 de Ritter se encuentra el trabajo de Chaekuk Na, Sun-Pilkim y Hyo- GyoungKwak (2009) en el cual se emplea un AG y datos modales incompletos de la estructura para deducir la extensión y localización de los daños estructurales. La evaluación de daño propuesto es demostrada a través de análisis numéricos utilizando OPENSEES.

En el trabajo de Ramsés Rodríguez y José Escobar titulado “*Detección de daño estructural en edificios sin parámetros modales base*” se propone el método de rigideces base que permite determinar la ubicación y magnitud de daño en edificios sin parámetros modales base (estado sin daño). A partir de la respuesta dinámica de la estructura (aceleraciones) se extraen parámetros modales con daño que a su vez sirven para determinar un estado de referencia. Al final cantidades de rigidez con y sin daño son comparadas para determinar índices de daño particulares para cada elemento de la estructura. Este método no afecta la estructura y no se requiere de inspección visual.

Un método de detección de daño basado en vibraciones estructurales armónicas se ha aplicado para la reconstrucción de patrones de daño típico en vigas de hormigón armado y pórticos 3D. Se ha demostrado que la comparación entre la estructura dañada y la intacta hace que sea posible formular criterios de optimización que pueden resolverse con eficacia por medio de la aplicación del algoritmo genético (AG) combinado con el algoritmo Levenberg Marquardt (LMA). Este método tiene una exactitud del 1% e identifica elementos con pérdidas de rigidez tan bajas como 5%, lo que lo hace muy eficaz durante la evaluación post-terremoto de estructuras civiles. (SewerynKokot, ZbigniewZembaty, 2009).

Debido a la variedad de tipologías estructurales, propiedades de materiales, condiciones de contorno y posibles patrones de daño, la aplicación de técnicas de detección de daño basadas en análisis modal y métodos de elementos finitos dejan todavía puntos que no son claros en caso de daño sísmico. Para comenzar con el diseño de posibles intervenciones de rehabilitación estructural después de la ocurrencia de un sismo importante, es necesario tener conocimiento previo de la distribución de daño y la capacidad estructural residual.

Por otro lado, el modelo de elementos finitos ajustado con base en mediciones de vibración, se utiliza para detectar, evaluar y cuantificar los daños en pórticos estructurales de hormigón armado con acero altamente dúctil. En el trabajo de Roeck y colaboradores (2010) se presenta un estudio de diferentes estrategias de modelamiento estructural para realizar detección de daños mediante la técnica de ajuste de modelos, lo que permite una descripción integral y la cuantificación de la degradación progresiva de las articulaciones viga-columna.

Las técnicas de optimización, conjuntamente con los sistemas informáticos se han convertido en una poderosa herramienta para el diagnóstico y solución de múltiples problemas complejos. La optimización o programación matemática intenta dar respuesta a un tipo general de problemas matemáticos donde se desea

elegir el mejor entre un conjunto de elementos y llegar a la solución con mayor rapidez. Este tipo de problemas a menudo representan una realidad física por lo que su aplicación puede encontrarse en prácticamente todos los campos de la ciencia. Un aspecto fundamental para lograr el objeto de optimizar es poder elegir que algoritmo y bajo que parámetros de ese algoritmo se ajustan mejor a las características del problema (Celis y Millán, 2009).

Los autores *Venter y Sobieszczanski-Sobieski (2003)* usaron el algoritmo de optimización con enjambre de partículas (PSO) para estudiar su funcionamiento en un problema estructural. El problema consistía en minimizar el volumen de material de una viga en voladizo de cinco segmentos para obtener un máximo esfuerzo de tensión para cada segmento. En el problema los parámetros a variar discretamente eran la altura y ancho del segmento, y se delimitaba dicha variación, dándole un rango determinado. El resultado indicó que el algoritmo PSO era capaz resolver problemas continuos y discretos.

Los algoritmos genéticos (AG) son llamados así porque se inspiran en la evolución biológica y su base genético-molecular. Estos algoritmos hacen evolucionar una población de individuos sometiéndola a acciones aleatorias semejantes a las que actúan en la evolución biológica (mutaciones y recombinaciones genéticas), así como también a una Selección de acuerdo con algún criterio, en función del cual se decide cuáles son los individuos más adaptados, que sobreviven, y cuáles los menos aptos, que son descartados. Este algoritmo se diferencia de los otros métodos de optimización tradicional debido a que son capaces de abordar la topología de solución multimodal que es típica de problemas de identificación de daño. El proceso de identificación de daño según algunos autores consiste en dos etapas: inicialmente se usa un VFR (variable frame rate o velocidad de cuadro variable) para localizar el daño y luego una segunda fase donde se usa el AG para cuantificar el daño en dichos elementos identificados. (*Carden y Fanning, 2004*).

El método de optimización de enjambres de partículas (PSO por sus siglas en inglés Particle Swarm Optimization) fue propuesta por Kennedy y Eberhart en el año 1995. Es una meta heurística evolutiva y de búsqueda, inspirada en el comportamiento social de las bandadas de pájaros o cardúmenes de peces. Las soluciones, llamadas partículas se “echan a volar” en el espacio de búsqueda, guiadas por la partícula que mejor solución ha encontrado hasta el momento y que hace de líder de la bandada. Cada partícula evoluciona teniendo en cuenta la mejor solución encontrada en su recorrido y al líder. En cada iteración, las partículas modifican su velocidad hacia la mejor solución de su entorno teniendo en cuenta la información del líder. Existen dos versiones diferentes de PSO, el algoritmo normal y una versión binaria. En el primero, las partículas pueden tomar valores reales en cada una de las dimensiones. En el algoritmo binario, cada dimensión de las partículas solo pueden tomar el valor 0 o 1. Para esta investigación se utilizó PSO normal.

De acuerdo con lo anterior las partículas o agentes representan cualquier tipo de individuos que muestren un comportamiento grupal. El movimiento de cada una de esas partículas se basa en dos fundamentos: la memoria autobiográfica de la partícula y la influencia social del enjambre. Matemáticamente esto se puede representar en N-dimensiones de acuerdo como lo requiera el problema y cada una de las posiciones de las partículas en dicho espacio N-dimensional representa una solución potencial. Fundamentalmente, el proceso evolutivo se reduce a mover cada partícula dentro del espacio de soluciones con una velocidad que variará de acuerdo a su velocidad actual, a la memoria de la partícula y a la información global que comparte el resto del enjambre, utilizando una función objetivo para cuantificar la calidad de cada partícula en función de la posición que ésta ocupe.

3. MODELADO DE LA ESTRUCTURA

Con el fin de desarrollar una técnica eficiente de detección de daño para pórticos en 3D con el uso de información modal se realizaron modificaciones en algoritmos básicos anteriormente propuestos (PSO Kennedy y Eberhart (1995) y Shi y Eberhard (2001), AG Holland 1975). El problema consiste en detectar posibles daños en los elementos que componen un pórtico 3D lineal elástico sin tener en cuenta ningún tipo de amortiguamiento y considerando una variación por ruido en la información modal.

El planteamiento del problema inicia con la definición de los parámetros geométricos de los elementos que conforman el pórtico 3D al que se le aborda mediante métodos de elementos finito.

La información modal necesaria para la realización de la detección de daño es obtenida por medio del software de análisis estructural OPENSEES el cual requiere de una programación explícita de los pórticos 3D usados en el estudio y un acople con los códigos de los algoritmos manejados.

Los parámetros geométricos e hipótesis usadas además de los diferentes códigos se encuentran explícitos en este capítulo.

3.1 PREDISEÑO

Para el desarrollo del presente trabajo se plantearon dos pórticos simples en tres dimensiones. Estos constan de cuatro columnas empotradas en la base y cuatro vigas, donde los nudos se modelaron como nudos rígidos.

Con la finalidad de realizar el estudio de sensibilidad al daño de cada algoritmo de optimización se consideró emplear un modelo 3D lineal elástico para todos los

caso presentados en este estudio. Como se sabe, en modelos 2D cualquier información sobre modos torsionales y de flexión en la dirección transversal se pierden. Por otro lado, en ensayos dinámicos, el nivel de excitación es generalmente menor en comparación con posibles cargas sísmicas, permitiendo que el modelo lineal satisfaga entonces los requerimientos de análisis de este trabajo.

3.1.1 DIMENSIONES

3.1.1.1 PORTICO COMPLETAMENTE SIMÉTRICO

- Las columnas tienen una altura de 3 metros y una sección transversal de 40x40 centímetros.
- Las vigas en dirección x y dirección y tienen una luz de 6 metros y una sección transversal de 40x40 centímetros.

3.1.1.2 PORTICO ASIMÉTRICO

- Dos columnas de longitud 3 metros y dos de longitud 4,5 metros, tienen una sección transversal de 20x20 centímetros. Simétrico
- Las vigas en dirección X tienen una longitud de 3 metros y una sección transversal de 20x20 centímetros, y las vigas en dirección Y tienen una luz de 4.5 metros y una sección transversal de 20x30 centímetros.

3.1.2 MATERIALES

Para el modelo se supuso un concreto reforzado con densidad de masa de 2400 kg/m³.

3.1.3 SECCION TRANSVERSAL DE ELEMENTOS

De la tabla 1 a la tabla 5 se especifican las características geométricas de las secciones transversales de las vigas y columnas usadas en los dos pórticos definidos en los numerales 4.1.1.1. Y 4.1.1.2.

Tabla 1. Sección transversal viga (20x20)

AREA	0.04 m ²
INERCIA X, INERCIA Y	0.0001333 m ⁴
DENSIDAD	2400 kg/m ³
MODULO DE ELASTISIDAD	21.5x10 ⁹ N/m ²
MODULO TORCIONAL	0.0002253 m ⁴

Tabla 2. Sección transversal viga (20x30)

AREA	0.06 m ²
INERCIA X	0.00045 m ⁴
INERCIA Y	0.0002 m ⁴
DENSIDAD	2400 kg/m ³
MODULO DE ELASTISIDAD	21.5x10 ⁹ N/m ²
MODULO TORCIONAL	0.0004695 m ⁴

Tabla 3. Sección transversal viga (40x40)

AREA	0.16 m ²
INERCIA X, INERCIA Y	0.00213 m ⁴
DENSIDAD	2400 kg/m ³
MODULO DE ELASTISIDAD	21.5x10 ⁹ N/m ²
MODULO TORCIONAL	0.0036 m ⁴

Tabla 4. Sección transversal columna (20x20)

AREA	0.04 m ²
INERCIA X, INERCIA Z	0.000133 m ⁴
DENSIDAD	2400 kg/m ³
MODULO DE ELASTICIDAD	21.5x10 ⁹ N/m ²
MODULO TORCIONAL	0.0002253 m ⁴

Tabla 5. Sección transversal columna (40x40)

AREA	0.16 m ²
INERCIA X, INERCIA Z	0.00213 m ⁴
DENSIDAD	2400 kg/m ³
MODULO DE ELASTICIDAD	21.5x10 ⁹ N/m ²
MODULO TORCIONAL	0.0036 m ⁴

3.2 METODO DE ELEMENTOS FINITOS

Los métodos de elementos finitos están basados en la representación de un cuerpo por el ensamble de subdivisiones llamados elementos finitos. Estos métodos son muy utilizados en diversos problemas de ingeniería y física. En este trabajo el problema involucra complejas propiedades de materiales y condiciones de contorno por lo cual permite tener soluciones más precisas.

En este proyecto se subdividió cada elemento en tres subelementos para mejorar los resultados de ubicación y magnitud de la falla. Detalles del modelado, de la validación del programa de análisis estructural y de los procedimientos de optimización se dan en los anexos 1 al 8.

3.3 ANALISIS MODAL

Para la realización del análisis modal de la estructura se utilizó OPENSEES (Open System for Earthquake Engineering Simulation). Con este programa se obtuvieron datos como: frecuencias, periodos y modos de vibración. Vale la pena mencionar que el daño en este trabajo fue modelado como pérdida de rigidez de los elementos y en ningún caso como pérdida de masa. Se supone también que el amortiguamiento de la estructura es bajo (menor del 5%). La validación matemática de los pórticos en 3D se encuentra en el anexo 4.

Para la detección de daño realizada en este estudio se tomaron como datos de entrada las cuatro primeras frecuencias y sus modos de vibración asociados para los nodos que simulan la unión entre viga – columna.

Se realizaron simulaciones de falla donde se aplicaron parámetros de ruido a las frecuencias y modos de vibración de 5% y 2% respectivamente. Estos valores se

encuentran dentro del rango admisible y se aplicaron con el fin de determinar la distorsión de los resultados de la detección de daño.

OPENSEES es una plataforma informática para el desarrollo de aplicaciones de simulación del comportamiento de sistemas estructurales y geotécnicos, sometidos a eventos sísmicos, posee capacidades avanzadas para la modelación y análisis de la respuesta no lineal de sistemas estructurales, para ello dispone de un amplio rango de modelos de materiales, elementos y algoritmos de solución. OpenSees utiliza métodos basados en elementos finitos, por lo tanto el primer paso para la modelación es subdividir el sistema en elementos y nudos, para de esta manera definir la acción de cargas, y las restricciones nodales. La característica principal de OpenSees es que dicha modelación y simulación se la realiza a través de una fuente abierta. Esto quiere decir que OpenSees está bajo constante desarrollo, de tal suerte que los diseñadores y los usuarios pueden actualizar sus bases permanentemente. El lenguaje de interpretación llamado lenguaje TCL, originado del acrónimo en inglés "Tool Command Language" o lenguaje de herramientas de comando, ha sido utilizado para soporte de los comandos de OPENSEES, los cuales son usados para la definición de la geometría del problema, estados de carga, formulación y solución.

Para este problema específico se realizó un código de análisis modal que se encuentra en el anexo 1, además el anexo 2 proporciona un manual del programa para comodidad del usuario y finalmente en el anexo 3 se encuentra el instalador del programa que es gratuito y se descarga en la página oficial www.opensees.com.

Para la realización del análisis modal dentro del proceso de detección de daño fueron acoplados OPENSEES y MATLAB de tal manera que escribiendo en código MATLAB se pudiera generar el Tcl necesitado por OPENSEES (anexo 5) para realizar el análisis modal de la estructura y arrojar las frecuencias y modos de

vibración necesarios para el proceso de optimización descrito en el capítulo 4 realizado en MATLAB (<http://www.mathworks.com>).

4. PRINCIPIOS DE LOS METODOS DE OPTIMIZACION USADOS

Debido a que el problema de Detección de Daños (DD) se caracteriza por tener múltiples puntos óptimos en este trabajo se emplearon los algoritmos PSO y AG para solucionarlo.

4.1 OPTIMIZACION DE ENJAMBRE DE PARTICULAS (PSO)

La optimización con enjambre de partículas más conocido como particle swarm optimization (PSO) imita y mimetiza el comportamiento de procesos naturales a partir de iteraciones de los individuos entre sí y con el entorno.

Los orígenes del PSO como método estocástico de optimización global se remontan a los estudios iniciados por Kennedy y Eberhart, quienes se fijan como objetivo inicial simular gráficamente el movimiento sincronizado e impredecible de grupos tales como los bancos de peces o las bandadas de aves, intrigados por la capacidad de estos grupos para separarse, reagruparse o encontrar alimento.

En el PSO las partículas o agentes representan individuos que exhiben un comportamiento social como grupo e interactúan entre sí. El movimiento de cada una de estas partículas hacia un objetivo común (solución del problema) está condicionado por dos factores básicos (c_1 y c_2), la memoria autobiográfica de la partícula o nostalgia y la influencia social de todo el enjambre. En este problema en específico se usó el PSO propuesto por Kennedy y Eberhard 1995, 2001 y su código se encuentra en el anexo 6.

A nivel computacional, esta filosofía puede extenderse a nivel N-dimensional de acuerdo con el problema analizado. La posición instantánea de cada una de las

partículas de la población en el espacio N -dimensional representa una solución potencial, siendo N el número de incógnitas del problema original.

Cada partícula se mueve dentro de un espacio de soluciones con una velocidad que varía de acuerdo a la velocidad actual, la memoria de la partícula y la información global que comparte con el resto del enjambre. Para esto se utiliza una función objetivo que cuantifica la calidad de cada una de las partículas en función de la posición que ocupe.

OPERADOR DE VELOCIDAD Y PARAMETROS DEL ALGORITMO

El operador de velocidad utilizado en el algoritmo PSO fue propuesto inicialmente por Kennedy y Eberhart en el año 1995, donde propone un único operador de velocidad que contiene memoria de la mejor de la partícula y la mejor posición de todas las partículas.

$$v_{k+1}^i = v_k^i + c_1 r_1 (p_k^i - x_k^i) + c_2 r_2 (p_k^g - x_k^i) \quad (4.1)$$

$$x_{k+1}^i = x_k^i + v_{k+1}^i \quad (4.2)$$

En (4.1), v_k^i y x_k^i representan, respectivamente, la velocidad y posición en la iteración o instante de tiempo k de la partícula i en la dimensión n -ésima del espacio de búsqueda. Los factores c_1 y c_2 son las denominadas constantes de aceleración cognitiva y social, que determinan en qué medida influyen sobre el movimiento de la partícula su propia memoria y la cooperación entre individuos, respectivamente. Los términos r_1 y r_2 son dos números aleatorios uniformemente distribuidos entre 0 y 1, cuyo objetivo es emular el comportamiento estocástico y un tanto impredecible que exhibe la población del enjambre. Después de calcular la nueva velocidad de la partícula i en la dimensión n , la nueva posición x_{k+1}^i se actualiza directamente de acuerdo con (4.2), donde se asume que la velocidad se aplica durante un cierto período de tiempo Δt , típicamente de valor unitario. El proceso descrito se extiende al espacio N -dimensional, de forma que se van

componiendo iterativamente nuevos vectores de posición X_i , utilizando, como en cualquier otro método de cómputo evolutivo, una función objetivo para ponderar la calidad de dicha solución parcial, actualizando los vectores p^i y p^g . El movimiento de los agentes sobre el espacio de soluciones y, en consecuencia, el rendimiento del algoritmo, está condicionado por el grado de contribución de las tres componentes de la velocidad en (4.1), ya mencionadas y denominadas por este orden como momento, hábito o inercia para considerar la tendencia de la partícula; memoria o autoaprendizaje para incluir la experiencia de la propia partícula, y cooperación, conocimiento social, conocimiento de grupo o información compartida, para reflejar el intercambio de información y el comportamiento social como grupo.

En el año 2001 Shi y Eberhart Incluyeron peso inercial (W) para controlar la tendencia de la partícula a continuar en la dirección en la que se estaba moviendo

$$v_{k+1}^i = wv_k^i + c_1r_1(p_k^i - x_k^i) + c_2r_2(p_k^g - x_k^i) \quad (4.3)$$

Los parámetros utilizados en este algoritmo involucran, un total de 100 partículas, de 24 dimensiones cada una y un espacio de búsqueda en cada dimensión entre 0 y 1 (límite inferior = 0, límite superior = 1), $c_1 = 2$, $c_2 = 2$ y $W = 1$ donde W es ajustado dinámicamente durante la optimización; La distribución inicial de la partícula se asigna de manera aleatoria así como los valores de r_1 y r_2 que oscilan entre 0 y 1.

Con el fin de evaluar y validar el desempeño del algoritmo se realizaron 5 pruebas con funciones extraídas de literatura internacional (ver anexo 8). Para dichas pruebas se usaron 100 partículas, una desviación estándar de 1×10^{-30} en las últimas 50 partículas, un máximo de 500 iteraciones. Estas funciones de prueba

fueron extraídas de la literatura internacional. (Georgieva A., Jordanov I., European Journal of Operational Research. 2009).

Para ajustar la función objetivo se realizaron pruebas con diferentes funciones encontradas en la literatura, donde se comprobó que optimizaban en función de las frecuencias y no de los modos de vibración.

La primera ecuación se basa en los cambios en las frecuencias y modos de vibración. (Sotelo Monroy, Freddy Saúl y Acevedo Gutiérrez, Javier Ernesto, identificación de daño en pórtico 2D empleando información modal y el método particle swarm optimization, universidad industrial de Santander, 2009,163p)

$$fobj1 = \sum_{i=1}^n \sum_{j=1}^m Abs(\phi_{c\ ij} - \phi_{e\ ij}) + \sum_i^n Abs(f_{c\ i} - f_{e\ i}) \quad (4.4)$$

Donde:

ϕ_{ij}^e = Elemento i del vector propio j experimental.

ϕ_{ij}^a = Elemento i del vector propio j analítica.

$f_{c\ i}$ = frecuencia calculada por el programa.

$f_{e\ i}$ = frecuencia experimental.

Se modificó la ecuación (4.4), en un intento de ajustarla para este problema en específico, obteniendo así las ecuaciones (4.5) y (4.6).

$$fobj2 = \sum_{i=1}^n \sum_{j=1}^m (\phi_{c\ ij} - \phi_{e\ ij})^2 + \sum_i^n (f_{c\ i} - f_{e\ i})^2 \quad (4.5)$$

$$fobj3 = \sqrt[2]{\left(\sum_{i=1}^n \sum_{j=1}^m (\phi_{c\ ij} - \phi_{e\ ij})^2\right)} + \sqrt[2]{\left(\sum_i^n (f_{c\ i} - f_{e\ i})^2\right)} \quad (4.6)$$

Como solución a este problema se implementó una función objetivo que da igual prioridad a los modos y frecuencias.

$$fobj4 = \sum_{i=1}^n \sum_{j=1}^m Abs \left(1 - Abs \left(\frac{\phi_{c_{ij}}}{\phi_{e_{ij}}} \right) \right) + \sum_i^n Abs \left(1 - Abs \left(\frac{f_{c_i}}{f_{e_i}} \right) \right) \quad (4.7)$$

La ecuación (4.7) se usó como función objetivo en el algoritmo PSO con el fin de minimizar los valores; cuando los modos de vibración y las frecuencias calculadas son iguales a las experimentales el resultado de esta función es igual a cero.

El algoritmo tiene dos criterios de parada, un número máximo de iteraciones (1000) o un valor en la desviación estándar calculada de 1×10^{-20} en los últimos 50 datos de los mejores globales.

Para el proceso de detección de daño se usan las frecuencias y modos de vibración experimentales como datos de entrada, con los cuales el PSO detecta la ubicación y magnitud de la falla.

Las frecuencias y modos de vibración de una estructura dependen de la masa y rigidez del sistema. En este modelo la pérdida de masa es despreciable por lo que se considera que la matriz de masa permanece constante y el daño se representa como un cambio en la matriz de rigidez. Por lo tanto la matriz de rigidez dañada de un elemento i estará dada por:

$$[K]_{di} = \alpha_i [K]_i \quad (4.7a)$$

En (4.7a) $[K]_{di}$ es la matriz de rigidez dañada del elemento, α_i es el factor de daño y $[K]_i$ es la matriz de rigidez sin daño del elemento i .

En el que α indica la severidad e i indica la posición de daño, simulando el cambio de las propiedades del elemento. Para este trabajo se consideró la variable de

daño como la disminución del módulo de elasticidad. El daño α_i estaría dado por valores entre cero y uno.

Teniendo la matriz de rigidez de cada elemento con simulación de daño, se ensambla la matriz de rigidez global.

$$[K_g] = \begin{bmatrix} [K]_{d1} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & [K]_{di} \end{bmatrix} \quad (4.7b)$$

En la ecuación 4.7b $[K_g]$ es la matriz de rigidez global de la estructura y $[K]_{di}$ es la matriz de rigidez con daño del elemento i .

También se debe tener la matriz de masa para así poder resolver la ecuación de movimiento de un sistema de múltiples grados de libertad ecuación 4.7c.

$$[M]\{\ddot{x}\} + [C]\{\dot{x}\} + [K]\{x\} = \{f(t)\} \quad (4.7c)$$

Donde $[M]$ es la matriz de masa, $[C]$ es la matriz de amortiguamiento, $[K]$ es la matriz de rigidez y $f(t)$ son las fuerzas dinámicas aplicadas al sistema.

Como el modelamiento de la estructura se realiza por medio de un análisis lineal, la matriz de amortiguamiento no es considerada (se considera amortiguamiento muy bajo). Además se considera que la estructura está en vibración libre (no está sometida a excitaciones externas) por lo tanto $f(t)$ es igual a cero y así se pueden obtener las propiedades más importantes de la estructura, que son las frecuencias naturales y sus correspondientes modos.

Por lo anterior la formulación de la ecuación de movimiento (4.7c) es la siguiente:

$$[M]\{\ddot{x}\} + [K]\{x\} = 0 \quad (4.7d)$$

La solución para el sistema anterior de ecuaciones diferenciales tiene la siguiente forma expresada de manera matricial.

$$\{x\} = \{a\} \text{sen}(\omega t - \delta) \quad (4.7e)$$

Donde $\{a\}$ es el vector de amplitudes del movimiento con un tamaño igual a los grados de libertad, δ es el ángulo de fase y ω es la frecuencia de vibración. Reemplazando la solución en la ecuación 4.7c tenemos:

$$[[K] - \omega^2[M]]\{a\} = 0 \quad (4.7f)$$

Esta ecuación (4.7f) representa el problema conocido como problema característico. La solución de esta formulación matemática, requiere que el determinante de la matriz del factor de $\{a\}$ sea igual a cero, es decir que la solución de las ecuaciones simultáneas lineales no homogéneas sea no trivial y requiere que el determinante de la matriz sea igual a cero:

$$|[K] - \omega^2[M]| = 0 \quad (4.7g)$$

Generalmente esta ecuación (4.7g) es conocida como ecuación característica del sistema. Las n raíces de esta ecuación se denominan valores propios que son siempre reales y positivas y su raíz cuadrada corresponde a las frecuencias de vibración (en la ecuación 4.7 y 4.8 corresponden a $f_{c_i} - f_{e_i}$) del sistema. Además podemos resolver para cada ω un vector $\{a\}$ llamado modo de vibración (en la ecuación 4.7 y 4.8 corresponden a $\phi_{c_{ij}} - \phi_{e_{ij}}$).

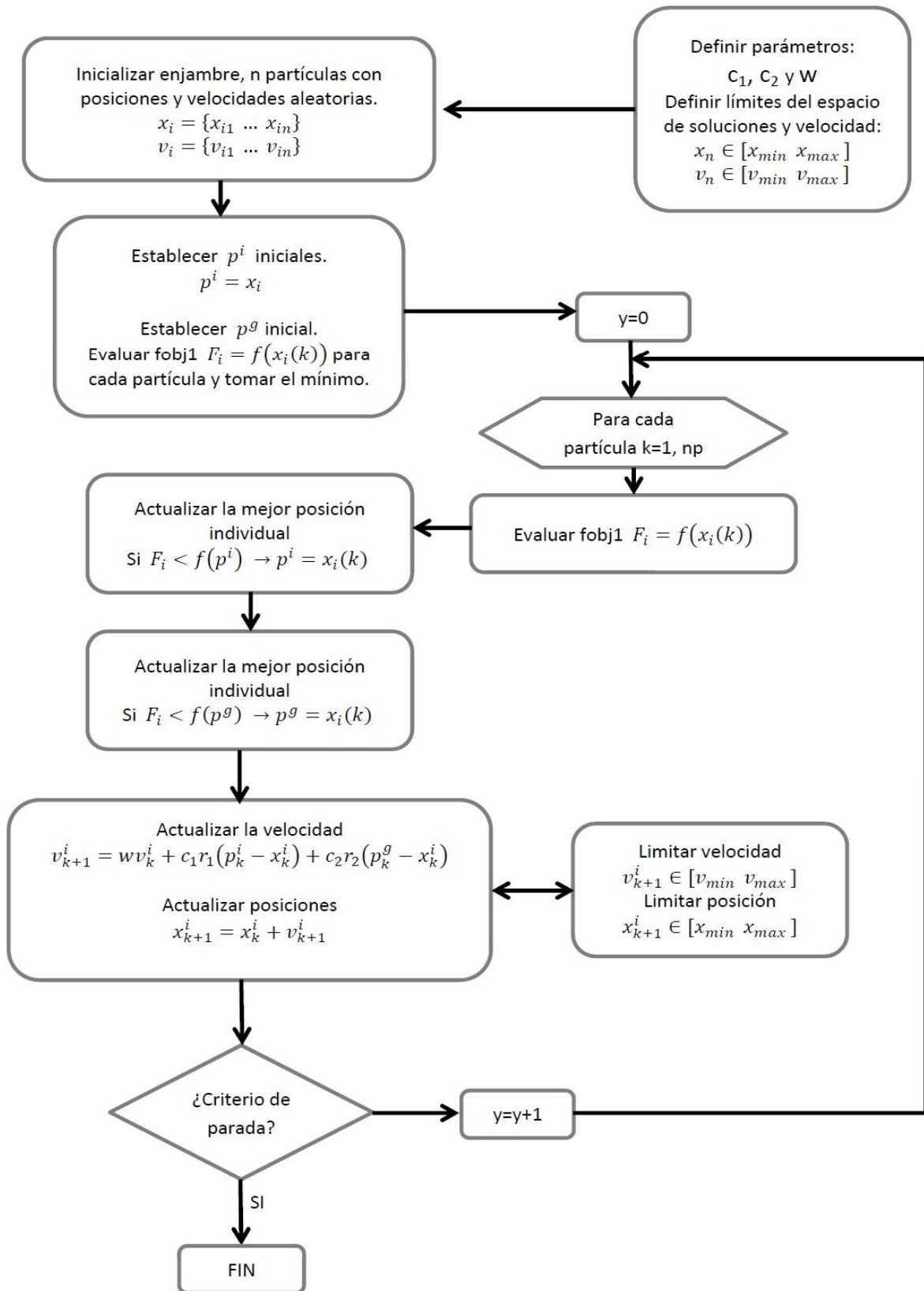


Figura 1. Diagrama de flujo del algoritmo de optimización de partículas.

4.2 ALGORITMO GENETICO (AG)

Los algoritmos genético (AG) son métodos usados para la búsqueda y optimización. Estos imitan los principios de selección natural de Darwin (1859), que consisten en la evolución natural de acuerdo a los principios de selección natural y la supervivencia de los más fuertes.

Los AG trabajan con una población de individuos que representan soluciones factibles a un problema determinado. Cada individuo tiene un valor o puntuación asignado relacionado con características de la solución. Cuanto mayor sea la capacidad de adaptación del individuo al problema, mayor será la probabilidad de que el mismo sea seleccionado para reproducirse y por tanto su material genético se propague en sucesivas generaciones.

De esta manera se produce una nueva generación de posibles soluciones que contiene mejores características y eventualmente convergerá a una solución óptima del problema.

El AG está basado en el teorema del esquema, el cual conjuga función objetivo, cruce y mutación que establece la afectación de la supervivencia y propagación de las soluciones. Tiene un esquema versátil que permite adaptarse con facilidad a cualquier tipo de problema y sus continuas modificaciones han mejorado su rendimiento.

Se utilizó el AG básico Holland 1975, el código se encuentra especificado en el ANEXO 7. Con el fin de evaluar el desempeño del algoritmo se realizaron 5 pruebas (ver ANEXO 8.), en las cuales se usaron 100 individuos, una probabilidad de mutación del 2%, una probabilidad de cruce del 80% y el criterio de para fue el número de generaciones (iteraciones) y una desviación estándar menor a 1×10^{-30} en los últimos 1000 resultados de las funciones objetivo. Estas funciones de

prueba fueron extraídas de la literatura internacional. (Georgieva A., Jordanov I., European Journal of Operational Research. 2009)

La función objetivo implementada es basada en la función objetivo utilizada en el PSO (ecuación 4.7) pero modificada de acuerdo al AG, debido que este código sirve para maximizar y lo que se necesita con la función objetivo del PSO es encontrar el mínimo.

$$fobj5 = C - \left[\sum_{i=1}^n \sum_{j=1}^m Abs \left(1 - abs \left(\frac{\phi_{c_{ij}}}{\phi_{e_{ij}}} \right) \right) + \sum_i^n Abs \left(1 - abs \left(\frac{f_{c_i}}{f_{e_i}} \right) \right) \right] \quad (4.8)$$

En la ecuación (4.8) se propone una constante C que garantiza que al maximizar la ecuación, dará un resultado óptimo igual a C y esto sucede cuando el resto de la ecuación es igual a cero. En este trabajo C tiene un valor de 20. Este valor fue seleccionado por medio del método de prueba y error.

En la selección natural se eligen los padres por el método de la ruleta dando así mayor oportunidad de ser seleccionado al individuo de mejor ubicación. En este caso el individuo con el valor de función objetivo más alto. En este proceso se ponderan todos los individuos dando la ponderación más alta al individuo con mayor valor de la función objetivo y menor ponderación al individuo con menor valor de la función objetivo, después de tener todos los individuos ponderados se eligen de manera aleatoria cuales pueden ser padres o posibles padres, se continúa con el cruce que se rige por la probabilidad de cruce, en este trabajo es del 80%, dando así la probabilidad de que un 20% de los individuos no se crucen y no continúen a la siguiente generación. Para este proceso de cruce se propuso una ecuación que simula la reproducción sexual así:

$$X_h = X_{p_{menor}}^i - r * a + rand * \left((X_{p_{mayor}}^j + r * a) - (X_{p_{menor}}^i - r * a) \right) \quad (4.9)$$

$$r = X_{p\ mayor}^j - X_{p\ menor}^i \quad (4.9a)$$

Donde:

X_h = Nuevo individuo generado por el cruce de X_p^j y X_p^i .

$X_{p\ menor}^i$ = Individuo padre de menor magnitud.

$X_{p\ mayor}^j$ = Individuo padre de mayor magnitud.

r = Espacio entre X_p^j y X_p^i .

a = Constante de expansión del espacio para la generación del nuevo individuo X_h .

$rand$ = Número aleatorio distribuido entre 0 y 1.

Después de haber realizado el cruce se forma el proceso de mutación el cual se rige por la probabilidad de mutación que en este trabajo es del 20% y consiste en que un 20% de la población puede ser mutada y genera un individuo completamente nuevo sin patrón alguno.

Después de terminados dichos procesos se realiza un paso llamado elitismo el cual consiste en garantizar que el mejor miembro de cada generación haga parte de la siguiente generación.

Los codigos de los algoritmos usados para el PSO y AG con el acople de OPENSEES se encuentran en los anexos 10 y 11 respectivamente.

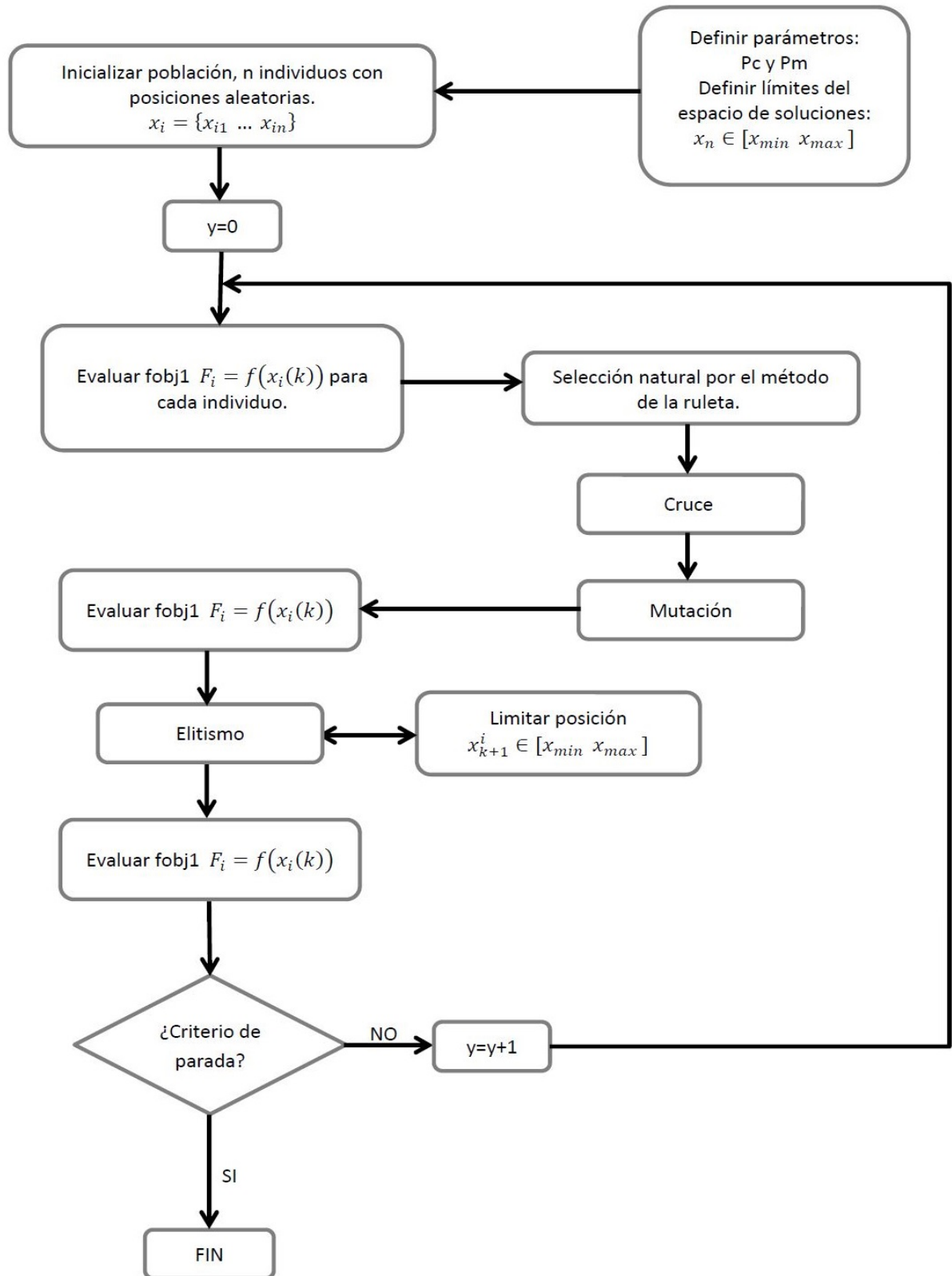


Figura 2. Diagrama de flujo del algoritmo genético.

5. EVALUACION DE DESEMPEÑO DEL ALGORITMO GENETICO (AG) Y EL ALGORITMO DE OPTIMIZACION DE PARTICULAS (PSO) EN DETECCION 3D.

SIMULACION DE DAÑO:

La matriz de rigidez de un elemento i con daño está dada por:

$$[K]_{di} = \alpha_i [K]_i \quad (4.7a)$$

Dónde:

α_i = Factor de daño del elemento i . Valores entre 0 y 1

$[K]_i$ = Matriz de rigidez sin daño del elemento i .

Esta ecuación se encuentra detallada en el capítulo 4.4.1

Para cada tipo de daño se evalúan los algoritmos (PSO Y AG) en pórticos en 3D. Se tienen en cuenta variables que pueden afectar el desempeño del programa como:

1. Simetría del pórtico (en longitudes y secciones transversales). Los pórticos usados para realizar la evaluación se encuentran descritos en el capítulo 4.1.1.1 y 4.1.1.2.
2. Porcentaje de daño: para cada caso se evalúa un daño del 20, 10, 5 y 1 por ciento. Esto con el fin de evaluar hasta que porcentaje de daño pueden detectar los algoritmos.
3. Calidad de datos: se evalúa el funcionamiento del programa con un porcentaje determinado de ruido, es decir cuando los datos de entrada

(frecuencias y modos de vibración) tienen algún margen de error. Los parámetros de ruido aplicados son los siguientes:

- Para las frecuencias un rango de (0-5%) de ruido. usando el valor crítico, el ruido aplicado a las frecuencias es de un 5%.
- Para los modos de vibración un rango de (0-2%) de ruido. usando el valor crítico, el ruido aplicado a los modos es de un 2%.

5.1 CASOS DE DAÑO

Se simulan dos tipos de daño para cada uno de los pórticos descritos en el capítulo 4.1.1.1 y 4.1.1.2 (simétrico y asimétrico).

1. Caso de daño único: se simulan los diferentes porcentajes de falla en un elemento del pórtico. Para saber si los resultados varían de acuerdo al tipo de elemento se simula una falla tipo columna en el elemento 1 (ver figuras 6.a y 6.d) y uno tipo viga en el elemento 20 (ver figuras 6.b y 6e).
2. Caso de daño múltiple: se simulan los diferentes porcentajes de falla para tres elementos del pórtico; en el caso del pórtico simétrico se fallan los elementos 9,10 y 18 mostrados en la figura 6.c y para el caso del pórtico asimétrico los elementos 9,13 y 18 mostrados en la figura 6.f.

Se utilizó el pórtico completamente simétrico para hacer las pruebas con ruido debido a que este obtuvo resultados más exactos en la detección de daño, fue más eficiente pues necesitaba menos iteraciones para llegar a una respuesta, lo que llevaba a que tomara menos tiempo. Las figuras que corresponden a los casos de daño con ruido son las figuras 6.a, 6.b y 6.c.

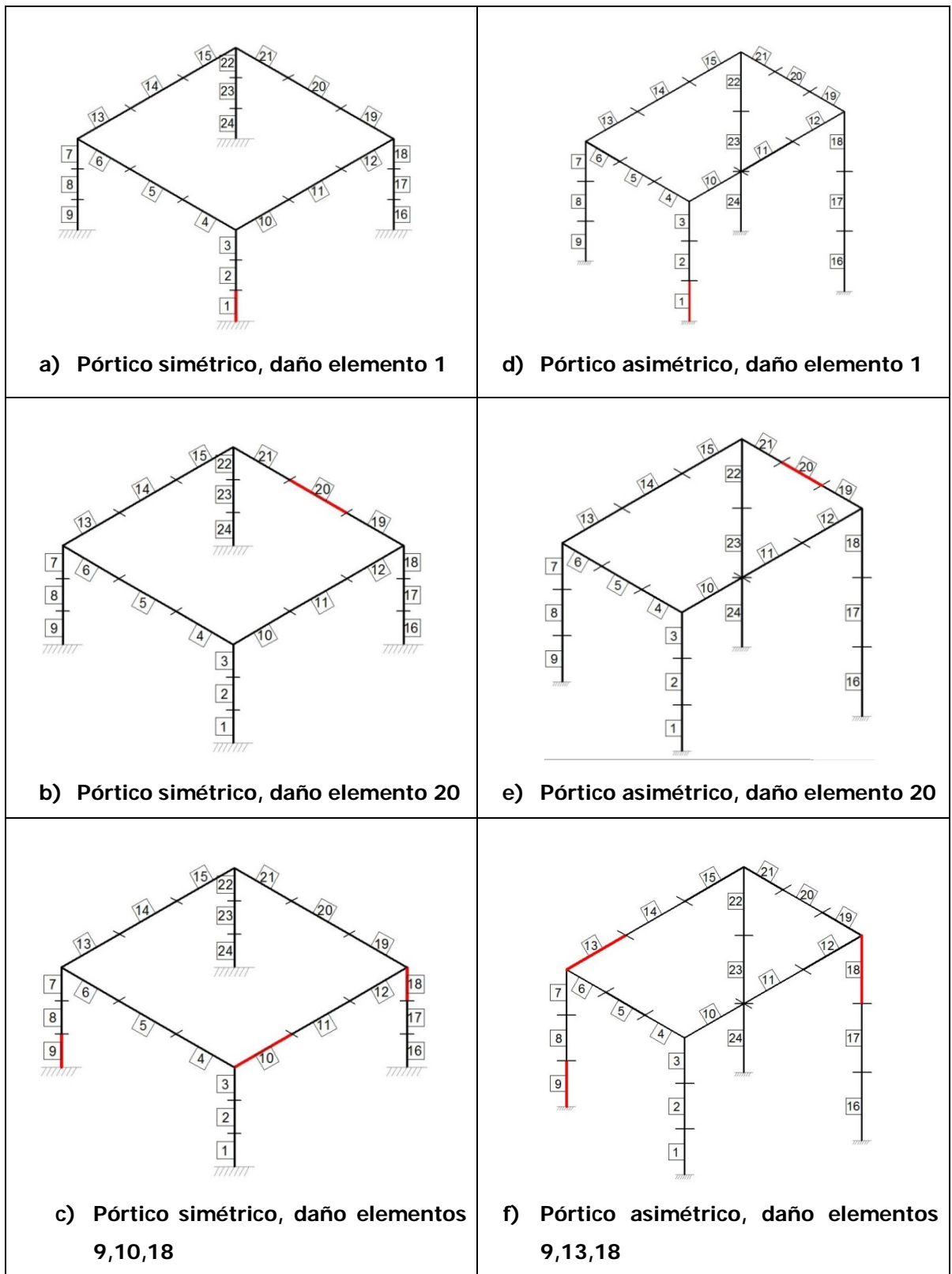


Figura 3. Casos de daño.

5.2 RESULTADOS OBTENIDOS USANDO PSO

Para cada uno de los casos de daño se corrió el algoritmo y se registraron los datos de tres pruebas con el fin de observar su comportamiento. Para el caso de daño único se introduce daño en el elemento 1 y en el elemento 20, y para el caso de daño múltiple se dañifican dos elementos de columna y un elemento de viga (elementos 9 13 y18 para pórtico asimétrico, elementos 9 10 y 18 para el pórtico simétrico). Para cada uno de los casos de daño se realizaron pruebas con porcentajes de daño del 20%, 10%, 5% y 1%.

Para los casos de daño donde se aplica ruido inicialmente se realizaron pruebas donde no había ningún tipo de daño para calcular la variación de los datos comparados con aquellas pruebas libres de ruido; debido a que el ruido puede generar una suposición de daño de hasta un 3% el valor mínimo de detección se modificó de 1% a 5% para los casos con ruido (Tabla 13).

Para cada caso de daño se propuso un número máximo de iteraciones según el comportamiento del modelo propuesto. Adicionalmente, en la tabla 6 se presentan los casos de daño para los pórticos estudiados con el PSO y sus porcentajes de daño correspondientes.

		Máximo de Iteraciones			
		20%	10%	5%	1%
Pórtico Simétrico	Elemento 1	100	100	100	100
	Elemento 20	100	100	100	100
	Elementos 9 10 18	100	100	100	100
Pórtico Asimétrico	Elemento 1	100	100	100	100
	Elemento 20	100	100	100	100
	Elementos 9 13 18	500	500	500	500
Pórtico Simétrico (Ruido)	Elemento 1	200	-	200	-
	Elemento 20	100	-	200	-
	Elementos 9 10 18	200	-	200	-

Tabla 6. Tabla de resultados PSO. Número máximo de iteraciones.

Los resultados completos de las pruebas realizadas donde se muestran las coordenadas de daño para cada uno de los elementos del pórtico 3D se encuentran en el ANEXO 11.

5.2.1 CASO DE DAÑO ÚNICO (DATOS SIN RUIDO)

En las tablas 7 y 8 se muestran los resultados de las tres pruebas para la simulación de daño único en el pórtico simétrico simulando el daño en los elementos 1 y 20 respectivamente con todos los porcentajes de daño (20%, 10%, 5% y 1%).

	CASO DE DAÑO	DATOS	PRUEBA	PRUEBA	PRUEBA
			1	2	3
PORTICO SIMETRICO	UNICO	# ITERACIONES	74.00	71.00	61.00
	ELEMENTO : 1	TIEMPO MINUTOS	22.63	23.40	19.17
	PORCENTAJE :20	FUNCION OBJETIVO	0.00	0.00	0.00
	UNICO	# ITERACIONES	69.00	39.00	85.00
	ELEMENTO : 1	TIEMPO MINUTOS	20.05	11.40	24.58
	PORCENTAJE :10	FUNCION OBJETIVO	0.00	0.00	0.00
	UNICO	# ITERACIONES	51.00	82.00	64.00
	ELEMENTO : 1	TIEMPO MINUTOS	15.15	24.60	18.75
	PORCENTAJE :5	FUNCION OBJETIVO	0.00	0.00	0.00
	UNICO	# ITERACIONES	55.00	76.00	55.00
	ELEMENTO : 1	TIEMPO MINUTOS	16.05	22.17	16.00
	PORCENTAJE :1	FUNCION OBJETIVO	0.00	0.00	0.00

Tabla 7. Resultados PSO – Pórtico simétrico – Simulación de daño en el elemento 1 con porcentajes de daño del 20%, 10%, 5% y 1%.

PORTICO SIMETRICO	CASO DE DAÑO	DATOS	PRUEBA 1	PRUEBA 2	PRUEBA 3
	UNICO ELEMENTO : 20 PORCENTAJE :20	# ITERACIONES	67.00	32.00	61.00
	TIEMPO MINUTOS	19.73	9.55	19.17	
	FUNCION OBJETIVO	0.00	0.00	0.00	
UNICO ELEMENTO : 1 PORCENTAJE :10	# ITERACIONES	89.00	77.00	76.00	
	TIEMPO MINUTOS	15.90	9.25	17.88	
	FUNCION OBJETIVO	0.00	0.00	0.00	
UNICO ELEMENTO : 20 PORCENTAJE :5	# ITERACIONES	51.00	31.00	61.00	
	TIEMPO MINUTOS	25.98	22.45	22.12	
	FUNCION OBJETIVO	0.00	0.00	0.00	
UNICO ELEMENTO : 20 PORCENTAJE :1	# ITERACIONES	30.00	63.00	39.00	
	TIEMPO MINUTOS	8.97	18.53	11.57	
	FUNCION OBJETIVO	0.00	0.00	0.16	

Tabla 8. Resultados PSO – Pórtico simétrico – Simulación de daño en el elemento 20 con porcentajes de daño del 20%, 10%, 5% y1%.

En las tablas 9 y 10 se muestran los resultados de las tres pruebas para la simulación de daño único para el pórtico asimétrico simulando el daño en los elementos 1 y 20 respectivamente con todos los porcentajes de daño (20%, 10%, 5% y 1%).

PORTICO ASIMETRICO	CASO DE DAÑO	DATOS	PRUEBA 1	PRUEBA 2	PRUEBA 3
	UNICO ELEMENTO : 1 PORCENTAJE :20	# ITERACIONES	65.00	82.00	89.00
	TIEMPO MINUTOS	23.33	28.64	30.57	
	FUNCION OBJETIVO	0.00	0.00	0.00	
UNICO ELEMENTO : 1 PORCENTAJE :10	# ITERACIONES	78.00	87.00	86.00	
	TIEMPO MINUTOS	27.29	30.29	30.68	
	FUNCION OBJETIVO	0.00	0.00	0.00	
UNICO ELEMENTO : 1 PORCENTAJE :5	# ITERACIONES	80.00	71.00	75.00	
	TIEMPO MINUTOS	28.65	25.68	27.42	
	FUNCION OBJETIVO	0.00	0.00	0.00	
UNICO ELEMENTO : 1 PORCENTAJE :1	# ITERACIONES	27.00	100.00	100.00	
	TIEMPO MINUTOS	11.17	37.07	36.31	
	FUNCION OBJETIVO	9.08	0.49	3.71	

Tabla 9. Resultados PSO – Pórtico asimétrico – Simulación de daño en el elemento 1 con porcentajes de daño del 20%, 10%, 5% y1%.

PORTICO ASIMETRICO	CASO DE DAÑO	DATOS	PRUEBA 1	PRUEBA 2	PRUEBA 3
	UNICO ELEMENTO : 20 PORCENTAJE :20	# ITERACIONES	61.00	69.00	68.00
		TIEMPO MINUTOS	21.79	23.42	23.11
		FUNCION OBJETIVO	0.00	0.00	0.00
	UNICO ELEMENTO : 1 PORCENTAJE :10	# ITERACIONES	71.00	31.00	73.00
		TIEMPO MINUTOS	25.98	12.74	26.39
FUNCION OBJETIVO		0.00	0.00	0.00	
UNICO ELEMENTO : 20 PORCENTAJE :5	# ITERACIONES	60.00	64.00	60.00	
	TIEMPO MINUTOS	20.50	21.77	20.53	
	FUNCION OBJETIVO	0.00	0.00	0.00	
UNICO ELEMENTO : 20 PORCENTAJE :1	# ITERACIONES	63.00	45.00	48.00	
	TIEMPO MINUTOS	21.30	16.21	18.24	
	FUNCION OBJETIVO	0.00	0.00	0.00	

Tabla 10. Resultados PSO – Pórtico asimétrico – Simulación de daño en el elemento 20 con porcentajes de daño del 20%, 10%, 5% y 1%.

5.2.2 CASO DE DAÑO MULTIPLE (DATOS SIN RUIDO)

En las tablas 11 y 12 se muestran los resultados de las tres pruebas para daño múltiple en los pórticos simétrico y asimétrico respectivamente con todos los porcentajes de daño (20%, 10%, 5% y 1%).

PORTICO SIMETRICO	CASO DE DAÑO	DATOS	PRUEBA 1	PRUEBA 2	PRUEBA 3
	MULTIPLE ELEMENTO : 9-10-18 PORCENTAJE :20	# ITERACIONES	91.00	91.00	85.00
		TIEMPO MINUTOS	26.27	26.08	24.33
		FUNCION OBJETIVO	0.00	0.00	0.00
	MULTIPLE ELEMENTO : 9-10-18 PORCENTAJE :10	# ITERACIONES	81.00	80.00	98.00
		TIEMPO MINUTOS	23.30	22.98	28.15
FUNCION OBJETIVO		0.00	0.00	0.00	
MULTIPLE ELEMENTO : 9-10-18 PORCENTAJE :5	# ITERACIONES	73.00	83.00	71.00	
	TIEMPO MINUTOS	23.48	24.15	22.82	
	FUNCION OBJETIVO	2.23	0.00	2.07	
MULTIPLE ELEMENTO : 9-10-18 PORCENTAJE :1	# ITERACIONES	100.00	53.00	82.00	
	TIEMPO MINUTOS	28.57	15.33	23.52	
	FUNCION OBJETIVO	0.27	0.23	0.23	

Tabla 11. Resultados PSO – Pórtico simétrico – Simulación de daño en los elemento 9 – 10 – 18 simultáneamente con porcentajes de daño del 20%, 10%, 5% y 1 %.

	CASO DE DAÑO	DATOS	PRUEBA	PRUEBA	PRUEBA
			1	2	3
PORTICO ASIMETRICO	MULTIPLE	# ITERACIONES	500.00	105.00	107.00
	ELEMENTO : 9-13-18	TIEMPO MINUTOS	195.35	38.74	36.57
	PORCENTAJE :20	FUNCION OBJETIVO	1.35	0.00	0.00
	MULTIPLE	# ITERACIONES	100.00	100.00	100.00
	ELEMENTO : 9-13-18	TIEMPO MINUTOS	50.43	39.57	36.46
	PORCENTAJE :10	FUNCION OBJETIVO	6.17	4.89	1.75
	MULTIPLE	# ITERACIONES	100.00	100.00	100.00
	ELEMENTO : 9-13-18	TIEMPO MINUTOS	36.75	36.67	37.06
	PORCENTAJE :5	FUNCION OBJETIVO	0.00	1.41	0.04
	MULTIPLE	# ITERACIONES	150.00	138.00	118.00
	ELEMENTO : 9-13-18	TIEMPO MINUTOS	53.76	48.78	41.53
	PORCENTAJE :1	FUNCION OBJETIVO	0.49	0.45	0.33

Tabla 12. Resultados PSO – Pórtico asimétrico – Simulación de daño en los elemento 9–13–18 simultáneamente con porcentajes de daño del 20%, 10%, 5% y 1%.

5.2.3 CASOS DE DAÑO APLICANDO RUIDO

5.2.3.1 CASO SIN DAÑO

Se realizaron tres pruebas aplicando ruido a los datos de entrada (frecuencias y modos) sin simular daño a ningún elemento. De esta manera se encontró que el ruido puede generar hasta una suposición de daño del 3% en cualquier elemento. A continuación se presentan los resultados de estas pruebas en la tabla 13.

		Prueba # 1	Prueba # 2	Prueba # 3
Función Objetivo =		2.31	2.22	2.21
Iteraciones realizadas =		63	300	183
Tiempo [seg] =		1114	5253	3208
Tiempo [min] =		18.6	87.5	53.4
Elemento	α_e	α_{c1}	α_{c2}	α_{c3}
1	1	1	1	1
2	1	1	0.98	1
3	1	1	1	1
4	1	1	1	1
5	1	1	1	1
6	1	1	1	1
7	1	1	1	1
8	1	1	1	1
9	1	1	1	1
10	1	1	1	1
11	1	1	1	1
12	1	1	1	0.97
13	1	1	1	1
14	1	1	1	1
15	1	1	0.99	1
16	1	1	0.99	1
17	1	1	1	1
18	1	0.98	0.97	0.99
19	1	1	0.99	1
20	1	1	1	1
21	1	1	1	0.98
22	1	1	1	1
23	1	1	1	1
24	1	1	1	1

Tabla 13. Resultados – PSO – Pórtico simétrico con ruido, sin daño.

5.2.3.2 CASO DE DAÑO UNICO (CON RUIDO)

En la tabla 14 se muestran los resultados de las tres pruebas realizadas para daño único en el pórtico simétrico aplicando ruido a los datos de entrada y para casos con magnitud de daño del 20% y 5%.

	CASO DE DAÑO	DATOS	PRUEBA	PRUEBA	PRUEBA
			1	2	3
PORTICO SIMETRICO CON RUIDO	UNICO	# ITERACIONES	189.00	147.00	111.00
	ELEMENTO : 1	TIEMPO MINUTOS	64.27	44.32	32.20
	PORCENTAJE :20	FUNCION OBJETIVO	2.54	2.56	2.60
	UNICO	# ITERACIONES	100.00	100.00	100.00
	ELEMENTO : 1	TIEMPO MINUTOS	37.06	37.15	36.74
	PORCENTAJE :5	FUNCION OBJETIVO	4.40	2.35	2.32
	UNICO	# ITERACIONES	100.00	100.00	96.00
	ELEMENTO : 20	TIEMPO MINUTOS	42.26	37.94	35.68
	PORCENTAJE :20	FUNCION OBJETIVO	2.25	2.17	2.26
	UNICO	# ITERACIONES	114.00	121.00	150.00
	ELEMENTO : 20	TIEMPO MINUTOS	40.85	43.73	56.65
	PORCENTAJE :5	FUNCION OBJETIVO	2.45	2.55	2.23

Tabla 14. Resultados PSO – Pórtico simétrico – Simulación de daño en los elementos 1 y 20 con porcentajes de daño del 20% y 5%.

5.2.3.3 CASO DE DAÑO MULTIPLE (CON RUIDO)

En la tabla 15 se muestran los resultados de las tres pruebas realizadas para daño múltiple en el pórtico simétrico aplicando ruido a los datos de entrada y para casos con magnitud de daño del 20% y 5%.

PORTICO SIMETRICO CON RUIDO	CASO DE DAÑO	DATOS	PRUEBA 1	PRUEBA 2	PRUEBA 3
	MULTIPLE ELEMENTO : 9-10-18 PORCENTAJE :20	ITERACIONES	104.00	109.00	178.00
		TIEMPO MIN.	30.43	31.80	51.97
		FUN OBJETIVO	2.43	2.43	2.38
	MULTIPLE ELEMENTO : 9-10-18 PORCENTAJE :5	ITERACIONES	125.00	200.00	194.00
		TIEMPO MIN.	42.95	68.08	67.95
FUN OBJETIVO		2.35	4.37	2.49	

Tabla 15. Resultados PSO – Pórtico simétrico – Simulación de daño en los elementos 9-10-18 simultáneamente con porcentajes de daño del 20% y 5%.

5.3 RESULTADOS OBTENIDOS USANDO AG

Para cada uno de los casos de daño se corre el algoritmo y se registran los datos de tres pruebas con el fin de observar su comportamiento. Para el caso de daño único se introduce una pérdida de rigidez de 20% para el elemento 1 y se repite el proceso para el elemento 20, y el caso de daño múltiple se simula una falla dos elementos de columna y un elemento de viga con una pérdida de rigidez de 20% (elementos 9, 13 y 18 para pórtico asimétrico, elementos 9, 10 y 18 para el pórtico simétrico).

Se decidió no efectuar las pruebas con los porcentajes de 10, 5 y 1 por ciento para el algoritmo AG debido a que los resultados obtenidos no convergen en el tiempo esperado con el número de iteraciones propuestas, siendo menos eficaz que el PSO (ver de la figura 7 a la 12). Cabe anotar que si el AG no se limita a un número de evaluaciones iguales a las realizadas con el PSO consigue llegar a resultados aceptables.

Con el fin de sustentar lo dicho anteriormente se realizaron las pruebas para un caso de daño único utilizando los mismos parámetros del algoritmo (probabilidad de cruce y mutación) y se modificó el número máximo de iteraciones, usando 100 individuos y 5000 generaciones. En esta prueba se puede mostrar que el algoritmo genético (AG) si realiza una búsqueda correcta pero toma un mayor número de iteraciones, haciendo que el tiempo de cálculo sea más extenso. (Ver tabla 22).

Para cada caso de daño se propuso un número máximo de iteraciones igual a las usadas en el PSO (Tabla 6). En la tabla 16 se presentan las iteraciones realizadas para el AG.

		Máximo de Iteraciones			
		20%	10%	5%	1%
Pórtico Simétrico	Elemento 1	100	-	-	-
	Elemento 20	100	-	-	-
	Elementos 9 10 18	100	-	-	-
Pórtico Asimétrico	Elemento 1	100	-	-	-
	Elemento 20	100	-	-	-
	Elementos 9 13 18	500	-	-	-
Pórtico Simétrico (Ruido)	Elemento 1	200	-	-	-
	Elemento 20	100	-	-	-
	Elementos 9 10 18	200	-	-	-

Tabla 16. Tabla de resultados AG. Número máximo de iteraciones.

5.3.1 CASO DE DAÑO ÚNICO (SIN RUIDO)

En las tablas 17 y 18 se muestran los resultados obtenidos con el AG para casos de daño único en los pórticos simétrico y asimétrico respectivamente.

PORTICO SIMETRICO	CASO DE DAÑO	DATOS	PRUEBA 1	PRUEBA 2	PRUEBA 3
	UNICO ELEMENTO : 1 PORCENTAJE :20	# ITERACIONES		100.00	100.00
TIEMPO MINUTOS			63.25	63.30	63.13
FUNCION OBJETIVO			14.29	13.58	14.12
UNICO ELEMENTO : 20 PORCENTAJE :20	# ITERACIONES		100.00	100.00	100.00
	TIEMPO MINUTOS		62.78	62.75	63.78
	FUNCION OBJETIVO		14.96	16.42	16.16

Tabla 17. Resultados AG – Pórtico simétrico – Simulación de daño en los elementos 1 y 20 con porcentajes de daño del 20%.

PORTICO ASIMETRICO	CASO DE DAÑO	DATOS	PRUEBA 1	PRUEBA 2	PRUEBA 3
	UNICO ELEMENTO : 1 PORCENTAJE :20	# ITERACIONES		100.00	100.00
TIEMPO MINUTOS			61.43	59.35	60.90
FUNCION OBJETIVO			7.68	10.69	13.32
UNICO ELEMENTO : 20 PORCENTAJE :20	# ITERACIONES		100.00	100.00	100.00
	TIEMPO MINUTOS		64.13	59.80	59.22
	FUNCION OBJETIVO		18.26	18.29	18.32

Tabla 18. Resultados AG – Pórtico asimétrico – Simulación de daño en los elementos 1 y 20 con porcentajes de daño del 20%.

5.3.2 CASO DE DAÑO MULTIPLE (SIN RUIDO)

En la tabla 19 se muestran los resultados obtenidos con el AG para los pórticos (simétrico y asimétrico) con simulación de daño múltiple.

	CASO DE DAÑO	DATOS	PRUEBA	PRUEBA	PRUEBA
			1	2	3
PORTICO SIMETRICO	MULTIPLE	ITERACIONES	63.23	63.18	63.40
	ELEMENTO : 9-10-18	TIEMPO MIN	100.00	100.00	100.00
	PORCENTAJE :20	FUN OBJETIVO	12.99	13.97	12.96
PORTICO ASIMETRICO	MULTIPLE	ITERACIONES	500.00	500.00	500.00
	ELEMENTO : 9-13-18	TIEMPO MIN	334.08	304.33	297.50
	PORCENTAJE :20	FUN OBJETIVO	9.09	7.58	8.56

Tabla 19. Resultados AG – Pórticos simétrico y asimétrico – Simulación de daño en los elementos 9-10-18 y 9-13-18 respectivamente y simultáneamente, con porcentaje de daño del 20%.

5.3.3 CASOS DE DAÑO APLICANDO RUIDO

5.3.3.1 CASO DE DAÑO UNICO (CON RUIDO)

En la tabla 20 se muestran los resultados obtenidos con el AG para el pórtico simétrico aplicando ruido en los datos de entrada para una simulación de daño del 20% en los elementos 1 y 20 respectivamente.

	CASO DE DAÑO	DATOS	PRUEBA	PRUEBA	PRUEBA
			1	2	3
PORTICO SIMETRICO CON RUIDO	UNICO	ITERACIONES	100.00	100.00	100.00
	ELEMENTO : 1	TIEMPO MIN	62.77	66.00	65.73
	PORCENTAJE :20	FUN OBJETIVO	14.21	13.05	15.06
	UNICO	ITERACIONES	100.00	100.00	100.00
	ELEMENTO : 20	TIEMPO MIN	62.50	62.75	62.58
	PORCENTAJE :20	FUN OBJETIVO	15.63	16.12	16.30

Tabla 20. Resultados AG – Pórticos simétrico con ruido – Simulación de daño en los elementos 1 y 20 respectivamente, con porcentajes de daño del 20%.

5.3.3.2 CASO DE DAÑO MULTIPLE (CON RUIDO)

En la tabla 21 se muestran los resultados obtenidos con el AG para el pórtico simétrico aplicando ruido a los datos de entrada para una simulación de daño del 20% en los elementos 9-10-18 simultáneamente.

CASO DE DAÑO	DATOS	PRUEBA 1	PRUEBA 2	PRUEBA 3
MULTIPLE	# ITERACIONES	100.00	100.00	100.00
ELEMENTO : 9-10-18	TIEMPO MINUTOS	73.65	68.57	66.82
PORCENTAJE :20	FUNCION OBJETIVO	12.97	13.45	12.58

Tabla 21. Resultados AG – Pórticos simétrico con ruido – Simulación de daño en los elementos 9-10-18 simultáneamente, con porcentajes de daño del 20%.

5.3.4 CASO DAÑO UNICO PORTICO SIMETRICO CON VARIACION EN EL NUMERO DE ITERACIONES

Con el fin de verificar el funcionamiento del AG en este problema se decide aumentar el número máximo de iteraciones a 5000, los resultados obtenidos son:

		Prueba # 1	Prueba # 2	Prueba # 3
Función Objetivo =		18.12	18.51	19.07
Iteraciones =		5000	5000	5000
Tiempo [seg] =		186385	193950	178820
Tiempo [min] =		3106.4	3232.5	2980.3
Tiempo [horas] =		51.77	53.87	49.67
Elemento	α_e	α_{c1}	α_{c2}	α_{c3}
1	0.80	0.81	0.81	0.81
2	1.00	0.99	0.98	0.99
3	1.00	0.96	0.97	0.92
4	1.00	0.93	0.97	0.97
5	1.00	0.98	0.98	0.97
6	1.00	0.95	0.97	0.97
7	1.00	0.92	0.97	0.97
8	1.00	0.97	0.97	0.97
9	1.00	0.99	0.98	0.97
10	1.00	0.93	0.97	0.97
11	1.00	1.00	0.97	0.98
12	1.00	0.96	0.98	0.98
13	1.00	0.98	0.98	0.98
14	1.00	0.94	0.98	0.98
15	1.00	0.99	0.97	0.98
16	1.00	0.99	0.97	0.97
17	1.00	0.98	0.98	0.97
18	1.00	0.98	0.98	0.98
19	1.00	0.97	0.98	0.97
20	1.00	0.97	0.97	0.98
21	1.00	0.98	0.97	0.98
22	1.00	0.99	0.98	0.98
23	1.00	0.99	0.98	0.98
24	1.00	0.98	0.97	0.98

Tabla 22. Resultados AG – Pórticos simétrico – Simulación de daño en el elemento 1, con porcentajes de daño del 20%.

Como de observa en la tabla 22 los resultados obtenidos con el AG después de 5000 iteraciones muestran mejoría con respecto a los vistos en la tabla 17 y un valor de la función objetivo muy cercano al óptimo.

5.4 ANALISIS DE RESULTADOS

Para el análisis de resultados se calcularon la media y la desviación estándar para las tres pruebas realizadas en cada caso mostrado en el numeral 5.2.

En las siguientes tablas se muestran la media y la desviación estándar calculada con las tres pruebas realizadas para la función objetivo, el número de iteraciones y el tiempo medido en minutos, en todos los porcentajes de daño corridos (Tablas de la 23 a la 26).

En estas tablas *elem* representa el nombre del elemento que presenta el daño, *fun objetivo* es el valor de la función objetivo y *tiempo* es la duración del proceso.

En las siguientes tablas (tabla 23 y 24) se muestran los cálculos realizados para los resultados obtenidos con el PSO.

		Media				Desviación Estándar		
	Elem	% Daño	Fun Objetivo	Iteraciones	Tiempo [min]	Fun Objetivo	Iteraciones	Tiempo [min]
PORTICO SIMÉTRICO	1	20%	0	68.67	21.73	0	5.56	1.84
		10%	0	64.33	18.68	0	19.07	5.47
		5%	0	65.67	19.5	0	12.71	3.89
		1%	0	62	18.07	0	9.9	2.9
	20	20%	0	53.33	16.15	0	15.28	4.67
		10%	0	80.67	14.34	0	5.91	3.69
		5%	0	47.67	23.52	0	12.47	1.75
		1%	0.05	44	13.02	0.08	13.93	4.04
	9-10-18	20%	0	89	25.56	0	2.83	0.87
		10%	0	86.33	24.81	0	8.26	2.36
		5%	1.43	75.67	23.48	1.02	5.25	0.54
		1%	0.24	78.33	22.47	0.02	19.36	5.45

Tabla 23. Media y Desviación Estándar – Pórtico simétrico – PSO.

	Elem	% Daño	Media			Desviación Estándar		
			Fun Objetivo	Iteraciones	Tiempo [min]	Fun Objetivo	Iteraciones	Tiempo [min]
PÓRTICO ASIMÉTRICO	1	20%	0	78.67	27.51	0	10.08	3.06
		10%	0	83.67	29.42	0	4.03	1.51
		5%	0	75.33	27.25	0	3.68	1.22
		1%	4.43	75.67	28.18	3.54	34.41	12.03
	20	20%	0	66	22.77	0	3.56	0.71
		10%	0	58.33	21.7	0	19.34	6.34
		5%	0	61.33	20.93	0	1.89	0.59
		1%	0	52	18.58	0	7.87	2.09
	9-13-18	20%	0.45	237.3	90.22	0.64	185.7	74.34
		10%	4.27	100	42.15	1.86	0	5.99
		5%	0.49	100	36.82	0.65	0	0.17
		1%	0.42	135.3	48.02	0.07	13.2	5.02

Tabla 24. Media y Desviación Estándar – Pórtico asimétrico – PSO.

Para seleccionar el pórtico al que se le aplicó el ruido, se comparó la media de la función objetivo y el número de iteraciones de los resultados obtenidos entre los dos pórticos (simétrico y asimétrico). Ver figuras 4, 5 y 6.

En la figura 4 se muestra que la detección de daño en el elemento 1 es más refinada cuando se hace en un pórtico simétrico. Esto se puede observar comparando los valores de la media calculada para la función objetivo en los dos pórticos, en el caso del pórtico simétrico los datos se acercan al valor óptimo (cero), mientras que en el asimétrico hubo una diferencia de hasta 4.43; es evidente también que la convergencia para todos los porcentajes de daño siempre requirió menos iteraciones en el pórtico simétrico.

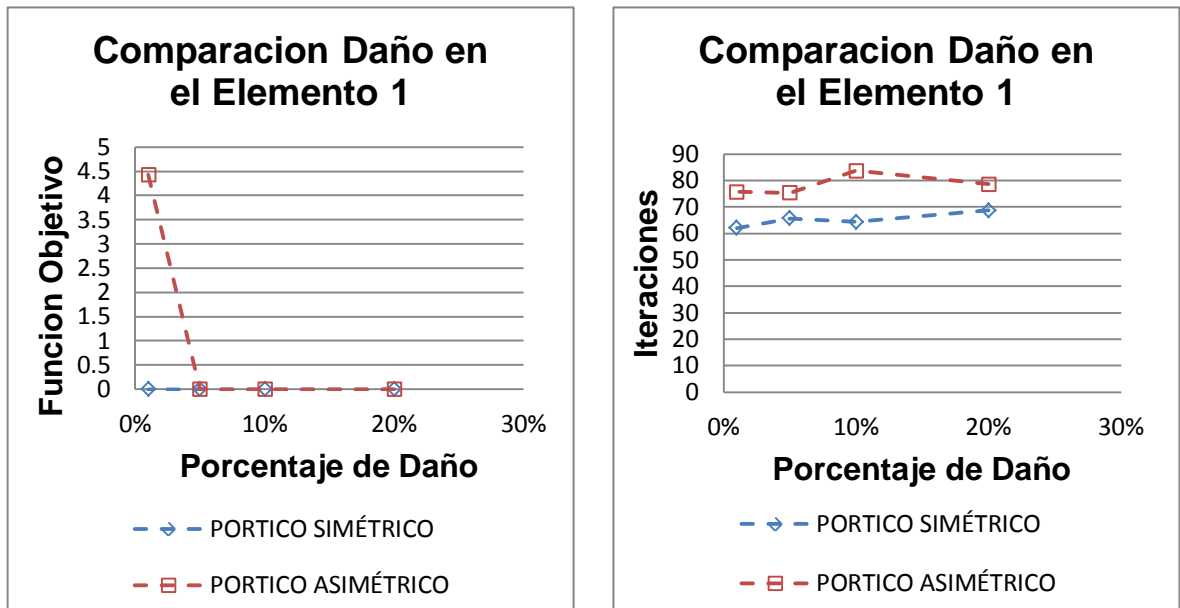


Figura 4. Comparación daño único elemento 1 – PSO.

En la figura 5 se observa que la detección de daño en el elemento 20 requirió un menor número de iteraciones en el pórtico simétrico en la mayoría de los casos y los valores de la función objetivo fueron consistentes y cercanos a cero para los dos pórticos.

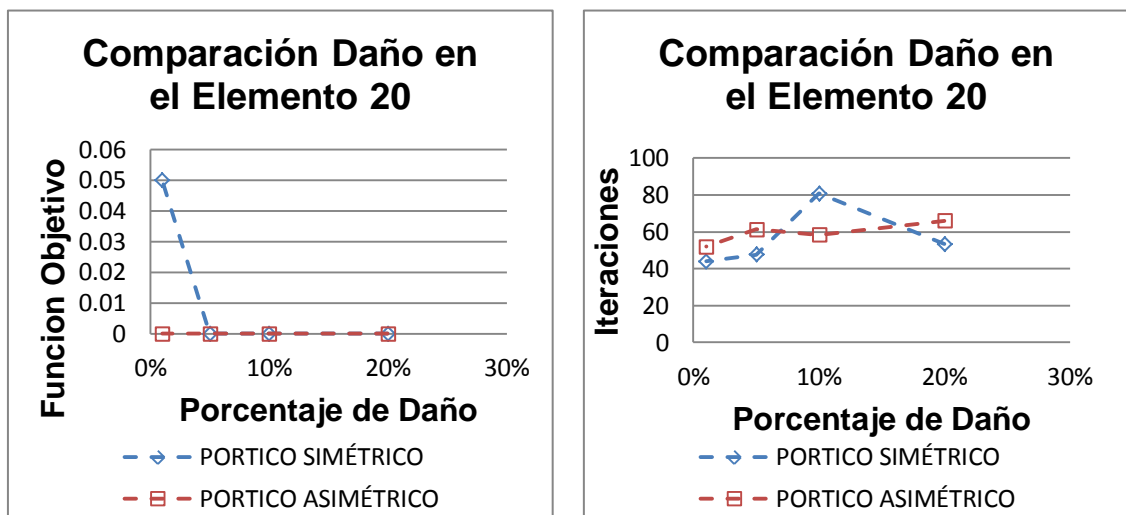


Figura 5. Comparación daño único elemento 20 – PSO.

En la figura 6 se observa una convergencia con un menor número de iteraciones y valores más cercanos a cero de la función objetivo para los diferentes porcentajes de daño en el pórtico simétrico.

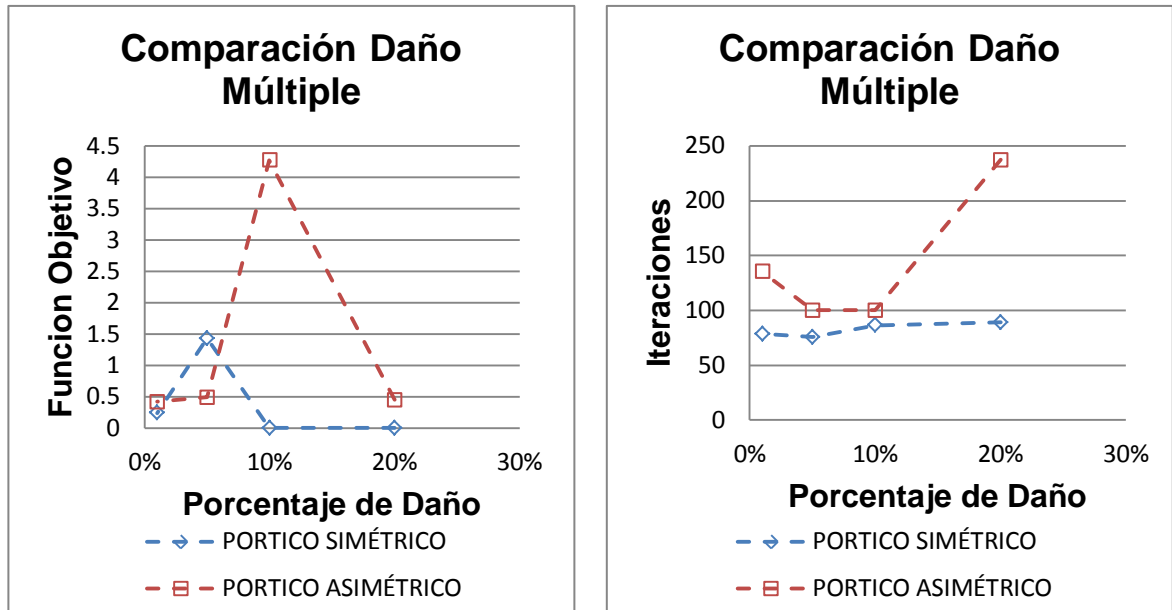


Figura 6. Comparación daño múltiple – PSO.

Los resultados obtenidos con el pórtico simétrico muestran que se requieren menos iteraciones para detectar el daño, por esta razón se decidió usar dicho pórtico para realizar las pruebas con ruido.

	Elem	% Daño	Media			Desviación Estándar		
			Fun Objetivo	Iteraciones	Tiempo [min]	Fun Objetivo	Iteraciones	Tiempo [min]
			PORTICO SIMÉTRICO CON RUIDO					
1	1	20%	2.57	149	46.93	0.03	31.87	13.22
		5%	3.02	100	36.99	0.98	0	0.18
20	20	20%	2.23	98.67	38.62	0.04	1.89	2.73
		5%	2.41	128.3	47.07	0.14	15.58	6.87
9-10-18	18	20%	2.41	130.3	38.07	0.02	33.77	9.84
		5%	3.07	173	59.66	0.92	34.03	11.82

Tabla 25. Media y Desviación Estándar – Pórtico simétrico con ruido – PSO.

Con el fin de poder comparar los valores de las funciones objetivo de los dos algoritmos (PSO y AG) se decide despejar la ecuación 4.8 para obtener el mismo valor de función objetivo sacado con la ecuación 4.7. La tabla 26 fue creada con estos valores modificados.

	Elem	% Daño	Media			Desviación Estándar		
			Fun Objetivo	Iteraciones	Tiempo [min]	Fun Objetivo	Iteraciones	Tiempo [min]
PORTICO SIMÉTRICO	1	20%	6	100	63.23	0.31	0	0.07
		20%	1.43	5000	3106.4	0.39	0	102.96
	20	20%	4.16	100	63.11	0.64	0	0.48
	9-10-18	20%	6.69	100	63.27	0.47	0	0.09
PORTICO ASIMÉTRICO	1	20%	9.43	100	60.56	2.3	0	0.88
	20	20%	1.71	100	61.05	0.02	0	2.19
	9-13-18	20%	11.59	500	311.97	0.63	0	15.88
PORTICO SIMÉTRICO CON RUIDO	1	20%	5.9	100	64.84	0.82	0	1.46
	20	20%	3.99	100	62.61	0.28	0	0.1
	9-10-18	20%	7	100	69.68	0.36	0	2.9

Tabla 26. Media y Desviación Estándar – AG.

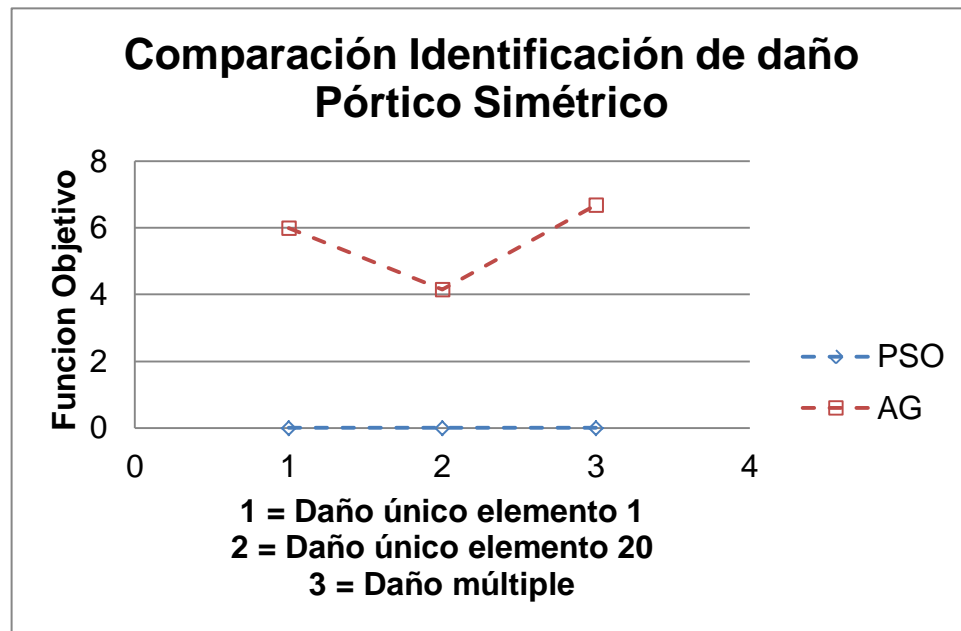


Figura 7. Comparación Valores Función Objetivo – Pórtico Simétrico sin ruido – AG y PSO.

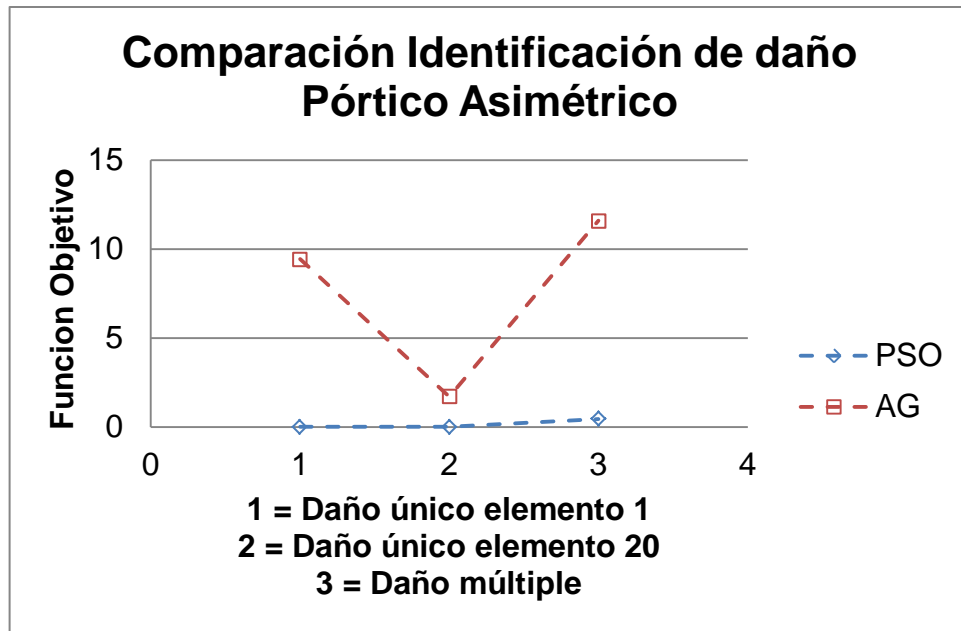


Figura 8. Comparación Valores Función Objetivo – Pórtico Asimétrico – AG y PSO.

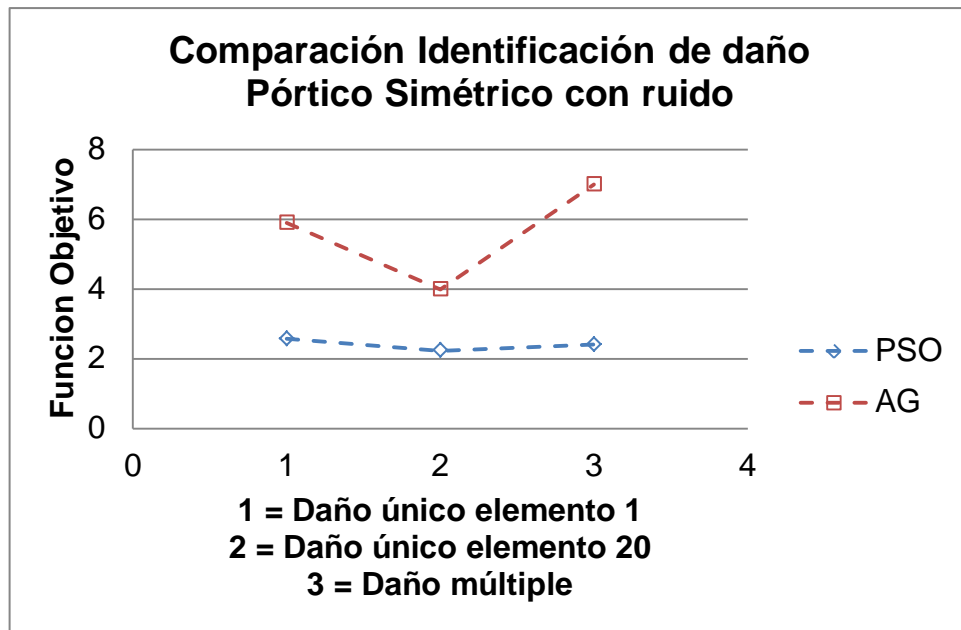


Figura 9. Comparación Valores Función Objetivo – Pórtico Simétrico con ruido – AG y PSO.

En las figuras 7,8 y 9 se puede observar la diferencia entre los valores de la función objetivo calculado con cada uno de los algoritmos usando un mismo número de iteraciones máximas para cada caso. Esto demuestra que para este problema en particular y con igualdad de condiciones el desempeño del PSO básico (shi y Eberhart) es superior al AG básico (Holland 1975).

En las figuras 10, 11 y 12 se compararon las medias de los tiempos que tomaron los algoritmos en detenerse. Se puede observar que el AG tarda más en detenerse que el PSO y tiene resultados no exactos.

Se puede ver que el PSO siempre alcanza el valor óptimo de la función objetivo (ver figura7 y 8) y toma menos tiempo en el proceso (figura 10 y 11) para los dos tipos de pórticos en todos los casos de daño, incluyendo aquellos casos donde se aplicó ruido (ver figura 12).

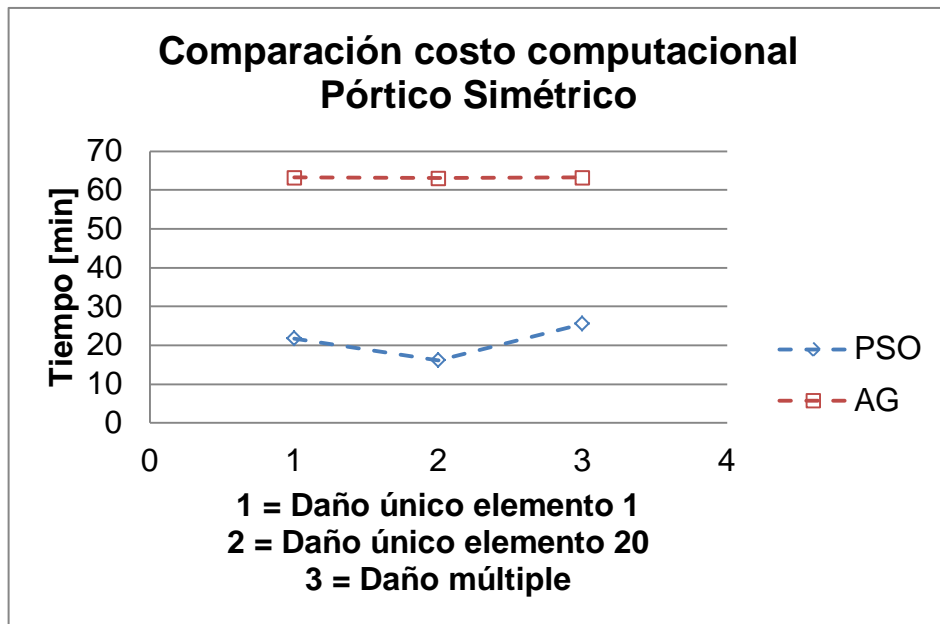


Figura 10. Comparación de tiempos promedio de computo – Pórtico Simétrico sin ruido – AG y PSO.

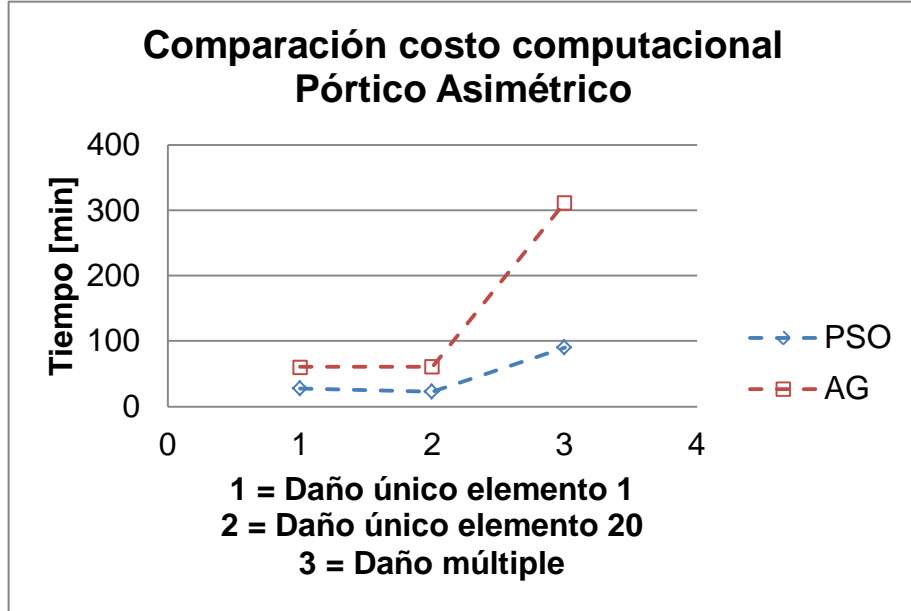


Figura 11. Comparación de tiempos promedio de computo – Pórtico Asimétrico – AG y PSO.

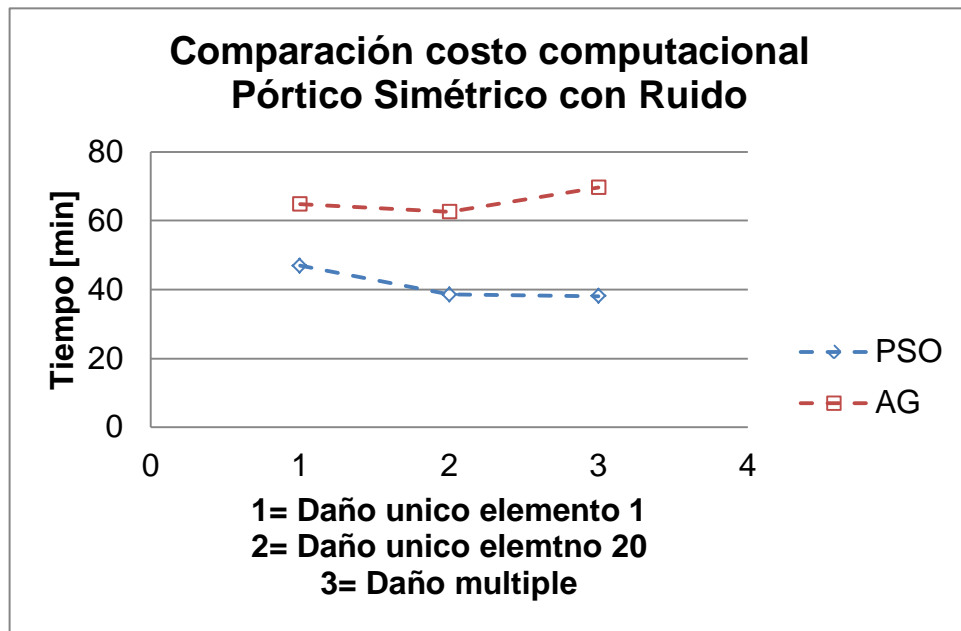


Figura 12. Comparación de tiempos promedio de computo – Pórtico Simétrico con ruido – AG y PSO.

6. CONCLUSIONES

Se programaron pórticos 3D en OPENSEES, con el fin de obtener su información modal. Los resultados modales obtenidos fueron validados mediante su comparación con los calculados mediante el programa SAP. La diferencia entre ellos fue de hasta 0.92%. (Ver anexo 4)

Se realizó un acople entre OPENSEES y MATLAB; para la utilización del OPENSEES como solucionador modal y los algoritmos PSO y AG como optimizadores (programados en MATLAB ver anexos 6 y 7). Esta herramienta permite la obtención de frecuencias y modos teóricos dentro de los códigos de los algoritmos de forma automática, sin intervención del usuario.

Se programó una versión básica del PSO propuesta por Kennedy y Eberhart (1995) y Shi y Ebarhard (2001) en MATLAB, la cual fue validada con cuatro funciones matemáticas reportadas en la literatura internacional determinando en todos los casos los puntos óptimos de las funciones empleadas. Estos resultados están en el anexo 8.

Se programó una versión del AG de holland (1975) en MATLAB, la cual fue validada con cuatro funciones matemáticas reportadas en la literatura internacional determinando en todos los casos los puntos óptimos de las funciones empleadas. Estos resultados se encuentran en el anexo 8.

Los resultados obtenidos con el PSO en este trabajo (Shi y Eberhart (2001)) muestran que el algoritmo tiene alta capacidad para la solución del problema de detección de daño para pórticos en 3D, basándose en la información modal, donde se emplearon las cuatro primeras frecuencias y sus modos asociados. Tanto en los casos de daño simple como de daño múltiple (sin ruido) el algoritmo puede detectar una simulación de daño de hasta un 1%. Para casos donde los datos de entrada simulan ruido detecta hasta un 5%. Ver tablas 14 y 15.

Las pruebas realizadas para el algoritmo AG muestran que generan un mayor costo computacional y por ende un mayor tiempo de trabajo. Esto se debe a su baja velocidad de convergencia por lo que dicho algoritmo realiza una mayor cantidad de iteraciones. En comparación, el PSO es más eficiente y genera resultados más exactos. Las versiones a comparar de los dos algoritmos son básicas, desarrolladas en el mismo ambiente de programación y la realización de sus pruebas fueron hechas en el mismo computador. En ambos casos se utilizaron las mismas funciones de la literatura internacional para su validación y tienen los mismos criterios de parada. Vale la pena mencionar que cuando se dejó al AG completar 5000 iteraciones su desempeño mejor considerablemente (ver tabla 22).

OBJETIVO	CAPITULOS
Modelar con elementos finitos (o análisis matricial) un pórtico en 3 dimensiones para realizar su análisis modal numérico.	Capítulo 4, secciones 4.3 y Anexos 1 y 4.
Emplear los algoritmos PSO y AG para identificar el daño en la estructura.	Capítulo 4, secciones 4.4.1 y 4.4.2 y anexos 5 y 8
Comparar el desempeño, desde el punto de vista estadístico, de los procedimientos empleados.	Capítulo 5, numeral 5.4

Tabla 27. Evidencia al cumplimiento de objetivos específicos.

7. RECOMENDACIONES

Emplear un AG con operador de mutación modificado para la detección de daño en pórticos 3D.

Futuros alumnos: profundización en la creación de modelos estructurales 3D teniendo en cuenta nudos flexibles.

Emplear OPENSEES (Open System for Earthquake Engineering Simulation) para la realización del cálculo de frecuencias y modos de vibración a cambio de hacer este proceso en MATLAB debido a algunas ventajas observadas en el proceso realizado en este trabajo. Como se mostró en el capítulo 4, sección 4.1 el cálculo de las frecuencias y modos de vibración dependen de la masa y rigidez del sistema lo cual hace que la programación en MATLAB para el cálculo de estas frecuencias y modos (cálculo de matrices de rigidez y masa) sea muy extensa y sea muy probable cometer errores, mientras que OPENSEES es un programa propicio para la realización de este proceso.

8. BIBLIOGRAFIA

ATIENZA PASCUAL, Ricardo. Técnicas de Actualización Aplicadas a la Detección de Daños. Tesis Doctoral. Madrid: Universidad Politécnica de Madrid. Escuela Técnica Superior de Ingenieros Aeronáuticos, 2004. 172p.

CARDEN, Peter E. and FANNING, Paul. Vibration Based Condition Monitoring: A Review. In: Sage Publications. December, 2004, vol. 3, no. 4, p. 355-377

CELIZ MELO, Camilo Eduardo y MILLAN PARAMO, Carlos Andrés. Evaluación del Desempeño de Algoritmos de Evolución Paramétrica para optimización. Trabajo de Grado. Bucaramanga: Universidad Industrial de Santander. Escuela de Ingeniería Civil, 2009, 200p.

Chaekuk Na, Sun-Pilkim y Hyo- Gyoung Kwak. Structural damage evaluation using genetic algorithm. Republic of Korea, Department of Civil and Environmental Engineering, Korea Advanced Institute of Science and Technology, 2009, p. 701.

CHELLINI, Giuseppe, DE ROECK, Guido, NARDINI, Luca, SALVATORE, Walter, Damage analysis of a steel_concrete composite frame by finite element model updating, Pisa, Italia: Universidad de Pisa, 2010.

EBERHART, R.C. y SHI, "Particle swarm optimization: developments, applications and resources", *Proceedings of the 2001 Congress on Evolutionary Computation*, Seoul (South Korea), Mayo 2001.

HOLLAND, J.H, "Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence", University of Michigan Press, 1975 – 183p.

KENNEDY, James and EBERHART Russell. Particle Swarm Optimization. Proceedings of the IEEE International Conference on Neural (December 1995: Perth, Australia), vol.4, pp. 1942-1948.

LI, Hong-Nan, LI, Dong-Sheng and SONG, Gang-Bing. Recent Applications of Fiber Optic Sensors to Health Monitoring in Civil Engineering. In: Engineering Structures. September, 2004, vol. 26, no. 11, p. 1647-1657.

MARULANDA A., Johannio, THOMSON, Peter. Monitoreo de salud estructural. En: Ingeniería y competitividad. Diciembre 2000, vol. 2, no. 2, p 40-46.

NAGY, E. ET.AL. (2003). 3D Analysis of Tomographic Images. 16th ASCE Engineering Mechanics Conference, July 16-18, University of Washington, Seattle.

RILEM, bookseries. Ensayo no destructivos en materiales y estructuras. Vol. 6, 2013.

RIVEROS, Carlos. Structural Health Monitoring Methodology for Simply Supported Bridges: Numerical Implementation. En: Revista facultad de ingeniería (Universidad de Antioquia). Marzo, 2007, vol. 39, p.42-55.

SEWERYNKOKOT, ZBIGNIEWZEMBATY. Damage reconstruction of 3D frames using genetic algorithms with Levenberg – Marquardt local search, Opole, Polonia: Universidad de tecnologia Opole, 2009.

SOTELO MONROY, Freddy Saul y ACEVEDO GUTIERREZ, Javier Ernesto. Identificación de daño en pórticos 2D empleando información modal y el método particle swarm optimization (PSO). Trabajo de grado. Bucaramanga: Universidad Industrial de Santander. Escuela de ingeniería civil, 2009, 163p.

TOLMOS RODRÍGUEZ, Piedad. Introducción a los algoritmos genéticos y sus aplicaciones. Madrid, 2005, p. 9.

VENTER, Gerhard and SOBIESZCZANSKI - SOBIESKI, Jaroslaw. Particle Swarm Optimization. In: Journal AIAA. August, 2003, vol. 41, no. 8, p. 1583-1589.

BEGAMBRE, Oscar J. Detecção de Dano a Partir da Resposta Dinâmica da Estrutura: Estudo Analítico com Aplicação a Estruturas do Tipo Viga. Tesis para a

obtenção do Título de Mestre em Engenharia de Estruturas. São Paulo: Universidade de São Paulo. Escola de Engenharia de São Carlos, 2004. 112p.

BEER, Ferdinand, JOHNSTON Russell y DEWLOF John. Mecánica de Materiales. 3 ed. México D.F.: McGraw-Hill Interamericana, 2004. 790p.

ESCAMILLA, Jairo Uribe. Análisis de Estructuras. 3 ed. Bogotá D. C.: Escuela Colombiana de Ingeniería, 2002. 790p.

HERNANDEZ, Miguel. Detección de Daño a Partir de Cambios en las Características Vibracionales Usando Redes Neuronales. Universidad de los Andes, Junio, 2005, p. 10.

MALDONADO RONDON, Esperanza y CHIO CHO, Gustavo. Análisis Sísmico de Edificaciones. Bucaramanga: Universidad Industrial de Santander, 2004. 297p.

RODRÍGUEZ, Ramsés y ESCOBAR, José. Detección de daño estructural en edificios sin parámetros modales base. August 20, 2011. 84p.

ANEXO A. Código de análisis modal de la estructura en lenguaje Tcl.

En este anexo están los códigos utilizados por OPENSEES para realizar el análisis modal.

Código de análisis modal de la estructura en lenguaje Tcl para pórtico simétrico.

```
1  # UNIDADES: Kg, m, sec, N
2  # ESCRITO: fssm
3  # FECHA: Enero 2013 Portico3D.tcl
4
5  wipe; #Limpiar memoria
6  model BasicBuilder -ndm 3 -ndf 6;
7  #Definicion del modelo 3D y 6GDL por nodo
8  set dataDir Portico3Def;
9  #Asignacion del nombre de Datos1 al directorio datadir
10 file mkdir $dataDir;
11 #Definicion del archivo de salida Datos1
12
13 #-----
14 set numModes 50; #Numero de modos a calcular
15 #-----
16
17 #-----
18 #COORDENADAS, RESTRICCIONES
19 #-----
20
21 set LCol 1; #Longitud Columnas paralelas al eje Y
22 set LBeam 2; #Longitud Vigas paralelas al eje X
23 set LGird 2; #Longitud Vigas paralelas al eje Z
24 set dmas 2400;
25 #Densidad de la masa Concreto Reforzado [Kg/m3]
26
27 #Definicion de coordenadas nodales
28 set X0 0.;
29 set X1 [expr $X0 + $LBeam];
30 set X2 [expr $X1 + $LBeam];
31 set X3 [expr $X2 + $LBeam];
32 set Y0 0.;
33 set Y1 [expr $Y0 + $LCol];
34 set Y2 [expr $Y1 + $LCol];
35 set Y3 [expr $Y2 + $LCol];
36 set Z0 0.;
37 set Z1 [expr $Z0 + $LGird];
```

```

34     set Z2 [expr $Z1 + $LGird];
35     set Z3 [expr $Z2 + $LGird];
36
37     #NODOS
38     # Portico 1
39     node 1 $X0 $Y0 $Z0
40     node 2 $X0 $Y1 $Z0
41     node 3 $X0 $Y2 $Z0
42     node 4 $X0 $Y3 $Z0
43     node 5 $X1 $Y3 $Z0
44     node 6 $X2 $Y3 $Z0
45     node 7 $X3 $Y3 $Z0
46     node 8 $X3 $Y2 $Z0
47     node 9 $X3 $Y1 $Z0
48     node 10 $X3 $Y0 $Z0
49
50     #Nodos vigas eje Z
51     node 11 $X0 $Y3 $Z1
52     node 12 $X0 $Y3 $Z2
53
54     node 13 $X3 $Y3 $Z1
55     node 14 $X3 $Y3 $Z2
56
57     # Poritco 2
58     node 15 $X0 $Y0 $Z3
59     node 16 $X0 $Y1 $Z3
60     node 17 $X0 $Y2 $Z3
61     node 18 $X0 $Y3 $Z3
62     node 19 $X1 $Y3 $Z3
63     node 20 $X2 $Y3 $Z3
64     node 21 $X3 $Y3 $Z3
65     node 22 $X3 $Y2 $Z3
66     node 23 $X3 $Y1 $Z3
67     node 24 $X3 $Y0 $Z3
68
69     # RESTRICCIONES
70     fix 1 1 1 1 1 1 1
71     fix 10 1 1 1 1 1 1
72     fix 15 1 1 1 1 1 1
73     fix 24 1 1 1 1 1 1
74
75     puts "NUDOS CREADOS Y RESTRINGIDOS!!"
76
77     #-----
78     #DEFINICION DE CARACTERITICAS DE LOS ELEMENTOS
79     #-----
80

```

```

81 # PROPIEDADES COLUMNAS
82 set A1 0.16;; #Area de la seccion transversal
83 set J1 0.003605; #Constante torsional
84 set Iz1 0.002133; #Momento de inercia en eje z
85 set Iy1 0.002133; #Momento de inercia en eje y
86
87 set E1 21.54e9; #Modulo de elasticidad
88 set G1 [expr $E1/(2./(1+0.2))]; #Modulo de rigidez
89 set M1 [expr $dmas*$A1];
#Masa por unidad de longitud
90
91
92 # PROPIEDADES VIGAS
93 set A2 0.16; #Area de la seccion transversal
94 set J2 0.003604; #Constante torsional
95 set Iz2 0.002133; #Momento de inercia en eje z
96 set Iy2 0.002133; #Momento de inercia en eje y
97
98 set E2 21.54e9; #Modulo de elasticidad
99 set G2 [expr $E2/(2./(1+0.2))]; #Modulo de rigidez
100 set M2 [expr $dmas*$A2];
#Masa por unidad de longitud
101
102
103 puts "SECCIONES CREADAS!!"
104 puts "masa1 $M1 masa2 $M2"
105
106 # TRANSFORMACION GEOMETRICA
107 set Columnas 1; #Columnas
108 set Vigasx 2; #Vigas eje x
109 set Vigasz 3; #Vigas eje z
110 geomTransf Linear $Columnas 0 0 1
111 geomTransf Linear $Vigasx 0 0 1
112 geomTransf Linear $Vigasx -1 0 0
113
114 # CREACION DE ELEMENTOS (TIPO VIGA-COLUMNA)
comando: element elasticBeamColumn
$seleTag $iNode $jNode $A $E $G $J $Iy $Iz $tra nsfTag
<-mass $massDens>
115 element elasticBeamColumn 1 1 2 $A1 $E1 $G1 $J1 $Iy1
$Iz1 $Columnas -mass $M1; #Columna - Elemento 1
116 element elasticBeamColumn 2 2 3 $A1 $E1 $G1 $J1 $Iy1
$Iz1 $Columnas -mass $M1; #Columna - Elemento 2
117 element elasticBeamColumn 3 3 4 $A1 $E1 $G1 $J1 $Iy1
$Iz1 $Columnas -mass $M1; #Columna - Elemento 3
118 element elasticBeamColumn 4 4 5 $A2 $E2 $G2 $J2 $Iy2
$Iz2 $Vigasx -mass $M2; #Viga x - Elemento 4

```

```

119 element elasticBeamColumn 5 5 6 $A2 $E2 $G2 $J2 $Iy2
    $Iz2 $Vigasx -mass $M2; #Viga x - Elemento 5
120 element elasticBeamColumn 6 6 7 $A2 $E2 $G2 $J2 $Iy2
    $Iz2 $Vigasx -mass $M2; #Viga x - Elemento 6
121 element elasticBeamColumn 7 8 7 $A1 $E1 $G1 $J1 $Iy1
    $Iz1 $Columnas -mass $M1; #Columna - Elemento 7
122 element elasticBeamColumn 8 9 8 $A1 $E1 $G1 $J1 $Iy1
    $Iz1 $Columnas -mass $M1; #Columna - Elemento /
123 element elasticBeamColumn 9 10 9 $A1 $E1 $G1 $J1 $Iy1
    $Iz1 $Columnas -mass $M1; #Columna - Elemento 9
124
125 element elasticBeamColumn 10 4 11 $A2 $E2 $G2 $J2 $Iy2
    $Iz2 $Vigasx -mass $M2; #Viga z - Elemento 10
126 element elasticBeamColumn 11 11 12 $A2 $E2 $G2 $J2 $Iy2
    $Iz2 $Vigasx -mass $M2; #Viga z - Elemento 11
127 element elasticBeamColumn 12 12 16 $A2 $E2 $G2 $J2 $Iy2
    $Iz2 $Vigasx -mass $M2; #Viga z - Elemento 12
128
129 element elasticBeamColumn 13 7 13 $A2 $E2 $G2 $J2 $Iy2
    $Iz2 $Vigasx -mass $M2; #Viga z - Elemento 13
130 element elasticBeamColumn 14 13 14 $A2 $E2 $G2 $J2 $Iy2
    $Iz2 $Vigasx -mass $M2; #Viga z - Elemento 14
131 element elasticBeamColumn 15 14 21 $A2 $E2 $G2 $J2 $Iy2
    $Iz2 $Vigasx -mass $M2; #Viga z - Elemento 15
132
133 element elasticBeamColumn 16 15 16 $A1 $E1 $G1 $J1 $Iy1
    $Iz1 $Columnas -mass $M1; #Columna - Elemento 16
134 element elasticBeamColumn 17 16 17 $A1 $E1 $G1 $J1 $Iy1
    $Iz1 $Columnas -mass $M1; #Columna - Elemento 17
135 element elasticBeamColumn 18 17 18 $A1 $E1 $G1 $J1 $Iy1
    $Iz1 $Columnas -mass $M1; #Columna - Elemento 18
136 element elasticBeamColumn 19 18 19 $A2 $E2 $G2 $J2 $Iy2
    $Iz2 $Vigasx -mass $M2; #Viga x - Elemento 19
137 element elasticBeamColumn 20 19 20 $A2 $E2 $G2 $J2 $Iy2
    $Iz2 $Vigasx -mass $M2; #Viga x - Elemento 20
138 element elasticBeamColumn 21 20 21 $A2 $E2 $G2 $J2 $Iy2
    $Iz2 $Vigasx -mass $M2; #Viga x - Elemento 21
139 element elasticBeamColumn 22 22 21 $A1 $E1 $G1 $J1 $Iy1
    $Iz1 $Columnas -mass $M1; #Columna - Elemento 22
140 element elasticBeamColumn 23 23 22 $A1 $E1 $G1 $J1 $Iy1
    $Iz1 $Columnas -mass $M1; #Columna - Elemento 23
141 element elasticBeamColumn 24 24 23 $A1 $E1 $G1 $J1 $Iy1
    $Iz1 $Columnas -mass $M1; #Columna - Elemento 24
142
143 puts "ELEMENTOS CREADOS!!"
144

```

```

145 recorder Node -file Modos1.txt -node 4 7 18 21
    -dof 1 2 3 4 5 6 "eigen 1"
146 recorder Node -file Modos2.txt -node 4 7 18 21
    -dof 1 2 3 4 5 6 "eigen 2"
147 recorder Node -file Modos3.txt -node 4 7 18 21
    -dof 1 2 3 4 5 6 "eigen 3"
148 recorder Node -file Modos4.txt -node 4 7 18 21
    -dof 1 2 3 4 5 6 "eigen 4"
149
150 #-----
151 #Realizacion de eigen-analisis
152 #-----
153 set lambda [eigen $numModes];
154
155 #-----
156 #Calculo de frecuencias y periodos de la estructura
157 #-----
158 set omega {}
159 set f {}
160 set T {}
161 set pi 3.141593
162
163 foreach lam $lambda {
164     lappend omega [expr sqrt($lam)]
165     lappend f [expr sqrt($lam)/(2*$pi)]
166     lappend T [expr (2*$pi)/sqrt($lam)]
167 }
168
169 puts "Las frecuencias son $f (Hz)"
170 puts "Los periodos son $T (seg)"
171
172 #-----
173 #Grabacion de los periodos
174 #-----
175 set periodo "Portico3Def/Periodos.txt"
176 set Periodos [open $periodo "w"]
177 foreach t $T {
178     puts $Periodos " $t"
179 }
180 close $Periodos
181

```

```

182  #-----
183  #Grabacion de las frecuencias
184  #-----
185  set frec "Portico3Def/Frecuencias.txt"
186  set Frecuencias [open $frec "w"]
187  foreach t $f {
188      puts $Frecuencias " $t"
189  }
190  close $Frecuencias
191
192  record

```

Código de análisis modal de la estructura en lenguaje Tcl para pórtico asimétrico.

```

1  # UNIDADES: Kg, m, sec, N
2  # FECHA: Febrero 2013 Portico.tcl
3
4  wipe; #Limpiar memoria
5  model BasicBuilder -ndm 3 -ndf 6;
6  #Definicion del modelo 3D y 6GDL por nodo
7  set dataDir Portico;
8  #Asignacion del nombre de Datos1 al directorio datadir
9  file mkdir $dataDir;
10 #Definicion del archivo de salida Datos1
11
12
13 #-----
14 #COORDENADAS, RESTRICCIONES
15 #-----
16
17 set LCol1 1;
18 #Longitud Columnas paralelas al eje Y
19 #ubicadas en eje X=0
20 set LCol2 1.5;
21 #Longitud Columnas paralelas al eje Y
22 #ubicadas en eje X=4.5
23 set LBeam 1.5;
24 #Longitud Vigas paralelas al eje X
25 set LGird 1;
26 #Longitud Vigas paralelas al eje Z
27 set dmas 2356.3122;

```

```

22      #Densidad de la masa Concreto Reforzado [Kg/m3]
23      #Definicion de coordenadas nodales
24      set X0 0.;
25      set X1 [expr $X0 + $LBeam];
26      set X2 [expr $X1 + $LBeam];
27      set X3 [expr $X2 + $LBeam];
28      set Y0 0.;
29      set Y1 [expr $Y0 + $LCol1];
30      set Y3 [expr $Y1 + $LCol1];
31      set Y5 [expr $Y3 + $LCol1];
32      set Y2 [expr $Y1 + $LCol2];
33      set Y4 [expr $Y2 + $LCol2];
34      set Z0 0.;
35      set Z1 [expr $Z0 + $LGird];
36      set Z2 [expr $Z1 + $LGird];
37      set Z3 [expr $Z2 + $LGird];
38
39      #NODOS
40
41      #Portico X=0
42
43      node 1 $X0 $Y1 $Z0
44      node 2 $X0 $Y2 $Z0
45      node 3 $X0 $Y4 $Z0
46      node 4 $X0 $Y5 $Z0
47      node 5 $X0 $Y5 $Z1
48      node 6 $X0 $Y5 $Z2
49      node 7 $X0 $Y5 $Z3
50      node 8 $X0 $Y4 $Z3
51      node 9 $X0 $Y2 $Z3
52      node 10 $X0 $Y1 $Z3
53
54      #Vigas paralelas a eje X
55
56      node 11 $X1 $Y5 $Z0
57      node 12 $X2 $Y5 $Z0
58
59      node 13 $X1 $Y5 $Z3
60      node 14 $X2 $Y5 $Z3
61
62      #Portico X=4.5
63
64      node 15 $X3 $Y0 $Z0
65      node 16 $X3 $Y1 $Z0
66      node 17 $X3 $Y3 $Z0
67      node 18 $X3 $Y5 $Z0

```

```

68 node 19 $X3 $Y5 $Z1
69 node 20 $X3 $Y5 $Z2
70 node 21 $X3 $Y5 $Z3
71 node 22 $X3 $Y3 $Z3
72 node 23 $X3 $Y1 $Z3
73 node 24 $X3 $Y0 $Z3
74
75 # RESTRICCIONES
76 fix 1 1 1 1 1 1
77 fix 10 1 1 1 1 1
78 fix 15 1 1 1 1 1
79 fix 24 1 1 1 1 1
80
81 puts "NUDOS CREADOS Y RESTRINGIDOS!!"
82
83 #-----
84 #DEFINICION DE CARACTERITICAS DE LOS ELEMENTOS
85 #-----
86
87 set E1 21.54e9; #Modulo de elasticidad
88 set G1 [expr $E1/(2./(1+0.2))]; #Modulo de rigidez
89
90 # PROPIEDADES COLUMNAS - ubicadas en X=0,Z=0 (20x20)
91 set A1 0.04; #Area de la seccion transversal
92
93 set J1 0.0002253; #Constante torsional
94 set Iz1 0.0001333; #Momento de inercia en eje z
95 set Iy1 0.0001333; #Momento de inercia en eje y
96
97 set M1 [expr $dmas*$A1];
98 #Masa por unidad de longitud
99
100 # PROPIEDADES COLUMNAS - ubicadas en X=0,Z=3 (20x20)
101 set A5 0.04; #Area de la seccion transversal
102 set J5 0.0002253; #Constante torsional
103 set Iz5 0.0001333; #Momento de inercia en eje z
104 set Iy5 0.0001333; #Momento de inercia en eje y
105
106 set M5 [expr $dmas*$A5];
107 #Masa por unidad de longitud
108
109 # PROPIEDADES COLUMNAS - ubicadas en X=4.5,Z=0 (20x20)
110 set A2 0.04; #Area de la seccion transversal
111 set J2 0.0002253; #Constante torsional
112 set Iz2 0.0001333; #Momento de inercia en eje z
113 set Iy2 0.0001333; #Momento de inercia en eje y

```

```

112
113 set M2 [expr $dmas*$A2];
      #Masa por unidad de longitud
114
115 # PROPIEDADES COLUMNAS - ubicadas en X=4.5,Z=3 (20x20)
116 set A6 0.04; #Area de la seccion transversal
117 set J6 0.0002253; #Constante torsional
118 set Iz6 0.0001333; #Momento de inercia en eje z
119 set Iy6 0.0001333; #Momento de inercia en eje y
120
121 set M6 [expr $dmas*$A2];
      #Masa por unidad de longitud
122
123
124 # PROPIEDADES VIGAS - paralelas eje x
125 set A3 0.06; #Area de la seccion transversal
126 set J3 0.0004695; #Constante torsional
127 set Iz3 0.00045; #Momento de inercia en eje z
128 set Iy3 0.0002; #Momento de inercia en eje y
129
130 set M3 [expr $dmas*$A3];
      #Masa por unidad de longitud
131
132 # PROPIEDADES VIGAS - paralelas eje z
133 set A4 0.04; #Area de la seccion transversal
134 set J4 0.0002253; #Constante torsional
135 set Iz4 0.0001333; #Momento de inercia en eje z
136 set Iy4 0.0001333; #Momento de inercia en eje y
137
138 set M4 [expr $dmas*$A3];
      #Masa por unidad de longitud
139
140 puts "SECCIONES CREADAS!!"
141
142 # TRANSFORMACION GEOMETRICA
143 set Columnas 1; #Columnas
144 set Vigasx 2; #Vigas eje x
145 set Vigasz 3; #Vigas eje z
146 geomTransf Linear $Columnas 0 0 1
147 geomTransf Linear $Vigasx 0 0 1
148 geomTransf Linear $Vigasx -1 0 0
149
150 # CREACION DE ELEMENTOS (TIPO VIGA-COLUMNA)
      comando: element elasticBeamColumn $eleTag
      $iNode $jNode $A $E $G $J $Iy $Iz $tra
      nsfTag <-mass $massDens>

```

```

151
152 element elasticBeamColumn 1 1 2 $A1 $E1 $G1 $J1 $Iy1
    $Iz1 $Columns -mass $M1; #Columna X0 - Elemento 1
153 element elasticBeamColumn 2 2 3 $A1 $E1 $G1 $J1 $Iy1
    $Iz1 $Columns -mass $M1; #Columna X0 - Elemento 2
154 element elasticBeamColumn 3 3 4 $A1 $E1 $G1 $J1 $Iy1
    $Iz1 $Columns -mass $M1; #Columna X0 - Elemento 3
155 element elasticBeamColumn 4 4 5 $A4 $E1 $G1 $J4 $Iy4
    $Iz4 $Vigasz -mass $M4;
    #Viga paralela eje z - Elemento 4
156 element elasticBeamColumn 5 5 6 $A4 $E1 $G1 $J4 $Iy4
    $Iz4 $Vigasz -mass $M4;
    #Viga paralela eje z - Elemento 5
157 element elasticBeamColumn 6 6 7 $A4 $E1 $G1 $J4 $Iy4
    $Iz4 $Vigasz -mass $M4;
    #Viga paralela eje z - Elemento 6
158 element elasticBeamColumn 7 8 7 $A5 $E1 $G1 $J5 $Iy5
    $Iz5 $Columns -mass $M5;
    #Columna X0 - Elemento 7
159 element elasticBeamColumn 8 9 8 $A5 $E1 $G1 $J5 $Iy5
    $Iz5 $Columns -mass $M5; #Columna X0 - Elemento 8
160 element elasticBeamColumn 9 10 9 $A5 $E1 $G1 $J5 $Iy5
    $Iz5 $Columns -mass $M5; #Columna X0 - Elemento 9
161
162 element elasticBeamColumn 10 4 11 $A3 $E1 $G1 $J3 $Iy3
    $Iz3 $Vigasx -mass $M3;
    #Viga paralela eje x - Elemento 10
163 element elasticBeamColumn 11 11 12 $A3 $E1 $G1 $J3 $Iy3
    $Iz3 $Vigasx -mass $M3;
    #Viga paralela eje x - Elemento 11
164 element elasticBeamColumn 12 12 18 $A3 $E1 $G1 $J3 $Iy3
    $Iz3 $Vigasx -mass $M3;
    #Viga paralela eje x - Elemento 12
165
166 element elasticBeamColumn 13 7 13 $A3 $E1 $G1 $J3 $Iy3
    $Iz3 $Vigasx -mass $M3;
    #Viga paralela eje x - Elemento 13
167 element elasticBeamColumn 14 13 14 $A3 $E1 $G1 $J3 $Iy3
    $Iz3 $Vigasx -mass $M3;
    #Viga paralela eje x - Elemento 14
168 element elasticBeamColumn 15 14 21 $A3 $E1 $G1 $J3 $Iy3
    $Iz3 $Vigasx -mass $M3;
    #Viga paralela eje x - Elemento 15
169
170 element elasticBeamColumn 16 15 16 $A2 $E1 $G1 $J2 $Iy2
    $Iz2 $Columns -mass $M2; #Columna X3 - Elemento 16
171 element elasticBeamColumn 17 16 17 $A2 $E1 $G1 $J2 $Iy2

```

```

172   $Iz2 $Columnas -mass $M2; #Columna X3 - Elemento 17
     element elasticBeamColumn 18 17 18 $A2 $E1 $G1 $J2 $Iy2
173   $Iz2 $Columnas -mass $M2; #Columna X3 - Elemento 18
     element elasticBeamColumn 19 18 19 $A4 $E1 $G1 $J4 $Iy4
     $Iz4 $Vigasz -mass $M4;
     #Viga paralela eje z - Elemento 19
174   element elasticBeamColumn 20 19 20 $A4 $E1 $G1 $J4 $Iy4
     $Iz4 $Vigasz -mass $M4;
     #Viga paralela eje z - Elemento 20
175   element elasticBeamColumn 21 20 21 $A4 $E1 $G1 $J4 $Iy4
     $Iz4 $Vigasz -mass $M4;
     #Viga paralela eje z - Elemento 21
176   element elasticBeamColumn 22 22 21 $A6 $E1 $G1 $J6 $Iy6
     $Iz6 $Columnas -mass $M6; #Columna X3 - Elemento 22
177   element elasticBeamColumn 23 23 22 $A6 $E1 $G1 $J6 $Iy6
     $Iz6 $Columnas -mass $M6; #Columna X3 - Elemento 23
178   element elasticBeamColumn 24 24 23 $A6 $E1 $G1 $J6 $Iy6
     $Iz6 $Columnas -mass $M6; #Columna X3 - Elemento 24
179
180   puts "ELEMENTOS CREADOS!!"
181
182   #-----
183   # Grabacion de modos de nodos principales
     (nodos 4-7-18-21)para 4 modos
184   #-----
185   recorder Node -file Modos1.txt -node 4 7 18 21
     -dof 1 2 3 4 5 6 "eigen 1"
186   recorder Node -file Modos2.txt -node 4 7 18 21
     -dof 1 2 3 4 5 6 "eigen 2"
187   recorder Node -file Modos3.txt -node 4 7 18 21
     -dof 1 2 3 4 5 6 "eigen 3"
188   recorder Node -file Modos4.txt -node 4 7 18 21
     -dof 1 2 3 4 5 6 "eigen 4"
189
190   #-----
191   #Realizacion de eigen-analisis
192   #-----
193   set lambda [eigen $numModes];
194
195   #-----
196   #Calculo de frecuencias y periodos de la estructura
197   #-----
198   set omega {}
199   set f {}
200   set T {}
201   set pi 3.141593
202

```

```

203     foreach lam $lambda {
204         lappend omega [expr sqrt($lam)]
205         lappend f [expr sqrt($lam)/(2*$pi)]
206         lappend T [expr (2*$pi)/sqrt($lam)]
207     }
208
209     puts "Las frecuencias son $f (Hz)"
210     puts "Los periodos son $T (seg)"
211
212     #-----
213     #Grabacion de los periodos
214     #-----
215     set periodo "Periodos.txt"
216     set Periodos [open $periodo "w"]
217     foreach t $T {
218         puts $Periodos " $t"
219     }
220     close $Periodos
221
222     #-----
223     #Grabacion de las frecuencias
224     #-----
225     set frec "Frecuencias.txt"
226     set Frecuencias [open $frec "w"]
227     foreach t $f {
228         puts $Frecuencias " $t"
229     }
230     close $Frecuencias
231
232     record

```

ANEXO B. Manual de OPENSEES (ver CD adjunto).

Este anexo es un manual OPENSEES en PDF descargado en la pagina <http://opensees.berkeley.edu/OpenSees/manuals/usermanual/OpenSeesCommandLanguageManual.pdf> y se encuentra en el CD adjunto (ANEXO 2 y 3) con el nombre de OpenSeesCommandLanguageManualJune2006.PDF

ANEXO C. Programa OPENSEES (ver CD adjunto).

Este anexo es el instalador del programa OPENSEES descargado en la pagina <http://opensees.berkeley.edu/OpenSees/user/download.php> y se encuentra en el CD adjunto (ANEXO 2 y 3) OpenSees2.4.0.exe.zip.

ANEXO D. Validación de pórticos.

Con el fin de validar el proceso realizado en OPENSEES se compararon frecuencias de los pórticos calculadas con OPENSEES y las dadas por SAP2000.

Pórtico Simétrico			Pórtico Asimétrico		
SAP2000 Periodos [seg]	OPENSEES Periodos [seg]	% Error	SAP2000 Periodos [seg]	OPENSEES Periodos [seg]	% Error
0.185184	0.1861039	0.50%	0.097931	0.098457585	0.54%
0.088030	0.0881886	0.18%	0.097931	0.097700405	0.24%
0.057767	0.0573567	0.71%	0.040176	0.040371303	0.49%
0.044983	0.0449014	0.18%	0.039348	0.039709991	0.92%
0.042436	0.0426686	0.55%	0.039348	0.039547213	0.51%
0.038945	0.0386015	0.88%	0.015338	0.015337129	0.01%
0.025008	0.0249979	0.04%	0.015007	0.015055774	0.33%
0.021862	0.0218915	0.13%	0.014365	0.014226929	0.96%
0.018098	0.0179762	0.67%	0.009393	0.009362528	0.32%
0.015790	0.0158682	0.50%	0.009319	0.009362528	0.47%
0.014159	0.0141938	0.25%	0.009319	0.009329134	0.11%
0.011598	0.0115693	0.25%	0.008406	0.008395372	0.13%
0.008475	0.0083969	0.92%	0.008276	0.008320996	0.54%
0.007343	0.0072898	0.72%	0.006303	0.006275553	0.44%
0.006818	0.0068532	0.52%	0.00623	0.006216894	0.21%
0.006721	0.0067660	0.67%	0.006161	0.006175611	0.24%
0.006667	0.0066605	0.10%	0.005692	0.005723113	0.55%
0.006493	0.0065074	0.22%	0.00569	0.00568673	0.06%
0.005759	0.0057585	0.01%	0.003489	0.003482888	0.18%
0.004041	0.0040430	0.05%	0.003489	0.003481911	0.20%
0.004007	0.0039973	0.24%	0.003483	0.003481911	0.03%
0.002225	0.0022257	0.03%	0.003481	0.003481911	0.03%
0.002216	0.0022240	0.36%	0.003407	0.003413385	0.19%

ANEXO E. Código de análisis modal de la estructura en código MATLAB.

Código de análisis modal de la estructura en lenguaje MATLAB para generación de los comandos Tcl utilizados por OPENSEES para la realización del análisis modal.

En este anexo se muestran los comandos utilizados para la creación y escritura del Tcl.

Comando para creación del archivo Tcl:

```
fid=fopen('Titulo.tcl', 'w');
```

Comando para la escritura del Tcl en filas de texto:

```
fprintf(fid, 'linea 1 codigo;\n');  
fprintf(fid, 'linea 2 codigo;\n');  
fprintf(fid, 'linea . codigo;\n');  
fprintf(fid, 'linea . codigo;\n');  
fprintf(fid, 'linea . codigo;\n');  
fprintf(fid, 'linea n codigo;\n');
```

Comando de finalización del Tcl:

```
fclose(fid);
```

Comando para correr el Tcl en OPENSEES:

```
dos('opensees Portico.tcl');
```

- Pórtico Simétrico

```
E(1:24,1)=21.54e9;
```

```
% BENCHMARK PROBLEM
% UNIDADES: Kg, m, sec, N
% ESCRITO: fssm
% FECHA: Enero 2013 Portico3D.tcl
fid=fopen('Portico.tcl', 'w');

fprintf(fid, 'wipe;\n'); %Limpiar memoria
fprintf(fid, 'model BasicBuilder -ndm 3 -ndf 6;\n'); %Definicion del
modelo 3D y 6GDL por nodo
fprintf(fid, 'set dataDir Portico3Def;\n'); %Asignacion del
nombre de Datos1 al directorio datadir
fprintf(fid, 'file mkdir $dataDir;\n'); %Definicion del
archivo de salida Datos1
fprintf(fid, '\n');
%-----
fprintf(fid, 'set numModos 12;\n'); %Numero de modos a calcular
%-----
fprintf(fid, '\n');
%-----
%COORDENADAS, RESTRICCIONES
%-----
fprintf(fid, '\n');
fprintf(fid, 'set LCol 1;\n'); %Longitud Columnas paraleasl al eje Y
fprintf(fid, 'set LBeam 2;\n'); %Longitud Vigas paraleasl al eje X
fprintf(fid, 'set LGird 2;\n'); %Longitud Vigas paraleasl al eje Z
fprintf(fid, 'set dmas 2400;\n'); %Densidad de la masa Concreto Reforzado
[Kg/m3]
fprintf(fid, '\n');
fprintf(fid, '\n');
%Definicion de coordenadas nodales
fprintf(fid, 'set X0 0.;\n');
fprintf(fid, 'set X1 [expr $X0 + $LBeam];\n');
fprintf(fid, 'set X2 [expr $X1 + $LBeam];\n');
fprintf(fid, 'set X3 [expr $X2 + $LBeam];\n');
fprintf(fid, 'set Y0 0.;\n');
fprintf(fid, 'set Y1 [expr $Y0 + $LCol];\n');
fprintf(fid, 'set Y2 [expr $Y1 + $LCol];\n');
fprintf(fid, 'set Y3 [expr $Y2 + $LCol];\n');
fprintf(fid, 'set Z0 0.;\n');
fprintf(fid, 'set Z1 [expr $Z0 + $LGird];\n');
fprintf(fid, 'set Z2 [expr $Z1 + $LGird];\n');
fprintf(fid, 'set Z3 [expr $Z2 + $LGird];\n');
fprintf(fid, '\n');
%NODOS
fprintf(fid, '\n');
% Portico 1
fprintf(fid, 'node 1 $X0 $Y0 $Z0\n');
fprintf(fid, 'node 2 $X0 $Y1 $Z0\n');
fprintf(fid, 'node 3 $X0 $Y2 $Z0\n');
```

```

fprintf(fid,'node 4 $X0 $Y3 $Z0\n');
fprintf(fid,'node 5 $X1 $Y3 $Z0\n');
fprintf(fid,'node 6 $X2 $Y3 $Z0\n');
fprintf(fid,'node 7 $X3 $Y3 $Z0\n');
fprintf(fid,'node 8 $X3 $Y2 $Z0\n');
fprintf(fid,'node 9 $X3 $Y1 $Z0\n');
fprintf(fid,'node 10 $X3 $Y0 $Z0\n');
fprintf(fid,'\n');
% Nodos vigas eje Z
fprintf(fid,'node 11 $X0 $Y3 $Z1\n');
fprintf(fid,'node 12 $X0 $Y3 $Z2\n');
fprintf(fid,'\n');
fprintf(fid,'node 13 $X3 $Y3 $Z1\n');
fprintf(fid,'node 14 $X3 $Y3 $Z2\n');
fprintf(fid,'\n');
% Poritco 2
fprintf(fid,'node 15 $X0 $Y0 $Z3\n');
fprintf(fid,'node 16 $X0 $Y1 $Z3\n');
fprintf(fid,'node 17 $X0 $Y2 $Z3\n');
fprintf(fid,'node 18 $X0 $Y3 $Z3\n');
fprintf(fid,'node 19 $X1 $Y3 $Z3\n');
fprintf(fid,'node 20 $X2 $Y3 $Z3\n');
fprintf(fid,'node 21 $X3 $Y3 $Z3\n');
fprintf(fid,'node 22 $X3 $Y2 $Z3\n');
fprintf(fid,'node 23 $X3 $Y1 $Z3\n');
fprintf(fid,'node 24 $X3 $Y0 $Z3\n');
fprintf(fid,'\n');
fprintf(fid,'\n');
% RESTRICCIONES
fprintf(fid,'fix 1 1 1 1 1 1 1\n');
fprintf(fid,'fix 10 1 1 1 1 1 1\n');
fprintf(fid,'fix 15 1 1 1 1 1 1\n');
fprintf(fid,'fix 24 1 1 1 1 1 1\n');
fprintf(fid,'\n');

fprintf(fid,'\n');
fprintf(fid,'\n');
%-----
%DEFINICION DE CARACTERITICAS DE LOS ELEMENTOS
%-----

% PROPIEDADDES COLUMNAS
fprintf(fid,'set A1 0.16;\n'); %Area de la seccion
transversal
fprintf(fid,'set J1 0.003605;\n'); %Constante torsional
fprintf(fid,'set Iz1 0.002133;\n'); %Momento de inercia en eje z
fprintf(fid,'set Iy1 0.002133;\n'); %Momento de inercia en eje y
fprintf(fid,'\n');
fprintf(fid,'set M1 [expr $dmas*$A1];\n'); %Masa por unidad de
longitud
fprintf(fid,'\n');
fprintf(fid,'\n');
% PROPIEDADDES VIGAS

```

```

fprintf(fid,'set A2 0.16;\n');           %Area de la seccion
transversal
fprintf(fid,'set J2 0.003604;\n');      %Constante torsional
fprintf(fid,'set Iz2 0.002133;\n');    %Momento de inercia en eje z
fprintf(fid,'set Iy2 0.002133;\n');    %Momento de inercia en eje y
fprintf(fid,'\n');
fprintf(fid,'set M2 [expr $dmas*$A2];\n'); %Masa por unidad de
longitud
fprintf(fid,'\n');
fprintf(fid,'\n');

fprintf(fid,'\n');
fprintf(fid,'\n');
% MODULOS DE ELASTICIDAD
fprintf(fid,['set E1 ' num2str(E(1)) ';' '\n']); %Modulo de elasticidad
elemento 1
fprintf(fid,['set E2 ' num2str(E(2)) ';' '\n']); %Modulo de elasticidad
elemento 2
fprintf(fid,['set E3 ' num2str(E(3)) ';' '\n']); %Modulo de elasticidad
elemento 3
fprintf(fid,['set E4 ' num2str(E(4)) ';' '\n']); %Modulo de elasticidad
elemento 4
fprintf(fid,['set E5 ' num2str(E(5)) ';' '\n']); %Modulo de elasticidad
elemento 5
fprintf(fid,['set E6 ' num2str(E(6)) ';' '\n']); %Modulo de elasticidad
elemento 6
fprintf(fid,['set E7 ' num2str(E(7)) ';' '\n']); %Modulo de elasticidad
elemento 7
fprintf(fid,['set E8 ' num2str(E(8)) ';' '\n']); %Modulo de elasticidad
elemento 8
fprintf(fid,['set E9 ' num2str(E(9)) ';' '\n']); %Modulo de elasticidad
elemento 9
fprintf(fid,['set E10 ' num2str(E(10)) ';' '\n']); %Modulo de elasticidad
elemento 10
fprintf(fid,['set E11 ' num2str(E(11)) ';' '\n']); %Modulo de elasticidad
elemento 11
fprintf(fid,['set E12 ' num2str(E(12)) ';' '\n']); %Modulo de elasticidad
elemento 12
fprintf(fid,['set E13 ' num2str(E(13)) ';' '\n']); %Modulo de elasticidad
elemento 13
fprintf(fid,['set E14 ' num2str(E(14)) ';' '\n']); %Modulo de elasticidad
elemento 14
fprintf(fid,['set E15 ' num2str(E(15)) ';' '\n']); %Modulo de elasticidad
elemento 15
fprintf(fid,['set E16 ' num2str(E(16)) ';' '\n']); %Modulo de elasticidad
elemento 16
fprintf(fid,['set E17 ' num2str(E(17)) ';' '\n']); %Modulo de elasticidad
elemento 17
fprintf(fid,['set E18 ' num2str(E(18)) ';' '\n']); %Modulo de elasticidad
elemento 18
fprintf(fid,['set E19 ' num2str(E(19)) ';' '\n']); %Modulo de elasticidad
elemento 19
fprintf(fid,['set E20 ' num2str(E(20)) ';' '\n']); %Modulo de elasticidad
elemento 20

```

```

fprintf(fid,['set E21 ' num2str(E(21)) ';\n']); %Modulo de elasticidad
elemento 21
fprintf(fid,['set E22 ' num2str(E(22)) ';\n']); %Modulo de elasticidad
elemento 22
fprintf(fid,['set E23 ' num2str(E(23)) ';\n']); %Modulo de elasticidad
elemento 23
fprintf(fid,['set E24 ' num2str(E(24)) ';\n']); %Modulo de elasticidad
elemento 24
fprintf(fid,'\n');
% MODULOS DE RIGIDEZ
fprintf(fid,'set G1 [expr $E1/(2./(1+0.2))];\n'); %Modulo de rigidez
elemento 1
fprintf(fid,'set G2 [expr $E2/(2./(1+0.2))];\n'); %Modulo de rigidez
elemento 2
fprintf(fid,'set G3 [expr $E3/(2./(1+0.2))];\n'); %Modulo de rigidez
elemento 3
fprintf(fid,'set G4 [expr $E4/(2./(1+0.2))];\n'); %Modulo de rigidez
elemento 4
fprintf(fid,'set G5 [expr $E5/(2./(1+0.2))];\n'); %Modulo de rigidez
elemento 5
fprintf(fid,'set G6 [expr $E6/(2./(1+0.2))];\n'); %Modulo de rigidez
elemento 6
fprintf(fid,'set G7 [expr $E7/(2./(1+0.2))];\n'); %Modulo de rigidez
elemento 7
fprintf(fid,'set G8 [expr $E8/(2./(1+0.2))];\n'); %Modulo de rigidez
elemento 8
fprintf(fid,'set G9 [expr $E9/(2./(1+0.2))];\n'); %Modulo de rigidez
elemento 9
fprintf(fid,'set G10 [expr $E10/(2./(1+0.2))];\n'); %Modulo de rigidez
elemento 10
fprintf(fid,'set G11 [expr $E11/(2./(1+0.2))];\n'); %Modulo de rigidez
elemento 11
fprintf(fid,'set G12 [expr $E12/(2./(1+0.2))];\n'); %Modulo de rigidez
elemento 12
fprintf(fid,'set G13 [expr $E13/(2./(1+0.2))];\n'); %Modulo de rigidez
elemento 13
fprintf(fid,'set G14 [expr $E14/(2./(1+0.2))];\n'); %Modulo de rigidez
elemento 14
fprintf(fid,'set G15 [expr $E15/(2./(1+0.2))];\n'); %Modulo de rigidez
elemento 15
fprintf(fid,'set G16 [expr $E16/(2./(1+0.2))];\n'); %Modulo de rigidez
elemento 16
fprintf(fid,'set G17 [expr $E17/(2./(1+0.2))];\n'); %Modulo de rigidez
elemento 17
fprintf(fid,'set G18 [expr $E18/(2./(1+0.2))];\n'); %Modulo de rigidez
elemento 18
fprintf(fid,'set G19 [expr $E19/(2./(1+0.2))];\n'); %Modulo de rigidez
elemento 19
fprintf(fid,'set G20 [expr $E20/(2./(1+0.2))];\n'); %Modulo de rigidez
elemento 20
fprintf(fid,'set G21 [expr $E21/(2./(1+0.2))];\n'); %Modulo de rigidez
elemento 21
fprintf(fid,'set G22 [expr $E22/(2./(1+0.2))];\n'); %Modulo de rigidez
elemento 22

```

```

fprintf(fid,'set G23 [expr $E23/(2./(1+0.2))];\n'); %Modulo de rigidez
elemendo 23
fprintf(fid,'set G24 [expr $E24/(2./(1+0.2))];\n'); %Modulo de rigidez
elemendo 24
fprintf(fid,'\n');
% TRANSFORMACION GEOMETRICA
fprintf(fid,'set Columnas 1;\n'); %Columnas
fprintf(fid,'set Vigasx 2;\n'); %Vigas eje x
fprintf(fid,'set Vigasz 3;\n'); %Vigas eje z
fprintf(fid,'geomTransf Linear $Columnas 0 0 1\n');
fprintf(fid,'geomTransf Linear $Vigasx 0 0 1\n');
fprintf(fid,'geomTransf Linear $Vigasx -1 0 0\n');
fprintf(fid,'\n');
fprintf(fid,'\n');
% CREACION DE ELEMENTOS (TIPO VIGA-COLUMNA)comando: element
elasticBeamColumn $eleTag $iNode $jNode $A $E $G $J $Iy $Iz $transfTag <-
mass $massDens>

fprintf(fid,'element elasticBeamColumn 1 1 2 $A1 $E1 $G1 $J1 $Iy1 $Iz1
$Columnas -mass $M1;\n'); %Columna - Elemento 1
fprintf(fid,'element elasticBeamColumn 2 2 3 $A1 $E2 $G2 $J1 $Iy1 $Iz1
$Columnas -mass $M1;\n'); %Columna - Elemento 2
fprintf(fid,'element elasticBeamColumn 3 3 4 $A1 $E3 $G3 $J1 $Iy1 $Iz1
$Columnas -mass $M1;\n'); %Columna - Elemento 3
fprintf(fid,'element elasticBeamColumn 4 4 5 $A2 $E4 $G4 $J2 $Iy2 $Iz2
$Vigasx -mass $M2;\n'); %Viga x - Elemento 4
fprintf(fid,'element elasticBeamColumn 5 5 6 $A2 $E5 $G5 $J2 $Iy2 $Iz2
$Vigasx -mass $M2;\n'); %Viga x - Elemento 5
fprintf(fid,'element elasticBeamColumn 6 6 7 $A2 $E6 $G6 $J2 $Iy2 $Iz2
$Vigasx -mass $M2;\n'); %Viga x - Elemento 6
fprintf(fid,'element elasticBeamColumn 7 8 7 $A1 $E7 $G7 $J1 $Iy1 $Iz1
$Columnas -mass $M1;\n'); %Columna - Elemento 7
fprintf(fid,'element elasticBeamColumn 8 9 8 $A1 $E8 $G8 $J1 $Iy1 $Iz1
$Columnas -mass $M1;\n'); %Columna - Elemento /
fprintf(fid,'element elasticBeamColumn 9 10 9 $A1 $E9 $G9 $J1 $Iy1 $Iz1
$Columnas -mass $M1;\n'); %Columna - Elemento 9
fprintf(fid,'\n');
fprintf(fid,'element elasticBeamColumn 10 4 11 $A2 $E10 $G10 $J2 $Iy2
$Iz2 $Vigasx -mass $M2;\n'); %Viga z - Elemento 10
fprintf(fid,'element elasticBeamColumn 11 11 12 $A2 $E11 $G11 $J2 $Iy2
$Iz2 $Vigasx -mass $M2;\n'); %Viga z - Elemento 11
fprintf(fid,'element elasticBeamColumn 12 12 16 $A2 $E12 $G12 $J2 $Iy2
$Iz2 $Vigasx -mass $M2;\n'); %Viga z - Elemento 12
fprintf(fid,'\n');
fprintf(fid,'element elasticBeamColumn 13 7 13 $A2 $E13 $G13 $J2 $Iy2
$Iz2 $Vigasx -mass $M2;\n'); %Viga z - Elemento 13
fprintf(fid,'element elasticBeamColumn 14 13 14 $A2 $E14 $G14 $J2 $Iy2
$Iz2 $Vigasx -mass $M2;\n'); %Viga z - Elemento 14
fprintf(fid,'element elasticBeamColumn 15 14 21 $A2 $E15 $G15 $J2 $Iy2
$Iz2 $Vigasx -mass $M2;\n'); %Viga z - Elemento 15
fprintf(fid,'\n');
fprintf(fid,'element elasticBeamColumn 16 15 16 $A1 $E16 $G16 $J1 $Iy1
$Iz1 $Columnas -mass $M1;\n'); %Columna - Elemento 16

```

```

fprintf(fid,'element elasticBeamColumn 17 16 17 $A1 $E17 $G17 $J1 $Iy1
$Iz1 $Columnas -mass $M1;\n'); %Columna - Elemento 17
fprintf(fid,'element elasticBeamColumn 18 17 18 $A1 $E18 $G18 $J1 $Iy1
$Iz1 $Columnas -mass $M1;\n'); %Columna - Elemento 18
fprintf(fid,'element elasticBeamColumn 19 18 19 $A2 $E19 $G19 $J2 $Iy2
$Iz2 $Vigasx -mass $M2;\n'); %Viga x - Elemento 19
fprintf(fid,'element elasticBeamColumn 20 19 20 $A2 $E20 $G20 $J2 $Iy2
$Iz2 $Vigasx -mass $M2;\n'); %Viga x - Elemento 20
fprintf(fid,'element elasticBeamColumn 21 20 21 $A2 $E21 $G21 $J2 $Iy2
$Iz2 $Vigasx -mass $M2;\n'); %Viga x - Elemento 21
fprintf(fid,'element elasticBeamColumn 22 22 21 $A1 $E22 $G22 $J1 $Iy1
$Iz1 $Columnas -mass $M1;\n'); %Columna - Elemento 22
fprintf(fid,'element elasticBeamColumn 23 23 22 $A1 $E23 $G23 $J1 $Iy1
$Iz1 $Columnas -mass $M1;\n'); %Columna - Elemento 23
fprintf(fid,'element elasticBeamColumn 24 24 23 $A1 $E24 $G24 $J1 $Iy1
$Iz1 $Columnas -mass $M1;\n'); %Columna - Elemento 24
fprintf(fid,'\n');

fprintf(fid,'\n');
fprintf(fid,'\n');
fprintf(fid,'recorder Node -file Modos1.txt -node 4 7 18 21 -dof 1 2 3 4
5 6 "eigen 1"\n');
fprintf(fid,'recorder Node -file Modos2.txt -node 4 7 18 21 -dof 1 2 3 4
5 6 "eigen 2"\n');
fprintf(fid,'recorder Node -file Modos3.txt -node 4 7 18 21 -dof 1 2 3 4
5 6 "eigen 3"\n');
fprintf(fid,'recorder Node -file Modos4.txt -node 4 7 18 21 -dof 1 2 3 4
5 6 "eigen 4"\n');

fprintf(fid,'\n');
%-----
%Realizacion de eigen- analisis
%-----
fprintf(fid,'set lambda [eigen $numModes];\n');
fprintf(fid,'\n');
fprintf(fid,'\n');
%-----
%Calculo de frecuencias y periodos de la estructura
%-----
fprintf(fid,'set omega {}\n');
fprintf(fid,'set f {}\n');
fprintf(fid,'set T {}\n');
fprintf(fid,'set pi 3.141593\n');
fprintf(fid,'\n');
fprintf(fid,'foreach lam $lambda {\n');
fprintf(fid,'  lappend omega [expr sqrt($lam)]\n');
fprintf(fid,'  lappend f [expr sqrt($lam)/(2*$pi)]\n');
fprintf(fid,'  lappend T [expr (2*$pi)/sqrt($lam)]\n');
fprintf(fid,'}\n');
fprintf(fid,'\n');

fprintf(fid,'\n');
fprintf(fid,'\n');

```

```

%-----
%Grabacion de lo periodos
%-----
fprintf(fid,'set periodo "Periodos.txt"\n');
fprintf(fid,'set Periodos [open $periodo "w"]\n');
fprintf(fid,'foreach t $T {\n');
fprintf(fid,'    puts $Periodos " $t"\n');
fprintf(fid,'}\n');
fprintf(fid,'close $Periodos\n');
fprintf(fid,'\n');
fprintf(fid,'\n');
%-----
%Grabacion de las frecuencias
%-----
fprintf(fid,'set frec "Frecuencias.txt"\n');
fprintf(fid,'set Frecuencias [open $frec "w"]\n');
fprintf(fid,'foreach t $f {\n');
fprintf(fid,'    puts $Frecuencias " $t"\n');
fprintf(fid,'}\n');
fprintf(fid,'close $Frecuencias\n');
fprintf(fid,'\n');
fprintf(fid,'\n');
fprintf(fid,'record\n');
fclose(fid);
%Ejecutar Opensees
dos('opensees Portico.tcl');

%Lectura de TXT
[Modos1]=textread('Modos1.txt', '%f');
[Modos2]=textread('Modos2.txt', '%f');
[Modos3]=textread('Modos3.txt', '%f');
[Modos4]=textread('Modos4.txt', '%f');

[fe]=textread('Frecuencias.txt', '%f');

%Ensamble de matriz de modos

mve(1:24,1)=Modos1;
mve(1:24,2)=Modos2;
mve(1:24,3)=Modos3;
mve(1:24,4)=Modos4;

```

- Pórtico Asimétrico

```
E(1:24,1)=21.54e9;
```

```
fid=fopen('Portico.tcl', 'w');
```

```
% UNIDADES: Kg, m, sec, N
% FECHA: Febrero 2013 Portico.tcl
```

```
fprintf(fid, 'wipe;\n'); %Limpiar memoria
fprintf(fid, 'model BasicBuilder -ndm 3 -ndf 6;\n'); %Definicion del
modelo 3D y 6GDL por nodo
fprintf(fid, 'set dataDir Portico;\n'); %Asignacion del
nombre de Datos1 al directorio datadir
fprintf(fid, 'file mkdir $dataDir;\n'); %Definicion del
archivo de salida Datos1
```

```
%-----
fprintf(fid, 'set numModes 4;\n'); %Numero de modos a calcular
%-----
```

```
%-----
%COORDENADAS, RESTRICCIONES
%-----
```

```
fprintf(fid, 'set LCol1 1;\n'); %Longitud Columnas paraleasl al
eje Y ubicadas en eje X=0
fprintf(fid, 'set LCol2 1.5;\n'); %Longitud Columnas paraleasl al
eje Y ubicadas en eje X=4.5
fprintf(fid, 'set LBeam 1.5;\n'); %Longitud Vigas paraleasl al eje
X
fprintf(fid, 'set LGird 1;\n'); %Longitud Vigas paraleasl al eje
Z
fprintf(fid, 'set dmas 2356.3122;\n'); %Densidad de la masa Concreto
Reforzado [Kg/m3]
```

```
%Definicion de coordenadas nodales
fprintf(fid, 'set X0 0.;\n');
fprintf(fid, 'set X1 [expr $X0 + $LBeam];\n');
fprintf(fid, 'set X2 [expr $X1 + $LBeam];\n');
fprintf(fid, 'set X3 [expr $X2 + $LBeam];\n');
fprintf(fid, 'set Y0 0.;\n');
fprintf(fid, 'set Y1 [expr $Y0 + $LCol1];\n');
fprintf(fid, 'set Y3 [expr $Y1 + $LCol1];\n');
fprintf(fid, 'set Y5 [expr $Y3 + $LCol1];\n');
fprintf(fid, 'set Y2 [expr $Y1 + $LCol2];\n');
fprintf(fid, 'set Y4 [expr $Y2 + $LCol2];\n');
fprintf(fid, 'set Z0 0.;\n');
fprintf(fid, 'set Z1 [expr $Z0 + $LGird];\n');
fprintf(fid, 'set Z2 [expr $Z1 + $LGird];\n');
fprintf(fid, 'set Z3 [expr $Z2 + $LGird];\n');
```

```

%NODOS

%Portico X=0

fprintf(fid,'node 1 $X0 $Y1 $Z0\n');
fprintf(fid,'node 2 $X0 $Y2 $Z0\n');
fprintf(fid,'node 3 $X0 $Y4 $Z0\n');
fprintf(fid,'node 4 $X0 $Y5 $Z0\n');
fprintf(fid,'node 5 $X0 $Y5 $Z1\n');
fprintf(fid,'node 6 $X0 $Y5 $Z2\n');
fprintf(fid,'node 7 $X0 $Y5 $Z3\n');
fprintf(fid,'node 8 $X0 $Y4 $Z3\n');
fprintf(fid,'node 9 $X0 $Y2 $Z3\n');
fprintf(fid,'node 10 $X0 $Y1 $Z3\n');

%Vigas paralelas a eje X

fprintf(fid,'node 11 $X1 $Y5 $Z0\n');
fprintf(fid,'node 12 $X2 $Y5 $Z0\n');

fprintf(fid,'node 13 $X1 $Y5 $Z3\n');
fprintf(fid,'node 14 $X2 $Y5 $Z3\n');

%Portico X=4.5

fprintf(fid,'node 15 $X3 $Y0 $Z0\n');
fprintf(fid,'node 16 $X3 $Y1 $Z0\n');
fprintf(fid,'node 17 $X3 $Y3 $Z0\n');
fprintf(fid,'node 18 $X3 $Y5 $Z0\n');
fprintf(fid,'node 19 $X3 $Y5 $Z1\n');
fprintf(fid,'node 20 $X3 $Y5 $Z2\n');
fprintf(fid,'node 21 $X3 $Y5 $Z3\n');
fprintf(fid,'node 22 $X3 $Y3 $Z3\n');
fprintf(fid,'node 23 $X3 $Y1 $Z3\n');
fprintf(fid,'node 24 $X3 $Y0 $Z3\n');

% RESTRICCIONES
fprintf(fid,'fix 1 1 1 1 1 1\n');
fprintf(fid,'fix 10 1 1 1 1 1\n');
fprintf(fid,'fix 15 1 1 1 1 1\n');
fprintf(fid,'fix 24 1 1 1 1 1\n');

%fprintf(fid,'puts "NUDOS CREADOS Y RESTRINGIDOS!!"\n');

%-----
%DEFINICION DE CARACTERITICAS DE LOS ELEMENTOS
%-----

% MODULOS DE ELASTICIDAD
fprintf(fid,['set E1 ' num2str(E(1)) ';\n']); %Modulo de elasticidad
elemento 1

```

```

fprintf(fid,['set E2 ' num2str(E(2)) ';\n']); %Modulo de elasticidad
elemento 2
fprintf(fid,['set E3 ' num2str(E(3)) ';\n']); %Modulo de elasticidad
elemento 3
fprintf(fid,['set E4 ' num2str(E(4)) ';\n']); %Modulo de elasticidad
elemento 4
fprintf(fid,['set E5 ' num2str(E(5)) ';\n']); %Modulo de elasticidad
elemento 5
fprintf(fid,['set E6 ' num2str(E(6)) ';\n']); %Modulo de elasticidad
elemento 6
fprintf(fid,['set E7 ' num2str(E(7)) ';\n']); %Modulo de elasticidad
elemento 7
fprintf(fid,['set E8 ' num2str(E(8)) ';\n']); %Modulo de elasticidad
elemento 8
fprintf(fid,['set E9 ' num2str(E(9)) ';\n']); %Modulo de elasticidad
elemento 9
fprintf(fid,['set E10 ' num2str(E(10)) ';\n']); %Modulo de elasticidad
elemento 10
fprintf(fid,['set E11 ' num2str(E(11)) ';\n']); %Modulo de elasticidad
elemento 11
fprintf(fid,['set E12 ' num2str(E(12)) ';\n']); %Modulo de elasticidad
elemento 12
fprintf(fid,['set E13 ' num2str(E(13)) ';\n']); %Modulo de elasticidad
elemento 13
fprintf(fid,['set E14 ' num2str(E(14)) ';\n']); %Modulo de elasticidad
elemento 14
fprintf(fid,['set E15 ' num2str(E(15)) ';\n']); %Modulo de elasticidad
elemento 15
fprintf(fid,['set E16 ' num2str(E(16)) ';\n']); %Modulo de elasticidad
elemento 16
fprintf(fid,['set E17 ' num2str(E(17)) ';\n']); %Modulo de elasticidad
elemento 17
fprintf(fid,['set E18 ' num2str(E(18)) ';\n']); %Modulo de elasticidad
elemento 18
fprintf(fid,['set E19 ' num2str(E(19)) ';\n']); %Modulo de elasticidad
elemento 19
fprintf(fid,['set E20 ' num2str(E(20)) ';\n']); %Modulo de elasticidad
elemento 20
fprintf(fid,['set E21 ' num2str(E(21)) ';\n']); %Modulo de elasticidad
elemento 21
fprintf(fid,['set E22 ' num2str(E(22)) ';\n']); %Modulo de elasticidad
elemento 22
fprintf(fid,['set E23 ' num2str(E(23)) ';\n']); %Modulo de elasticidad
elemento 23
fprintf(fid,['set E24 ' num2str(E(24)) ';\n']); %Modulo de elasticidad
elemento 24

% MODULOS DE RIGIDEZ
fprintf(fid,'set G1 [expr $E1/(2./(1+0.2))];\n'); %Modulo de rigidez
elemento 1
fprintf(fid,'set G2 [expr $E2/(2./(1+0.2))];\n'); %Modulo de rigidez
elemento 2
fprintf(fid,'set G3 [expr $E3/(2./(1+0.2))];\n'); %Modulo de rigidez
elemento 3

```

```

fprintf(fid,'set G4 [expr $E4/(2./(1+0.2))];\n'); %Modulo de rigidez
elemendo 4
fprintf(fid,'set G5 [expr $E5/(2./(1+0.2))];\n'); %Modulo de rigidez
elemendo 5
fprintf(fid,'set G6 [expr $E6/(2./(1+0.2))];\n'); %Modulo de rigidez
elemendo 6
fprintf(fid,'set G7 [expr $E7/(2./(1+0.2))];\n'); %Modulo de rigidez
elemendo 7
fprintf(fid,'set G8 [expr $E8/(2./(1+0.2))];\n'); %Modulo de rigidez
elemendo 8
fprintf(fid,'set G9 [expr $E9/(2./(1+0.2))];\n'); %Modulo de rigidez
elemendo 9
fprintf(fid,'set G10 [expr $E10/(2./(1+0.2))];\n'); %Modulo de rigidez
elemendo 10
fprintf(fid,'set G11 [expr $E11/(2./(1+0.2))];\n'); %Modulo de rigidez
elemendo 11
fprintf(fid,'set G12 [expr $E12/(2./(1+0.2))];\n'); %Modulo de rigidez
elemendo 12
fprintf(fid,'set G13 [expr $E13/(2./(1+0.2))];\n'); %Modulo de rigidez
elemendo 13
fprintf(fid,'set G14 [expr $E14/(2./(1+0.2))];\n'); %Modulo de rigidez
elemendo 14
fprintf(fid,'set G15 [expr $E15/(2./(1+0.2))];\n'); %Modulo de rigidez
elemendo 15
fprintf(fid,'set G16 [expr $E16/(2./(1+0.2))];\n'); %Modulo de rigidez
elemendo 16
fprintf(fid,'set G17 [expr $E17/(2./(1+0.2))];\n'); %Modulo de rigidez
elemendo 17
fprintf(fid,'set G18 [expr $E18/(2./(1+0.2))];\n'); %Modulo de rigidez
elemendo 18
fprintf(fid,'set G19 [expr $E19/(2./(1+0.2))];\n'); %Modulo de rigidez
elemendo 19
fprintf(fid,'set G20 [expr $E20/(2./(1+0.2))];\n'); %Modulo de rigidez
elemendo 20
fprintf(fid,'set G21 [expr $E21/(2./(1+0.2))];\n'); %Modulo de rigidez
elemendo 21
fprintf(fid,'set G22 [expr $E22/(2./(1+0.2))];\n'); %Modulo de rigidez
elemendo 22
fprintf(fid,'set G23 [expr $E23/(2./(1+0.2))];\n'); %Modulo de rigidez
elemendo 23
fprintf(fid,'set G24 [expr $E24/(2./(1+0.2))];\n'); %Modulo de rigidez
elemendo 24

%fprintf(fid,'set E1 21.54e9;\n'); %Modulo de elasticidad
%fprintf(fid,'set G1 [expr $E1/(2./(1+0.2))];\n'); %Modulo de rigidez

% PROPIEDADES COLUMNAS - ubicadas en X=0,Z=0 (20x20)
fprintf(fid,'set A1 0.04;\n'); %Area de la seccion
transversal
fprintf(fid,'set J1 0.0002253;\n'); %Constante torsional
fprintf(fid,'set Iz1 0.0001333;\n'); %Momento de inercia en eje z
fprintf(fid,'set Iy1 0.0001333;\n'); %Momento de inercia en eje y

```

```

fprintf(fid,'set M1 [expr $dmas*$A1];\n');           %Masa por unidad de
longitud

% PROPIEDADDES COLUMNAS - ubicadas en X=0,Z=3 (20x20)
fprintf(fid,'set A5 0.04;\n');                       %Area de la seccion transversal
fprintf(fid,'set J5 0.0002253;\n');                 %Constante torsional
fprintf(fid,'set Iz5 0.0001333;\n');                %Momento de inercia en eje z
fprintf(fid,'set Iy5 0.0001333;\n');                %Momento de inercia en eje y

fprintf(fid,'set M5 [expr $dmas*$A1];\n');           %Masa por unidad de
longitud

% PROPIEDADDES COLUMNAS - ubicadas en X=4.5,Z=0 (20x20)
fprintf(fid,'set A2 0.04;\n');                       %Area de la seccion
transversal
fprintf(fid,'set J2 0.0002253;\n');                 %Constante torsional
fprintf(fid,'set Iz2 0.0001333;\n');                %Momento de inercia en eje z
fprintf(fid,'set Iy2 0.0001333;\n');                %Momento de inercia en eje y

fprintf(fid,'set M2 [expr $dmas*$A2];\n');           %Masa por unidad de
longitud

% PROPIEDADDES COLUMNAS - ubicadas en X=4.5,Z=3 (21x21)
fprintf(fid,'set A6 0.0441;\n');                     %Area de la seccion
transversal
fprintf(fid,'set J6 0.0002739;\n');                 %Constante torsional
fprintf(fid,'set Iz6 0.0001621;\n');                %Momento de inercia en eje z
fprintf(fid,'set Iy6 0.0001621;\n');                %Momento de inercia en eje y

fprintf(fid,'set M6 [expr $dmas*$A2];\n');           %Masa por unidad de
longitud

% PROPIEDADDES VIGAS - paralelas eje x
fprintf(fid,'set A3 0.06;\n');                       %Area de la seccion
transversal
fprintf(fid,'set J3 0.0004695;\n');                 %Constante torsional
fprintf(fid,'set Iz3 0.00045;\n');                  %Momento de inercia en eje z
fprintf(fid,'set Iy3 0.0002;\n');                   %Momento de inercia en eje y

fprintf(fid,'set M3 [expr $dmas*$A3];\n');           %Masa por unidad de
longitud

% PROPIEDADDES VIGAS - paralelas eje z
fprintf(fid,'set A4 0.04;\n');                       %Area de la seccion
transversal
fprintf(fid,'set J4 0.0002253;\n');                 %Constante torsional
fprintf(fid,'set Iz4 0.0001333;\n');                %Momento de inercia en eje z
fprintf(fid,'set Iy4 0.0001333;\n');                %Momento de inercia en eje y

```

```

fprintf(fid,'set M4 [expr $dmas*$A3];\n');           %Masa por unidad de
longitud

%fprintf(fid,'puts "SECCIONES CREADAS!!"\n');

% TRANSFORMACION GEOMETRICA
fprintf(fid,'set Columnas 1;\n'); %Columnas
fprintf(fid,'set Vigasx 2;\n'); %Vigas eje x
fprintf(fid,'set Vigasz 3;\n'); %Vigas eje z
fprintf(fid,'geomTransf Linear $Columnas 0 0 1\n');
fprintf(fid,'geomTransf Linear $Vigasx 0 0 1\n');
fprintf(fid,'geomTransf Linear $Vigasz -1 0 0\n');

% CREACION DE ELEMENTOS (TIPO VIGA-COLUMNA)comando: element
elasticBeamColumn $eleTag $iNode $jNode $A $E $G $J $Iy $Iz $transfTag <-
mass $massDens>

fprintf(fid,'element elasticBeamColumn 1 1 2 $A1 $E1 $G1 $J1 $Iy1 $Iz1
$Columnas -mass $M1;\n'); %Columna X0 - Elemento 1
fprintf(fid,'element elasticBeamColumn 2 2 3 $A1 $E2 $G2 $J1 $Iy1 $Iz1
$Columnas -mass $M1;\n'); %Columna X0 - Elemento 2
fprintf(fid,'element elasticBeamColumn 3 3 4 $A1 $E3 $G3 $J1 $Iy1 $Iz1
$Columnas -mass $M1;\n'); %Columna X0 - Elemento 3
fprintf(fid,'element elasticBeamColumn 4 4 5 $A4 $E4 $G4 $J4 $Iy4 $Iz4
$Vigasz -mass $M4;\n'); %Viga paralela eje z - Elemento 4
fprintf(fid,'element elasticBeamColumn 5 5 6 $A4 $E5 $G5 $J4 $Iy4 $Iz4
$Vigasz -mass $M4;\n'); %Viga paralela eje z - Elemento 5
fprintf(fid,'element elasticBeamColumn 6 6 7 $A4 $E6 $G6 $J4 $Iy4 $Iz4
$Vigasz -mass $M4;\n'); %Viga paralela eje z - Elemento 6
fprintf(fid,'element elasticBeamColumn 7 8 7 $A5 $E7 $G7 $J5 $Iy5 $Iz5
$Columnas -mass $M5;\n'); %Columna X0 - Elemento 7
fprintf(fid,'element elasticBeamColumn 8 9 8 $A5 $E8 $G8 $J5 $Iy5 $Iz5
$Columnas -mass $M5;\n'); %Columna X0 - Elemento 8
fprintf(fid,'element elasticBeamColumn 9 10 9 $A5 $E9 $G9 $J5 $Iy5 $Iz5
$Columnas -mass $M5;\n'); %Columna X0 - Elemento 9

fprintf(fid,'element elasticBeamColumn 10 4 11 $A3 $E10 $G10 $J3 $Iy3
$Iz3 $Vigasx -mass $M3;\n'); %Viga paralela eje x - Elemento 10
fprintf(fid,'element elasticBeamColumn 11 11 12 $A3 $E11 $G11 $J3 $Iy3
$Iz3 $Vigasx -mass $M3;\n'); %Viga paralela eje x - Elemento 11
fprintf(fid,'element elasticBeamColumn 12 12 18 $A3 $E12 $G12 $J3 $Iy3
$Iz3 $Vigasx -mass $M3;\n'); %Viga paralela eje x - Elemento 12

fprintf(fid,'element elasticBeamColumn 13 7 13 $A3 $E13 $G13 $J3 $Iy3
$Iz3 $Vigasx -mass $M3;\n'); %Viga paralela eje x - Elemento 13
fprintf(fid,'element elasticBeamColumn 14 13 14 $A3 $E14 $G14 $J3 $Iy3
$Iz3 $Vigasx -mass $M3;\n'); %Viga paralela eje x - Elemento 14
fprintf(fid,'element elasticBeamColumn 15 14 21 $A3 $E15 $G15 $J3 $Iy3
$Iz3 $Vigasx -mass $M3;\n'); %Viga paralela eje x - Elemento 15

fprintf(fid,'element elasticBeamColumn 16 15 16 $A2 $E16 $G16 $J2 $Iy2
$Iz2 $Columnas -mass $M2;\n'); %Columna X3 - Elemento 16

```

```

fprintf(fid,'element elasticBeamColumn 17 16 17 $A2 $E17 $G17 $J2 $Iy2
$Iz2 $Columnas -mass $M2;\n'); %Columna X3 - Elemento 17
fprintf(fid,'element elasticBeamColumn 18 17 18 $A2 $E18 $G18 $J2 $Iy2
$Iz2 $Columnas -mass $M2;\n'); %Columna X3 - Elemento 18
fprintf(fid,'element elasticBeamColumn 19 18 19 $A4 $E19 $G19 $J4 $Iy4
$Iz4 $Vigas $ -mass $M4;\n'); %Viga paralela eje z - Elemento 19
fprintf(fid,'element elasticBeamColumn 20 19 20 $A4 $E20 $G20 $J4 $Iy4
$Iz4 $Vigas $ -mass $M4;\n'); %Viga paralela eje z - Elemento 20
fprintf(fid,'element elasticBeamColumn 21 20 21 $A4 $E21 $G21 $J4 $Iy4
$Iz4 $Vigas $ -mass $M4;\n'); %Viga paralela eje z - Elemento 21
fprintf(fid,'element elasticBeamColumn 22 22 21 $A6 $E22 $G22 $J6 $Iy6
$Iz6 $Columnas -mass $M6;\n'); %Columna X3 - Elemento 22
fprintf(fid,'element elasticBeamColumn 23 23 22 $A6 $E23 $G23 $J6 $Iy6
$Iz6 $Columnas -mass $M6;\n'); %Columna X3 - Elemento 23
fprintf(fid,'element elasticBeamColumn 24 24 23 $A6 $E24 $G24 $J6 $Iy6
$Iz6 $Columnas -mass $M6;\n'); %Columna X3 - Elemento 24

%fprintf(fid,'puts "ELEMENTOS CREADOS!!"\n');

%-----
% Grabacion de modos de nodos principales (nodos 4-7-18-21)para 4 modos
%-----

fprintf(fid,'recorder Node -file Modos1.txt -node 4 7 18 21 -dof 1 2 3 4
5 6 "eigen 1"\n');
fprintf(fid,'recorder Node -file Modos2.txt -node 4 7 18 21 -dof 1 2 3 4
5 6 "eigen 2"\n');
fprintf(fid,'recorder Node -file Modos3.txt -node 4 7 18 21 -dof 1 2 3 4
5 6 "eigen 3"\n');
fprintf(fid,'recorder Node -file Modos4.txt -node 4 7 18 21 -dof 1 2 3 4
5 6 "eigen 4"\n');

%-----
%Realizacion de eigen-analisis
%-----
fprintf(fid,'set lambda [eigen $numModes];\n');

%-----
%Calculo de frecuencias y periodos de la estructura
%-----
fprintf(fid,'set omega {\}\n');
fprintf(fid,'set f {\}\n');
fprintf(fid,'set T {\}\n');
fprintf(fid,'set pi 3.141593\n');

fprintf(fid,'foreach lam $lambda {\n');
fprintf(fid,'  lappend omega [expr sqrt($lam)]\n');
fprintf(fid,'  lappend f [expr sqrt($lam)/(2*$pi)]\n');
fprintf(fid,'  lappend T [expr (2*$pi)/sqrt($lam)]\n');
fprintf(fid,'}\n');

%fprintf(fid,'puts "Las frecuencias son $f (Hz)"\n');
%fprintf(fid,'puts "Los periodos son $T (seg)"\n');

```

```

%-----
%Grabacion de los periodos
%-----
fprintf(fid,'set periodo "Periodos.txt"\n');
fprintf(fid,'set Periodos [open $periodo "w"]\n');
fprintf(fid,'foreach t $T {\n');
fprintf(fid,'  puts $Periodos " $t"\n');
fprintf(fid,'}\n');
fprintf(fid,'close $Periodos\n');

%-----
%Grabacion de las frecuencias
%-----
fprintf(fid,'set frec "Frecuencias.txt"\n');
fprintf(fid,'set Frecuencias [open $frec "w"]\n');
fprintf(fid,'foreach t $f {\n');
fprintf(fid,'  puts $Frecuencias " $t"\n');
fprintf(fid,'}\n');
fprintf(fid,'close $Frecuencias\n');

fprintf(fid,'record\n');

fclose(fid);
dos('opensees Portico.tcl');

%Lectura de TXT
[Modos1]=textread('Modos1.txt', '%f');
[Modos2]=textread('Modos2.txt', '%f');
[Modos3]=textread('Modos3.txt', '%f');
[Modos4]=textread('Modos4.txt', '%f');

[fe]=textread('Frecuencias.txt', '%f');

%Ensamble de matriz de modos

mve(1:24,1)=Modos1;
mve(1:24,2)=Modos2;
mve(1:24,3)=Modos3;
mve(1:24,4)=Modos4;

```

ANEXO F. Código PSO.

Código PSO propuesto por Kennedy y Eberhard 1995, 2001 y programado por los autores.

```
%%
clc;clear all;format compact;

%% ----- PSO -----
disp('
disp('
disp('
disp('          HHHHHHHH          HHHH          HHHHHHHH          ')
disp('          HHH   HHH          HHHHHHHHHH          HHHHHHHHHHHH          ')
disp('          HHH   HHH          HHH   HHH          HHH          HHH          ')
disp('          HHH   HHH          HHH          HH          HHH          HHH          ')
disp('          HHH   HHH          HHHH          HHH          HHH          ')
disp('          HHHHHHHH          HHHH          HHH          HHH          ')
disp('          HHH          HHHH          HHH          HHH          ')
disp('          HHH          HH   HHH          HHH          HHH          ')
disp('          HHH          HHH          HHH          HHH          ')
disp('          HHH          HHHHHHHHHH          HHHH HHHH          ')
disp('          HHH          HHHH          HHHHHH          ')
disp('
disp('
disp('
disp('

%% DATOS DE ENTRADA

d=50;          %input('Numero de dimensiones de la funcion = ');
a=-2;          %input('Ingrese el limite inferior = ');
b=2;           %input('Ingrese el limite superior = ');
np=500;        %input('Ingrese el # de particulas = ');
rd=1000;       %input('Numero de datos para determinar la Desviación Standar
= ');
Es=1E-10;     %input('Desviación Standar admisible = ');
mite=10000;   %input('Numero maximo de iteraciones = ');

%% NUMERO DE PRUEBAS
q=0;
o=1; %input('Ingrese el # de pruebas = ');
while q<o
    q=q+1;

    %% PARAMETROS DEL ALGORITMO

p=abs((b-a))/3;
c1=2;
c2=2;
w=1;
```

```

%% CONTADOR DE TIEMPO
tic;

%% DISTRIBUCION DE LA VELOCIDAD(vi) Y LA POSICION(xi) INICIAL

[xi,vi]=dis_inic(np,d,a,b);
[xi]=check_x(xi,a,b);
[vi]=check_v(vi,p);

%% CALCULO FUNCION OBJETIVO-SELECCION DE MEJORES LOCALES Y GLOBALES

for k=1:1:np
    F(k,1)=my_fun(k,xi,d);
end

pbest=F;
coord_pbest=xi;

%% SELECCION DEL MEJOR GLOBAL
gbest=min(pbest);
[coord_gbest]=coord_cgbest(np,gbest,pbest,coord_pbest);

%% PARAMEROS PARA EL CONTROL DE LA EJECUCION DEL ALGORITMO
e=0;
des=10;

%% EJECUCION DEL ALGORITMO
while e<mite && des>Es
    e=e+1;
    x0=xi;
    v0=vi;
    vi=[];

    %% CALCULO VELOCIDAD vi
    r1=rand(np,d);
    r2=rand(np,d);
    for k=1:1:np
        vi(k,:)=w*v0(k,:)+c1*r1(k,:).*(coord_pbest(k,:)-
x0(k,:))+c2*r2(k,:).*(coord_gbest-x0(k,:));
    end

    %% CHEQUEO DE LA VELOCIDAD
    [vi]=check_v(vi,p);
    %% CALCULO DE xi A PARTIR DE vi
    xi=x0+vi;
    %% CHEQUEO DE LA POSICION
    [xi]=check_x(xi,a,b);
    %% CALCULO FUNCION OBJETIVO

    for k=1:1:np

```

```

        FM(k,1)=my_fun(k,xi,d);
    end

    %% SELECCION gbest,pbest,coord_pbest Y coord_gbest
    [gbest,pbest,coord_pbest]=select_pbest(np,pbest,FM,xi,coord_pbest,e);
    [coord_gbest]=coord_cgbest(np,gbest,pbest,coord_pbest);

    %% REDUCCION LINEAL DE w
    w=w*0.98;

    %% ALMACENAMIENTO DE DATOS PARA DETERMINAR EL CRITERIO DE PARADA
    ds(e)=gbest;

    %% CALCULO DEL CRITERIO DE PARADA
    if e>rd
        des=std(ds(1,e-rd:e));
    end
end
tiempo=toc;
mejorglobal(1,q)=gbest
mejorglobalcoord(:,q)=coord_gbest
iteraciones(1,q)=e
tiempos(1,q)=tiempo
des_est(1,q)=des;

end

```

Las fórmulas utilizadas en el código PSO son las siguientes:

- Distribución inicial de las partículas y velocidades.

```

function [xi,vi]=dis_inic(np,d,a,b)

xi=a+rand(np,d)*(b-a);
vi=a+rand(np,d)*(b-a);

```

- Chequeo limites de partículas y velocidades.

```

function [xi]=check_x(xi,a,b)

c=(a<=xi) & (xi<=b);%ones=están dentro del intervalo
m=xi.*c;

```

```

p=not(c);
t=p.*xi;%ones=estén por fuera del intervalo
e1=a*(t<=a);
e2=b*(t>=b);
xi=m+e1+e2;
function [vi]=check_v(vi,p)

c=(-p<=vi) & (vi<=p);%ones=están dentro del intervalo
m=vi.*c;
pd=not(c);
t=pd.*vi;%ones=estén por fuera del intervalo
e1=-p*(t<=-p);
e2=p*(t>=p);
vi=m+e1+e2;

```

- Funciones Objetivo.

```

function [z] = my_fun(k,xi,d)
z=0;

```

```

%FUNCION DE ROSENBROCK
for i=1:1:d-1
    z=z+(100*((xi(k,i+1)-xi(k,i))^2)^2)+((xi(k,i)-1)^2));
end

```

```

%FUNCION DE SCHWEFEL
% for i=1:1:d
%     z=z-xi(k,i)*sin(sqrt(abs(xi(k,i))));
% end

```

```

%FUNCION N-DIMENSIONAL
% for i=1:1:d
%     z=z+(xi(k,i)^4-16*xi(k,i)^2+5*xi(k,i));
% end
% z=z/d;

```

```

%FUNCION DE BROWN
% for i=1:1:d-1
%
z=z+(((xi(k,i)^2)^(xi(k,i+1)^2+1))+((xi(k,i+1)^2)^(xi(k,i)^2+1)));
% end

```

- Selección de los mejores locales, coordenadas de los mejores locales, mejor global y coordenadas del mejor global.

```
function
[gbest,pbest,coord_pbest]=select_pbest(np,pbest,FM,xi,coord_pbest,e)
for k=1:1:np
    if FM(k,1)<pbest(k,1)
        pbest(k,1)=FM(k,1);
        coord_pbest(k,:)=xi(k,:);
    else
        pbest(k,1)=pbest(k,1);
        coord_pbest(k,:)=coord_pbest(k,:);
    end
end
gbest=min(pbest);
```

```
function [coord_gbest]=coord_cgbest(np,gbest,pbest,coord_pbest)
for k=1:1:np
    if gbest==pbest(k,1)
        coord_gbest=coord_pbest(k,:);
        break
    end
end
end
```

ANEXO G. Código AG.

Código AG básico propuesto por Holland 1975, y programado por los autores.

```
clc;clear all;format compact;

%% ----- AG -----
disp(' ')
disp(' ')
disp(' ')
disp('          HHHHHHH          HHHHHH ')
disp('        HHHHHHHHH          HHHHHHHHHHH ')
disp('      HHH      HHH      HHH      HHH ')
disp('    HHH      HHH      HHH      HHH ')
disp('  HHHHHHHHHHHHHHHHHHH      HHH      HHHHHH ')
disp(' HHHHHHHHHHHHHHHHHHHHHHH      HHH      HHHHHHHH ')
disp('   HHH      HHH      HHH      HHH      HHH ')
disp('   HHH      HHH      HHH      HHH      HHH ')
disp('  HHH      HHH      HHH      HHHHHHHHHHHH ')
disp(' HHHH      HHHH      HHHHHH ')
disp(' ')
disp(' ')
disp(' ')

%% INGRESO DE DATOS GENERALES

M=100;      %NUMERO DE INDIVIDUOS !DEBE SER UN NUMERO PAR!
N=5;        %NUMERO DE DIMENSIONES
XL=-5;      %LIMITE INFERIOR
XU=5;       %LIMITE SUPERIOR
NN=5000;    %NUMERO MAXIMO DE ITERACIONES
Pc=0.8;     %PROBABILIDAD DE CRUCE
Pm=0.1;     %PROBABILIDAD DE MUTACION
a=0.2;      %VALOR PARA CRUCE BLX-A
toler=1e-30;%TOLERNCIA PARA EL CRITERIO DE PARADA
JC=1000;    %NUMERO DE DATOS PARA CALCULAR LA DESVIACION ESTANDAR
NC=1;
NI=0;

%%
while NC>NI
    NI=NI+1;

%% GENERACION DE INDIVIDUOS ALEATORIOS
X=XL+(XU-XL)*rand(M,N);

ite=1;
des=10;
tic

while des>toler
```

```

%% EVALUAR FUNCION OBJETIVO
for k=1:1:M
    F(k,1)=my_fun(k,X,N);
end

%% SELECCION POR RULETA
S=sum(F);
PROB(:,1)=F(:,1)/S;
PROB_ACUM=0.;
PROB_ACUM(1,1)=PROB(1,1);

for k=2:1:M
    PROB_ACUM(k,1)=PROB(k,1)+PROB_ACUM(k-1,1);
end

%% GENERACION DE NUMEROS ALEATORIOS
SR(:,1)=rand(M,1);

%% ASIGNACION DE NUEVOS PADRES

for k=1:1:M
    if SR(k,1)<=PROB_ACUM(1,1)
        XS(k,:)=X(1,:);
    end
    for h=2:1:M
        if PROB_ACUM(h-1,1)<SR(k,1)&& SR(k,1)<=PROB_ACUM(h,1);
            XS(k,:)=X(h,:);
        end
    end
end

%% CRUCE

CR(:,1)=rand(M/2,1);

for k=1:1:M/2
    if CR(k,1)<Pc
        for l=1:1:N
            Imin(k,l)=min(XS(2*k-1,N),XS(2*k,N));
            Imax(k,l)=max(XS(2*k-1,N),XS(2*k,N));
            r=Imax(k,l)-Imin(k,l);
            XH(k,l)=Imin(k,l)-r*a+rand()*(Imax(k,l)+r*a-Imin(k,l)-r*a);
        end
    else
        XH(k,:)=XS(2*k-1,:);
    end
end

for k=1:1:M/2
    if CR(k,1)<Pc
        for l=1:1:N

```

```

        Imin(k+M/2,1)=min(XS(2*k-1,1),XS(2*k,1));
        Imax(k+M/2,1)=max(XS(2*k-1,1),XS(2*k,1));
        r=Imax(k+M/2,1)-Imin(k+M/2,1);
        XH(k+M/2,1)=Imin(k+M/2,1)-r*a+rand()*(Imax(k+M/2,1)+r*a-
Imin(k+M/2,1)-r*a);
        end
    else
        XH(k+M/2,:)=XS(2*k,:);
    end
end

%% MUTACION

MR(:,1)=rand(M,1);

for k=1:1:M
    if MR(k,1)<Pm
        XM(k,:)=XL+(XU-XL)*rand(1,N);
    else
        XM(k,:)=XH(k,:);
    end
end

for k=1:1:M
    FM(k,1)=my_fun(k,XM,N);
end

%% ELITISMO

Fmax=max(F);
Fmin=min(FM);

for k=1:1:M
    if F(k,1)==Fmax;
        Xmax=X(k,:);
    end
end

for k=1:1:M
    if FM(k,1)==Fmin;
        X2(k,:)=Xmax(1,:);
    else
        X2(k,:)=XM(k,:);
    end
end

X=X2;

for k=1:1:M
    for h=1:1:N
        if X(k,h)>XU

```

```

        X(k,h)=XL+rand()*(XU-XL);
    end
    if X(k,h)<XL
        X(k,h)=XL+rand()*(XU-XL);
    end
end
end

Xopt=Xmax;
Fopt(1,1)=-my_fun(1,Xopt,N)+100000;

%% ALMACENAR LOS MEJORES DE CADA ITERACION
Fmej(1,ite)=Fopt;

%% CALCULO DEL CRITERIO DE PARADA
if ite>JC
    des=std(Fmej(1,ite-JC:ite));
end

if ite>NN
    break
end

ite=ite+1;
end

Coordenadas(NI,:)=Xopt
Valor_Funcion_Objetivo(NI,1)=Fopt
Numero_de_iteraciones(NI,1)=ite
Tiempo_de_Ejecucion(NI,1)=toc
end

```

- Funciones Objetivo.

```

function [z] = my_fun(k,X,N)
z=0;

%FUNCION DE ROSENBROCK
for i=1:1:N-1
    z=z+(100*((X(k,i+1)-X(k,i)^2)^2)+(X(k,i)-1)^2));
end

%FUNCION DE SCHWEFEL
for i=1:1:N
    z=z-X(k,i)*sin(sqrt(abs(X(k,i))));
end

%FUNCION N-DIMENSIONAL
for i=1:1:N

```

```

        z=z+(X(k,i)^4-16*X(k,i)^2+5*X(k,i));
    end
    z=z/N;

%FUNCION DE BROWN
% for i=1:1:N-1
%     z=z+(((X(k,i)^2)^(X(k,i+1)^2+1))+((X(k,i+1)^2)^(X(k,i)^2+1)));
% end

z=100000-z;

```

ANEXO H. Validación PSO y AG. Funciones de prueba.

VALIDACION PSO Y AG CON FUNCIONES DE PRUEBA

Validación del PSO

Con el fin de evaluar el desempeño del algoritmo se realizaron 5 pruebas, en las cuales se usaron 100 partículas, un número máximo de iteraciones de 1000 y una desviación estándar menor a 1×10^{-30} en los últimos 100 resultados de las funciones objetivo. Los parámetros del algoritmo usados son: $c1=2$, $c2=2$ y $w=1$ donde w es ajustado dinámicamente durante la optimización. Estas funciones de prueba fueron extraídas de la literatura internacional (Georgieva A., Jordanov I., European Journal of Operational Research. 2009).

Función de Brown 20D

$$F(x) = \sum_{i=1}^{19} (X_i^2)^{(X_{i+1}^2+1)} + (X_{i+1}^2)^{(X_i^2+1)},$$

$$\text{Límites } [-10,10] \quad , \quad F_x(0, \dots, 0) = 0$$

Función de Brown 20D			
	Prueba # 1	Prueba # 2	Prueba # 3
Fobjetivo	0.000	0.000	0.000
# Iteraciones	374	291	317
Tiempo [seg]	0.7926	0.62	0.68
Coordenadas Teóricas	Coordenadas Experimentales	Coordenadas Experimentales	Coordenadas Experimentales
0	0.00000	0.00000	0.00000
0	0.00000	0.00000	0.00000
0	0.00000	0.00000	0.00000
0	0.00000	0.00000	0.00000
0	0.00000	0.00000	0.00000

0	0.00000	0.00000	0.00000
0	0.00000	0.00000	0.00000
0	0.00000	0.00000	0.00000
0	0.00000	0.00000	0.00000
0	0.00000	0.00000	0.00000
0	0.00000	0.00000	0.00000
0	0.00000	0.00000	0.00000
0	0.00000	0.00000	0.00000
0	0.00000	0.00000	0.00000
0	0.00000	0.00000	0.00000
0	0.00000	0.00000	0.00000
0	0.00000	0.00000	0.00000
0	0.00000	0.00000	0.00000
0	0.00000	0.00000	0.00000
0	0.00000	0.00000	0.00000
0	0.00000	0.00000	0.00000
0	0.00000	-0.00001	0.00000

Función Schwefel 5D

$$F(x) = \sum_{i=1}^N -X_i \sin(\sqrt{|X_i|}),$$

Limites [-500,500] , $F_x(420.968, \dots, 420.968) = -2094.914$

Función Schwefel 5D			
	Prueba # 1	Prueba # 2	Prueba # 3
Fobjetivo	-2094.900	-2094.900	-2094.900
# Iteraciones	1079	1083	1087
Tiempo [seg]	1.48	1.489	1.498
Coordenadas Teóricas	Coordenadas Experimentales	Coordenadas Experimentales	Coordenadas Experimentales
420.968	420.9687	420.9687	420.969
420.968	420.9687	420.9687	420.969
420.968	420.9687	420.9687	420.969
420.968	420.9687	420.9687	420.969
420.968	420.9687	420.9687	420.969

N-dimensional Test Function 5D

$$F(x) = \frac{1}{N} \sum_{i=1}^N (X_i^4 - 16X_i^2 + 5X_i),$$

$$\text{Limites } [-5,5] \quad , \quad F_x(-2.9, \dots, -2.9) = -78.332$$

N-dimensional Test Function 5D			
	Prueba # 1	Prueba # 2	Prueba # 3
Fobjetivo	-78.332	-78.332	-78.332
# Iteraciones	1077	1085	1075
Tiempo [seg]	1.049	1.063	1.042
Coordenadas Teóricas	Coordenadas Experimentales	Coordenadas Experimentales	Coordenadas Experimentales
-2.90	-2.904	-2.904	-2.904
-2.90	-2.904	-2.904	-2.904
-2.90	-2.904	-2.904	-2.904
-2.90	-2.904	-2.904	-2.904
-2.90	-2.904	-2.904	-2.904

Función de Rosenbrock 2D y 50D

$$F(x) = \sum_{i=1}^{N-1} [100(X_{i+1} - X_i^2)^2 + (X_i - 1)^2],$$

$$\text{Limites } [-2,2] \quad , \quad F_{x_{opt}}(1, \dots, 1) = 0$$

Función de Rosenbrock 2D			
	Prueba # 1	Prueba # 2	Prueba # 3
Fobjetivo	0.000	0.000	0.000
# Iteraciones	100	100	100
Tiempo [seg]	0.1104	0.111	0.1187
Coordenadas Teóricas	Coordenadas Experimentales	Coordenadas Experimentales	Coordenadas Experimentales
1	0.9999	1.0000	0.9991
1	0.9999	1.0000	0.9981

Función de Rosenbrock 50D			
	Prueba # 1	Prueba # 2	Prueba # 3
Fobjetivo	43.151	44.082	43.776
# Iteraciones	10000	10000	10000
Tiempo [seg]	150.9	151.82	152.73
Coordenadas Teóricas	Coordenadas Experimentales	Coordenadas Experimentales	Coordenadas Experimentales
1	0.96050	-0.98690	0.93970
1	0.92360	0.98430	0.88530
1	0.85540	0.97530	0.79190
1	0.73930	0.95680	0.63150
1	0.54990	0.92470	0.40510
1	0.31500	0.86100	0.16880
1	0.10620	0.75030	0.03130
1	0.01650	0.57080	0.01170
1	0.01060	0.33180	0.01020
1	0.01030	0.12010	0.01010
1	0.00950	0.02370	0.01020
1	0.01000	0.01040	0.01030
1	0.00980	0.01000	0.01040
1	0.01020	0.01030	0.00860
1	0.01030	0.01140	0.01030
1	0.01040	0.00990	0.01090
1	0.00960	0.01000	0.01010
1	0.01060	0.01080	0.00880
1	0.00960	0.00960	0.01580
1	0.01030	0.00910	0.01060
1	0.01020	0.00970	0.01110
1	0.01020	0.00990	0.01080
1	0.00970	0.00710	0.01020
1	0.01010	0.00720	0.01100
1	0.01120	0.00910	0.01070
1	0.01040	0.01040	0.01050
1	0.01020	0.01080	0.01170
1	0.01010	0.01040	0.01020
1	0.00970	0.00970	0.00960
1	0.01020	0.01010	0.01000
1	0.01010	0.00980	0.01050
1	0.01000	0.00930	0.01120
1	0.01010	0.01010	0.01400

1	0.01000	0.00720	0.01010
1	0.01030	0.01090	0.01000
1	0.01020	0.01080	0.01070
1	0.01020	0.01130	0.01250
1	0.00980	0.00980	0.01320
1	0.01010	0.01120	0.01010
1	0.01060	0.00970	0.01000
1	0.01030	0.00970	0.01060
1	0.01030	0.00960	0.01170
1	0.01110	0.01000	0.01000
1	0.00970	0.01100	0.00980
1	0.01030	0.00960	0.01030
1	0.01030	0.01030	0.01030
1	0.01050	0.00970	0.00990
1	0.01050	0.00860	0.01120
1	0.01000	0.00980	0.01010
1	-0.00020	-0.00120	0.00080

Validación del AG

Con el fin de evaluar el desempeño del algoritmo se realizaron 5 pruebas, en las cuales se usaron 100 individuos, una probabilidad de mutación del 2%, una probabilidad de cruce del 80% y el criterio de para fue el número de generaciones (iteraciones) y una desviación estándar menor a 1×10^{-30} en los últimos 1000 resultados de las funciones objetivo. Estas funciones de prueba fueron extraídas de la literatura internacional (Georgieva A., Jordanov I., European Journal of Operational Research. 2009).

Función de Brown 20D

$$F(x) = \sum_{i=1}^{19} (X_i^2)^{(X_{i+1}^2+1)} + (X_{i+1}^2)^{(X_i^2+1)},$$

$$\text{Limites } [-10,10] \quad , \quad F_x(0, \dots, 0) = 0$$

Función Schwefel 5D			
	Prueba # 1	Prueba # 2	Prueba # 3
Fobjetivo	-2.0948E+03	-2.0949E+03	-2.0949E+03
# Iteraciones	10001	10001	20001
Tiempo [seg]	47.59	47.75	101.04
Coordenadas Teóricas	Coordenadas Experimentales	Coordenadas Experimentales	Coordenadas Experimentales
420.968	420.304	421.028	420.928
420.968	420.440	421.028	420.928
420.968	420.638	421.028	420.928
420.968	420.744	421.028	420.928
420.968	420.354	421.028	420.928

N-dimensional Test Function 5D

$$F(x) = \frac{1}{N} \sum_{i=1}^N (X_i^4 - 16X_i^2 + 5X_i),$$

Limites $[-5,5]$, $F_x(-2.9, \dots, -2.9) = -78.332$

N-dimensional Test Function 5D			
	Prueba # 1	Prueba # 2	Prueba # 3
Fobjetivo	-78.332	-78.332	-78.332
# Iteraciones	238	1170	5001
Tiempo [seg]	0.92	4.28	18.21
Coordenadas Teóricas	Coordenadas Experimentales	Coordenadas Experimentales	Coordenadas Experimentales
-2.90	-2.902	-2.903	-2.904
-2.90	-2.902	-2.903	-2.904
-2.90	-2.902	-2.903	-2.904
-2.90	-2.902	-2.903	-2.904
-2.90	-2.902	-2.903	-2.904

Función de Rosenbrock 2D y 50D

$$F(x) = \sum_{i=1}^{N-1} [100(X_{i+1} - X_i^2)^2 + (X_i - 1)^2],$$

Limites $[-2,2]$, $F_{x_{opt}}(1, \dots, 1) = 0$


```

%% DATOS DE ENTRADA

d=24; %input('Numero de dimensiones de la funcion = ');
a=0; %input('Ingrese el limite inferior = ');
b=1; %input('Ingrese el limite superior = ');
np=input('Ingrese el # de particulas = ');
rd=20;%input('Numero de datos para determinar la Desviación Standar = ');
Es=1E-10; %input('Desviación Standar admisible = ');
mite=input('Numero maximo de iteraciones = ');

%% NUMERO DE PRUEBAS
q=0;
o=input('Ingrese el # de pruebas = ');
while q<o
    q=q+1;

    %% PARAMETROS DEL ALGORITMO

    p=abs((b-a))/3;
    c1=2;
    c2=2;
    w=1;
    %% CONTADOR DE TIEMPO
    tic;

    %% DISTRIBUCION DE LA VELOCIDAD(vi) Y LA POSICION(xi) INICIAL

    [xi,vi]=dis_inic(np,d,a,b);
    [xi]=check_x(xi,a,b);
    [vi]=check_v(vi,p);

    %% CALCULO FUNCION OBJETIVO-SELECCION DE MEJORES LOCALES Y GLOBALES

    ee(1:24,1)=21.54e9;

    for i=1:1:np

        % Vector de modulos de elasticidad
        E=ee.*xi(:,i); %Vector de E con daños

        [fe mve]=Portico(E);
        [fobj]=fobj1(fa,fe,mva,mve);

        fob(1,i)=fobj;
    end
    pbest=fob;
    coord_pbest=xi;

    %% SELECCION DEL MEJOR GLOBAL

```

```

gbest=min(pbest);
[coord_gbest]=coord_cgbest(np,gbest,pbest,coord_pbest);

%% PARAMEROS PARA EL CONTROL DE LA EJECUCION DEL ALGORITMO
e=0;
des=10;

%% EJECUCION DEL ALGORITMO
while e<mite && des>Es
    e=e+1;
    x0=xi;
    v0=vi;
    vi=[];

    %% PORCENTAJE DE AVANCE
    %por=e*100/mite;
    %disp(['avance de ', num2str(por),'%']);
    %% CALCULO VELOCIDAD vi
    r1=rand(d,np);
    r2=rand(d,np);
    for k=1:1:np
        vi(:,k)=w*v0(:,k)+c1*r1(:,k).*(coord_pbest(:,k)-
x0(:,k))+c2*r2(:,k).*(coord_gbest-x0(:,k));
    end

    %% CHEQUEO DE LA VELOCIDAD
    [vi]=check_v(vi,p);
    %% CALCULO DE xi A PARTIR DE vi
    xi=x0+vi;
    %% CHEQUEO DE LA POSICION
    [xi]=check_x(xi,a,b);
    %% CALCULO FUNCION OBJETIVO
    for i=1:1:np
        E=ee.*xi(:,i);
        [fe mve]=Portico(E);
        [fobj]=fobj1(fa,fe,mva,mve);
        fob(e+1,i)=fobj;
    end

    %% SELECCION gbest,pbest,coord_pbest Y coord_gbest

[gbest,pbest,coord_pbest]=select_pbest(np,pbest,fob,xi,coord_pbest,e);
[coord_gbest]=coord_cgbest(np,gbest,pbest,coord_pbest);

%% REDUCCION LINEAL DE w
w=w*0.98;

%% ALMACENAMIENTO DE DATOS PARA DETERMINAR EL CRITERIO DE PARADA
ds(e)=gbest;

%% CALCULO DEL CRITERIO DE PARADA
if e>rd
    des=std(ds(1,e-rd:e));

```

```

        end
    end
    tiempo=toc;
    mejorglobal(1,q)=gbest;
    mejorglobalcoord(:,q)=coord_gbest;
    iteraciones(1,q)=e;
    tiempos(1,q)=tiempo;
    des_est(1,q)=des;
    if gbest<1e-3
        disp(['Prueba # ', num2str(q),' listo ']);
    else
        disp(['Prueba # ', num2str(q),' no valida ']);
        disp(['Valor mejor Global = ', num2str(gbest), ' No admisible
    ]]);
        %q=q-1;
        o=o+1;
    end

    fob=[];
end

```

- Portico Asimetrico Longitudes – PSO

```

%%
clc;clear all;format compact;

%% DAÑO INICIAL
disp(' ')
q=input('Numero de elementos dañados = ');
disp(' ')
xa=[];
xa(1:24,1)=1;

for i=1:1:q
    n=input('Nombre del elemento dañado = ');
    xa(n)=input('Daño (valor entre 0 y 1) = ');
    disp(' ')
end
disp(' ')

%%CALCULO DE FRECUENCIAS MODOS ANALITICOS

[fa mva]=Portico_inicial(xa);

%% ----- PSO -----

```

```

disp(' ')
disp(' ')
disp(' ')
disp('          HHHHHHHHH          HHHH          HHHHHHHH          ')
disp('          HHH   HHH          HHHHHHHHHHH          HHHHHHHHHHHH          ')
disp('          HHH   HHH          HHH   HHH          HHH          HHH          ')
disp('          HHH   HHH          HHH          HH          HHH          HHH          ')
disp('          HHH   HHH          HHHH          HHH          HHH          ')
disp('          HHHHHHHHH          HHHH          HHH          HHH          ')
disp('          HHH          HHHH          HHH          HHH          ')
disp('          HHH          HH          HHH          HHH          HHH          ')
disp('          HHH          HHH          HHH          HHH          HHH          ')
disp('          HHH          HHHHHHHHHHH          HHHHH HHHHH          ')
disp('          HHH          HHHH          HHHHHHH          ')
disp(' ')
disp(' ')
disp(' ')

```

```

%% DATOS DE ENTRADA

```

```

d=24;%input('Numero de dimensiones de la funcion = ');
a=0;%input('Ingrese el limite inferior = ');
b=1;%input('Ingrese el limite superior = ');
np=input('Ingrese el # de particulas = ');
rd=20;%input('Numero de datos para determinar la Desviación Standar = ');
Es=1E-10;%input('Desviación Standar admisible = ');
mite=input('Numero maximo de iteraciones = ');

```

```

%% NUMERO DE PRUEBAS

```

```

q=0;
o=input('Ingrese el # de pruebas = ');
while q<o
    q=q+1;

```

```

    %% PARAMETROS DEL ALGORITMO

```

```

p=abs((b-a))/3;
c1=2;
c2=2;
w=1;
    %% CONTADOR DE TIEMPO
    tic;

```

```

%% DISTRIBUCION DE LA VELOCIDAD(vi) Y LA POSICION(xi) INICIAL

```

```

[xi,vi]=dis_inic(np,d,a,b);
[xi]=check_x(xi,a,b);
[vi]=check_v(vi,p);

```

```

%% CALCULO FUNCION OBJETIVO-SELECCION DE MEJORES LOCALES Y GLOBALES

```

```

ee(1:24,1)=21.54e9;

for i=1:1:np

    % Vector de modulos de elasticidad
    E=ee.*xi(:,i);           %Vector de E con daños

    [fe mve]=Portico(E);
    [fobj]=fobj1(fa,fe,mva,mve);

    fob(1,i)=fobj;
end
pbest=fob;
coord_pbest=xi;

%% SELECCION DEL MEJOR GLOBAL
gbest=min(pbest);
[coord_gbest]=coord_cgbest(np,gbest,pbest,coord_pbest);

%% PARAMEROS PARA EL CONTROL DE LA EJECUCION DEL ALGORITMO
e=0;
des=10;

%% EJECUCION DEL ALGORITMO
while e<mite && des>Es
    e=e+1;
    x0=xi;
    v0=vi;
    vi=[];

    %% PORCENTAJE DE AVANCE
    %por=e*100/mite;
    %disp(['avance de ', num2str(por),'%']);
    %% CALCULO VELOCIDAD vi
    r1=rand(d,np);
    r2=rand(d,np);
    for k=1:1:np
        vi(:,k)=w*v0(:,k)+c1*r1(:,k).*(coord_pbest(:,k)-
x0(:,k))+c2*r2(:,k).*(coord_gbest-x0(:,k));
    end

    %% CHEQUEO DE LA VELOCIDAD
    [vi]=check_v(vi,p);
    %% CALCULO DE xi A PARTIR DE vi
    xi=x0+vi;
    %% CHEQUEO DE LA POSICION
    [xi]=check_x(xi,a,b);
    %% CALCULO FUNCION OBJETIVO
    for i=1:1:np
        E=ee.*xi(:,i);
        [fe mve]=Portico(E);
        [fobj]=fobj1(fa,fe,mva,mve);
        fob(e+1,i)=fobj;
    end
end

```

```

end

%% SELECCION gbest,pbest,coord_pbest Y coord_gbest

[gbest,pbest,coord_pbest]=select_pbest(np,pbest,fob,xi,coord_pbest,e);
[coord_gbest]=coord_cgbest(np,gbest,pbest,coord_pbest);

%% REDUCCION LINEAL DE w
w=w*0.98;

%% ALMACENAMIENTO DE DATOS PARA DETERMINAR EL CRITERIO DE PARADA
ds(e)=gbest;

%% CALCULO DEL CRITERIO DE PARADA
if e>rd
    des=std(ds(1,e-rd:e));
end
end
tiempo=toc;
mejorglobal(1,q)=gbest;
mejorglobalcoord(:,q)=coord_gbest;
iteraciones(1,q)=e;
tiempos(1,q)=tiempo;
des_est(1,q)=des;
if gbest<1e-3
    disp(['Prueba # ', num2str(q), ' listo ']);
else
    disp(['Prueba # ', num2str(q), ' a repetir ']);
    disp(['Valor mejor Global = ', num2str(gbest), ' No admisible
']);
    %q=q-1;
    o=o+1;
end

fob=[];
end

```

- Portico Simetrico Total - Ruido – PSO

```

%%
clc;clear all;format compact;

%% DAÑO INICIAL
disp(' ')
q=input('Numero de elementos dañados = ');
disp(' ')
xa=[];
xa(1:24,1)=1;

for i=1:1:q

```

```

n=input('Nombre del elemento dañado = ');
xa(n)=input('Daño (valor entre 0 y 1) = ');
disp(' ')
end
disp(' ')

%CALCULO DE FRECUENCIAS MODOS ANALITICOS

[fa mva]=Portico_inicial(xa);

%% CALCULO DE FRECUENCIAS Y MODOS CON RUIDO

errorfa=(-0.02+0.04*rand(4,1)).*fa; %Error en medida de +- un 2%
fa=fa+errorfa; %Frecuencias con error
errormva=(-0.05+0.1*rand(24,4)).*mva; %Error en medida de +- un 5%
mva=mva+errormva; %Modos con error
%% ----- PSO -----

disp(' ')
disp(' ')
disp(' ')
disp('          HHHHHHHH          HHHH          HHHHHHHH          ')
disp('          HHH   HHH          HHHHHHHHHHH          HHHHHHHHHHHH          ')
disp('          HHH   HHH          HHH   HHH          HHH          HHH          ')
disp('          HHH   HHH          HHH          HH          HHH          HHH          ')
disp('          HHH   HHH          HHHH          HHH          HHH          ')
disp('          HHHHHHHHH          HHHH          HHH          HHH          ')
disp('          HHH          HHHH          HHH          HHH          ')
disp('          HHH          HH   HHH          HHH          HHH          ')
disp('          HHH          HHH          HHH          HHH          HHH          ')
disp('          HHH          HHHHHHHHHHH          HHHHH HHHHH          ')
disp('          HHH          HHHH          HHHHHHH          ')
disp(' ')
disp(' ')
disp(' ')

%% DATOS DE ENTRADA

d=24;%input('Numero de dimensiones de la funcion = ');
a=0;%input('Ingrese el limite inferior = ');
b=1;%input('Ingrese el limite superior = ');
np=input('Ingrese el # de particulas = ');
rd=20;%input('Numero de datos para determinar la Desviación Standar = ');
Es=1E-10;%input('Desviación Standar admisible = ');
mite=input('Numero maximo de iteraciones = ');

%% NUMERO DE PRUEBAS
q=0;
o=input('Ingrese el # de pruebas = ');
while q<o
    q=q+1;

```

```

%% PARAMETROS DEL ALGORITMO

p=abs((b-a))/3;
c1=2;
c2=2;
w=1;
%% CONTADOR DE TIEMPO
tic;

%% DISTRIBUCION DE LA VELOCIDAD(vi) Y LA POSICION(xi) INICIAL

[xi,vi]=dis_inic(np,d,a,b);
[xi]=check_x(xi,a,b);
[vi]=check_v(vi,p);

%% CALCULO FUNCION OBJETIVO-SELECCION DE MEJORES LOCALES Y GLOBALES

ee(1:24,1)=21.54e9;

for i=1:1:np

    % Vector de modulos de elasticidad
    E=ee.*xi(:,i);          %Vector de E con daños

    [fe mve]=Portico(E);
    [fobj]=fobj1(fa,fe,mva,mve);

    fob(1,i)=fobj;
end
pbest=fob;
coord_pbest=xi;

%% SELECCION DEL MEJOR GLOBAL
gbest=min(pbest);
[coord_gbest]=coord_cgbest(np,gbest,pbest,coord_pbest);

%% PARAMEROS PARA EL CONTROL DE LA EJECUCION DEL ALGORITMO
e=0;
des=10;

%% EJECUCION DEL ALGORITMO
while e<mite && des>Es
    e=e+1;
    x0=xi;
    v0=vi;
    vi=[];

    %% PORCENTAJE DE AVANCE
    %por=e*100/mite;
    %disp(['avance de ', num2str(por),'%']);
    %% CALCULO VELOCIDAD vi

```

```

r1=rand(d,np);
r2=rand(d,np);
for k=1:1:np
    vi(:,k)=w*v0(:,k)+c1*r1(:,k).*(coord_pbest(:,k)-
x0(:,k))+c2*r2(:,k).*(coord_gbest-x0(:,k));
end

%% CHEQUEO DE LA VELOCIDAD
[vi]=check_v(vi,p);
%% CALCULO DE xi A PARTIR DE vi
xi=x0+vi;
%% CHEQUEO DE LA POSICION
[xi]=check_x(xi,a,b);
%% CALCULO FUNCION OBJETIVO
for i=1:1:np
    E=ee.*xi(:,i);
    [fe mve]=Portico(E);
    [fobj]=fobj1(fa,fe,mva,mve);
    fob(e+1,i)=fobj;
end

%% SELECCION gbest,pbest,coord_pbest Y coord_gbest
[gbest,pbest,coord_pbest]=select_pbest(np,pbest,fob,xi,coord_pbest,e);
[coord_gbest]=coord_cgbest(np,gbest,pbest,coord_pbest);

%% REDUCCION LINEAL DE w
w=w*0.98;

%% ALMACENAMIENTO DE DATOS PARA DETERMINAR EL CRITERIO DE PARADA
ds(e)=gbest;

%% CALCULO DEL CRITERIO DE PARADA
if e>rd
    des=std(ds(1,e-rd:e));
end
end
tiempo=toc;
mejorglobal(1,q)=gbest;
mejorglobalcoord(:,q)=coord_gbest;
iteraciones(1,q)=e;
tiempos(1,q)=tiempo;
des_est(1,q)=des;
if gbest<1e-3
    disp(['Prueba # ', num2str(q), ' listo ']);
else
    disp(['Prueba # ', num2str(q), ' no valida ']);
    disp(['Valor mejor Global = ', num2str(gbest), ' No admisible
']);
    %q=q-1;
    o=o+1;
end

```

```
fob=[];  
end
```

La función objetivo utilizada en para la optimización con PSO es:

```
function [fobj]=fobj4(fa,fe,mva,mve)  
f=abs(fe-fa);  
diff=0;  
for k=1:1:12  
    diff=diff+f(k);  
end  
mv=abs(mve-mva);  
difmv=0;  
for k=1:1:4  
    for l=1:1:24  
        difmv=difmv+mv(l,k);  
    end  
end  
fobj=diff+difmv;
```

ANEXO J. Código pórticos (OPENSEES)- AG.

Codigos Porticos (OPENSEES) – PSO.

- Portico Simetrico Total – AG

```
%%
clc;clear all;format compact;

%% DAÑO INICIAL
disp(' ')
q=input('Numero de elementos dañados = ');
disp(' ')
xa=[];
xa(1,1:24)=1;

for i=1:1:q
    n=input('Nombre del elemento dañado = ');
    xa(n)=input('Daño (valor entre 0 y 1) = ');
    disp(' ')
end
disp(' ')

%CALCULO DE FRECUENCIAS MODOS ANALITICOS

[fa mva]=Portico_inicial(xa);

%% ----- AG -----
disp(' ')
disp(' ')
disp(' ')
disp('          HHHHHHHH          HHHHHHHH          ')
disp('          HHHHHHHHHH          HHHHHHHHHHHH          ')
disp('          HHH      HHH          HHH          HHH          ')
disp('          HHH      HHH          HHH          HHH          ')
disp('          HHH      HHH          HHH          HHH          ')
disp('          HHHHHHHHHHHHHHHHHHH          HHHHHHH          ')
disp('          HHHHHHHHHHHHHHHHHHH          HHHHHHHHH          ')
disp('          HHH          HHH          HHH          HHH          ')
disp('          HHH          HHH          HHH          HHH          ')
disp('          HHH          HHH          HHHHHHHHHHHHH          ')
disp('          HHHH          HHHH          HHHHHHH          ')
disp(' ')
disp(' ')
disp(' ')

%% INGRESO DE DATOS GENERALES

d=24;%input('Numero de dimensiones de la funcion = ');
a=0;%input('Ingrese el limite inferior = ');
b=1;%input('Ingrese el limite superior = ');
```

```

np=input('Ingrese el # de individuos (debe se un numero par) = ');
rd=50;%input('Numero de datos para determinar la Desviación Standar = ');
Es=1E-30;%input('Desviación Standar admisible = ');
mite=input('Numero maximo de iteraciones = ');

%% PARAMETROS DEL ALGORITMO

Pc=0.8;  %PROBABILIDAD DE CRUCE
Pm=0.2;  %PROBABILIDAD DE MUTACION
ac=0.2;  %VALOR PARA CRUCE BLX-A

%% NUMERO DE PRUEBAS
q=0;
o=input('Ingrese el # de pruebas = ');
while q<o
    q=q+1;

%% GENERACION DE INDIVIDUOS ALEATORIOS

[xi]=dis_inic(np,d,a,b);
[xi]=check_x(xi,a,b);

e=0;
des=10;
tic

%% EJECUCION DEL ALGORITMO
while e<mite && des>Es
    e=e+1;
%% CALCULO FUNCION OBJETIVO

ee(1,1:24)=21.54e9;

for i=1:1:np
    E=ee.*xi(i,:);           %Vector de E con daños
    [fe mve]=Portico(E);
    [fobj]=fobj5(fa,fe,mva,mve);

    fob(i,1)=fobj;
end

%% SELECCION POR RULETA
S=sum(fob);

PROB(:,1)=fob(:,1)/S;
PROB_ACUM=0.;
PROB_ACUM(1,1)=PROB(1,1);

```

```

for k=2:1:np
    PROB_ACUM(k,1)=PROB(k,1)+PROB_ACUM(k-1,1);
end

%% GENERACION DE NUMEROS ALEATORIOS
SR(:,1)=rand(np,1);

%% ASIGNACION DE NUEVOS PADRES

for k=1:1:np
    if SR(k,1)<=PROB_ACUM(1,1)
        XS(k,:)=xi(1,:);
    end
    for h=2:1:np
        if PROB_ACUM(h-1,1)<SR(k,1)&& SR(k,1)<=PROB_ACUM(h,1);
            XS(k,:)=xi(h,:);
        end
    end
end

%% CRUCE

CR(:,1)=rand(np/2,1);

for k=1:1:np/2
    if CR(k,1)<Pc
        for l=1:1:d
            Imin(k,l)=min(XS(2*k-1,d),XS(2*k,d));
            Imax(k,l)=max(XS(2*k-1,d),XS(2*k,d));
            r=Imax(k,l)-Imin(k,l);
            XH(k,l)=Imin(k,l)-r*ac+rand()*(Imax(k,l)+r*ac-Imin(k,l)-
r*ac);
        end
    else
        XH(k,:)=XS(2*k-1,:);
    end
end

for k=1:1:np/2
    if CR(k,1)<Pc
        for l=1:1:d
            Imin(k+np/2,l)=min(XS(2*k-1,d),XS(2*k,d));
            Imax(k+np/2,l)=max(XS(2*k-1,d),XS(2*k,d));
            r=Imax(k+np/2,l)-Imin(k+np/2,l);
            XH(k+np/2,l)=Imin(k+np/2,l)-r*ac+rand()*(Imax(k+np/2,l)+r*ac-
Imin(k+np/2,l)-r*ac);
        end
    else
        XH(k+np/2,:)=XS(2*k,:);
    end
end

%% MUTACION

```

```

MR(:,1)=rand(np,1);

for k=1:1:np
    if MR(k,1)<Pm
        XM(k,:)=a+(b-a)*rand(1,d);
    else
        XM(k,:)=XH(k,:);
    end
end

for k=1:1:np
    for h=1:1:d
        if XM(k,h)>b
            XM(k,h)=b; %a+rand()*(b-a);
        end
        if XM(k,h)<a
            XM(k,h)=a+rand()*(b-a);
        end
    end
end

for i=1:1:np
    E=ee.*XM(i,:); %Vector de E con daños
    [fe mve]=Portico(E);
    [fobj]=fobj5(fa,fe,mva,mve);

    FM(i,1)=fobj;
end

%% ELITISMO

Fmax=max(fob);
Fmin=min(FM);

for k=1:1:np
    if fob(k,1)==Fmax;
        Xmax=xi(k,:);
    end
end

for k=1:1:np
    if FM(k,1)==Fmin;
        X2(k,:)=Xmax(1,:);
    else
        X2(k,:)=XM(k,:);
    end
end

%% CAMBIO DE GENERACION

```

```

xi=X2;

[xi]=check_x(xi,a,b);

%% ALMACENAR LOS MEJORES DE CADA ITERACION
Xopt=Xmax;
E=ee.*Xopt;           %Vector de E con daños
[fe mve]=Portico(E);
[fobj]=fobj5(fa,fe,mva,mve);
Fobt=fobj;

Fmej(1,e)=Fobt;

%% CALCULO DEL CRITERIO DE PARADA
if e>rd
    des=std(Fmej(1,e-rd:e));
end

    end
    tiempo=toc;

Coordenadas(q,:)=Xopt;
Valor_Funcion_Objetivo(q,1)=Fobt;
Numero_de_iteraciones(q,1)=e;
Tiempo_de_Ejecucion(q,1)=tiempo;

if Fobt>19.9
    disp(['Prueba # ', num2str(q),' listo ']);
else
    disp(['Prueba # ', num2str(q),' a repetir ']);
    disp(['Valor Funcion Objetivo = ', num2str(Fobt), ' No admisible
    ']);
    %q=q-1;
    o=o+1;
end
end

```

- Portico Asimetrico – AG

```

%%
clc;clear all;format compact;

%% DAÑO INICIAL
disp(' ')
q=input('Numero de elementos dañados = ');
disp(' ')
xa=[];
xa(1,1:24)=1;

```



```

%% GENERACION DE INDIVIDUOS ALEATORIOS

[xi]=dis_inic(np,d,a,b);
[xi]=check_x(xi,a,b);

e=0;
des=10;
tic

%% EJECUCION DEL ALGORITMO
while e<mite && des>Es
    e=e+1;
%% CALCULO FUNCION OBJETIVO

ee(1,1:24)=21.54e9;

for i=1:1:np
    E=ee.*xi(i,:);           %Vector de E con daños
    [fe mve]=Portico(E);
    [fobj]=fobj5(fa,fe,mva,mve);

    fob(i,1)=fobj;
end

%% SELECCION POR RULETA
S=sum(fob);

PROB(:,1)=fob(:,1)/S;
PROB_ACUM=0.;
PROB_ACUM(1,1)=PROB(1,1);

for k=2:1:np
    PROB_ACUM(k,1)=PROB(k,1)+PROB_ACUM(k-1,1);
end

%% GENERACION DE NUMEROS ALEATORIOS
SR(:,1)=rand(np,1);

%% ASIGNACION DE NUEVOS PADRES

for k=1:1:np
    if SR(k,1)<=PROB_ACUM(1,1)
        XS(k,:)=xi(1,:);
    end
    for h=2:1:np
        if PROB_ACUM(h-1,1)<SR(k,1)&& SR(k,1)<=PROB_ACUM(h,1);
            XS(k,:)=xi(h,:);
        end
    end
end

```

```

    end
end

%% CRUCE

CR(:,1)=rand(np/2,1);

for k=1:1:np/2
    if CR(k,1)<Pc
        for l=1:1:d
            Imin(k,l)=min(XS(2*k-1,d),XS(2*k,d));
            Imax(k,l)=max(XS(2*k-1,d),XS(2*k,d));
            r=Imax(k,l)-Imin(k,l);
            XH(k,l)=Imin(k,l)-r*ac+rand()*(Imax(k,l)+r*ac-Imin(k,l)-
r*ac);
        end
    else
        XH(k,:)=XS(2*k-1,:);
    end
end

for k=1:1:np/2
    if CR(k,1)<Pc
        for l=1:1:d
            Imin(k+np/2,l)=min(XS(2*k-1,d),XS(2*k,d));
            Imax(k+np/2,l)=max(XS(2*k-1,d),XS(2*k,d));
            r=Imax(k+np/2,l)-Imin(k+np/2,l);
            XH(k+np/2,l)=Imin(k+np/2,l)-r*ac+rand()*(Imax(k+np/2,l)+r*ac-
Imin(k+np/2,l)-r*ac);
        end
    else
        XH(k+np/2,:)=XS(2*k,:);
    end
end

%% MUTACION

MR(:,1)=rand(np,1);

for k=1:1:np
    if MR(k,1)<Pm
        XM(k,:)=a+(b-a)*rand(1,d);
    else
        XM(k,:)=XH(k,:);
    end
end

for k=1:1:np
    for h=1:1:d
        if XM(k,h)>b

```

```

        XM(k,h)=b; %a+rand()*(b-a);
    end
    if XM(k,h)<a
        XM(k,h)=a+rand()*(b-a);
    end
end
end

for i=1:1:np
    E=ee.*XM(i,:); %Vector de E con daños
    [fe mve]=Portico(E);
    [fobj]=fobj5(fa,fe,mva,mve);

    FM(i,1)=fobj;
end

%% ELITISMO

Fmax=max(fob);
Fmin=min(FM);

for k=1:1:np
    if fob(k,1)==Fmax;
        Xmax=xi(k,:);
    end
end

for k=1:1:np
    if FM(k,1)==Fmin;
        X2(k,:)=Xmax(1,:);
    else
        X2(k,:)=XM(k,:);
    end
end

%% CAMBIO DE GENERACION
xi=X2;

[xi]=check_x(xi,a,b);

%% ALMACENAR LOS MEJORES DE CADA ITERACION
Xopt=Xmax;
E=ee.*Xopt; %Vector de E con daños
[fe mve]=Portico(E);
[fobj]=fobj5(fa,fe,mva,mve);
Fobt=fobj;

Fmej(1,e)=Fobt;

%% CALCULO DEL CRITERIO DE PARADA

```

```

if e>rd
    des=std(Fmej(1,e-rd:e));
end

    end
    tiempo=toc;

Coordenadas(q,:)=Xopt;
Valor_Funcion_Objetivo(q,1)=Fobt;
Numero_de_iteraciones(q,1)=e;
Tiempo_de_Ejecucion(q,1)=tiempo;

if Fobt>19.9
    disp(['Prueba # ', num2str(q),' listo ']);
else
    disp(['Prueba # ', num2str(q),' a repetir ']);
    disp(['Valor Funcion Objetivo = ', num2str(Fobt), ' No admisible
']);
    %q=q-1;
    o=o+1;
end
end

```

- Portico Simetrico Total - Ruido – AG

```

%%
clc;clear all;format compact;

%% DAÑO INICIAL
disp(' ')
q=input('Numero de elementos dañados = ');
disp(' ')
xa=[];
xa(1,1:24)=1;

for i=1:1:q
    n=input('Nombre del elemento dañado = ');
    xa(n)=input('Daño (valor entre 0 y 1) = ');
    disp(' ')
end
disp(' ')

%CALCULO DE FRECUENCIAS MODOS ANALITICOS

[fa mva]=Portico_inicial(xa);

%% ----- AG -----
disp(' ')
disp(' ')

```



```

ee(1,1:24)=21.54e9;

for i=1:1:np
    E=ee.*xi(i,:);           %Vector de E con daños
    [fe mve]=Portico(E);
    [fobj]=fobj5(fa,fe,mva,mve);

    fob(i,1)=fobj;
end

%% SELECCION POR RULETA
S=sum(fob);

PROB(:,1)=fob(:,1)/S;
PROB_ACUM=0.;
PROB_ACUM(1,1)=PROB(1,1);

for k=2:1:np
    PROB_ACUM(k,1)=PROB(k,1)+PROB_ACUM(k-1,1);
end

%% GENERACION DE NUMEROS ALEATORIOS
SR(:,1)=rand(np,1);

%% ASIGNACION DE NUEVOS PADRES

for k=1:1:np
    if SR(k,1)<=PROB_ACUM(1,1)
        XS(k,:)=xi(1,:);
    end
    for h=2:1:np
        if PROB_ACUM(h-1,1)<SR(k,1)&& SR(k,1)<=PROB_ACUM(h,1);
            XS(k,:)=xi(h,:);
        end
    end
end

%% CRUCE

CR(:,1)=rand(np/2,1);

for k=1:1:np/2
    if CR(k,1)<Pc
        for l=1:1:d
            Imin(k,l)=min(XS(2*k-1,d),XS(2*k,d));
            Imax(k,l)=max(XS(2*k-1,d),XS(2*k,d));
            r=Imax(k,l)-Imin(k,l);
            XH(k,l)=Imin(k,l)-r*ac+rand()*(Imax(k,l)+r*ac-Imin(k,l))-
r*ac);
        end
    end
end

```

```

        end
    else
        XH(k,:) = XS(2*k-1,:);
    end
end

for k=1:1:np/2
    if CR(k,1) < Pc
        for l=1:1:d
            Imin(k+np/2,l) = min(XS(2*k-1,d), XS(2*k,d));
            Imax(k+np/2,l) = max(XS(2*k-1,d), XS(2*k,d));
            r = Imax(k+np/2,l) - Imin(k+np/2,l);
            XH(k+np/2,l) = Imin(k+np/2,l) - r*ac + rand() * (Imax(k+np/2,l) + r*ac -
Imin(k+np/2,l) - r*ac);
        end
    else
        XH(k+np/2,:) = XS(2*k,:);
    end
end

%% MUTACION

MR(:,1) = rand(np,1);

for k=1:1:np
    if MR(k,1) < Pm
        XM(k,:) = a + (b-a) * rand(1,d);
    else
        XM(k,:) = XH(k,:);
    end
end

for k=1:1:np
    for h=1:1:d
        if XM(k,h) > b
            XM(k,h) = b; %a+rand()*(b-a);
        end
        if XM(k,h) < a
            XM(k,h) = a + rand() * (b-a);
        end
    end
end

for i=1:1:np
    E = ee.*XM(i,:); %Vector de E con daños
    [fe mve] = Portico(E);
    [fobj] = fobj5(fa, fe, mva, mve);

    FM(i,1) = fobj;
end

```

```

%% ELITISMO

Fmax=max(fob);
Fmin=min(FM);

for k=1:1:np
    if fob(k,1)==Fmax;
        Xmax=xi(k,:);
    end
end

for k=1:1:np
    if FM(k,1)==Fmin;
        X2(k,:)=Xmax(1,:);
    else
        X2(k,:)=XM(k,:);
    end
end

%% CAMBIO DE GENERACION
xi=X2;

[xi]=check_x(xi,a,b);

%% ALMACENAR LOS MEJORES DE CADA ITERACION
Xopt=Xmax;
E=ee.*Xopt;           %Vector de E con daños
[fe mve]=Portico(E);
[fobj]=fobj5(fa,fe,mva,mve);
Fobt=fobj;

Fmej(1,e)=Fobt;

%% CALCULO DEL CRITERIO DE PARADA
if e>rd
    des=std(Fmej(1,e-rd:e));
end

    end
    tiempo=toc;

Coordenadas(q,:)=Xopt;
Valor_Funcion_Objetivo(q,1)=Fobt;
Numero_de_iteraciones(q,1)=e;
Tiempo_de_Ejecucion(q,1)=tiempo;

if Fobt>19.9
    disp(['Prueba # ', num2str(q),' listo ']);
else

```

```

disp(['Prueba # ', num2str(q), ' a repetir ']);
disp(['Valor Funcion Objetivo = ', num2str(Fobt), ' No admisible
']);
    %q=q-1;
    o=o+1;
end
end

```

La función objetivo utilizada en para la optimización con AG es:

```

function [fobj]=fobj5(fa,fe,mva,mve)

diff=0;
for k=1:1:4
    division=abs((fe(k))/(fa(k)));
    diferencia=abs(1-division);
    diff=diff+diferencia;
end

difmv=0;
for k=1:1:4
    for l=1:1:24
        division=abs((mve(l,k))/(mva(l,k)));
        diferencia=abs(1-division);
        difmv=difmv+diferencia;
    end
end
fobj=20-(diff+difmv);

```

ANEXO K. Tablas de resultados pruebas algoritmos.

Resultaos PSO y AG.

En este anexo se muestran los resultados obtenidos con los algoritmos AG y PSO.

Los resultados obtenidos con el PSO se encuentran divididos en tres grupos, los resultados para los pórticos sin ruido y con daño en un único elemento (ver tabla 1 a la 16), los resultados para los pórticos sin ruido y con daño múltiple (ver tabla 17 a 24) y los resultados para el pórtico con ruido (ver tabla 25 a 31).

Los resultados obtenidos con el AG se encuentran en tres grupos, los resultados para los pórticos sin ruido y con daño en un único elemento (ver tabla 32 a 35), los resultados para los pórticos sin ruido con daño múltiple (ver tabla 36 a 37), los resultados para el pórtico con ruido (ver tabla 38 a 40) y en la tabla 41 se encuentran los resultados obtenidos para el caso del pórtico simétrico con daño en el elemento 1 y magnitud del 20% con un número máximo de 5000 iteraciones.

En las tablas se muestran los siguientes datos:

α_e = Coordenadas experimentales de daño.

α_{ci} = Coordenadas calculadas en la prueba i.

Fun_Obj = Valor de la Función Objetivo

Iteraciones = Iteraciones Realizadas

Tiempo [seg] =Tiempo en segundo.

Tiempo [min] = Tiempo en minutos.

Elemento dañado:	1
Magnitud de daño:	20%

		Prueba		
		1	2	3
Fun_Obj =		0	0	0
Iteraciones =		74	71	61
Tiempo [seg] =		1358	1404	1150
Tiempo [min] =		22.63	23.40	19.17

Elem	α_e	α_{c1}	α_{c2}	α_{c3}
1	0.80	0.80	0.80	0.80
2	1.00	1.00	1.00	1.00
3	1.00	1.00	1.00	1.00
4	1.00	1.00	1.00	1.00
5	1.00	1.00	1.00	1.00
6	1.00	1.00	1.00	1.00
7	1.00	1.00	1.00	1.00
8	1.00	1.00	1.00	1.00
9	1.00	1.00	1.00	1.00
10	1.00	1.00	1.00	1.00
11	1.00	1.00	1.00	1.00
12	1.00	1.00	1.00	1.00
13	1.00	1.00	1.00	1.00
14	1.00	1.00	1.00	1.00
15	1.00	1.00	1.00	1.00
16	1.00	1.00	1.00	1.00
17	1.00	1.00	1.00	1.00
18	1.00	1.00	1.00	1.00
19	1.00	1.00	1.00	1.00
20	1.00	1.00	1.00	1.00
21	1.00	1.00	1.00	1.00
22	1.00	1.00	1.00	1.00
23	1.00	1.00	1.00	1.00
24	1.00	1.00	1.00	1.00

Tabla 1. Resultados PSO – Pórtico Simétrico – Daño en elemento 1 con 20%.

Elemento dañado:	1
Magnitud de daño:	10%

		Prueba		
		1	2	3
Fun_Obj =		0	0	0
Iteraciones =		69	39	85
Tiempo [seg] =		1203	684	1475
Tiempo [min] =		20.05	11.40	24.58
Elem	α_e	α_{c1}	α_{c2}	α_{c3}
1	0.90	0.90	0.90	0.90
2	1.00	1.00	1.00	1.00
3	1.00	1.00	1.00	1.00
4	1.00	1.00	1.00	1.00
5	1.00	1.00	1.00	1.00
6	1.00	1.00	1.00	1.00
7	1.00	1.00	1.00	1.00
8	1.00	1.00	1.00	1.00
9	1.00	1.00	1.00	1.00
10	1.00	1.00	1.00	1.00
11	1.00	1.00	1.00	1.00
12	1.00	1.00	1.00	1.00
13	1.00	1.00	1.00	1.00
14	1.00	1.00	1.00	1.00
15	1.00	1.00	1.00	1.00
16	1.00	1.00	1.00	1.00
17	1.00	1.00	1.00	1.00
18	1.00	1.00	1.00	1.00
19	1.00	1.00	1.00	1.00
20	1.00	1.00	1.00	1.00
21	1.00	1.00	1.00	1.00
22	1.00	1.00	1.00	1.00
23	1.00	1.00	1.00	1.00
24	1.00	1.00	1.00	1.00

Tabla 2. Resultados PSO – Pórtico Simétrico – Daño en elemento 1 con 10%.

Elemento dañado:	1
Magnitud de daño:	5%

Elem	α_e	Prueba			
		1	2	3	
		Fun_Obj =	0	0	0
		Iteraciones =	51	82	64
		Tiempo [seg] =	909	1476	1125
		Tiempo [min] =	15.15	24.60	18.75
		α_{c1}	α_{c2}	α_{c3}	
1	0.95	0.95	0.95	0.95	
2	1.00	1.00	1.00	1.00	
3	1.00	1.00	1.00	1.00	
4	1.00	1.00	1.00	1.00	
5	1.00	1.00	1.00	1.00	
6	1.00	1.00	1.00	1.00	
7	1.00	1.00	1.00	1.00	
8	1.00	1.00	1.00	1.00	
9	1.00	1.00	1.00	1.00	
10	1.00	1.00	1.00	1.00	
11	1.00	1.00	1.00	1.00	
12	1.00	1.00	1.00	1.00	
13	1.00	1.00	1.00	1.00	
14	1.00	1.00	1.00	1.00	
15	1.00	1.00	1.00	1.00	
16	1.00	1.00	1.00	1.00	
17	1.00	1.00	1.00	1.00	
18	1.00	1.00	1.00	1.00	
19	1.00	1.00	1.00	1.00	
20	1.00	1.00	1.00	1.00	
21	1.00	1.00	1.00	1.00	
22	1.00	1.00	1.00	1.00	
23	1.00	1.00	1.00	1.00	
24	1.00	1.00	1.00	1.00	

Tabla 3. Resultados PSO – Pórtico Simétrico – Daño en elemento 1 con 5%.

Elemento dañado:	1
Magnitud de daño:	1%

		Prueba		
		1	2	3
Fun_Obj =		0	0	0
Iteraciones =		55	76	55
Tiempo [seg] =		963	1330	960
Tiempo [min] =		16.05	22.17	16
Elem	α_e	α_{c1}	α_{c2}	α_{c3}
1	0.99	0.99	0.99	0.99
2	1.00	1.00	1.00	1.00
3	1.00	1.00	1.00	1.00
4	1.00	1.00	1.00	1.00
5	1.00	1.00	1.00	1.00
6	1.00	1.00	1.00	1.00
7	1.00	1.00	1.00	1.00
8	1.00	1.00	1.00	1.00
9	1.00	1.00	1.00	1.00
10	1.00	1.00	1.00	1.00
11	1.00	1.00	1.00	1.00
12	1.00	1.00	1.00	1.00
13	1.00	1.00	1.00	1.00
14	1.00	1.00	1.00	1.00
15	1.00	1.00	1.00	1.00
16	1.00	1.00	1.00	1.00
17	1.00	1.00	1.00	1.00
18	1.00	1.00	1.00	1.00
19	1.00	1.00	1.00	1.00
20	1.00	1.00	1.00	1.00
21	1.00	1.00	1.00	1.00
22	1.00	1.00	1.00	1.00
23	1.00	1.00	1.00	1.00
24	1.00	1.00	1.00	1.00

Tabla 4. Resultados PSO – Pórtico Simétrico – Daño en elemento 1 con 1%.

Elemento dañado:	20
Magnitud de daño:	20%

		Prueba		
		1	2	3
Fun_Obj =		0	0.00	0
Iteraciones =		67	32	61
Tiempo [seg] =		1184	573	1150
Tiempo [min] =		19.73	9.55	19.17
Elem	α_e	α_{c1}	α_{c2}	α_{c3}
1	1.00	1.00	1.00	1.00
2	1.00	1.00	1.00	1.00
3	1.00	1.00	1.00	1.00
4	1.00	1.00	1.00	1.00
5	1.00	1.00	1.00	1.00
6	1.00	1.00	1.00	1.00
7	1.00	1.00	1.00	1.00
8	1.00	1.00	1.00	1.00
9	1.00	1.00	1.00	1.00
10	1.00	1.00	1.00	1.00
11	1.00	1.00	1.00	1.00
12	1.00	1.00	1.00	1.00
13	1.00	1.00	1.00	1.00
14	1.00	1.00	1.00	1.00
15	1.00	1.00	1.00	1.00
16	1.00	1.00	1.00	1.00
17	1.00	1.00	1.00	1.00
18	1.00	1.00	1.00	1.00
19	1.00	1.00	1.00	1.00
20	0.80	0.80	0.80	0.80
21	1.00	1.00	1.00	1.00
22	1.00	1.00	1.00	1.00
23	1.00	1.00	1.00	1.00
24	1.00	1.00	1.00	1.00

Tabla 5. Resultados PSO – Pórtico Simétrico – Daño en elemento 20 con 20%.

Elemento dañado:	20
Magnitud de daño:	10%

		Prueba		
		1	2	3
	Fun_Obj =	0	0	0
	Iteraciones =	89	77	76
	Tiempo [seg] =	1559	1347	1327
	Tiempo [min] =	25.98	22.45	22.12
Elem	α_e	α_{c1}	α_{c2}	α_{c3}
1	1.00	1.00	1.00	1.00
2	1.00	1.00	1.00	1.00
3	1.00	1.00	1.00	1.00
4	1.00	1.00	1.00	1.00
5	1.00	1.00	1.00	1.00
6	1.00	1.00	1.00	1.00
7	1.00	1.00	1.00	1.00
8	1.00	1.00	1.00	1.00
9	1.00	1.00	1.00	1.00
10	1.00	1.00	1.00	1.00
11	1.00	1.00	1.00	1.00
12	1.00	1.00	1.00	1.00
13	1.00	1.00	1.00	1.00
14	1.00	1.00	1.00	1.00
15	1.00	1.00	1.00	1.00
16	1.00	1.00	1.00	1.00
17	1.00	1.00	1.00	1.00
18	1.00	1.00	1.00	1.00
19	1.00	1.00	1.00	1.00
20	0.90	0.90	0.90	0.90
21	1.00	1.00	1.00	1.00
22	1.00	1.00	1.00	1.00
23	1.00	1.00	1.00	1.00
24	1.00	1.00	1.00	1.00

Tabla 6. Resultados PSO – Pórtico Simétrico – Daño en elemento 20 con 10%.

Elemento dañado:	20
Magnitud de daño:	5%

		Prueba		
		1	2	3
	Fun_Obj =	0	0.003	0
	Iteraciones =	51	31	61
	Tiempo [seg] =	954	555	1073
	Tiempo [min] =	15.9	9.25	17.88
Elem	α_e	α_{c1}	α_{c2}	α_{c3}
1	1.00	1.00	1.00	1.00
2	1.00	1.00	1.00	1.00
3	1.00	1.00	1.00	1.00
4	1.00	1.00	1.00	1.00
5	1.00	1.00	1.00	1.00
6	1.00	1.00	1.00	1.00
7	1.00	1.00	1.00	1.00
8	1.00	1.00	1.00	1.00
9	1.00	1.00	1.00	1.00
10	1.00	1.00	1.00	1.00
11	1.00	1.00	1.00	1.00
12	1.00	1.00	1.00	1.00
13	1.00	1.00	1.00	1.00
14	1.00	1.00	1.00	1.00
15	1.00	1.00	1.00	1.00
16	1.00	1.00	1.00	1.00
17	1.00	1.00	1.00	1.00
18	1.00	1.00	1.00	1.00
19	1.00	1.00	1.00	1.00
20	0.95	0.95	0.95	0.95
21	1.00	1.00	1.00	1.00
22	1.00	1.00	1.00	1.00
23	1.00	1.00	1.00	1.00
24	1.00	1.00	1.00	1.00

Tabla 7. Resultados PSO – Pórtico Simétrico – Daño en elemento 20 con 5%.

Elemento dañado:	20
Magnitud de daño:	1%

		Prueba		
		1	2	3
	Fun_Obj =	0.00	0.00	0.16
	Iteraciones =	30	63	39
	Tiempo [seg] =	538	1112	694
	Tiempo [min] =	8.97	18.53	11.57
Elem	α_e	α_{c1}	α_{c2}	α_{c3}
1	1.00	1.00	1.00	1.00
2	1.00	1.00	1.00	1.00
3	1.00	1.00	1.00	1.00
4	1.00	1.00	1.00	1.00
5	1.00	1.00	1.00	1.00
6	1.00	1.00	1.00	1.00
7	1.00	1.00	1.00	1.00
8	1.00	1.00	1.00	1.00
9	1.00	1.00	1.00	1.00
10	1.00	1.00	1.00	1.00
11	1.00	1.00	1.00	1.00
12	1.00	1.00	1.00	1.00
13	1.00	1.00	1.00	1.00
14	1.00	1.00	1.00	0.99
15	1.00	1.00	1.00	1.00
16	1.00	1.00	1.00	1.00
17	1.00	1.00	1.00	1.00
18	1.00	1.00	1.00	1.00
19	1.00	1.00	1.00	1.00
20	0.99	0.99	0.99	1.00
21	1.00	1.00	1.00	1.00
22	1.00	1.00	1.00	1.00
23	1.00	1.00	1.00	0.98
24	1.00	1.00	1.00	1.00

Tabla 8. Resultados PSO – Pórtico Simétrico – Daño en elemento 20 con 1%.

Elemento dañado:	1
Magnitud de daño:	20%

		Prueba		
		1	2	3
	Fun_Obj =	0	0	0
	Iteraciones =	65	82	89
	Tiempo [seg] =	1400	1718	1834
	Tiempo [min] =	23.33	28.64	30.57
Elem	α_e	α_{c1}	α_{c2}	α_{c3}
1	0.8	0.80	0.80	0.80
2	1	1.00	1.00	1.00
3	1	1.00	1.00	1.00
4	1	1.00	1.00	1.00
5	1	1.00	1.00	1.00
6	1	1.00	1.00	1.00
7	1	1.00	1.00	1.00
8	1	1.00	1.00	1.00
9	1	1.00	1.00	1.00
10	1	1.00	1.00	1.00
11	1	1.00	1.00	1.00
12	1	1.00	1.00	1.00
13	1	1.00	1.00	1.00
14	1	1.00	1.00	1.00
15	1	1.00	1.00	1.00
16	1	1.00	1.00	1.00
17	1	1.00	1.00	1.00
18	1	1.00	1.00	1.00
19	1	1.00	1.00	1.00
20	1	1.00	1.00	1.00
21	1	1.00	1.00	1.00
22	1	1.00	1.00	1.00
23	1	1.00	1.00	1.00
24	1	1.00	1.00	1.00

Tabla 9. Resultados PSO – Pórtico Asimétrico – Daño en elemento 1 con 20%.

Elemento dañado:	1
Magnitud de daño:	10%

		Prueba		
		1	2	3
	Fun_Obj =	0	0	0
	Iteraciones =	78	87	86
	Tiempo [seg] =	1638	1817	1841
	Tiempo [min] =	27.29	30.29	30.68
Elem	α_e	α_{c1}	α_{c2}	α_{c3}
1	0.9	0.9	0.9	0.9
2	1	1	1	1
3	1	1	1	1
4	1	1	1	1
5	1	1	1	1
6	1	1	1	1
7	1	1	1	1
8	1	1	1	1
9	1	1	1	1
10	1	1	1	1
11	1	1	1	1
12	1	1	1	1
13	1	1	1	1
14	1	1	1	1
15	1	1	1	1
16	1	1	1	1
17	1	1	1	1
18	1	1	1	1
19	1	1	1	1
20	1	1	1	1
21	1	1	1	1
22	1	1	1	1
23	1	1	1	1
24	1	1	1	1

Tabla 10. Resultados PSO – Pórtico Asimétrico – Daño en elemento 1 con 10%.

Elemento dañado:	1
Magnitud de daño:	5%

		Prueba		
		1	2	3
	Fun_Obj =	0	0	0
	Iteraciones =	80	71	75
	Tiempo [seg] =	1719	1541	1645
	Tiempo [min] =	28.65	25.68	27.42
Elem	α_e	α_{c1}	α_{c2}	α_{c3}
1	0.95	0.95	0.95	0.95
2	1	1.00	1.00	1.00
3	1	1.00	1.00	1.00
4	1	1.00	1.00	1.00
5	1	1.00	1.00	1.00
6	1	1.00	1.00	1.00
7	1	1.00	1.00	1.00
8	1	1.00	1.00	1.00
9	1	1.00	1.00	1.00
10	1	1.00	1.00	1.00
11	1	1.00	1.00	1.00
12	1	1.00	1.00	1.00
13	1	1.00	1.00	1.00
14	1	1.00	1.00	1.00
15	1	1.00	1.00	1.00
16	1	1.00	1.00	1.00
17	1	1.00	1.00	1.00
18	1	1.00	1.00	1.00
19	1	1.00	1.00	1.00
20	1	1.00	1.00	1.00
21	1	1.00	1.00	1.00
22	1	1.00	1.00	1.00
23	1	1.00	1.00	1.00
24	1	1.00	1.00	1.00

Tabla 11. Resultados PSO – Pórtico Asimétrico – Daño en elemento 1 con 5%.

Elemento dañado:	1
Magnitud de daño:	1%

		Prueba		
		1	2	3
Fun_Obj =		0.91	0.49	0.37
Iteraciones =		27	100	100
Tiempo [seg] =		670	2224	2179
Tiempo [min] =		11.17	37.07	36.31
Elem	α_e	α_{c1}	α_{c2}	α_{c3}
1	0.99	0.92	1.00	1.00
2	1	1.00	0.97	1.00
3	1	0.70	1.00	0.84
4	1	1.00	1.00	1.00
5	1	1.00	1.00	1.00
6	1	0.94	1.00	0.90
7	1	0.68	1.00	0.98
8	1	1.00	1.00	0.81
9	1	1.00	1.00	1.00
10	1	0.95	1.00	0.92
11	1	1.00	1.00	0.88
12	1	0.65	0.99	0.78
13	1	0.75	1.00	0.85
14	1	1.00	1.00	1.00
15	1	0.75	1.00	0.86
16	1	1.00	1.00	1.00
17	1	0.67	1.00	0.90
18	1	0.66	1.00	0.91
19	1	1.00	1.00	1.00
20	1	0.76	1.00	0.99
21	1	1.00	1.00	1.00
22	1	1.00	1.00	1.00
23	1	0.51	1.00	0.97
24	1	0.84	1.00	0.96

Tabla 12. Resultados PSO – Pórtico Asimétrico – Daño en elemento 1 con 1%.

Elemento dañado:	20
Magnitud de daño:	20%

Elem	α_e	Prueba		
		1	2	3
	Fun_Obj =	0	0	0
	Iteraciones =	61	69	68
	Tiempo [seg] =	1307	1405	1387
	Tiempo [min] =	21.79	23.42	23.11
Elem	α_e	α_{c1}	α_{c2}	α_{c3}
1	1	1	1	1
2	1	1	1	1
3	1	1	1	1
4	1	1	1	1
5	1	1	1	1
6	1	1	1	1
7	1	1	1	1
8	1	1	1	1
9	1	1	1	1
10	1	1	1	1
11	1	1	1	1
12	1	1	1	1
13	1	1	1	1
14	1	1	1	1
15	1	1	1	1
16	1	1	1	1
17	1	1	1	1
18	1	1	1	1
19	1	1	1	1
20	0.8	0.80	0.80	0.80
21	1	1	1	1
22	1	1	1	1
23	1	1	1	1
24	1	1	1	1

Tabla 13. Resultados PSO – Pórtico Asimétrico – Daño en elemento 20 con 20%.

Elemento dañado:	20
Magnitud de daño:	5%

		Prueba		
		1	2	3
Fun_Obj =		0	0	0
Iteraciones =		60	64	60
Tiempo [seg] =		1,230	1306	1232
Tiempo [min] =		20.5	21.77	20.53
Elem	α_e	α_{c1}	α_{c2}	α_{c3}
1	1	1	1	1
2	1	1	1	1
3	1	1	1	1
4	1	1	1	1
5	1	1	1	1
6	1	1	1	1
7	1	1	1	1
8	1	1	1	1
9	1	1	1	1
10	1	1	1	1
11	1	1	1	1
12	1	1	1	1
13	1	1	1	1
14	1	1	1	1
15	1	1	1	1
16	1	1	1	1
17	1	1	1	1
18	1	1	1	1
19	1	1	1	1
20	0.95	0.95	0.95	0.95
21	1	1	1	1
22	1	1	1	1
23	1	1	1	1
24	1	1	1	1

Tabla 15. Resultados PSO – Pórtico Asimétrico – Daño en elemento 20 con 5%.

Elemento dañado:	20
Magnitud de daño:	1%

		Prueba		
		1	2	3
Fun_Obj =		0	0	0
Iteraciones =		63	45	48
Tiempo [seg] =		1278	972.3	1095
Tiempo [min] =		21.3	16.21	18.24
Elem	α_e	α_{c1}	α_{c2}	α_{c3}
1	1	1	1	1
2	1	1	1	1
3	1	1	1	1
4	1	1	1	1
5	1	1	1	1
6	1	1	1	1
7	1	1	1	1
8	1	1	1	1
9	1	1	1	1
10	1	1	1	1
11	1	1	1	1
12	1	1	1	1
13	1	1	1	1
14	1	1	1	1
15	1	1	1	1
16	1	1	1	1
17	1	1	1	1
18	1	1	1	1
19	1	1	1	1
20	0.99	0.99	0.99	0.99
21	1	1	1	1
22	1	1	1	1
23	1	1	1	1
24	1	1	1	1

Tabla 16. Resultados PSO – Pórtico Asimétrico – Daño en elemento 20 con 1%.

Elemento dañado:	9 - 10 - 18
Magnitud de daño:	20%

Elem	α_e	Prueba		
		1	2	3
	Fun_Obj =	0	0	0
	Iteraciones =	91	91	85
	Tiempo [seg] =	1576	1565	1460
	Tiempo [min] =	26.27	26.08	24.33
Elem	α_e	α_{c1}	α_{c2}	α_{c3}
1	1	1	1	1
2	1	1	1	1
3	1	1	1	1
4	1	1	1	1
5	1	1	1	1
6	1	1	1	1
7	1	1	1	1
8	1	1	1	1
9	0.8	0.80	0.8	0.8
10	0.8	0.80	0.8	0.8
11	1	1	1	1
12	1	1	1	1
13	1	1	1	1
14	1	1	1	1
15	1	1	1	1
16	1	1	1	1
17	1	1	1	1
18	0.8	0.80	0.8	0.8
19	1	1	1	1
20	1	1	1	1
21	1	1	1	1
22	1	1	1	1
23	1	1	1	1
24	1	1	1	1

Tabla 17. Resultados PSO – Pórtico Simétrico – Daño en elementos 9-10-18 con 20%.

Elemento dañado:	9 - 10 - 18
Magnitud de daño:	10%

Elem	α_e	Prueba		
		1	2	3
	Fun_Obj =	0	0	0
	Iteraciones =	81	80	98
	Tiempo [seg] =	1398	1379	1689
	Tiempo [min] =	23.3	22.98	28.15
		α_{c1}	α_{c2}	α_{c3}
1	1	1	1	1
2	1	1	1	1
3	1	1	1	1
4	1	1	1	1
5	1	1	1	1
6	1	1	1	1
7	1	1	1	1
8	1	1	1	1
9	0.9	0.90	0.9	0.9
10	0.9	0.90	0.9	0.9
11	1	1	1	1
12	1	1	1	1
13	1	1	1	1
14	1	1	1	1
15	1	1	1	1
16	1	1	1	1
17	1	1	1	1
18	0.9	0.90	0.9	0.9
19	1	1	1	1
20	1	1	1	1
21	1	1	1	1
22	1	1	1	1
23	1	1	1	1
24	1	1	1	1

Tabla 18. Resultados PSO – Pórtico Simétrico – Daño en elementos 9-10-18 con 10%.

Elemento dañado:	9 - 10 - 18
Magnitud de daño:	1%

Elem	α_e	Prueba		
		1	2	3
	Fun_Obj =	0.27	0.23	0.23
	Iteraciones =	100	53	82
	Tiempo [seg] =	1714	920	1411
	Tiempo [min] =	28.57	15.33	23.52
		α_{c1}	α_{c2}	α_{c3}
1	1	1	1	1
2	1	1	1	1
3	1	1	1	1
4	1	1	1	1
5	1	1	1	1
6	1	1	1	1
7	1	1	1	1
8	1	1	1	1
9	0.99	1	0.99	0.99
10	0.99	0.99	1	1
11	1	1	1	1
12	1	1	1	1
13	1	1	1	1
14	1	1	1	1
15	1	1	1	1
16	1	1	1	1
17	1	1	1	1
18	0.99	0.99	0.991	0.991
19	1	1	1	1
20	1	1	1	1
21	1	1	1	1
22	1	1	1	1
23	1	1	1	1
24	1	1	1	1

Tabla 20. Resultados PSO – Pórtico Simétrico – Daño en elementos 9-10-18 con 1%.

Elemento dañado:	9 - 13 - 18
Magnitud de daño:	20%

Elem	α_e	Prueba		
		1	2	3
	Fun_Obj =	1.35	0.00	0.00
	Iteraciones =	500	105	107
	Tiempo [seg] =	11721	2324	2194
	Tiempo [min] =	195.35	38.74	36.57
Elem	α_e	α_{c1}	α_{c2}	α_{c3}
1	1	0.90	1.00	1.00
2	1	1.00	1.00	1.00
3	1	0.88	1.00	1.00
4	1	0.94	1.00	1.00
5	1	0.76	1.00	1.00
6	1	1.00	1.00	1.00
7	1	1.00	1.00	1.00
8	1	1.00	1.00	1.00
9	0.8	0.75	0.80	0.80
10	1	0.95	1.00	1.00
11	1	0.84	1.00	1.00
12	1	0.95	1.00	1.00
13	0.8	0.80	0.80	0.80
14	1	0.76	1.00	1.00
15	1	0.98	1.00	1.00
16	1	1.00	1.00	1.00
17	1	0.97	1.00	1.00
18	0.8	0.73	0.80	0.80
19	1	1.00	1.00	1.00
20	1	0.62	1.00	1.00
21	1	1.00	1.00	1.00
22	1	1.00	1.00	1.00
23	1	1.00	1.00	1.00
24	1	0.84	1.00	1.00

Tabla 21. Resultados PSO – Pórtico Asimétrico – Daño en elementos 9-13-18 con 20%.

Elemento dañado:	9 - 13 - 18
Magnitud de daño:	10%

Elem	α_e	Prueba		
		1	2	3
	Fun_Obj =	6.17	4.89	1.75
	Iteraciones =	100	100	100
	Tiempo [seg] =	3026	2374	2188
	Tiempo [min] =	50.43	39.57	36.46
Elem	α_e	α_{c1}	α_{c2}	α_{c3}
1	1	1.00	1.00	1.00
2	1	1.00	1.00	1.00
3	1	1.00	1.00	1.00
4	1	1.00	1.00	1.00
5	1	1.00	1.00	1.00
6	1	1.00	1.00	1.00
7	1	1.00	1.00	1.00
8	1	1.00	1.00	1.00
9	0.9	0.91	0.90	0.96
10	1	1.00	1.00	0.89
11	1	1.00	1.00	1.00
12	1	1.00	1.00	1.00
13	0.9	1.00	0.90	0.91
14	1	0.95	1.00	1.00
15	1	1.00	1.00	1.00
16	1	1.00	1.00	1.00
17	1	1.00	1.00	1.00
18	0.9	0.92	0.90	1.00
19	1	1.00	1.00	1.00
20	1	1.00	1.00	1.00
21	1	1.00	1.00	0.99
22	1	1.00	1.00	1.00
23	1	1.00	1.00	1.00
24	1	1.00	1.00	1.00

Tabla 22. Resultados PSO – Pórtico Asimétrico – Daño en elementos 9-13-18 con 10%.

Elemento dañado:	9 - 13 - 18
Magnitud de daño:	5%

		Prueba		
		1	2	3
	Fun_Obj =	0.00	1.41	0.04
	Iteraciones =	100	100	100
	Tiempo [seg] =	2205	2200	2223
	Tiempo [min] =	36.75	36.67	37.06
Elem	α_e	α_{c1}	α_{c2}	α_{c3}
1	1	1.00	1.00	1.00
2	1	1.00	1.00	1.00
3	1	1.00	1.00	1.00
4	1	1.00	0.99	1.00
5	1	1.00	1.00	1.00
6	1	1.00	1.00	1.00
7	1	1.00	1.00	1.00
8	1	1.00	1.00	1.00
9	0.95	0.95	0.98	0.95
10	1	1.00	0.96	1.00
11	1	1.00	1.00	1.00
12	1	1.00	1.00	1.00
13	0.95	0.95	0.97	0.95
14	1	1.00	1.00	1.00
15	1	1.00	1.00	1.00
16	1	1.00	1.00	1.00
17	1	1.00	1.00	1.00
18	0.95	0.95	1.00	0.95
19	1	1.00	1.00	1.00
20	1	1.00	1.00	1.00
21	1	1.00	1.00	1.00
22	1	1.00	1.00	1.00
23	1	1.00	1.00	1.00
24	1	1.00	1.00	1.00

Tabla 23. Resultados PSO – Pórtico Asimétrico – Daño en elementos 9-13-18 con 5%.

Elemento dañado:	9 - 13 - 18
Magnitud de daño:	1%

Elem	α_e	Prueba		
		1	2	3
	Fun_Obj =	150	138	118
	Iteraciones =	3225	2927	2492
	Tiempo [seg] =	53.76	48.78	41.53
	Tiempo [min] =			
		α_{c1}	α_{c2}	α_{c3}
1	1	1.00	1.00	1.00
2	1	1.00	1.00	1.00
3	1	1.00	1.00	1.00
4	1	1.00	1.00	1.00
5	1	1.00	1.00	1.00
6	1	1.00	1.00	1.00
7	1	1.00	1.00	1.00
8	1	1.00	1.00	1.00
9	0.95	1.00	1.00	1.00
10	1	1.00	0.99	0.99
11	1	1.00	1.00	1.00
12	1	1.00	1.00	1.00
13	0.95	1.00	1.00	1.00
14	1	0.99	1.00	1.00
15	1	0.99	1.00	1.00
16	1	1.00	1.00	1.00
17	1	1.00	1.00	1.00
18	0.95	1.00	1.00	1.00
19	1	1.00	1.00	1.00
20	1	1.00	1.00	1.00
21	1	1.00	1.00	1.00
22	1	1.00	1.00	1.00
23	1	1.00	1.00	1.00
24	1	1.00	1.00	1.00

Tabla 24. Resultados PSO – Pórtico Asimétrico – Daño en elementos 9-13-18 con 1%.

		Prueba # 1	Prueba # 2	Prueba # 3
Función Objetivo =		2.31	2.22	2.21
Iteraciones realizadas =		63	300	183
Tiempo [seg] =		1114	5253	3208
Tiempo [min] =		18.6	87.5	53.4
Elemento	α_e	α_{c1}	α_{c2}	α_{c3}
1	1	1	1	1
2	1	1	0.98	1
3	1	1	1	1
4	1	1	1	1
5	1	1	1	1
6	1	1	1	1
7	1	1	1	1
8	1	1	1	1
9	1	1	1	1
10	1	1	1	1
11	1	1	1	1
12	1	1	1	0.97
13	1	1	1	1
14	1	1	1	1
15	1	1	0.99	1
16	1	1	0.99	1
17	1	1	1	1
18	1	0.98	0.97	0.99
19	1	1	0.99	1
20	1	1	1	1
21	1	1	1	0.98
22	1	1	1	1
23	1	1	1	1
24	1	1	1	1

Tabla 25. Resultados PSO – Pórtico Simétrico con ruido – sin daño.

Elemento dañado:	1
Magnitud de daño:	20%

Elem	α_e	Prueba		
		1	2	3
	Fun_Obj =	2.54	2.56	2.60
	Iteraciones =	217	147	111
	Tiempo [seg] =	3856	2659	1932
	Tiempo [min] =	64.27	44.32	32.20
		α_{c1}	α_{c2}	α_{c3}
1	0.8	0.81	0.80	0.80
2	1	1.00	1.00	1.00
3	1	1.00	1.00	1.00
4	1	1.00	1.00	1.00
5	1	1.00	1.00	1.00
6	1	1.00	1.00	1.00
7	1	1.00	1.00	1.00
8	1	1.00	1.00	1.00
9	1	1.00	1.00	1.00
10	1	1.00	1.00	1.00
11	1	1.00	1.00	1.00
12	1	1.00	0.99	1.00
13	1	1.00	1.00	1.00
14	1	1.00	1.00	1.00
15	1	1.00	1.00	1.00
16	1	1.00	1.00	0.99
17	1	1.00	1.00	1.00
18	1	0.98	1.00	1.00
19	1	1.00	1.00	1.00
20	1	1.00	1.00	1.00
21	1	0.98	0.97	1.00
22	1	1.00	1.00	1.00
23	1	1.00	1.00	1.00
24	1	1.00	1.00	1.00

Tabla 26. Resultados PSO – Pórtico Simétrico con ruido – Daño elemento 1 con 20%.

Elemento dañado:	1
Magnitud de daño:	5%

		Prueba		
		1	2	3
	Fun_Obj =	4.40	2.35	2.32
	Iteraciones =	100	100	100
	Tiempo [seg] =	2224	2229	2205
	Tiempo [min] =	37.06	37.15	36.74
Elem	α_e	α_{c1}	α_{c2}	α_{c3}
1	0.95	0.65	1.00	0.96
2	1	1.00	1.00	1.00
3	1	1.00	0.95	1.00
4	1	1.00	0.96	1.00
5	1	1.00	1.00	1.00
6	1	0.84	1.00	1.00
7	1	1.00	0.97	1.00
8	1	1.00	0.91	1.00
9	1	0.68	1.00	1.00
10	1	1.00	1.00	1.00
11	1	1.00	1.00	1.00
12	1	0.65	1.00	1.00
13	1	1.00	1.00	1.00
14	1	0.82	1.00	1.00
15	1	0.86	1.00	1.00
16	1	0.81	0.98	1.00
17	1	1.00	1.00	1.00
18	1	0.87	0.96	0.96
19	1	0.82	1.00	1.00
20	1	0.69	1.00	1.00
21	1	1.00	1.00	1.00
22	1	1.00	1.00	1.00
23	1	1.00	1.00	1.00
24	1	0.67	1.00	1.00

Tabla 27. Resultados PSO – Pórtico Simétrico con ruido – Daño elemento 1 con 5%.

Elemento dañado:	20
Magnitud de daño:	20%

Elem	α_e	Prueba		
		1	2	3
	Fun_Obj =	2.25	2.17	2.26
	Iteraciones =	100	100	96
	Tiempo [seg] =	2535	2276	2141
	Tiempo [min] =	42.26	37.94	35.68
Elem	α_e	α_{c1}	α_{c2}	α_{c3}
1	1	1.00	1.00	1.00
2	1	1.00	1.00	1.00
3	1	1.00	1.00	1.00
4	1	1.00	1.00	1.00
5	1	1.00	0.99	1.00
6	1	1.00	1.00	1.00
7	1	0.99	1.00	1.00
8	1	1.00	1.00	1.00
9	1	1.00	0.98	1.00
10	1	1.00	1.00	1.00
11	1	1.00	0.99	1.00
12	1	1.00	1.00	0.98
13	1	1.00	1.00	1.00
14	1	1.00	1.00	1.00
15	1	1.00	1.00	1.00
16	1	1.00	0.98	1.00
17	1	0.99	1.00	1.00
18	1	1.00	1.00	1.00
19	1	1.00	1.00	1.00
20	0.8	0.80	0.79	0.80
21	1	1.00	1.00	1.00
22	1	1.00	1.00	1.00
23	1	1.00	1.00	1.00
24	1	1.00	0.99	1.00

Tabla 28. Resultados PSO – Pórtico Simétrico con ruido – Daño elemento 20 con 20%.

Elemento dañado:	20
Magnitud de daño:	5%

Elem	α_e	Prueba		
		1	2	3
	Fun_Obj =	2.45	2.55	2.23
	Iteraciones =	114	121	150
	Tiempo [seg] =	2451	2624	3399
	Tiempo [min] =	40.85	43.73	56.65
Elem	α_e	α_{c1}	α_{c2}	α_{c3}
1	1	1.00	1.00	1.00
2	1	1.00	1.00	1.00
3	1	1.00	1.00	1.00
4	1	1.00	1.00	1.00
5	1	1.00	1.00	1.00
6	1	1.00	1.00	1.00
7	1	1.00	1.00	1.00
8	1	1.00	0.92	1.00
9	1	1.00	1.00	0.99
10	1	1.00	1.00	1.00
11	1	0.93	0.96	1.00
12	1	1.00	1.00	0.98
13	1	1.00	1.00	1.00
14	1	0.99	1.00	1.00
15	1	0.98	1.00	0.99
16	1	1.00	1.00	1.00
17	1	1.00	1.00	1.00
18	1	1.00	1.00	0.99
19	1	1.00	1.00	0.99
20	0.95	1.00	1.00	0.96
21	1	1.00	1.00	1.00
22	1	1.00	1.00	1.00
23	1	1.00	0.92	1.00
24	1	1.00	1.00	1.00

Tabla 29. Resultados PSO – Pórtico Simétrico con ruido – Daño elemento 20 con 5%.

Elemento dañado:	9 - 10 - 18
Magnitud de daño:	20%

Elem	α_e	Prueba		
		1	2	3
	Fun_Obj =	2.43	2.43	2.38
	Iteraciones =	104	109	178
	Tiempo [seg] =	1826	1908	3118
	Tiempo [min] =	30.433	#####	#####
		α_{c1}	α_{c2}	α_{c3}
1	1.00	1.00	1.00	1.00
2	1.00	1.00	1.00	1.00
3	1.00	1.00	1.00	1.00
4	1.00	1.00	1.00	1.00
5	1.00	1.00	1.00	1.00
6	1.00	1.00	1.00	0.99
7	1.00	1.00	1.00	0.93
8	1.00	1.00	1.00	0.99
9	0.80	0.81	0.81	0.82
10	0.80	0.80	0.80	0.81
11	1.00	1.00	1.00	0.97
12	1.00	1.00	1.00	1.00
13	1.00	1.00	1.00	0.98
14	1.00	1.00	1.00	1.00
15	1.00	1.00	1.00	1.00
16	1.00	1.00	1.00	0.99
17	1.00	1.00	1.00	1.00
18	0.80	0.79	0.79	0.78
19	1.00	1.00	1.00	1.00
20	1.00	1.00	1.00	1.00
21	1.00	1.00	1.00	1.00
22	1.00	1.00	1.00	0.97
23	1.00	1.00	1.00	1.00
24	1.00	1.00	1.00	1.00

Tabla 30. Resultados PSO – Pórtico Simétrico con ruido – Daño elementos 9-10-18 con 20%.

Elemento dañado:	9 - 10 - 18
Magnitud de daño:	5%

		Prueba		
		1	2	3
	Fun_Obj =	2.35	4.37	2.49
	Iteraciones =	125	200	194
	Tiempo [seg] =	2577	4085	4077
	Tiempo [min] =	42.95	68.08	67.95
Elem	α_e	α_{c1}	α_{c2}	α_{c3}
1	1.00	1.00	0.70	1.00
2	1.00	1.00	1.00	1.00
3	1.00	1.00	0.79	1.00
4	1.00	1.00	0.99	1.00
5	1.00	1.00	0.89	1.00
6	1.00	1.00	0.68	1.00
7	1.00	1.00	1.00	1.00
8	1.00	1.00	1.00	1.00
9	0.95	0.94	0.61	0.93
10	1.00	0.98	0.76	0.97
11	1.00	1.00	0.54	1.00
12	1.00	1.00	0.85	1.00
13	0.95	1.00	0.70	1.00
14	1.00	1.00	0.59	1.00
15	1.00	1.00	0.98	1.00
16	1.00	1.00	0.76	1.00
17	1.00	0.95	0.67	0.96
18	0.95	0.94	0.93	1.00
19	1.00	1.00	0.77	1.00
20	1.00	1.00	0.82	1.00
21	1.00	1.00	0.75	1.00
22	1.00	1.00	1.00	1.00
23	1.00	1.00	1.00	1.00
24	1.00	1.00	0.62	0.99

Tabla 31. Resultados PSO – Pórtico Simétrico con ruido – Daño elementos 9-10-18 con 5%.

Elemento dañado:	1
Magnitud de daño:	20%

Elem	α_e	Prueba			
		1	2	3	
		Fun_Obj =	14.29	13.58	14.12
		Iteraciones =	100	100	100
		Tiempo [seg] =	3795	3798	3788
		Tiempo [min] =	63.25	63.3	63.13
		α_{c1}	α_{c2}	α_{c3}	
1	0,80	0.78	0.63	0.72	
2	1	0.91	0.64	0.78	
3	1	0.79	0.63	0.75	
4	1	0.83	0.64	0.73	
5	1	0.83	0.70	0.75	
6	1	0.87	0.60	0.71	
7	1	0.80	0.70	0.74	
8	1	0.91	0.64	0.74	
9	1	0.86	0.67	0.73	
10	1	0.82	0.71	0.73	
11	1	0.92	0.61	0.73	
12	1	0.80	0.60	0.77	
13	1	0.94	0.63	0.76	
14	1	0.90	0.66	0.77	
15	1	0.86	0.74	0.77	
16	1	0.86	0.67	0.70	
17	1	0.82	0.69	0.78	
18	1	0.94	0.62	0.71	
19	1	0.83	0.62	0.77	
20	1	0.85	0.76	0.76	
21	1	0.87	0.71	0.72	
22	1	0.91	0.61	0.75	
23	1	0.85	0.75	0.76	
24	1	0.93	0.74	0.72	

Tabla 32. Resultados AG – Pórtico Simétrico – Daño elemento 1 con 20%.

Elemento dañado:	20
Magnitud de daño:	20%

Elem	α_e	Prueba		
		1	2	3
	Fun_Obj =	14.96	16.42	16.16
	Iteraciones =	100	100	100
	Tiempo [seg] =	3767	3765	3827
	Tiempo [min] =	62.78	62.75	63.78
Elem	α_e	α_{c1}	α_{c2}	α_{c3}
1	1	0.87	0.95	0.76
2	1	0.89	0.96	0.74
3	1	0.85	0.94	0.78
4	1	0.87	0.96	0.79
5	1	0.84	0.96	0.76
6	1	0.85	0.96	0.78
7	1	0.88	0.93	0.78
8	1	0.86	0.95	0.78
9	1	0.87	0.91	0.74
10	1	0.85	0.94	0.73
11	1	0.86	0.93	0.79
12	1	0.84	0.95	0.76
13	1	0.86	0.93	0.79
14	1	0.83	0.92	0.75
15	1	0.86	0.92	0.75
16	1	0.85	0.94	0.78
17	1	0.86	0.95	0.74
18	1	0.88	0.95	0.76
19	1	0.86	0.92	0.78
20	0.8	0.87	0.92	0.77
21	1	0.87	0.95	0.78
22	1	0.88	0.93	0.76
23	1	0.86	0.96	0.73
24	1	0.86	0.95	0.79

Tabla 33. Resultados AG – Pórtico Simétrico – Daño elemento 20 con 20%.

Elemento dañado:	1
Magnitud de daño:	20%

		Prueba		
		1	2	3
	Fun_Obj =	7.68	10.69	13.32
	Iteraciones =	100	100	100
	Tiempo [seg] =	3686	3561	3654
	Tiempo [min] =	61.43	59.35	60.90
Elem	α_e	α_{c1}	α_{c2}	α_{c3}
1	0.8	0.86	0.09	0.67
2	1	0.90	0.10	0.75
3	1	0.87	0.09	0.86
4	1	0.86	0.09	0.82
5	1	0.86	0.09	0.75
6	1	0.89	0.08	0.79
7	1	0.86	0.09	0.89
8	1	0.93	0.09	0.83
9	1	0.91	0.10	0.81
10	1	0.85	0.09	0.87
11	1	0.92	0.09	0.84
12	1	0.87	0.09	0.77
13	1	0.87	0.08	0.87
14	1	0.93	0.09	0.76
15	1	0.90	0.09	0.71
16	1	0.86	0.09	0.89
17	1	0.88	0.09	0.81
18	1	0.84	0.09	0.76
19	1	0.85	0.10	0.74
20	1	0.92	0.08	0.74
21	1	0.85	0.09	0.82
22	1	0.85	0.09	0.87
23	1	0.90	0.09	0.89
24	1	0.93	0.10	0.84

Tabla 34. Resultados AG – Pórtico Asimétrico – Daño elemento 1 con 20%.

Elemento dañado:	20
Magnitud de daño:	20%

		Prueba		
		1	2	3
Fun_Obj =		18.26	18.29	18.32
Iteraciones =		100	100	100
Tiempo [seg] =		3848	3588	3553
Tiempo [min] =		64.133	59.800	59.217
Elem	α_e	α_{c1}	α_{c2}	α_{c3}
1	1	0.97	0.98	1.00
2	1	0.97	0.98	1.00
3	1	0.97	0.98	1.00
4	1	0.97	0.98	1.00
5	1	0.97	0.98	1.00
6	1	0.97	0.98	1.00
7	1	0.97	0.98	1.00
8	1	0.97	0.98	1.00
9	1	0.97	0.98	1.00
10	1	0.97	0.98	1.00
11	1	0.97	0.98	1.00
12	1	0.97	0.98	1.00
13	1	0.97	0.98	1.00
14	1	0.97	0.98	1.00
15	1	0.97	0.98	1.00
16	1	0.97	0.98	1.00
17	1	0.97	0.98	1.00
18	1	0.97	0.98	1.00
19	1	0.97	0.98	1.00
20	0.8	0.97	0.98	1.00
21	1	0.97	0.98	1.00
22	1	0.97	0.98	1.00
23	1	0.97	0.98	1.00
24	1	0.97	0.98	1.00

Tabla 35. Resultados AG – Pórtico Asimétrico – Daño elemento 20 con 20%.

Elemento dañado:	9 - 10 - 18
Magnitud de daño:	20%

		Prueba		
		1	2	3
	Fun_Obj =	12.99	13.97	12.96
	Iteraciones =	100	100	100
	Tiempo [seg] =	3794	3791	3804
	Tiempo [min] =	63.23	63.18	63.40
Elem	α_e	α_{c1}	α_{c2}	α_{c3}
1	1	0.49	0.73	0.53
2	1	0.49	0.73	0.50
3	1	0.49	0.73	0.51
4	1	0.51	0.75	0.54
5	1	0.51	0.74	0.50
6	1	0.51	0.76	0.55
7	1	0.52	0.71	0.55
8	1	0.53	0.72	0.53
9	0.8	0.47	0.71	0.50
10	0.8	0.47	0.71	0.54
11	1	0.47	0.76	0.54
12	1	0.50	0.71	0.52
13	1	0.50	0.73	0.52
14	1	0.50	0.75	0.54
15	1	0.49	0.72	0.53
16	1	0.52	0.74	0.53
17	1	0.48	0.75	0.50
18	0.8	0.48	0.74	0.50
19	1	0.52	0.76	0.55
20	1	0.51	0.73	0.54
21	1	0.47	0.75	0.52
22	1	0.52	0.76	0.54
23	1	0.47	0.72	0.52
24	1	0.50	0.76	0.52

Tabla 36. Resultados AG – Pórtico Simétrico – Daño elementos 9-10-18 con 20%.

Elemento dañado:	9 - 13 - 18
Magnitud de daño:	20%

		Prueba		
		1	2	3
	Fun_Obj =	9.09	7.58	8.56
	Iteraciones =	500.00	500.00	500.00
	Tiempo [seg] =	20045	18260	17850
	Tiempo [min] =	334.08	304.33	297.50
Elem	α_e	α_{c1}	α_{c2}	α_{c3}
1	1	0.58	0.40	0.73
2	1	0.72	0.52	0.80
3	1	0.56	0.53	0.73
4	1	0.56	0.55	0.72
5	1	0.69	0.39	0.70
6	1	0.73	0.50	0.68
7	1	0.52	0.56	0.71
8	1	0.69	0.48	0.71
9	0.8	0.60	0.41	0.66
10	1	0.64	0.57	0.69
11	1	0.64	0.52	0.79
12	1	0.70	0.38	0.69
13	0.8	0.52	0.41	0.76
14	1	0.64	0.49	0.74
15	1	0.59	0.39	0.77
16	1	0.62	0.52	0.68
17	1	0.64	0.39	0.76
18	0.8	0.55	0.42	0.70
19	1	0.64	0.45	0.73
20	1	0.58	0.50	0.71
21	1	0.65	0.41	0.70
22	1	0.63	0.55	0.73
23	1	0.68	0.49	0.72
24	1	0.55	0.43	0.68

Tabla 37. Resultados AG – Pórtico Asimétrico – Daño elementos 9-13-18 con 20%.

Elemento dañado:	1
Magnitud de daño:	20%

Elem	α_e	Prueba		
		1	2	3
	Fun_Obj =	14.21	13.05	15.06
	Iteraciones =	100	100	100
	Tiempo [seg] =	37664	3960	3944
	Tiempo [min] =	627.73	66.00	65.73
		α_{c1}	α_{c2}	α_{c3}
1	0.8	0.50	0.64	0.37
2	1	0.56	0.68	0.43
3	1	0.48	0.59	0.37
4	1	0.47	0.59	0.44
5	1	0.51	0.67	0.46
6	1	0.52	0.59	0.50
7	1	0.49	0.63	0.47
8	1	0.52	0.69	0.48
9	1	0.49	0.59	0.38
10	1	0.54	0.60	0.46
11	1	0.49	0.58	0.44
12	1	0.54	0.63	0.44
13	1	0.53	0.61	0.42
14	1	0.52	0.68	0.37
15	1	0.56	0.67	0.48
16	1	0.50	0.66	0.40
17	1	0.49	0.70	0.47
18	1	0.48	0.57	0.43
19	1	0.52	0.60	0.40
20	1	0.54	0.59	0.45
21	1	0.52	0.60	0.40
22	1	0.56	0.65	0.42
23	1	0.54	0.61	0.48
24	1	0.51	0.70	0.43

Tabla 38. Resultados AG – Pórtico Simétrico con ruido – Daño elemento 1 con 20%.

Elemento dañado:	20
Magnitud de daño:	20%

		Prueba		
		1	2	3
Fun_Obj =		15.63	16.12	16.30
Iteraciones =		100	100	100
Tiempo [seg] =		3750	3765	3755
Tiempo [min] =		62.50	62.75	62.58
Elem	α_e	α_{c1}	α_{c2}	α_{c3}
1	1	0.68	0.79	0.89
2	1	0.69	0.81	0.86
3	1	0.68	0.77	0.85
4	1	0.70	0.77	0.91
5	1	0.68	0.80	0.92
6	1	0.69	0.77	0.93
7	1	0.67	0.78	0.86
8	1	0.68	0.78	0.92
9	1	0.69	0.78	0.86
10	1	0.68	0.77	0.92
11	1	0.69	0.80	0.87
12	1	0.68	0.75	0.85
13	1	0.67	0.77	0.91
14	1	0.67	0.78	0.86
15	1	0.67	0.77	0.88
16	1	0.70	0.78	0.84
17	1	0.68	0.77	0.86
18	1	0.69	0.79	0.91
19	1	0.68	0.80	0.85
20	0.8	0.70	0.76	0.84
21	1	0.69	0.80	0.89
22	1	0.68	0.76	0.84
23	1	0.67	0.75	0.85
24	1	0.69	0.77	0.86

Tabla 39. Resultados AG – Pórtico Simétrico con ruido – Daño elemento 20 con 20%.

Elemento dañado:	9 - 10 - 18
Magnitud de daño:	20%

Elem	α_e	Prueba		
		1	2	3
	Fun_Obj =	12.97	13.45	12.58
	Iteraciones =	100	100	100
	Tiempo [seg] =	4419	4114	4009
	Tiempo [min] =	73.65	68.57	66.82
Elem	α_e	α_{c1}	α_{c2}	α_{c3}
1	1	0.60	0.66	0.84
2	1	0.69	0.66	0.85
3	1	0.68	0.75	0.83
4	1	0.63	0.65	0.84
5	1	0.62	0.72	0.85
6	1	0.71	0.75	0.85
7	1	0.59	0.64	0.82
8	1	0.57	0.71	0.82
9	0.8	0.63	0.67	0.82
10	0.8	0.65	0.67	0.82
11	1	0.69	0.76	0.81
12	1	0.67	0.66	0.81
13	1	0.67	0.75	0.83
14	1	0.61	0.79	0.84
15	1	0.59	0.65	0.83
16	1	0.66	0.68	0.83
17	1	0.63	0.68	0.83
18	0.8	0.70	0.67	0.84
19	1	0.73	0.74	0.84
20	1	0.67	0.62	0.81
21	1	0.70	0.69	0.81
22	1	0.72	0.66	0.86
23	1	0.57	0.71	0.82
24	1	0.65	0.70	0.84

Tabla 40. Resultados AG – Pórtico Simétrico con ruido – Daño elementos 9-10-18 con 20%.

Elemento dañado:	1
Magnitud de daño:	20%

		Prueba # 1	Prueba # 2	Prueba # 3
Función Objetivo =		18.12	18.51	19.07
Iteraciones =		5000	5000	5000
Tiempo [seg] =		186385	193950	178820
Tiempo [min] =		3106.4	3232.5	2980.3
Tiempo [horas] =		51.77	53.87	49.67
Elemento	α_e	α_{c1}	α_{c2}	α_{c3}
1	0.80	0.81	0.81	0.81
2	1.00	0.99	0.98	0.99
3	1.00	0.96	0.97	0.92
4	1.00	0.93	0.97	0.97
5	1.00	0.98	0.98	0.97
6	1.00	0.95	0.97	0.97
7	1.00	0.92	0.97	0.97
8	1.00	0.97	0.97	0.97
9	1.00	0.99	0.98	0.97
10	1.00	0.93	0.97	0.97
11	1.00	1.00	0.97	0.98
12	1.00	0.96	0.98	0.98
13	1.00	0.98	0.98	0.98
14	1.00	0.94	0.98	0.98
15	1.00	0.99	0.97	0.98
16	1.00	0.99	0.97	0.97
17	1.00	0.98	0.98	0.97
18	1.00	0.98	0.98	0.98
19	1.00	0.97	0.98	0.97
20	1.00	0.97	0.97	0.98
21	1.00	0.98	0.97	0.98
22	1.00	0.99	0.98	0.98
23	1.00	0.99	0.98	0.98
24	1.00	0.98	0.97	0.98

Tabla 41. Resultados AG – Pórtico Simétrico – Daño elemento 1 con 20% y máximo de 5000 iteraciones.