

# DETECCIÓN DE DAÑOS EN PÓRTICOS 3D EMPLEANDO INFORMACIÓN EN EL DOMINIO DE LA FRECUENCIA Y TÉCNICAS HEURÍSTICAS DE OPTIMIZACIÓN

Ing. FREDY SAÚL SOTELO MONROY

UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIRÍAS FÍSICO-MECÁNICAS  
ESCUELA DE INGENIERÍA CIVIL  
MAESTRÍA EN INGENIERÍA CIVIL  
BUCARAMANGA

2013

# DETECCIÓN DE DAÑOS EN PÓRTICOS 3D EMPLEANDO INFORMACIÓN EN EL DOMINIO DE LA FRECUENCIA Y TÉCNICAS HEURÍSTICAS DE OPTIMIZACIÓN

Ing. FREDY SAÚL SOTELO MONROY

Trabajo de grado para optar el título  
MÁSTER EN INGENIERÍA CIVIL

ORIENTADOR

Oscar Javier Begambre Carrillo IC, Msc, PhD

UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIRÍAS FÍSICO-MECÁNICAS  
ESCUELA DE INGENIERÍA CIVIL  
MAESTRÍA EN INGENIERÍA CIVIL  
BUCARAMANGA

2013

*A mis padres, Saúl y Alba*  
*A mis hermanos Yesit y Zamir*  
*A mi novia Janes*  
*A mis sobrinos, Mafe y Matías*

---

# AGRADECIMIENTOS

---

Agradezco a Dios, por darme la perseverancia y constancia para poder lograr cada uno de mis objetivos propuestos.

Al Profesor Oscar Begambre que me alentó a seguir en el estudio, el apoyo brindado en la orientación y el desarrollo de este trabajo, además sus aportes y sabios consejos.

Agradezco también a todos mis compañeros de maestría, en especial a Camilo Celis, Ludwing Pérez, Jackson Monroy y José Miguel Benjumea por sus comentarios y ayudas al desarrollo de esta investigación, además de por la compañía y amistad que me brindaron durante todos estos años.

Finalmente, expreso mi gratitud a la Universidad Industrial de Santander y a la Escuela de ingeniería Civil por brindarme el apoyo y la oportunidad de poder finalizar una etapa más en mi vida académica.

---

# ÍNDICE GENERAL

---

<b>INTRODUCCIÓN</b>	<b>15</b>
<b>OBJETIVO Y ORGANIZACIÓN DEL TRABAJO</b>	<b>16</b>
<b>1. DETECCIÓN DE DAÑOS EN ESTRUCTURAS: ANTECEDENTES</b>	<b>17</b>
1.1. TÉCNICAS BASADAS EN MODELOS (TBM) . . . . .	21
1.2. TÉCNICAS BASADAS EN SEÑALES (TBS) . . . . .	22
1.3. IDENTIFICACIÓN DE DAÑOS: PROBLEMAS Y ERRORES . . . . .	23
1.4. MÉTODOS DE IDENTIFICACIÓN DE DAÑOS . . . . .	25
<b>2. PROBLEMA DE REFERENCIA: PÓRTICO UBC IASC–ASCE</b>	<b>35</b>
2.1. DESCRIPCIÓN GENERAL DEL PROBLEMA DE REFERENCIA . .	35
2.2. PATRONES DE DAÑO . . . . .	38
2.3. FASE NUMÉRICA . . . . .	38
2.4. FASE EXPERIMENTAL . . . . .	41
<b>3. TÉCNICAS DE OPTIMIZACIÓN PARA DETECCIÓN DE DAÑOS</b>	<b>47</b>
3.1. PRINCIPIOS DE LAS METAHEURÍSTICAS MÁS CONOCIDAS . .	50
3.2. EVALUACIÓN DE METAHEURÍSTICAS . . . . .	59
<b>4. METODOLOGÍA PROPUESTA</b>	<b>71</b>
4.1. OBTENCIÓN DE PARÁMETROS MODALES A PARTIR DE LA IN- FORMACIÓN EXPERIMENTAL DE LA ESTRUCTURA . . . . .	72
4.2. MODELO NUMÉRICO ADAPTADO . . . . .	76
4.3. FORMULACIÓN DEL PROBLEMA DE OPTIMIZACIÓN . . . . .	82
4.4. AJUSTE DEL MODELO NUMÉRICO EMPLEANDO EL SL-PSO . .	84
4.5. IDENTIFICACIÓN DE LOS CASOS DE DAÑO DISPONIBLE A PAR- TIR DEL MODELO AJUSTADO, LA INFORMACIÓN EXPERIMEN- TAL Y EL SL-PSO . . . . .	86
4.6. MÉTODO DE ÍNDICE DE DAÑO . . . . .	89
<b>5. RESULTADOS</b>	<b>94</b>
5.1. AJUSTE DEL MODELO NUMÉRICO . . . . .	94

## ÍNDICE GENERAL

5.2. IDENTIFICACIÓN DE LOS CASOS DE DAÑOS . . . . .	106
<b>6. CONCLUSIONES</b>	<b>111</b>
<b>7. TRABAJOS FUTUROS</b>	<b>113</b>
<b>BIBLIOGRAFÍA</b>	<b>114</b>
<b>A. FUNCIONES DE PRUEBA</b>	<b>120</b>
A.1. FUNCIONES OPTIMIZACIÓN CONTINUA SIN RESTRICCIONES . .	120
A.2. FUNCIONES OPTIMIZACIÓN CONTINUA CON RESTRICCIONES .	123
<b>B. ALGORITMOS DE OPTIMIZACIÓN</b>	<b>129</b>
B.1. ALGORITMOS PARA OPTIMIZACIÓN CONTINUA . . . . .	129
B.2. ALGORITMOS PARA OPTIMIZACIÓN CON RESTRICCIONES . .	151
<b>C. ALGORITMO PARA LA DETECCIÓN DE DAÑOS</b>	<b>164</b>

---

## ÍNDICE DE FIGURAS

---

2.1. Estructura de referencia [1] . . . . .	36
2.2. Conexión viga-columna [2] . . . . .	37
2.3. Configuración de la masa en la estructura [1] . . . . .	37
2.4. Shaker sobre la estructura [2] . . . . .	41
2.5. Martillo de impacto usado [1] . . . . .	42
3.1. Forma de la función objetivo de un problema de optimización . . . . .	49
3.2. Diagrama de flujo del algoritmo genético . . . . .	52
3.3. Diagrama de flujo del algoritmo Simulated Annealing . . . . .	53
3.4. Diagrama de flujo del algoritmo PSO . . . . .	56
3.5. selección de $pbest_{f_i(d)}^d$ [3] . . . . .	58
3.6. EL mínimo de la función $f(\bar{x})$ es igual al maximo de la función $-f(\bar{x})$ . . . . .	62
3.7. Ejemplo hipotético para un problema con dos variables de diseño y cinco restricciones de desigualdad. . . . .	63
3.8. Esquema de manejo de restricciones. . . . .	66
4.1. Configuración de la prueba de Impacto [4] . . . . .	74
4.2. Configuración de la prueba de Excitador [4] . . . . .	74
4.3. Fuentes de parámetros modales [4] . . . . .	75
4.4. Elemento tipo viga de un pórtico espacial [5] . . . . .	78
4.5. Deformada de una viga con conexiones semi-rígidas [6] . . . . .	81
4.6. Esquema general del ajuste del modelo numérico . . . . .	85
4.7. Configuración II . . . . .	86
4.8. Configuración III . . . . .	87
4.9. Configuración IV . . . . .	87
4.10. Configuración V . . . . .	88
4.11. Configuración VI . . . . .	88
4.12. Configuración VII . . . . .	89
5.1. Frecuencias ajustadas modelo rígido 116D E[0-1] . . . . .	96
5.2. Factores de ajuste modelo rígido 116D E[0-1] . . . . .	96
5.3. Frecuencias ajustadas modelo rígido 116D E[0.9-1.1] . . . . .	97
5.4. Factores de ajuste modelo rígido 116D E[0.9-1.1] . . . . .	98

## ÍNDICE DE FIGURAS

5.5.	Frecuencias ajustadas modelo semi-rígido 116D E[0-1] . . . . .	99
5.6.	Factores de ajuste modelo semi-rígido 116D E[0-1] . . . . .	100
5.7.	Frecuencias ajustadas ajuste modelo semi-rígido 116D E[0.9-1.1] . . . .	101
5.8.	Factores de ajuste modelo semi-rígido 116D E[0.9-1.1] . . . . .	101
5.9.	Ajuste modelo semi-rígido 17D E[0.9-1.1] L[0.9-1.1] $\mu$ [6-24] . . . . .	103
5.10.	Ajuste modelo semi-rígido 6D E[0.9-1.1] L[0.9-1.1] $\mu$ [6-24] . . . . .	105



---

## ÍNDICE DE TABLAS

---

2.1.	Propiedades de la estructura de referencia [1] . . . . .	36
2.2.	Descripción de los patrones de daño . . . . .	39
2.3.	Características de la instrumentación [1] . . . . .	42
2.4.	Determinación de frecuencias naturales $[Hz]$ usando vibración ambiental [7] . . . . .	43
2.5.	Determinación de frecuencias naturales $[Hz]$ usando prueba de impacto de martillo [7] . . . . .	43
2.6.	Determinación de frecuencias naturales $[Hz]$ usando vibración ambiental [8] . . . . .	44
2.7.	Determinación de frecuencias naturales $[Hz]$ usando prueba de impacto de martillo [8] . . . . .	44
2.8.	Determinación de frecuencias naturales $[Hz]$ usando prueba de impacto de martillo [9] . . . . .	45
2.9.	Determinación de frecuencias naturales $[Hz]$ [2] . . . . .	45
3.1.	Funciones de prueba para optimización continua sin restricciones . . . .	61
3.2.	Funciones de prueba para optimización continua con restricciones . . . .	64
3.3.	Resultados AG . . . . .	67
3.4.	Resultados SA . . . . .	67
3.5.	Resultados PSO . . . . .	68
3.6.	Resultados SL-PSO . . . . .	68
3.7.	Comparación de las mejores soluciones encontradas por el AG, SA, PSO y S-LPSO . . . . .	68
3.8.	Resultados SL-PSO . . . . .	69
3.9.	Comparación de las mejores soluciones encontradas por el SL-PSO con el AG [10], SA [11] y PSO [12]. . . . .	70
4.1.	Descripción de los códigos fuente del programa Datagen . . . . .	77
4.2.	Clasificación de la conexiones [6] . . . . .	82
5.1.	Elementos que componen la estructura . . . . .	95
5.2.	Clasificación de la conexiones [6] . . . . .	99
5.3.	Variables para el ajuste modelo semi-rígido 17D . . . . .	102

## ÍNDICE DE TABLAS

5.4. Factores ajuste modelo semi-rígido 17D . . . . .	103
5.5. Variables de ajuste para el ajuste modelo semi-rígido 6D . . . . .	104
5.6. Factores ajuste modelo semi-rígido 6D . . . . .	104
5.7. Resultados modelos ajustados . . . . .	105
5.8. Frecuencias ajustadas para los casos de daño . . . . .	107
5.9. Modos de vibración . . . . .	108
5.10. Indices de daño . . . . .	109
5.11. Resultados casos de daño . . . . .	110
5.12. Indices de daño [13] . . . . .	110

---

## RESUMEN

---

**TITULO:**

DETECCIÓN DE DAÑOS EN PÓRTICOS 3D EMPLEANDO INFORMACIÓN EN EL DOMINIO DE LA FRECUENCIA Y TÉCNICAS HEURÍSTICAS DE OPTIMIZACIÓN \*

**AUTOR:**

FREDY SAUL SOTELO MONROY. \*\*

**PALABRAS CLAVES:**

Detección de Daños, técnicas heurísticas, ajuste de modelos.

**DESCRIPCIÓN:**

La identificación de daños es de gran importancia dado que, en la ingeniería civil, uno de los campos de investigación de mayor relevancia es la búsqueda de estructuras más seguras, donde la detección de daños se orienta a determinar si la estructura bajo estudio ha sufrido alguna alteración que pueda comprometer su funcionalidad y seguridad. En este contexto, el objetivo de esta investigación es desarrollar una metodología para la identificación de daños en pórticos tridimensionales empleando información en el dominio de la frecuencia y técnicas heurísticas de optimización, dicha metodología fue implementada en la estructura de referencia propuesta por la IASC y la ASCE, donde el daño fue simulado removiendo varios elementos de la estructura y la respuesta dinámica del sistema para cada caso de daño fue medida experimentalmente. El ajuste de modelos fue empleado tanto para la calibración del modelo inicial como para los casos de daño, los cuales fueron abordados como un problema de optimización, donde se utilizó el algoritmo SL-PSO dado que mostro mejor desempeño en funciones de prueba comparado con tres técnicas (SA, Ag y PSO) de amplio conocimiento en la literatura. La identificación fue realizada por medio de un índice de daño por piso donde se logró determinar una adecuada estimación de la presencia de daño en cada uno de los casos estudiados. Finalmente, se compararon los resultados con un trabajo reportado en la literatura especializada.

---

\*Trabajo de Grado para optar el título de Máster en Ingeniería civil

\*\*Facultad de Ingenierías Físico-mecánicas, Escuela de Ingeniería Civil. Director: Oscar Begambre

---

# ABSTRACT

---

**TITLE:**

DAMAGE DETECTION IN THREE DIMENSIONAL FRAMES USING INFORMATION IN FREQUENCY DOMAIN AND HEURISTIC OPTIMIZATION TECHNIQUES \*

**AUTHOR:**

FREDY SAUL SOTELO MONROY \*\*

**KEYWORDS:**

Damage detection, heuristics, updating models.

**DESCRIPTION:**

The damage identification is very important because, in civil engineering, one of the research areas most relevant is the search for safe structures where damage detection guide to determine whether the structure under study has experiment some changes which would compromise their functionality and security. In this context, the aim of this research is to develop a methodology for damage identification in three dimensional frames using information in the frequency domain and heuristic optimization techniques, this methodology was implemented in the reference structure proposed by the IASC- ASCE, where damage was simulated by removing various elements of the structure and the dynamic response of the system for each case of damage was measured experimentally. The updating model was used for both the calibration of the initial model and the case of damage, which were formulated as an optimization problem, where the algorithm SL-PSO was used since it showed better performance on test functions compared with three techniques (SA, AG and PSO) of great knowledge in the literature. The identification did do by means of a story damage index where it was possible to determine an appropriate estimate of the presence of damage in each of the cases studied. Finally, the results were compared with a work reported in the literature.

---

\*Graduate Project to obtain the degree of Master in Civil Engineering

\*\*Department of Physical and Mechanical Engineering. School of Civil Engineering. Thesis advisor: Oscar Begambre

---

# INTRODUCCIÓN

---

En la ingeniería civil, uno de los campos de investigación de mayor relevancia es la búsqueda de estructuras más seguras. En este contexto, la detección de daños (D.D.) se orienta a determinar si la estructura bajo estudio ha sufrido alguna alteración que pueda comprometer su funcionalidad y seguridad. Además, la temprana detección y ubicación del daño permite minimizar los costos de mantenimiento y reparación dado que estas actividades pueden ser programadas adecuadamente.

Los primeros métodos de detección de daños eran experimentales, sin embargo, dichas técnicas requerían que la localización de los daños fuera conocida con anticipación y que la parte de la estructura a inspeccionar fuera accesible [14]. Además de estas limitaciones, estos métodos solo podían detectar daños en o cerca de la superficie de la estructura.

Actualmente, gran parte de las metodologías de monitoreo y detección de daños, se fundamentan en técnicas de ajuste de modelos (tanto numéricos como matemáticos), donde se realiza la comparación entre las respuestas del modelo numérico con las mediciones de la estructura experimental y así poder determinar si ha ocurrido daño y ubicar su posible localización.

Un enfoque innovador para solucionar problemas de ajuste de modelos y de D.D. es a través de técnicas de programación no lineal. Por este motivo, la optimización numérica empleando metaheurísticas (AG, SA, PSO, etc) vienen siendo empleadas con éxito en los últimos años. La ventaja de la optimización estocástica sobre las técnicas tradicionales es que pueden abordar problemas con soluciones multimodales, típicas de la identificación de daños

Debido a la multiplicidad de estudios sobre D. D. la IASC y la ASCE decidieron proponer un problema de referencia que posibilitara la evaluación y comprensión de diversas estrategias para identificación de daños. Por lo tanto, en este trabajo, se propondrá una metodología para D. D. validada con el problema de referencia del IASC-ASCE. Básicamente, la estructura de referencia consiste en un pórtico espacial de acero al cual se le realizaron mediciones experimentales para varios casos de daños. De acuerdo a lo anterior, y con la información disponible de la estructura (respuesta dinámica experimental, configuración, geometría y tipología) se desarrollara una metodología que incluirá una técnica de optimización con el fin de realizar la identificación de daños.

---

# OBJETIVO Y ORGANIZACIÓN DEL TRABAJO

---

Este trabajo tiene como finalidad desarrollar una metodología para ser implementada en la detección de daños en estructuras de acero a partir de la información en el dominio de la frecuencia y técnicas de optimización, y se encuentra organizado en los siguiente capítulos.

En el primer capítulo se presenta el estado del arte en la identificación de daños. En este sentido, se realiza una descripción general de la importancia de una temprana identificación junto con sus potencialidades y dificultades al momento de ser implementada, además se realiza una recopilación de los métodos más utilizados para abordar esta tipología de problemas.

El segundo capítulo está dedicado a la descripción detallada de problema de referencia del IASC-ASCE. Adicionalmente se presenta una recopilación de los trabajos más relevantes que se han realizado en dicha estructura.

En el tercer capítulo se presenta una descripción de las técnicas de optimización estocásticas empleadas para realizar la D. D. Se establecen diversas funciones matemáticas de benchmark seleccionadas de la literatura especializada en problemas de optimización (con y sin restricciones), las cuales son utilizadas para establecer la técnica de mejor desempeño que será empleada en el problema de detección de daños.

El cuarto capítulo corresponde a la metodología desarrollada la cual es implementada en el problema de identificación de daños. Además, en esta sección se realiza la formulación del problema de optimización, se definen las funciones objetivos que serán utilizadas y se establece el procedimiento de estimación del daño.

En el quinto capítulo se presentan los resultados identificación de daños de acuerdo a la metodología establecida. Se realiza una comparación con otras investigaciones realizadas en la misma estructura con el fin de determinar las bondades y dificultades del procedimiento abordado.

Finalmente se presentan las conclusiones de esta investigación, y algunas orientaciones y recomendaciones para la realización de futuros trabajos de investigación.

---

# DETECCIÓN DE DAÑOS EN ESTRUCTURAS: ANTECEDENTES

---

En términos generales, se puede considerar que los daños se inician al nivel del material, por lo tanto se puede definir como un defecto que se encuentra presente en algún grado en todo el material, donde bajo apropiados escenarios de carga, estos defectos pueden crecer y causar daños al nivel del sistema estructural, sin embargo, el termino de daño no implica necesariamente una total perdida de la funcionalidad del sistema, sino que el sistema ya no esta operando en su manera óptima, y si el daño crece, se llegará a un punto donde afecta la funcionalidad del sistema donde ya no es aceptado por el usuario. Este punto se conoce como falla.

En escalas de tiempo, el daño se puede acumular gradualmente durante largos períodos de tiempo, como lo es en el caso de daños por acumulación de fatiga o corrosión. Por otro lado, a escalas cortas de tiempo, el daño también puede resultar en eventos discretos programados, tales como aterrizajes de aviones, y de eventos discretos no programados, como los fenómenos naturales [15].

Dada la importancia de este campo de investigación, se ha desarrollado la implementación de estrategias para la identificación de daños, que hacen referencia al monitoreo de la salud estructural “*Structural Health Monitoring*” (SHM), las cuales han sido enfocadas principalmente a infraestructuras civiles, al sector espacial, a la industria del petróleo y al campo de la ingeniería mecánica. En este tipo de metodologías, el daño es definido como cambios en: las propiedades del material, geometría, condiciones de contorno y de conectividad; los cuales afectan adversamente el desempeño del sistema estructural [15]. Una amplia variedad de herramientas se han desarrollado en las ultimas décadas debido al rápido incremento en las investigaciones relacionadas con el SHM, dados los beneficios económicos y de seguridad que provee [14].

La identificación de daños teniendo en cuenta los cambios de la respuesta del sistema, fue inicialmente practicada de manera cualitativa, empleando métodos visuales o métodos experimentales como: métodos acústicos o ultrasónicos, métodos de campo magnético, radiografías, métodos de corriente inducida “*eddy-current*” y métodos de campo térmico[14]. Para la implementación de dichas técnicas, se requiere que la ubi-

## DETECCIÓN DE DAÑOS EN ESTRUCTURAS: ANTECEDENTES

cación del daño sea conocida con anticipación y que la porción de la estructura a inspeccionar sea de fácil acceso. Sujetos a estas limitaciones, los métodos experimentales solo pueden detectar daños en o cerca de la superficie de la estructura. Por lo tanto, la necesidad de métodos adicionales para la detección de daños a nivel global que puedan ser aplicados a estructuras complejas ha dado lugar al desarrollo de métodos que examinen los cambios en las características dinámicas o respuesta de la estructura, donde la idea básica es emplear la relación que existe entre los parámetros modales (frecuencias, formas modales y amortiguamiento modal) y las propiedades físicas de la estructura (masa, amortiguamiento y rigidez) [14], es decir, que el daño causa cambios en la distribución de las propiedades físicas de la estructura, los cuales podrían ser detectados por medio de la medición de la respuesta dinámica [16].

Inicialmente, este tipo de problemas comenzó a ser estudiado durante los años setenta y ochenta, donde la industria del petróleo hizo un considerable esfuerzo para desarrollar metodologías de identificación de daños basadas en vibraciones para las plataformas en alta mar, pero dado que la localización del daño era desconocida, la metodología adoptada por esta industria fue la simulación de escenarios de daños con modelos numéricos, donde se examinaban los cambios en las frecuencias de resonancia que eran producidas en los modelos, los cuales posteriormente se correlacionaban con las mediciones experimentales realizadas en las plataformas, sin embargo, un número de problemas prácticos fueron encontrados, incluyendo: dificultades en las mediciones causadas por el ruido, dificultades de la instrumentación en ambientes hostiles, cambios de masa por el crecimiento del oleaje, variación de los niveles de almacenamiento de fluidos, variabilidad temporal en la condiciones de la fundación y la incapacidad de excitación de los modos mas altos; por tal motivo se impidió la adaptación de esta tecnología y los esfuerzos de seguir desarrollando esta tecnología para las plataformas en alta mar fueron abandonadas en gran medida a principios de los años ochenta [17].

Por otro lado, la comunidad aeroespacial comenzó a estudiar el uso de la identificación de daños basadas en vibraciones durante los finales de los setenta y principios de los ochenta en conjunto con el desarrollo del transbordador espacial, desarrollando el sistema de inspección modal de la lanzadera espacial “*shuttle modal inspection system*” (SMIS) para identificar el daño por fatiga en: los componentes, las superficies de control, paneles de fuselaje y las superficies de elevación; y dado que estas áreas estaban recubiertas con un sistema de protección térmica eran inaccesibles y, poco prácticos para examinación convencional por medio de métodos locales no destructivos, por lo tanto, los enfoques desarrollados estaban basados en la correlación de modelos analíticos de la estructura sin daño con las mediciones de las propiedades modales de la estructura sin daño y con daño, donde los cambios en la rigidez según la evaluación de los dos modelos ajustados se usaban para localizar y cuantificar el daño [15].

La comunidad de la ingeniería civil ha estudiado la identificación de daños basada en vibraciones en estructuras de puentes y edificios desde los principios de los ochenta. En estos estudios, las propiedades modales y cantidades derivadas de estas propiedades, como la curvatura de las formas modales y la medición de los índices de la flexibilidad dinámica, fueron las primeras características usadas para identificar el daño. Sin em-



## DETECCIÓN DE DAÑOS EN ESTRUCTURAS: ANTECEDENTES

bargo la variabilidad de las condiciones ambientales y de operación presentaron inconvenientes significativos para la aplicación del monitoreo de puentes, además el tamaño físico de la estructura presentaba muchas dificultades prácticas para la evaluación de daños basado en vibraciones [15].

Idealmente, un esquema robusto de detección de daños debe ser capaz de: identificar la ocurrencia del daño en un estado muy temprano, localizar el daño de acuerdo con los sensores empleados, proporcionar una severidad del daño y realizar una predicción de la vida útil restante de la estructura. Además, es fundamental que se realice una comparación entre dos estados diferentes de la estructura, en la cual se deba realizar una medición inicial de la respuesta de sistema en buen estado (sin daño) y tomarlo como referencia para futuras comparaciones. Adicionalmente, los métodos deben ser capaces de tomar en cuenta las limitaciones operacionales, por ejemplo, una suposición común en la mayoría de los métodos de detección de daños es que la masa de la estructura no cambia apreciablemente como resultado del daño; sin embargo, hay cierto tipo de estructuras como las plataformas petroleras en alta mar donde esta suposición no es válida. Otra importante característica de los métodos de identificación de daños, y específicamente los métodos que emplean modelos, es su habilidad para discriminar entre la discrepancia modelo/datos causada por errores en la modelación y como resultado del daño estructural [14].

Muchos métodos de detección de daños están basados en la suposición de que la estructura puede ser modelada con un comportamiento lineal elástico después de la ocurrencia del daño, aunque también existen algunos métodos no lineales [14]. En el caso de técnicas lineales de detección de daños, las variaciones de las características resultan en cambios de la rigidez, condiciones de contorno o propiedades de los materiales, donde la respuesta estructural en el estado de daño puede ser analizada usando un modelo lineal elástico diferente del inicial sin daño [18]. El caso no lineal es definido cuando la estructura con comportamiento lineal elástico se comporta de una manera no lineal después de que el daño ha sido introducido, un ejemplo de daño no lineal es la formación de grietas por fatiga que posteriormente se abren y cierran bajo la condición normal de funcionamiento [19].

La identificación de daños es llevada a cabo a través de un conjunto de cinco disciplinas estrechamente relacionadas que incluye: el SHM, el monitoreo de la condición “*Condition Monitoring*” (CM), la evaluación no destructiva “*Non-Destructive evaluation*” (NDE), el control de procesos estadísticos “*Statistical Process control*” (SPC) y el pronóstico de daños “*Damage Prognosis*” (DP). Típicamente, el SHM está asociado con la identificación global de daños en sistemas estructurales tales como edificios y aviones. El CM es análogo al SHM, pero la identificación de daños está dirigido a máquinas rotativas y oscilatorias. La NDE es usualmente llevada a cabo de manera local después de que el daño ha sido localizado. Hay excepciones en esta regla, dado que NDE también es usada como herramienta de monitoreo para estructuras in situ como recipientes a presión o tuberías. La NDE es por lo tanto principalmente usada para la caracterización y la revisión de la severidad del daño cuando hay un conocimiento a priori de su localización. El SPC está basado en procesos estadísticos y no basado en

## DETECCIÓN DE DAÑOS EN ESTRUCTURAS: ANTECEDENTES

modelos y emplea una variedad de sensores para el monitoreo de los cambios de dichos procesos, uno de los cuales puede resultar del daño estructural. Una vez el daño ha sido detectado, el DP es usado para predecir la vida útil restante del sistema [15].

Generalmente, las metodologías de detección de daños se aplican teniendo en cuenta dos tipos de técnicas las cuales pueden ser clasificadas como: técnicas basadas en modelos y técnicas basadas en señales, llamadas paramétricas y no paramétricas respectivamente. En el primer caso se supone que la estructura responde de una manera predeterminada por lo que puede ser modelada por medio de un análisis de elementos finitos (EF), mientras que en el segundo caso no es necesario un modelo detallado de la estructura pero se debe poseer la disponibilidad de los datos de las mediciones de la estructura sin daño. En ocasiones, la falta de disponibilidad de este tipo de datos pueden hacer a un método poco práctico para ciertas aplicaciones [15, 18].

Un sistema de clasificación para los métodos de identificación de daños, es presentado por Ritter [20], definiendo cuatro niveles de identificación de daños de la siguiente manera:

- Nivel 1: Determinación de que el daño esta presente dentro de la estructura.
- Nivel 2: Determinación de la localización geométrica del daño.
- Nivel 3: Cuantificación de la severidad del daño.
- Nivel 4: Predicción de la vida útil de la estructura.

En el caso de emplearse una metodología de detección de daños usando solamente la respuesta de la estructura, se puede llegar a determinar la existencia del daño en el sistema. Sin embargo, si se tiene acceso tanto a mediciones de la respuesta experimental como del modelo analítico de la estructura, los procedimientos se puede utilizar para determinar el tipo, la extensión del daño y la vida útil restante de la estructura [15, 14].

La literatura en el campo del SHM puede ser clasificada en métodos de nivel 1, nivel 2 y nivel 3 porque estos están más relacionados directamente con las pruebas de dinámica estructural y problemas de modelado. El nivel 4 de predicción esta generalmente categorizado con el campo de la mecánica de la fractura, análisis de la vida de fatiga, o evaluación del diseño estructural. De esta manera, la revisión de la literatura concerniente a la detección, localización y caracterización del daño estructural a través de métodos que examinan cambios en las mediciones de la respuesta de la vibración de la estructura incluye: cambios en las frecuencias, cambios en las mediciones de las formas modales y sus derivadas, cambios en las mediciones de los coeficientes de flexibilidad, métodos que emplean la propiedad de ajuste de modelos, detección de la respuesta no lineal, y detección de daños por medio de redes neuronales, entre otros [14].

A continuación, se presenta una breve reseña de algunas de las técnicas y metodologías que han sido ampliamente implementadas en el problema de detección de daños.

## 1.1. TÉCNICAS BASADAS EN MODELOS (TBM)

En situaciones prácticas es casi imposible tener suficientes datos experimentales para caracterizar completamente los estados de la estructura, ya que el costo de estas pruebas, generalmente impide su realización. La anterior es quizá la razón principal por la que los modelos numéricos son ampliamente utilizados en la ingeniería debido a que permiten el estudio, con un costo relativamente bajo, de los fenómenos de interés que no puede medirse directamente. Este argumento puede ser reforzado por el aumento de la eficiencia de los procesadores, la memoria y la disponibilidad de técnicas de computación paralela de la que se dispone actualmente, debido a que los métodos numéricos son empleados para simular, paso a paso, el comportamiento de los grandes sistemas estructurales, lo que permite la predicción, comparación y validación de los resultados del modelo permitiendo su uso en la solución del problema de detección de daños.

Los datos medidos durante el monitoreo de una estructura se encuentran en el dominio del tiempo, sin embargo, se puede emplear la Transformada discreta de Fourier para la transformar datos en el dominio del tiempo al dominio de la frecuencia. La respuesta en el dominio del tiempo (series de tiempo) puede ser obtenido de acelerómetros o sensores piezoeléctricos. Las funciones de respuesta de frecuencias (FRFs) se obtienen en el caso de conocer tanto la excitación como la respuesta, utilizando la Transformada Rápida de Fourier “*Fast Fourier Transform*” (FFT). La FRFs en sistemas lineales, sirven de base para el cálculo de los parámetros modales (frecuencias naturales, formas modales y amortiguación), de tal manera que el modelo teórico reproduce de la mejor manera posible los datos experimentales.

Generalmente el método del Elementos Finitos (EF) es utilizado como herramienta para analizar el comportamiento dinámico del sistema estructural. Sin embargo, como las predicciones efectuadas con el uso del modelo difieren de los resultados experimentales obtenidos a partir de los ensayos de vibración de la estructura existente, es necesario realizar lo que se conoce como el ajuste del modelo “*Model Updating*” el cual busca efectuar correcciones paramétricas en el modelo de elementos finitos con el fin de lograr que las propiedades dinámicas analíticas se asemejen a sus contrapartes experimentales [21], para así poder realizar una mejor predicción de daños a partir del modelo numérico.

En este ámbito Zapico et al. [22] propusieron un ajuste de modelos de elementos finitos en el campo de dinámica estructural, en el cual se plantea la minimización de una función definida en el dominio del tiempo, donde la función error entre datos experimentales y datos teóricos es minimizada mediante un algoritmo de muestreo adaptativo. Los autores emplearon los algoritmos en dos problemas diferentes, una simulación numérica y un caso experimental. En los casos estudiados, el desempeño del algoritmo adaptativo propuesto fue mejor que otros algoritmos estocásticos con los que se comparó.

## 1.2. TÉCNICAS BASADAS EN SEÑALES (TBS)

La finalidad de estos métodos es emplear solamente datos experimentales para resolver el problema de la detección de daños, y por lo tanto poder prescindir de la utilización de modelos numéricos. El propósito de esta técnica es establecer un conjunto de características para definir la condición normal de la estructura y utilizarlos como referencia, para así poder establecer un nivel de funcionamiento normal (sin daño) del sistema. Durante un proceso de monitoreo posterior a la estructura, las nuevas características se estiman y se realiza una comparación con las de referencia, y en el momento de la generación de cualquier desviación significativa, un algoritmo indicara la presencia de la anomalía.

En este campo, el proceso de SHM se puede definir en términos de un paradigma de reconocimiento de patrones estadísticos de cuatro pasos, los cuales incluyen [15]:

- Evaluación operativa.
- Adquisición de datos, normalización y limpieza.
- Selección de las características y condensación de la información.
- Desarrollo de modelos estadísticos para la discriminación de las características.

La evaluación operativa es la que fija los límites para el seguimiento y el control que se llevará a cabo teniendo en cuenta las condiciones, tanto operativas como medioambientales, en las cuales el sistema a monitorear está funcionando y cuales son las limitaciones en la adquisición de datos [15]. La parte de la obtención de los datos consiste en seleccionar los métodos de excitación, el tipo, número, ubicación, de los sensores, y el hardware para la adquisición, almacenamiento y transmisión de la información [23], sin embargo, las consideraciones económicas desempeñarán un papel importante en la toma de estas decisiones, además el intervalo en el que los datos deben recogerse es otra consideración que debe ser tomada en cuenta. Como los datos se puede medir en diversas condiciones, se vuelve muy importante para el proceso de identificación que se normalicen, con la finalidad de separar las lecturas de las condiciones operativas de la ambientales. Por otro lado, la limpieza de la información, consiste en la selección de los datos a transmitir o rechazar para el proceso de la identificación de las características, el cual es el aspecto que recibe la mayor atención en la literatura técnica, dado que permite distinguir entre la estructura en buen estado y la estructura dañada. Inherente al proceso de selección de características está la condensación de los datos. Dentro de los métodos de extracción de las características de sistema, un aspecto muy común, es la correlación de las medidas de la respuesta del sistema (amplitud de la vibración o frecuencia), con las observaciones del sistema con daño. El desarrollo de modelos estadísticos se refiere a la implementación de algoritmos que operan sobre las características extraídas para cuantificar el estado de daño de la estructura. También dichos modelos son utilizados para reducir al mínimo las falsas indicaciones de daños,

las cuales caen en dos categorías: indicación de los daños de falsos positivos (indicación del daño cuando no está presente) y la indicación de los daños de falsos negativos (no hay indicación de daño cuando el daño está presente) [15].

Entre los temas importantes están las consideraciones de la excitación y la medición, incluyendo la selección del tipo y ubicación tanto de los sensores como de la excitación [24]. Otro tema importante es el procesamiento de señales, el cual incluye métodos como el análisis de Fourier, el análisis tiempo-frecuencia y el análisis de wavelet [14].

Posenato, et al. [25] presentaron dos metodologías para la interpretación de datos, para identificar y localizar algún comportamiento anómalo en la estructura. Los autores tuvieron en cuenta dos métodos estadísticos basados en: análisis de componentes principales móviles “*moving principal component analysis*” y análisis de regresión robusta “*robust regression analysis*”, ya que habían demostrado ser útiles para la detección de daños durante el monitoreo continuo estático de estructuras civiles. Las metodologías fueron aplicadas a elementos simulados numéricamente con sensores con un rango de ruido en las mediciones. Además realizaron un estudio comparativo con otros análisis estadísticos, en los cuales, la metodología propuesta presentó un mejor desempeño para la detección de daños. Adicionalmente, realizaron un enfoque para manejar los valores atípicos y faltantes, comunes en el SHM. Las metodologías fueron validadas en dos estructuras a gran escala. Los resultados demostraron la capacidad de la metodología para identificar abruptos cambios permanentes en el comportamiento del sistema estructural.

### 1.3. IDENTIFICACIÓN DE DAÑOS: PROBLEMAS Y ERRORES

Dentro de la literatura, se han indicado algunos de los problemas con la identificación de daños. Siempre hay errores en los datos medidos y en el modelo numérico que afectan las metodologías de identificación. Por lo tanto estos errores, y la adecuación de los datos, deben ser discutidos. Además las metodologías de identificación de daños deberían ser probados en ejemplos reales de experimentación, dado que muchos métodos que funcionan bien en los datos simulados a menudo fallan debido a los problemas señalados, sin embargo, antes de la implementación experimental se podría considerar la simulación de dichos problemas incluyendo ruido en los modelos numéricos.

#### 1.3.1. ERRORES EN MODELACIÓN

Uno de los principales problemas en la localización del daño es la dependencia en el modelo de elementos finitos. Este modelo es también un factor importante, porque un conjunto muy incompleto de datos medidos requiere información adicional del modelo para poder identificar la localización de los daños. Sin duda habrán errores, incluso en el modelo de la estructura en buen estado. Así, si las mediciones de la estructura dañada son usadas para identificar la localización del daño, los métodos tendrían gran dificultad

## DETECCIÓN DE DAÑOS EN ESTRUCTURAS: ANTECEDENTES

para distinguir entre las ubicaciones del daño real y los generados por los errores en el modelo original. Por lo tanto, si los parámetros adecuados no son incluidos para evitar los errores del modelo en buen estado, el resultado será un error sistemático entre el modelo y los datos. Además, es muy probable que los errores en el modelo original produzcan cambios en las frecuencias que son mucho mayores que las producidas por el daño. Hay dos enfoques básicos para reducir este problema, aunque ambos se basan en los datos medidos a partir de una estructura en buen estado. El primero es ajustar el modelo de elementos finitos de la estructura en buen estado para producir un modelo confiable. Obviamente la calidad de la evaluación de la ubicación del daño depende fundamentalmente del modelo ajustado. La segunda alternativa emplea las diferencias entre los datos de la respuesta con daño y sin daño en la estrategia de identificación [26, 27].

Otra fuente potencial de error es la discrepancia entre los lugares de medición y los grados de libertad del modelo. Esta discrepancia hace que la comparación directa de las funciones de respuesta de frecuencia y formas modales sea imposible, y la generación de residuos, inexacta. La magnitud de los errores involucra la dependencia de la densidad de la malla en la región del sensor y la complejidad de las formas modales. La mejor solución es asegurarse que en los nodos del modelo exista la ubicación de los sensores. Alternativamente se pueden utilizar técnicas de interpolación [26].

### 1.3.2. EFECTOS NO ESTACIONARIOS Y MEDIOAMBIENTALES

Un aspecto muy difícil en la evaluación de los daños es el cambio en los datos medidos debido a los efectos medioambientales (cambios de temperatura, mediciones de ruido, daño progresivo, etc) [16], que presentan efectos indeseables no estacionarios y que hace muy difícil la ubicación del daño [26].

Los efectos medioambientales típicos se manifiestan en puentes, especialmente en los construidos con hormigón, los cuales han sido objeto de muchos estudios en la ubicación de los daños. Por ejemplo, los cambios de temperatura pueden causar cambios significativos en las propiedades de rigidez de un puente, y la dificultad es predecir los efectos de la temperatura de las mediciones disponibles [26].

### 1.3.3. EFECTOS DEL RANGO DE FRECUENCIAS

El rango de las frecuencias empleadas en la identificación del daño tiene una gran influencia sobre los resultados. La gran ventaja en el uso de la medición de vibraciones de baja frecuencia, es que los modos asociados a dichas frecuencias suelen ser globales [17]. El problema con los modos de baja frecuencia es que las longitudes de onda de los modos son grandes, y típicamente son mucho más grandes que la magnitud de los daños, por lo tanto, el sistema para la identificación de daños requiere que haya un cambio significativo en la respuesta entre dos sitios adyacentes a posibles daños. Por

lo tanto, si los modos de baja frecuencia son usados, entonces esta resolución está estrechamente relacionada con las longitudes de onda de los modos. Usando excitaciones de alta frecuencia se puede llegar a utilizar modos muy locales los cuales son capaces de localizar con precisión los daños, pero sólo muy cerca de la posición del sensor y actuador [26].

### 1.3.4. MAGNITUD DEL DAÑO

Un problema frecuente que se presenta en la metodologías de detección de daños paramétricas, es la necesidad de un modelo matemático muy preciso, de modo que capte correctamente el comportamiento dinámico estructural real en una cierta gama de frecuencia predeterminadas. A menudo, en el SHM los cambios en las cantidades medidas causados por el daño estructural son más pequeñas que las observadas entre los de la estructura en buen estado y el modelo matemático. En consecuencia, se presenta gran dificultad en discernir entre los modelos inadecuados y los cambios reales debido a los daños. Hay dos enfoques alternativos para abordar este problema [26].

El primero es ajustar el modelo sin daño de modo que la correlación entre el modelo y los datos medidos sea mejorado. Este enfoque requiere que los errores que permanecen después del ajuste sean de menor magnitud que los cambios debido a los daños. Además, los cambios en el modelo deben ser físicamente significativos, de modo que el proceso de ajuste pueda corregir los errores del modelo real, y no se limite a reproducir los datos medidos [26].

El segundo enfoque se basa en el uso de la diferencia entre los datos medidos en la estructura sin daño y potencialmente dañada. En este caso, se asume que los únicos cambios en la estructura se deben a los daños, y por lo tanto el problema puede reducirse a la búsqueda de los parámetros que reproducen los cambios medidos [26].

## 1.4. MÉTODOS DE IDENTIFICACIÓN DE DAÑOS

Un resumen de la literatura pertinente de varios métodos para la identificación de daños y el monitoreo de la salud estructural basado en los cambios en las mediciones de sus propiedades dinámicas es presentado en esta sección. Los métodos están clasificados de acuerdo al tipo de datos medidos usados y la técnica empleada para la identificación. A continuación se presenta una descripción de los métodos con cada una de sus características.

### 1.4.1. MÉTODOS BASADOS EN LOS CAMBIOS DE FRECUENCIA

La mayoría de la literatura relacionada con la detección de daños usando cambios en las frecuencias naturales es bastante extensa. Además, la relación que existe entre los cambios en las propiedades físicas de la estructura con los cambios en las frecuencias,



junto con su facilidad de medición (solo un sensor es necesario en muchas aplicaciones) impulsaron el uso de métodos modales para la identificación de daños y el monitoreo de la salud estructural [17, 28].

Los métodos que se basan en cambios de las frecuencias modales tienen limitaciones prácticas importantes, ya que estas son una propiedad global de la estructura, y por ende, no es muy claro que cambios en este tipo de propiedades sean útiles para determinar mas allá de la existencia de daños. Por lo anterior, es de esperarse que los métodos basados en este criterio, no sean capaces de proporcionar información espacial de los cambios en la estructura sobre la que se implementan. Bajo este criterio de identificación se han desarrollado métodos que caen en dos categorías principales, las cuales se han designado como el problema hacia adelante y como el problema inverso [14, 29].

### Problema hacia adelante

Estos métodos caen usualmente en el Nivel 1 (determinación de la presencia de daño en la estructura) de la identificación de daños y consisten en calcular el cambio en las frecuencias a partir de un tipo de daño conocido. Comúnmente el daño se modela matemáticamente y posteriormente, para determinar el daño, las frecuencias medidas son comparadas con las frecuencias predichas [29].

### Problema inverso

Métodos inversos combinan un modelo inicial de la estructura y la medición de datos para mejorar el modelo o probar una hipótesis. En la práctica, el modelo se basa en el análisis de elementos finitos y las medidas son datos de la aceleración y la fuerza, a menudo en forma de una base de datos, aunque la técnica de función de respuesta de frecuencia (FRF) también puede ser utilizada. En este sentido, generalmente se emplean los métodos de ajuste de modelos, los cuales han tenido cierto éxito en mejorar los modelos [26, 30].

Estos métodos se ubican por lo general en los Niveles 2 (determinación de la localización geométrica del daño) y 3 (cuantificación de la severidad del daño) de la identificación de daños, consisten en calcular los parámetros de daño, por ejemplo, la longitud y/o localización de una grieta, la cual se realiza a partir de cambios en las frecuencias [29].

### 1.4.2. MÉTODOS BASADOS EN LA COMPARACIÓN DIRECTA DE LAS FORMAS MODALES

Son numerosos los investigadores que han utilizado este criterio para desarrollar métodos de identificación de daños, donde el indicador se define a partir de los cambios en las formas modales.

Una medida común de una sola forma modal es el Criterio de Confianza Modal o “*Modal Assurance Criterion*” (MAC), para calcular el nivel de correlación entre los



## DETECCIÓN DE DAÑOS EN ESTRUCTURAS: ANTECEDENTES

modos y así poder determinar la presencia de daño, pero ha resultado ser un criterio bastante insensible. El valor MAC de dos modos  $\phi^A$  y  $\phi^B$ , donde A y B indican las formas modales del estado sin daño y con daño respectivamente, se define como [31]:

$$MAC_{A,B} = \frac{\left| \sum_{i=1}^n \phi_i^A \phi_i^B \right|^2}{\sum_{i=1}^n (\phi_i^A)^2 * \sum_{i=1}^n (\phi_i^B)^2} \quad (1.1)$$

Con  $n$  grados de libertad, el valor del MAC se puede considerar como el grado de correlación entre los dos formas modales. Un valor de MAC igual a cero indicaría que no existe una correlación entre los modos, mientras que un valor de uno, tendría una correlación total. La desviación a partir de 1 puede ser interpretado como el indicador de daño en la estructura [17].

El MAC difiere de la definición del Criterio de Confianza de Coordenadas Modales “*Coordinate Modal Assurance Criterion*” (COMAC), ya que este último ofrece información local junto con la combinación de la información de los diferentes modos. El valor de COMAC en la coordenada modal  $i$  para los modos  $m$  se define como [31]:

$$MAC_{i,A,B} = \frac{\left| \sum_{j=1}^m \phi_{i,j}^A \phi_{i,j}^B \right|^2}{\sum_{j=1}^m (\phi_{i,j}^A)^2 * \sum_{j=1}^m (\phi_{i,j}^B)^2} \quad (1.2)$$

El COMAC es una medida puntual de la diferencia entre dos conjuntos de formas modales y toma un valor entre 1 y 0, donde un bajo valor del COMAC indicaría discordancia en un punto y por lo tanto también es un posible indicador de ubicación del daño [17].

Una alternativa de utilización de las formas modales para obtener información espacial sobre los cambios en la respuesta del sistema, es el de utilizar las derivadas de las formas modales, tales como la curvatura. Los resultados obtenidos mediante este criterio son extremos, la diferencia depende básicamente del tipo de señal que se utiliza; sin ser concluyentes se puede decir que los resultados no han sido muy buenos cuando se utilizan datos experimentales, ya que uno de los factores más importantes en la utilización de estos métodos es la selección de cuales modos utilizar para el análisis, ya que la información que se pueda obtener de cualquiera de los modos utilizados, dependerá de la localización de los sensores de medición respecto a la amplitud de la forma modal asociada [29].

### 1.4.3. MÉTODOS BASADOS EN LA CURVATURA DE LAS FORMAS MODALES

Mientras que los desplazamientos modales a menudo cambian levemente por daños reales con respecto a las formas modales iniciales de la estructura intacta, la curvatura modal es más sensible a los daños [31].

## DETECCIÓN DE DAÑOS EN ESTRUCTURAS: ANTECEDENTES

El uso de la curvatura en la identificación de daños se basa en el supuesto de que los cambios en las curvaturas de las formas modales son muy localizados en la región de los daños, de esta manera la reducción del  $EI$  (modulo de elasticidad e inercia) producen un aumento de la curvatura en la sección. Dado que las variaciones de la curvatura son locales y dependen de la reducción del  $EI$ , se pueden utilizar para detectar, localizar y cuantificar los daños. Por lo tanto, se espera que la diferencia absoluta entre la curvatura de las formas modales de la estructura sin daño y la estructura dañada proporcionará la región dañada. Además, la máxima diferencia se incrementa a medida que el grado de reducción de la rigidez aumenta en la zona dañada. La curvatura se puede calcular como [17, 32]:

$$\phi_{ji}'' = \frac{\phi_{(j+1)i} - 2\phi_{ji} + \phi_{(j-1)i}}{L^2} \quad (1.3)$$

Donde:

$\phi$  = Forma modal.

$i$  = Número de la forma modal.

$j$  = Número del nodo.

$L$  = Distancia entre nodos.

Como indicador de la presencia de daños presentes dentro de la estructura se puede calcular la diferencia absoluta de la curvatura (DC) entre los estados con daño y sin daño respectivamente, como:

$$DC = \left| \phi_k'' - \phi_k^{d''} \right| \quad (1.4)$$

Donde:

$\phi_k''$  = Curvatura de la forma modal  $k$  de la estructura sin daño.

$\phi_k^{d''}$  = Curvatura de la forma modal  $k$  de la estructura con daño.

Para tener en cuenta todos los modos medidos, De Roeck Wahab (1999) propone un indicador de daño llamado Factor de Daño de Curvatura (FDC), que se refiere a la diferencia de la curvatura antes y después de los daños haciendo un promedio sobre el número de modos, es decir [32]:

$$FDC = \frac{1}{n} \sum_{k=1}^n \left| \phi_k'' - \phi_k^{d''} \right| \quad (1.5)$$

El número de modos y de curvaturas utilizado por las rutinas de identificación de daños es, naturalmente, limitado al número de desplazamientos disponibles de la forma modal. En conclusión, las formas modales y sus derivadas han sido ampliamente utilizadas para identificar los daños. Sin embargo, existen algunas contradicciones sobre la

utilidad de las formas modales por si solas en la detección de daños. Esta incertidumbre ha dado lugar a la investigación de otros métodos [17].

Ratcliffe [33] presentó un estudio para localizar el daño estructural a partir de datos experimentales de vibración. El método empleaba mediciones de las funciones de respuesta en frecuencia para obtener desplazamientos como función de la frecuencia, y estas funciones de desplazamientos fueron convertidas a funciones de curvatura, las cuales fueron transformadas para obtener un índice de daño. Finalmente, se demostró que este método de detección de daños propuesto era muy sensible, ya que pudo encontrar un daño muy pequeño.

### 1.4.4. MÉTODOS BASADOS EN LA MEDICIÓN DINÁMICA DE LA FLEXIBILIDAD

Otra clase de métodos de identificación de daños emplea el uso de la medición dinámica de la matriz de flexibilidad para estimar cambios en el comportamiento estático de la estructura. La matriz de flexibilidad está definida como el inverso de la matriz de rigidez estática, de esta forma la matriz de flexibilidad relaciona la fuerza estática aplicada y el desplazamiento estructural resultante. Por lo tanto, cada columna de la matriz de flexibilidad representa el patrón de desplazamiento de la estructura asociada con una fuerza unitaria aplicada en los correspondientes grados de libertad (GDL). La medición de la matriz de flexibilidad puede ser estimada, a partir de las mediciones de la formas modales  $\varphi$  y frecuencias  $\Lambda$  como [17, 31]:

$$F = \varphi \Lambda^{-1} \phi^T \quad (1.6)$$

La formulación de la matriz de flexibilidad por este método es aproximado debido al hecho de que solo los primeros modos de la estructura (los modos de menor frecuencia) son medidos [14].

Típicamente, el daño es detectado por medio de la comparación de la matriz de flexibilidad usando los modos de la estructura dañada con la matriz de flexibilidad empleando los modos de la estructura sin daño o la matriz de flexibilidad de un modelo de elementos finitos (MEF). La ventaja de esta metodología radica en que la matriz de flexibilidad es mas sensitiva a los cambios en los modos de menor frecuencia dada la relación inversa al cuadrado de las frecuencias modales de la estructura [14].

En este ámbito, Reynders y Roeck [18], presentaron un método fundamentado en la flexibilidad cuasi-estática para la localización y cuantificación de daños a partir de las vibraciones. El método fue validado con ejemplos de simulación de daños en vigas isostáticas e hiperestáticas, y además se implementó en pruebas experimentales con una viga de hormigón armado libre-libre y un puente de hormigón pretensado de tres vanos, que fueron sometidos a una prueba de daño progresivo. La matriz de flexibilidad que era determinada experimentalmente fue combinada con una carga virtual para generar un esfuerzo no nulo en una pequeña parte de la estructura, determinando la posibilidad de algun cambio de rigidez local. Los autores demostraron que, si la relación

esfuerzo–deformación para la carga es proporcional, la proporción de una combinación de deformaciones antes y después de un cambio de rigidez que se ha producido, esta es igual a la relación inversa de la rigidez local, por esto, el método se denominó de flexibilidad local (FL). Dado que la matriz de flexibilidad cuasi-estática se puede componer directamente en los parámetros modales, el método FL permitió determinar las variaciones locales de rigidez directamente de los parámetros modales medidos, incluso si se determinará a partir sólo de los datos de salida.

#### 1.4.5. MÉTODOS BASADOS EN EL VECTOR DE FUERZA RESIDUAL

Con el acceso a las formas modales y frecuencias naturales es posible formular lo que se conoce como Vector de Fuerza Residual (VFR). Como se sabe, las frecuencias naturales y formas modales satisfacen la ecuación del valor propio. De acuerdo a lo anterior considerando el modo  $i$  de la estructura con daño tenemos que [17]:

$$(K_d - \lambda_{di} M_d) \phi_{di} = 0 \quad (1.7)$$

Donde  $\lambda_{di}$  es el cuadrado de la frecuencia natural, y  $\phi_{di}$  es la  $i$ -ésima forma modal de la estructura dañada, siendo estos parámetros conocidos para varios modos a partir de mediciones experimentales. Además teniendo en cuenta que las matrices de rigidez ( $K_d$ ) y masa ( $M_d$ ) asociadas con la estructura dañada se pueden definir como:

$$K_d = K_a + \Delta K \quad (1.8)$$

$$M_d = M_a + \Delta M \quad (1.9)$$

Donde:

$K_a$  = Matriz de rigidez de la estructura sin daño.

$\Delta K$  = Variación de la matriz de rigidez causada por los daños.

$M_a$  = Matriz de masa de la estructura sin daño.

$\Delta M$  = Variación de la matriz de masa causada por los daños.

Sustituyendo las ecuaciones 1.8 y 1.9 en la ecuación 1.7 y reorganizando, se obtiene que para el modo  $i$ , el VFR  $R$  para la  $i$ -ésima forma modal está dado por:

$$R_i = (K_a - \lambda_{di} M_a) \phi_{di} \quad (1.10)$$

Cada modo proporciona un solo VFR el cual representa un solo grado de libertad de modelo numérico de la estructura. Cuando un daño ocurre en un elemento asociado a este grado de libertad, el valor del VFR se hace muy grande comparado con los

demás valores correspondientes a los restantes grados de libertad donde ningún daño ha ocurrido. Al elegir dichas condiciones se proporciona un método para la identificación de daños [17].

Yang y Liu [34] estudiaron el método basado en el vector de fuerza residual. Los autores emplearon dicho método para localizar los supuestos elementos dañados preliminarmente. Luego para la correcta identificación de daños estudiaron tres técnicas. La primera fue la solución algebraica de la ecuación de fuerza residual, la segunda fue la técnica “*minimum-rank elemental update*” (MREU) y la tercera fue el método de sensibilidad de frecuencia natural. Además, para resolver el problema de medición incompleta emplearon una técnica de expansión de las formas modales, el cual se basó en el concepto del mejor vector propio. Estos métodos fueron aplicados en un ejemplo numérico teniendo en cuenta la medición de ruido.

### 1.4.6. MÉTODOS BASADOS EN EL AJUSTE DE LOS PARÁMETROS DEL MODELO ESTRUCTURAL

Otra clase de métodos de identificación de daños está basado en la modificación de las matrices del modelo estructural como por ejemplo la masa, rigidez y amortiguamiento para reproducir lo más cerca posible las mediciones de la respuesta estática o dinámica procedente de los datos. Estos métodos utilizados para resolver el ajuste de las matrices formulan un problema de optimización con restricciones basado en la ecuación de movimiento estructural y la medición de los datos. Comparaciones de la correlación de las matrices ajustadas con las matrices originales proporcionan una indicación de daño y puede ser usada para cuantificar la localización y extensión de los daños. La principal característica de la solución, es la minimización entre los residuos del modelo experimental y el modelo numérico que describe la dinámica de la estructura. Además, es necesario elegir los parámetros adecuados para el ajuste del modelo numérico [14, 17, 31].

Los métodos de ajuste de modelos se pueden clasificar como métodos de sensibilidad o directos. Los métodos de sensibilidad se basan en un modelo paramétrico de la estructura y la minimización de alguna función de penalidad teniendo en cuenta el error entre los datos medidos y las predicciones del modelo. Estos métodos ofrecen una amplia gama de parámetros para ser ajustados que tiene un significado físico y permiten un grado de control sobre el proceso de optimización. Los alternativos son métodos de ajuste directo que cambian la matriz de masa y/o de rigidez [14, 26].

Chellini G, et al. [35] estudiaron el proceso del ajuste del modelo de elementos finitos a partir de las mediciones de las vibraciones, las cuales fueron usadas para detectar, evaluar y cuantificar el daño estructural de un marco altamente dúctil compuesto por concreto-acero, el cual fue sometido a crecientes daños sísmicos mediante una prueba pseudo-dinámica y una prueba cíclica. El proceso de ajuste fue repetido para tres niveles de daño, siendo aplicados a diferentes modelos de elementos finitos, lo que permitió una amplia descripción y cuantificación de la degradación progresiva en las uniones viga-

columna, que generalmente se dedican a disipar la energía sísmica de diseño.

### 1.4.7. MÉTODOS DE OPTIMIZACIÓN

La optimización se puede definir como aquella ciencia encargada de determinar las mejores soluciones a problemas matemáticos, que a menudo modelan una realidad física. Los problemas complejos de optimización multidimensional, no lineal y altamente multimodal, pueden encontrarse en ingeniería, economía, geofísica y, prácticamente, en todos los campos de la ciencia. En este aspecto, la optimización numérica ha adquirido mucha atención entre la comunidad científica, durante las últimas décadas.

La principal característica de las técnicas de optimización es combinar técnicas clásicas de modelación con nuevos paradigmas de computación, tales como los algoritmos genéticos “*Genetic Algorithm*”, optimización con enjambre de partículas “*Particle Swarm Optimization*”, recocido simulado “*Simulated Annealing*” entre otros. Esto debido a que las metodologías clásicas, a pesar de su extenso uso, sufren de algunas limitaciones como: problemas de divergencia, inestabilidad y susceptibilidad de quedar atrapados en puntos óptimos locales. Las metodologías heurísticas forman un campo de investigación muy prometedor. El desarrollo de tales algoritmos generalmente siguen un patrón regular y previsible, donde las versiones iniciales de estos algoritmos son innovadores pero no muy efectivos y estos son mejorados por propuestas de variantes en algunos parámetros y finalmente estos son hibridizados con otros esquemas, para así poder ser implementados en problemas más complejos [36].

La ventaja sobre los algoritmos tradicionales de optimización es que son capaces de abordar topologías de soluciones multimodales, que son típicas en el problema de identificación de daños [17].

A continuación se presentan algunos estudios encontrados en la revisión de la literatura, los cuales abordan el problema de la detección de daños empleando técnicas de optimización.

Villalba y Laier [37] aplicaron los algoritmos genéticos para resolver el problema de detección de daño estructural, comparando tres tipos de codificación de individuos: binaria, real y binaria con código redundante. Los algoritmos de código binario y real calcularon la extensión del daño para cada elemento en la estructura, y para mejorar su desempeño se utilizó un proceso de re-inicialización de individuos. En cuanto al algoritmo de código redundante este busca en forma dinámica cuáles son los elementos dañados y cuantifica el daño solo para esos elementos. Para determinar el desempeño de los algoritmos se analizó una armadura sobre diversos escenarios de daño simple y múltiple, y el daño fue considerado como una reducción en el módulo de elasticidad de los elementos dañados. Los resultados muestran que el algoritmo que mejor localiza y cuantifica el daño corresponde al algoritmo genético de código redundante.

Begambre y Laier [38] propusieron un algoritmo híbrido entre el Particle Swarm Optimization y el simplex (PSOS), para la identificación de daños a partir de los datos del dominio de la frecuencia. La formulación de la función objetivo para el problema de minimización se basó en las funciones de respuesta en frecuencia (FRF) del sistema. La

## DETECCIÓN DE DAÑOS EN ESTRUCTURAS: ANTECEDENTES

estrategia planteada fue el control de los parámetros del particle swarm optimizations (PSO) a partir del algoritmo de Nelder-Mead (método Simplex), de esta manera, la convergencia del PSO se independiza de las constantes heurísticas, mejorando su estabilidad y confianza. El método híbrido formulado funcionó mejor en diferentes funciones de prueba que el simulated annealing (SA) y el PSO básico. Además, se solucionaron dos problemas de identificación de daños, teniendo en cuenta los efectos de los datos con ruido e incompletos. Los problemas estudiados fueron: en primer lugar, una armadura de 10 elementos y la segunda fue una viga libre-libre, las cuales fueron modeladas con elementos finitos. En cada uno de estos casos, la ubicación y la magnitud del daño se determinó con éxito.

Na et al. [39], presentaron un método identificación de daños, el cual fue implementado en un edificio de corte. Dicho método está basado en un algoritmo genético y hace uso de la matriz de flexibilidad estructural y el análisis dinámico. El método propuesto permitió la deducción de la extensión y localización del daño estructural, incluso cuando no se tenían los datos suficientes de las características dinámicas y cuando no se tenían suficientes mediciones exactas de la rigidez estructural y la masa. La validez del método de evaluación de los daños se demostró a través del análisis numérico utilizando OpenSees.

Nobahari y Seyedpoor [40], propusieron un proceso para la detectar daños en múltiples sistemas estructurales. Para la detección de la presencia de daños se consideraron los cambios en las frecuencias naturales de la estructura. Con el fin de evaluar las frecuencias naturales necesarias, se utilizó un análisis de elementos finitos. Los autores presentaron un algoritmo genético modificado con dos nuevos operadores para detectar con precisión la ubicación y magnitud de los daños. Además, para la función objetivo del algoritmo de optimización se introdujo un índice basado en la correlación entre la estructura con y sin daños. Los resultados numéricos de dos ejemplos de referencia teniendo en cuenta la medición del ruido demostraron las ventajas computacionales del método propuesto para determinar con precisión la ubicación y la magnitud de los daños estructurales múltiples.

Meruane y Heylen [41], realizaron la implementación de un algoritmo genético con codificación real y con penalización del daño para localizar y cuantificar el daño estructural. Además los autores estudiaron el desempeño del algoritmo en cinco funciones basadas en datos modales. En este estudio propusieron el uso de una penalización, que evitó de manera satisfactoria la falsa detección de daños debido al ruido experimental o errores numéricos. Un pórtico tridimensional con escenarios de daños simples y múltiples fue empleado para la verificación del enfoque propuesto, donde se realizaron pruebas con diferentes niveles de mediciones incompletas en los grados de libertad. Los resultados mostraron que este enfoque logró determinar a una solución mucho más precisa que los métodos convencionales de optimización.

Begambre et al. [42], emplearon el algoritmo Simulated Annealing (SA) para solucionar el problema inverso de detección de daño en vigas usando información modal contaminada con ruido. La formulación de la función objetivo para el procedimiento de optimización, basado en el SA, se fundamentó en el método de la fuerza residual

## DETECCIÓN DE DAÑOS EN ESTRUCTURAS: ANTECEDENTES

modificada. El desempeño del SA empleado en el estudio superó al de un algoritmo genético (AG) en dos funciones de prueba reportadas en la literatura internacional. El procedimiento de evaluación de integridad propuesto se confirmó y validó numéricamente empleando la teoría de vigas de Euler-Bernoulli y un Modelo de Elementos Finitos (MEF) para las vigas en voladizo y apoyadas.

He y Hwang [43], propusieron un algoritmo para detectar daños en estructuras tipo viga, el cual combina un algoritmo genético de parámetro real con el simulated annealing denominado ARSAGA. Para realizar la detección, el algoritmo propuesto emplea los desplazamientos de la respuesta estática y las frecuencias naturales de análisis modal, que se obtienen por el software de análisis de elementos finitos ANSYS. El desempeño del algoritmo, fue puesto a prueba en tres diferentes tipos de vigas. En cada tipo de estructura se tomaron diferentes condiciones de frontera y diferentes escenarios de daño. De los resultados, fue demostrado que el algoritmo propuesto fue eficiente en la identificación daños en las estructuras tipo viga en condición libre de ruido. Incluso en el caso del ruido, los resultados mostraron que la búsqueda de soluciones se encuentran con una precisión razonable.



---

## PROBLEMA DE REFERENCIA: PÓRTICO UBC IASC–ASCE

---

Actualmente, numerosos métodos globales de SHM han sido estudiados por varios investigadores y se encuentran documentados en la revisión de la literatura [14], pero estas técnicas no deberían ser comparadas entre si, debido principalmente a que los diferentes métodos desarrollados generalmente son probados en diferentes estructuras con variaciones en las condiciones de contorno. Por tal motivo, un problema común proporcionaría una plataforma para la evaluación de los métodos SHM [44]. De acuerdo con lo anterior, la International Association for Structural Control (IASC) y la American Society of Civil Engineers (ASCE) formaron el comité SHM para desarrollar una serie de problemas de referencia [44]. A continuación se presenta un resumen del trabajo realizado en este campo.

### 2.1. DESCRIPCIÓN GENERAL DEL PROBLEMA DE REFERENCIA

El problema propuesto por la IASC–ASCE consiste en una estructura de referencia, la cual fue sometida a varios escenarios de daño. Inicialmente el problema fue formulado numéricamente y posteriormente se realizó una fase experimental. En la fase numérica, se crearon modelos numéricos a partir de una estructura a escala existente, los cuales fueron usados para simular los registros de aceleración de la estructura con varios escenarios de daño, donde la identificación del daño de la estructura es determinado a partir de la respectiva respuesta dinámica. Por otro lado, en la fase experimental, los registros de aceleración fueron obtenidos de un modelo físico, donde fueron considerados diferentes casos de excitación y patrones de daño. A continuación se realiza una descripción detallada de cada uno de los aspectos que componen el problema de referencia.

### 2.1.1. ESTRUCTURA DE REFERENCIA

La estructura de referencia seleccionada por la IASC-ASCE, consta de una estructura metálica escalada 1:3 de 2 vanos por 2 vanos y 4 pisos (figura 2.1), la cual fue sometida varios casos de daño, la cual fue construida en el laboratorio de ingeniería sísmica de la universidad de la Columbia Británica (UBC-Canadá).

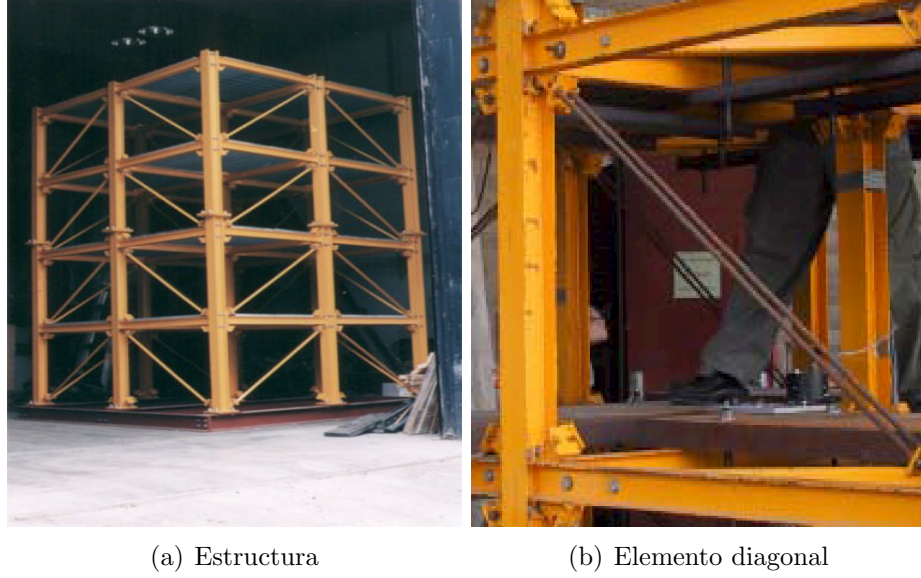


Figura 2.1: Estructura de referencia [1]

La estructura es de 2.5m x 2.5m de longitud en planta y de 3.6m de alto. Las vigas y columnas son de acero laminado en caliente de 300W (esfuerzo nominal de fluencia de 300Mpa). Las secciones fueron diseñadas para este modelo en especial y las propiedades de los diferentes elementos que componen la estructura se encuentran en la tabla 2.1.

Tabla 2.1: Propiedades de la estructura de referencia [1]

Propiedad	Columnas	Vigas	Rigidizadores
Nombre de la sección	B100X9	S75X11	L25X25X3
Area de la sección transversal ( $m^2$ )	$1,133 \times 10^{-3}$	$1,430 \times 10^{-3}$	$1,410 \times 10^{-4}$
Momento de inercia en la dir. fuerte ( $m^4$ )	$1,970 \times 10^{-6}$	$1,220 \times 10^{-6}$	0
Momento de inercia en la dir. débil ( $m^4$ )	$6,640 \times 10^{-7}$	$2,490 \times 10^{-7}$	0
Constante de torsión ( $m^4$ )	$8,010 \times 10^{-9}$	$3,820 \times 10^{-9}$	0

## PROBLEMA DE REFERENCIA: PÓRTICO UBC IASC-ASCE

Un aspecto importante a resaltar, según [2], es la conexión realizada entre los elementos viga-columna (ver figura 2.2), ya que este tipo de unión permite un grado de flexibilidad en la estructura, por lo cual [2], recomienda tener en cuenta este aspecto cuando se pretenda realizar un modelo numérico del sistema.



Figura 2.2: Conexión viga-columna [2]

Adicionalmente, para hacer la distribución de masa más realista, se colocaron losas adicionales en cada piso de la siguiente forma: cuatro losas de 1000Kg en cada uno de los tres primeros niveles, mientras que para el cuarto nivel se dispusieron de cuatro losas de 750Kg [1], como se ilustra en la figura 2.3.

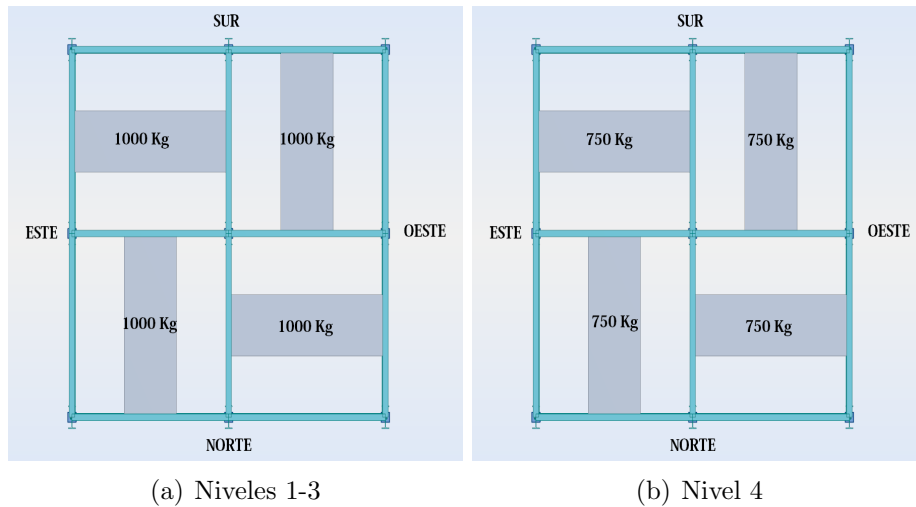


Figura 2.3: Configuración de la masa en la estructura [1]

Una vez detallada la estructura, se procede a la descripción de los casos de daño

propuestos y de las fases realizadas para la formulación del problema de referencia de detección de daños.

## 2.2. PATRONES DE DAÑO

Dentro del problema de referencia formulado, se establecieron dos tipos de daños presentes en la estructura. El primer tipo de daño abarca los casos I al VII de la tabla 2.2, los cuales fueron simulados removiendo algunos rigidizadores de la estructura (casos estudiados en el presente trabajo). El segundo tipo de daño es abarcado por los casos VIII al X de la tabla 2.2 los cuales fueron representados como perdidas de rigidez en las conexiones viga–columna, esto se logró soltando los tornillos de las conexiones. Una completa descripción de las pruebas realizadas puede ser encontradas en la referencia [45].

## 2.3. FASE NUMÉRICA

En esta fase, dos modelos numéricos fueron desarrollados basados en la estructura de referencia. El primero es un modelo de 12 grados de libertad (GDL) el cual permite el movimiento de cada piso en el plano horizontal, y rotaciones con respecto al eje vertical. El segundo modelo tiene un total de 120 GDL, donde los nodos de cada piso son restringidos para tener el mismo desplazamiento horizontal y la misma rotación con respecto al eje vertical (piso rígido). A diferencia de la estructura real, ambos modelos numéricos poseen una masa total de  $3200Kg$  en el primer piso,  $2400Kg$  en el segundo y tercer piso, y para el último piso  $1600Kg$  ó  $1700Kg$  dependiendo del caso a ser simulado [46]. Estos dos modelos fueron usados para la primera y segunda fase numérica del problema de referencia. Cabe resaltar, que para el presente trabajo el modelo numérico empleado posee la distribución de masa de la estructura real (ver figura 2.3).

El principal objetivo de la primera fase numérica fue estudiar los efectos de ruido en los sensores y los efectos de los errores de modelación en las metodologías de SHM. En esta fase, el daño fue inducido en la estructura removiendo los rigidizadores en la estructura o reduciendo la rigidez en las conexiones viga–columna [46].

El enfoque de la segunda fase numérica fue la investigación de las diferencias entre el modelo considerado para ser la estructura real (modelo de 120 GDL) y el modelo de identificación (modelo de 12GDL). Aunque en la primera fase existieron errores de modelado entre la estructura de identificación de 12 GDL y el modelo estructural de 120 GDL, la modelación fue demasiado ideal. Por lo tanto en la segunda fase numérica, el comité del IASC-ASCE decidió usar el modelo estructural 120 GDL con algunas modificaciones, las cuales se encuentran en la referencia [47].

A continuación se presenta una revisión de la literatura relevante relacionada con la fase numérica del problema de referencia.

## PROBLEMA DE REFERENCIA: PÓRTICO UBC IASC–ASCE

Tabla 2.2: Descripción de los patrones de daño

Configuración	Descripción
<i>I</i>	Estructura sin daño – Todos los rigidizadores presentes.
<i>II</i>	Todos los rigidizadores del costado este son removidos.
<i>III</i>	Rigidizadores de la mitad sur del costado este son removidos
<i>IV</i>	Rigidizadores en el primer y cuarto piso de la mitad sur del costado este son removidos
<i>V</i>	Rigidizadores en el primer piso de la mitad sur del costado este son removidos
<i>VI</i>	Rigidizadores en el segundo piso del costado este son removidos
<i>VII</i>	Todos los rigidizadores en todos los costados fueron removidos. Esta configuración se considera como la estructura en buen estado, sin rigidizadores
<i>VIII</i>	Estructura sin rigidizadores. Todas las conexiones del vano norte del costado este son aflojadas
<i>IX</i>	Estructura sin rigidizadores. Las conexiones del primer y segundo piso del vano norte del costado este son aflojadas
<i>X</i>	Estructura sin rigidizadores. Las conexiones del primer piso del vano norte del costado este son aflojadas
<i>XI</i>	Esta configuración de un “último minuto”. Es el mismo de la configuración VII, pero con una masa de 100Kg añadido a la parte superior de la esquina noroeste.

En este campo, Caicedo [48], desarrolló dos metodologías, las cuales fueron implementadas en el problema de referencia de la IASC-ASCE. La primera metodología utiliza “*Natural Excitation Technique*” (NexT) y el “*Eigensystem Realization Algorithm*” (ERA) para identificar los parámetros modales de la estructura. La segunda metodología fue utilizada de acuerdo a las funciones de transferencia de la estructura, donde dichas funciones se calcularon a partir de los datos de aceleración. Para las dos metodologías, el daño fue identificado mediante la comparación de la rigidez de la estructura sin daños con la estructura dañada.

En el estudio presentado por Barroso y Rodríguez [49], los autores utilizaron un método de índice de daño para detectar la localización y severidad del daño en problema de referencia de la ASCE. El método se basa en la comparación entre el radio de curvatura de la forma modal de la estructura dañada con la estructura sin daño. Como

## PROBLEMA DE REFERENCIA: PÓRTICO UBC IASC–ASCE

resultado, los autores lograron determinar con exactitud la presencia de daños, pero falsas localizaciones fueron encontradas. Los autores atribuyen este comportamiento al hecho que se usó un modelo de un edificio a cortante como modelo de identificación, que no es tan sensible a la información de la rotación.

Ching y Beck [50], usaron un enfoque probabilístico para los patrones de daño I al III de la estructura de referencia de la IASC–ASCE. Los casos involucrados en la detección de daños y evaluación de la estructura usan datos simulados de vibración ambiental generados del modelo numérico de referencia. El enfoque que usaron los autores implica la identificación modal, seguida de la evaluación de daños usando los parámetros modales antes y después del daño, basándose en una metodología de ajuste bayesiana. De esta manera, los autores propusieron un algoritmo de Expectación-Maximización para encontrar los valores más probables de los parámetros estructurales. Los resultados del análisis muestran que el enfoque probabilístico es capaz de detectar con éxito los lugares de daño, así como estimaciones razonablemente satisfactorias su gravedad.

Preston [51], investigó la detección de daños en el problema de referencia del IASC–ASCE mediante la identificación de los cambios de rigidez de la estructura. Para tal fin, el autor empleó el algoritmo denominado “*Least Mean Square*” (LMS). La investigación se centró principalmente en la Fase II del problema de referencia, dado que en dicha fase se incluyeron errores de modelado y se introdujo ruido en la estructura, haciendo que el problema fuera más realista. El autor encontró que el enfoque del filtro LMS se puede utilizar tanto para la identificación como para determinar la gravedad relativa del daño presente en la estructura en tiempo real. Aunque la investigación encontró que existe un umbral por debajo del cual el daño es difícil de identificar. Así, si la rigidez global cambia menos de 10 %, entonces la identificación de la presencia y localización del daño es difícil. Pero si el tipo de daño es conocido, entonces la presencia y ubicación se puede determinar.

Dharap [52], implemento una técnica basada las observaciones de ARMarkov y observaciones de “*Input Error Function*” (IRF) fue propuesta para la estimación de la extensión del daño en tiempo real, las cuales fueron evaluadas en la estructura de referencia del IASC–ASCE, donde la formulación propuesta tuvo un buen desempeño para los diferentes casos de daño establecidos.

Calderón [53], realizó un estudio numérico de la estructura de referencia, en los que, a partir de la respuesta dinámica simulada numéricamente, identificó las principales frecuencias de vibración. Para tal fin, el autor inicialmente efectuó un tratamiento de la señal para su posterior estudio en el dominio de la frecuencia, donde se empleó el método de Welch para la identificación de los cuatro modos fundamentales en las dos direcciones mediante el espectro de potencia. Como resultado se obtuvo una buena identificación de las frecuencias comparadas con las teóricas de un modelo de elementos finitos.



## 2.4. FASE EXPERIMENTAL

Luego de los respectivos estudios realizados con los modelos numéricos, se procedió a la realización de pruebas con la estructura física. Para dichas pruebas, la estructura fue ubicada de tal forma que el lado fuerte fuera paralelo al costado Norte-Sur. Como fue descrito anteriormente, en cada vano de la estructura, el sistema de rigidización consistió en barras de acero roscadas de  $12,7\text{mm}$  (0,5") de diámetro colocadas a lo largo de la diagonal [1]. Además para el caso experimental, la distribución de masa fue realizada de acuerdo a la figura 2.3

Al igual que en la fases numéricas, el daño fue simulado removiendo rigidizadores de la estructura o soltando los pernos de la conexión viga-columna. En las pruebas realizadas experimentales se consideró la vibración ambiental y dos tipos de excitaciones forzadas. La vibración ambiental incluye las excitaciones presentes del ambiente debido a viento, peatones y tráfico. Los casos considerados de excitación forzada fueron: prueba de impacto de martillo (ver figura 2.5) y excitaciones proporcionadas por un shaker electrodinámico (ver figura 2.4). Adicionalmente, fueron colocados acelerómetros en toda la estructura para proporcionar mediciones de la respuesta experimental [1]



Figura 2.4: Shaker sobre la estructura [2]

Varios sensores se colocaron a lo largo de la estructura (ver Tabla 2.3). Cuatro tipos de acelerómetros se utilizaron en la estructura. Además, se instaló un acelerómetro en el shaker para medir la aceleración de la masa en movimiento y para determinar la fuerza de entrada a la estructura en los casos de excitación forzada. También se utilizó un transformador de desplazamiento lineal variable (LVDT siglas en inglés) para medir el desplazamiento de la masa del shaker con respecto a la estructura.

Se utilizaron tres sistemas de adquisición de datos para registrar las respuestas estructurales: un sistema DasyLab, un sistema LabView y un SigLab. Los sistemas de

## PROBLEMA DE REFERENCIA: PÓRTICO UBC IASC–ASCE



Figura 2.5: Martillo de impacto usado [1]

DasyLab y LabView fueron capaces de medir hasta 16 canales, y el sistema SigLab midió cuatro canales de aceleración. En DasyLab y LabView, los datos se tomaron con una mayor frecuencia a 250 [Hz] y en SigLab la tasa de muestreo se estableció a 526 [Hz]. El sistema SigLab tiene un filtro anti-aliasing en 100 [Hz] (2.3). Los sistemas DasyLab y LabView no tiene filtro analógicos o digitales anti-aliasing, aunque varios de los sensores incluyen dicho filtro [1].

Tabla 2.3: Características de la instrumentación [1]

		Características de sensor		
Modelo Sensor		Rango de frecuencias	Sistema Anti-aliasing	Sensibilidad
Acelerómetros	Sensor FBA	DC–50 [Hz]	–	5 [Volts/sg]
	Sensor EPI	DC–200 [Hz]	–	5 [Volts/s]
	Sensor Kistler	AC–1 [KHz]	50 [Hz]	5 [Volts/s]
	Sensor IC	DC–350 [Hz]	50 [Hz]	0.667–1.43 [V/g]
	Sensor IC	DC–1 [KHz]	50 [Hz]	0.222 [V/g]
LDTV	Solartron Metrology	DC–100 [KHz]	50 [Hz]	0.611 [mm]

A continuación se describen algunos de los trabajos realizados en la fase experimental del problema de referencia de la AISC-ASCE, los cuales se encuentran reportados en la literatura.



## PROBLEMA DE REFERENCIA: PÓRTICO UBC IASC–ASCE

Caicedo [7], propuso una técnica de identificación de daños para estructuras flexibles a partir de las metodologías desarrolladas en [48], la cual fue implementada en varios tipos de estructuras. La primera estructura corresponde al problema de referencia de la IASC-ASCE, donde la implementación se aplicó a las primeras dos fases numéricas y la fase experimental. Además, se realizó una implementación a un modelo numérico del “*Bill Emerson Memorial Bridge*”. Los estudios realizados demostraron el potencial de la técnica propuesta para ser implementado en estructuras flexibles. En las tablas 2.4 y 2.5, se encuentran registrados los datos correspondientes a la determinación de las frecuencias naturales de las primeras cinco configuraciones de daño descritas en la tabla 2.2. Dichas frecuencias fueron determinadas a partir de la vibración ambiental y la prueba de impacto de martillo.

Tabla 2.4: Determinación de frecuencias naturales  $[Hz]$  usando vibración ambiental [7]

Patrón de daño	Modo 1 (E–W)	Modo 2 (N–S)	Modo 3 (Torsión)	Modo 4 (E–W)	Modo 5 (N–S)
<i>I</i>	7,50	7,77	14,48	19,88	21,01
<i>II</i>	7,73	5,56	12,74	20,13	15,02
<i>III</i>	7,65	6,65	13,48	20,02	18,91
<i>IV</i>	7,60	7,35	13,97	20,12	19,71
<i>V</i>	7,56	7,45	14,03	19,89	20,63
<i>VI</i>	5,96	7,79	13,19	19,87	20,99

Tabla 2.5: Determinación de frecuencias naturales  $[Hz]$  usando prueba de impacto de martillo [7]

Patrón de daño	Modo 1 (E–W)	Modo 2 (N–S)	Modo 3 (Torsión)	Modo 4 (E–W)	Modo 5 (N–S)	Modo 6 (Torsión)
<i>I</i>	7,47	7,64	14,45	19,90	20,87	22,52
<i>II</i>	7,94	5,11	12,69	19,97	14,91	22,14
<i>III</i>	7,56	6,50	13,39	19,90	18,76	22,22
<i>IV</i>	7,53	7,25	13,94	20,02	19,65	22,41
<i>V</i>	7,53	7,37	13,99	19,83	20,46	22,56
<i>VI</i>	–	7,67	13,06	19,80	20,79	22,34

Ching y Beck [8], continuaron con el estudio realizado en [50], y lo aplicaron en la fase experimental de problema de referencia. Dicha implementación involucra la evaluación y detección de daños usando datos experimentales generados tanto por el martillo de impacto como por las vibraciones ambientales. Al finalizar este estudio, los autores demostraron que el daño de rigidizadores fue detectado exitosamente tanto para el martillo de impacto como para la vibración ambiental. Sin embargo, el daño en la

## PROBLEMA DE REFERENCIA: PÓRTICO UBC IASC–ASCE

conexión fue más difícil de detectar dado que la identificación de los parámetros modales son menos sensibles a este tipo de daños. Teniendo, en cuenta que el estudio realizado de los autores fue de carácter experimental, la definición de las propiedades dinámicas de la estructura (frecuencias de vibración) se efectuó a partir de las mediciones hechas a la estructura real. Los resultados que los autores obtuvieron de dicho proceso, se encuentra registrados en las tablas 2.6 y 2.7.

Tabla 2.6: Determinación de frecuencias naturales  $[Hz]$  usando vibración ambiental [8]

Patrón de daño	Modo 1 (E–W)	Modo 2 (N–S)	Modo 3 (Torsión)	Modo 4 (E–W)	Modo 5 (N–S)
<i>I</i>	7,48	7,76	14,48	19,89	21,01
<i>II</i>	7,73	5,19	12,74	20,12	15,02
<i>III</i>	7,63	6,65	13,44	20,03	18,87
<i>IV</i>	7,60	7,36	13,98	20,11	19,68
<i>V</i>	7,61	7,46	14,02	19,89	20,00
<i>VI</i>	5,97	7,77	13,20	19,89	21,00

Tabla 2.7: Determinación de frecuencias naturales  $[Hz]$  usando prueba de impacto de martillo [8]

Patrón de daño	Modo 1 (E–W)	Modo 2 (N–S)	Modo 3 (Torsión)	Modo 4 (E–W)	Modo 5 (N–S)
<i>I</i>	7,45	7,67	14,45	19,84	20,87
<i>II</i>	7,68	5,12	12,69	19,99	14,92
<i>III</i>	7,56	6,52	13,40	19,91	18,78
<i>IV</i>	7,56	7,26	13,95	20,05	19,67
<i>V</i>	7,55	7,37	13,99	19,84	20,48
<i>VI</i>	5,93	7,68	13,07	19,81	20,80
<i>VII</i>	2,62	3,60	4,31	8,44	11,94

Wang et al. [13], desarrollaron un método de índice de daño denominado “*Story Damage Index*” (SDI), junto con una aproximación llamada “*Approximate Story Damage Index*” (ASDI) el cual es útil cuando no se tiene información de la masa de la estructura. La formulación de los índices se realizó en términos de la frecuencias naturales y las formas modales. Los autores implementaron dicho método en simulaciones numéricas y en el análisis experimental del modelo de referencia del IASC–ASCE. De los resultados obtenidos se demostró que el índice de daño concuerda bastante con los resultados de la inspección visual, además de la aplicación a la fase experimental se determinó que el índice ASDI es más adecuado para la identificación de daños en edificio simétricos, aunque podría sobre-estimar el grado de daño presente.

## PROBLEMA DE REFERENCIA: PÓRTICO UBC IASC-ASCE

Wu y Li [9], desarrollaron dos etapas basadas en la sensibilidad del ajuste del modelo de elementos finitos para la identificación de parámetros y detección de daños en la estructura de referencia de la IASC-ASCE, basado en mediciones de vibración ambiental. En la primera etapa, los métodos de "*weighted least squares*" y estimación Bayesiana fueron adoptados para la identificación tanto de la rigidez de las conexiones viga-columna como el módulo de Young de la estructura. La segunda etapa corresponde a la detección de los diferentes patrones de daños de la estructura a partir del ajuste de modelos. Las comparaciones entre los resultados y los datos experimentales mostraron que el proceso de ajuste de modelos de elementos finitos basados en la sensibilidad es una herramienta efectiva para la identificación de parámetros y detección de daños en estructuras de acero. En este trabajo, los parámetros de la estructura de referencia que los autores reportaron, se encuentran en la tabla 2.8

Tabla 2.8: Determinación de frecuencias naturales  $[Hz]$  usando prueba de impacto de martillo [9]

Patron de daño	Modo 1 (E-W)	Modo 2 (N-S)	Modo 3 (Torsión)	Modo 4 (E-W)	Modo 5 (N-S)
<i>I</i>	7,495	7,764	14,475	19,898	21,025
<i>II</i>	7,732	5,653	12,742	20,132	15,018
<i>III</i>	7,653	6,647	13,475	20,026	18,905
<i>IV</i>	7,603	7,348	13,974	20,126	19,713
<i>V</i>	7,560	7,452	14,028	19,895	20,632

Ventura et al. [2], presentaron una recopilación de diferentes pruebas de vibración efectuadas a la estructura de referencia del IASC-ASCE. Dichas pruebas incluyeron vibración ambiental, vibración forzada y prueba de impacto, las cuales fueron realizadas para diversas configuraciones de la estructura, que simulaban diferentes niveles de daño. Los autores realizaron una descripción de las pruebas, así como análisis modal de configuraciones daños seleccionados para el estudio. Adicionalmente, describieron una implementación de un sistema basado en Internet para el monitoreo en tiempo real a distancia de la estructura de referencia. Este sistema permitió la adquisición de los datos en tiempo real a partir de pruebas de vibración ambientales o forzadas. Algunos de los resultados reportados por los autores se pueden observar en la tabla 2.9.

Tabla 2.9: Determinación de frecuencias naturales  $[Hz]$  [2]

Patron de daño	Modo 1 (E-W)	Modo 2 (N-S)	Modo 3 (Torsión)	Modo 4 (E-W)	Modo 5 (N-S)
<i>I</i>	7,670	7,910	14,420	19,940	25,48
<i>II</i>	5,180	7,740	12,750	15,020	20,13
<i>VII</i>	2,640	3,630	4,340	8,490	12,05

## PROBLEMA DE REFERENCIA: PÓRTICO UBC IASC–ASCE

Como se puede observar en las tablas 2.4 a 2.9, los estudios realizados de la fase experimental mencionados anteriormente, los cuales están reportados en la literatura especializada, se ha logrado determinar algunos parámetros modales del sistema estructural de referencia, que generalmente corresponden a las primeras frecuencias de vibración, para las configuraciones I al VII, las cuales fueron determinadas a partir de mediciones experimentales del modelo físico.

Cabe señalar que el presente trabajo pretende emplear el pórtico de referencia del IASC–ASCE para la detección de daños a partir de la información suministrada experimentalmente, para tal fin se empleará un modelo numérico de elementos finitos para realizar la respectiva identificación. Dado que se va a combinar un modelo inicial de la estructura con la medición de datos, es importante realizar el ajuste del modelo numérico, teniendo en cuenta que este es un problema inverso, es decir que a partir de los datos obtenidos de las mediciones se ajustan algunos parámetros del modelo numérico como la modificación de las matrices de masa, rigidez y amortiguamiento.

Generalmente los métodos para resolver el ajuste de las matrices formulan un problema de optimización con restricciones basado en la ecuación de movimiento estructural y la medición de los datos [31], por tal motivo en el capítulo 3, se describen algunas técnicas para resolver este tipo de problemas.

Dentro de los métodos de detección de daños mencionados en el presente trabajo, el implementado por Wang et al. [13] ha llamado la atención, dado su enfoque innovador de implementación directa, además de ser adecuado para el uso de datos incompletos que es el caso típico de una identificación global a partir información experimental.

---

## TÉCNICAS DE OPTIMIZACIÓN PARA DETECCIÓN DE DAÑOS

---

Cada día, los ingenieros están enfrentando problemas de gran complejidad, los cuales emergen en diversos sectores tales como: investigación de operaciones, el diseño de sistemas mecánicos, procesamiento de imágenes, entre otros. El problema para ser resuelto puede ser generalmente expresado como un problema de optimización. Aquí se debe definir una función objetivo, o función de costo, que busca ser minimizada o maximizada de acuerdo a ciertos parámetros. En algunos casos, la definición del problema de optimización se encuentra complementado con la información suministrada por las restricciones. Todos los parámetros de las soluciones adoptadas deben satisfacer estas restricciones, o de otra forma las soluciones no son factibles.

Dos tipos de problemas de optimización pueden ser identificados: el problema con variables *discretas* y el problema con variables *continuas*. Entre los problemas discretos, se puede estudiar el famoso problema del viajero de comercio, en el que se busca la minimización de la longitud del recorrido de un *viajero de comercio*, el cual debería visitar un cierto número de ciudades, antes de llegar a la ciudad de partida. Por otro lado, un problema continuo es la búsqueda de los valores que se asignan a los parámetros de un modelo en un proceso, para que este modelo reproduzca el comportamiento real observado con la mayor precisión posible. Sin embargo, en la práctica, se podrían también encontrar *problemas mixtos*, los cuales comprenden simultáneamente variables continuas y variables discretas. Dos tipos de problemas son referidos, en la literatura, como los problemas difíciles de optimizar[36]:

- Algunos problemas de optimización discreta, para los cuales no hay conocimiento de un algoritmo polinomial exacto (es decir, que su tiempo de cálculo es proporcional a  $N^n$ , donde  $N$  es el número de parámetros desconocidos del problema, y  $n$  es una constante entera).
- Algunos problemas de optimización con variables continuas, para los cuales no hay conocimiento de un posible algoritmo para definir el óptimo global, es decir, la mejor posible solución.

## TÉCNICAS DE OPTIMIZACIÓN PARA DETECCIÓN DE DAÑOS

Inicialmente se realizaron muchos esfuerzos por un largo tiempo para resolver estos dos tipos de problemas separadamente. En el campo de optimización continua, se puede encontrar un significativo arsenal de métodos tradicionales usados para la optimización global, pero estas técnicas son generalmente inefectivas si la función objetivo no posee una propiedad particular, como por ejemplo convexidad. En el campo de optimización discreta, un gran número de heurísticas, las cuales producen soluciones para el óptimo, fueron desarrolladas, pero la mayoría de estas fueron concebidas específicamente para un problema dado.

Sin embargo con la llegada de la metaheurística se marcó una reconciliación entre ambos dominios: dado que estas pueden ser empleadas a toda clase de problemas discretos y también pueden ser adaptadas para problemas continuos. Además, estos métodos tienen en común las siguientes características[36]:

- Son en cierta medida estocásticos.
- Pueden ser de origen discreto o continuo y no requieren la determinación de gradientes de la función objetivo.
- Son inspirados por analogías: con física (simulated annealing), con biología (evolutionary algorithms) o con etología (ant colony, particle swarms).
- Desventajas: gran tiempo de computación.

Estos métodos no son mutuamente excluyentes: de hecho, en el estado actual de la investigación, generalmente es imposible prever con certeza la eficacia de un método determinado, cuando se aplica a un problema específico. Por otra parte, la tendencia actual es la aparición de métodos híbridos, que tratan de beneficiarse de las ventajas específicas de los diferentes enfoques mediante la combinación de ellos. Finalmente un aspecto importante de las metaheurísticas es que se prestan a todo tipo de extensiones, es decir[36]:

- Optimización multiobjetivo: se realiza la optimización de varias funciones objetivo simultáneamente.
- Optimización multimodal: se requiere localizar un conjunto de óptimos globales o locales (Detección de daños y ajustes de modelos).
- Optimización dinámica: la función objetivo posee variaciones temporales.
- Se pueden realizar implementaciones en paralelo.

Estas técnicas de optimización requieren, la definición de ciertas propiedades específicas o parámetros, los cuales para cada metaheurística son diferentes. Por esta razón, el ajuste y la comparación de las metaheurísticas se han llevado a cabo empíricamente, mediante la explotación de sistemas de análisis de funciones de prueba, cuyos mínimos globales y locales son conocidos.

Para facilitar la discusión, vamos a considerar un ejemplo sencillo del problema de optimización: la colocación de los componentes de un circuito electrónico. La función objetivo a minimizar es la longitud de las conexiones, y los factores desconocidos llamados *variables de decisión*, son los sitios de los componentes del circuito. La forma de la función objetivo de este problema puede ser representada esquemáticamente como se muestra en la figura 3.1. De acuerdo con la *configuración*: cada una de estas es una ubicación en particular, asociado con una opción de valor para cada variable de decisión. Cuando el espacio de las posibles configuraciones tiene una estructura tan variable, es difícil encontrar el mínimo global  $C^*$ . Explicamos a continuación el fracaso de un algoritmo clásico iterativo, antes de comentar la ventaja de que podemos obtener mediante el empleo de una metaheurística[36].

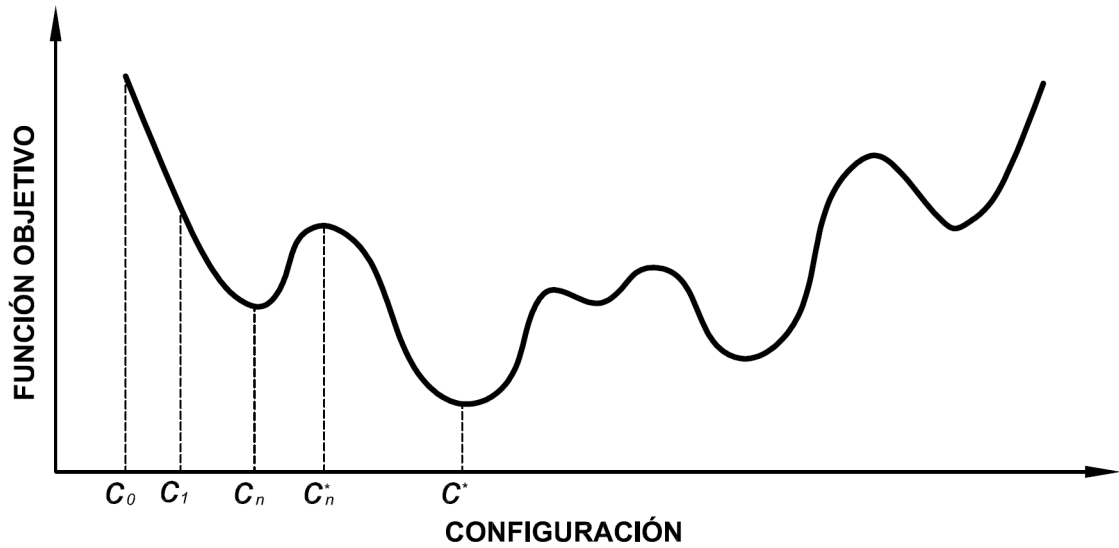


Figura 3.1: Forma de la función objetivo de un problema de optimización

El principio de un algoritmo tradicional de *mejoramiento iterativo* es el siguiente: se parte de una configuración inicial  $C_0$ , que pueden ser seleccionados al azar, o puede ser determinada por el diseñador. Una modificación inicial es realizada (por ejemplo, dos componentes elegidos al azar son permutados, o uno de ellos es trasladado) y luego examinada, y posteriormente los valores de la función objetivo antes y después de esta modificación son comparados. Si el cambio realizado llevó a una reducción en la función objetivo, se acepta, y la configuración  $C_1$  es obtenida, que es un *vecino* de la anterior, y esta nueva configuración se emplea como punto de partida para una nueva prueba. En el caso contrario, se vuelve a la configuración anterior, antes de hacer otro intento. El proceso iterativo es realizado hasta que alguna modificación haga que el resultado sea peor. La figura 3.1 muestra que este algoritmo de mejora iterativa (también indicado como método clásico, o el método de descenso) no conduce, en general, al óptimo global, excepto a una configuración  $C_n$  que es mínimo local, que constituye la mejor solución

de acceso teniendo en cuenta la hipótesis inicial.

Para mejorar la eficacia del método, se puede aplicar varias veces, seleccionado arbitrariamente diferentes condiciones iniciales, y conservando como solución final los mínimos de los mejores locales obtenidos. Sin embargo, este procedimiento aumenta considerablemente el tiempo de cálculo del algoritmo, y no puede encontrar la configuración óptima  $C^*$ . La aplicación repetida del método de descenso no es garantía de su determinación y es especialmente ineficaz cuando el número de mínimos locales crece exponencialmente con el tamaño del problema.

Para superar el obstáculo de los mínimos locales, otra idea demostró ser muy rentable, y que es utilizada por todas las metaheurísticas basadas en vecindades (simulated annealing, tabú search), la cual se trata de permitir, de vez en cuando, movimientos de incremento, es decir, la aceptación de una degradación temporal de la situación, durante un cambio en la configuración actual. Es el caso si se pasa de  $C_n$  a  $C_n^*$  (ver figura 3.1). Un mecanismo para el control de la degradación (específico para cada metaheurística) hace que sea posible evitar la divergencia del proceso. En consecuencia, es posible salir de un mínimo local, para poder explorar otro *lugar* más prometedor. Las metaheurísticas (tales como los algoritmos evolutivos) también tienen mecanismos que permitan la salida de una solución particular. Estos mecanismos (como la mutación en los algoritmos evolutivos) afectan a una solución, en este caso, para ayudar al mecanismo colectivo para luchar contra los mínimos locales, que representa el control de un conjunto de soluciones de la *población*.

### 3.1. PRINCIPIOS DE LAS METAHEURÍSTICAS MÁS CONOCIDAS

#### 3.1.1. ALGORITMO GENÉTICO “GENETIC ALGORITHM”

Los algoritmos evolutivos (AE) son técnicas de búsqueda inspiradas en la evolución biológica de las especies y aparecieron a finales de los años 50. Entre los varios métodos, los algoritmos genéticos (AG) son sin duda los más conocidos. Los métodos evolutivos inicialmente despertaron un interés limitado, debido a su costo significativo en términos de ejecución. Pero ellos han experimentado en los últimos diez años, un desarrollo considerable, que puede atribuirse al aumento significativo en la potencia de cálculo de los ordenadores y, en particular al desarrollo de arquitecturas paralelas [36].

El principio de un algoritmo evolutivo puede ser descrito de manera simple [54, 55]. Un conjunto de  $N$  puntos en un espacio de búsqueda, elegido al azar, constituye la población inicial, cada individuo  $x$  de la población tiene un valor de aptitud determinados, que mide su grado de adaptación al objetivo establecido (minimización o maximización). Los AE consisten en la evolución gradual, en sucesivas generaciones, manteniendo su tamaño constante. Durante las generaciones, el objetivo general es mejorar la aptitud de



## TÉCNICAS DE OPTIMIZACIÓN PARA DETECCIÓN DE DAÑOS

los individuos, esto se obtiene mediante la simulación de los dos principales mecanismos que gobiernan la evolución de los seres vivos, de acuerdo con la teoría de C. Darwin:

- La selección que apoya la reproducción y la supervivencia de los individuos más aptos.
- La reproducción que permite la mezcla, la recombinación y las variaciones de las características hereditarias de los padres, para formar una descendencia con nuevas potencialidades.

En la práctica, una representación debe ser elegida para los individuos de una población. Clásicamente, un individuo podría ser una lista de números enteros para los problemas combinatoriales, un vector de números reales para los problemas numéricos en espacios continuos, una cadena de dígitos binarios para los problemas de Boole, o serán capaces de combinar estas representaciones, incluso en estructuras complejas, si se requiere. El paso de una generación a la siguiente se procede en cuatro fases: una fase de selección, una fase de reproducción (o variación), una fase de evaluación de la aptitud y una fase de la sustitución. La fase de selección designa a los individuos que participan en la reproducción. Los individuos seleccionados estarán disponibles para la fase de reproducción, que consiste en la aplicación de los operadores de variación a las copias de los individuos seleccionados previamente para generar nuevos individuos, la mayoría de los operadores de cruce o recombinación son de uso frecuente, los cuales produce uno o dos hijos de dos padres, y finalmente, la mutación que genera un nuevo individuo a partir de un sólo individuo. La estructura de los operadores de variación depende en gran medida de la representación elegida para los individuos. La aptitud de los nuevos individuos son evaluadas. Por último, la fase de sustitución consiste en la selección de los miembros de las nuevas generaciones: se puede, por ejemplo, sustituir a los individuos más bajos de fitness de la población por los mejores ejemplares producidos, en un número igual. El algoritmo se termina después de un cierto número de generaciones, de acuerdo con un criterio de terminación arbitraria especificado por el usuario. El principio de un algoritmo evolutivo es representado en la figura 3.2.

### 3.1.2. RECOCIDO SIMULADO “SIMULATED ANNEALING”

El Simulated Annealing (SA) es una técnica de optimización combinatorial que se usa para afrontar problemas de gran complejidad matemática, de modo que se puedan obtener soluciones cercanas a la óptima. Esta técnica se fundamenta en el proceso de enfriamiento lento de un material que se encuentra a alta temperatura. El método de SA incorpora el proceso de recocido para la solución de un problema de optimización, donde la función objetivo del problema, es similar a la energía de un material, entonces se realiza la minimización con la ayuda de la introducción de una temperatura ficticia, que es, en este caso, un simple parámetro controlable del algoritmo [56].

En la práctica, la técnica del SA emplea el algoritmo de metrópolis [57], que permite describir el comportamiento de un sistema termodinámico en *equilibrio* a una cierta

## TÉCNICAS DE OPTIMIZACIÓN PARA DETECCIÓN DE DAÑOS

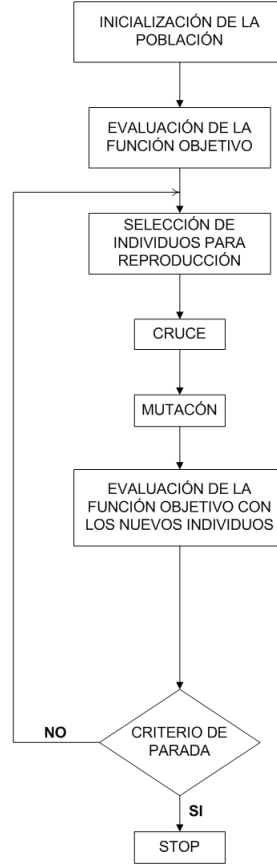


Figura 3.2: Diagrama de flujo del algoritmo genético

temperatura. Por lo tanto teniendo una configuración dada (por ejemplo, una ubicación inicial de todos los componentes), el sistema es sometido a una modificación elemental (por ejemplo, se traslada un componente, o se realiza un intercambio de dos componentes). Si esta transformación logra disminuir la función objetivo (o energía) del sistema, es aceptada. Por el lado contrario, si se produce un aumento de la función objetivo también puede ser aceptada, pero con una probabilidad dada por:

$$e^{-\frac{\Delta E}{k_B T}} \quad (3.1)$$

Donde  $k_B$  es la constante de Boltzmann,  $T$  es la temperatura y  $\Delta E$  es la variación de la temperatura del sistema. Con esta estrategia se evita el quedar atrapado en mínimos locales. Inicialmente cuando la temperatura  $T$  es grande, se permiten configuraciones que generan grandes valores en la función objetivo. A medida que disminuye la temperatura, la probabilidad de aceptar soluciones peores que la que se tiene es cada vez menor. Este proceso se repite de manera iterativa, manteniendo la temperatura constante, hasta que se alcanza el equilibrio termodinámico, concretamente al final de

## TÉCNICAS DE OPTIMIZACIÓN PARA DETECCIÓN DE DAÑOS

un *suficiente* número de modificaciones. Luego se disminuye la temperatura  $T$ , antes de implementar una nueva serie de transformaciones. Cabe resaltar, que la ley de disminución por etapas de la temperatura es a menudo empírica, al igual que el criterio de la finalización del programa. El diagrama de flujo del algoritmo de recocido simulado se presentan esquemáticamente en la figura 3.3.

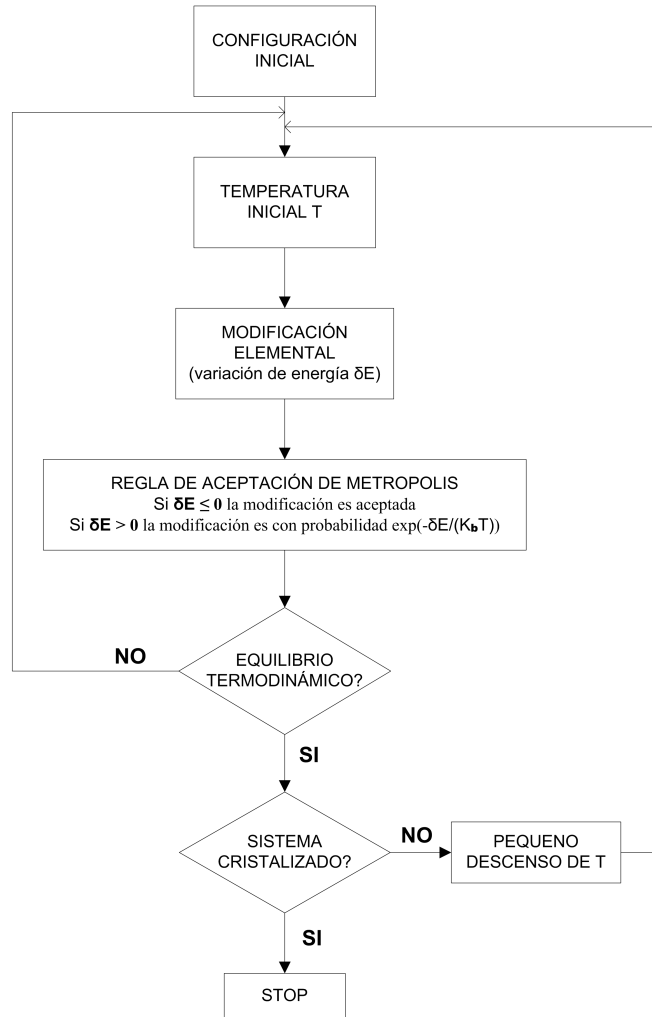


Figura 3.3: Diagrama de flujo del algoritmo Simulated Annealing

Para el correcto funcionamiento del SA, es clave la determinación de la temperatura inicial  $T_0$ , así como la velocidad de enfriamiento y la cantidad de modificaciones permitidas para cada nivel de temperatura  $T_k$ . Estos parámetros se calibran para adaptarse al tipo y tamaño del problema, de tal manera que se consigan soluciones satisfactorias con el SA.

### 3.1.3. OPTIMIZACIÓN CON ENJAMBRE DE PARTÍCULAS “PARTICLE SWARM OPTIMIZATION”

Muchos algoritmos de optimización han sido desarrollados a partir de la simulación de la interacción social de los miembros de una especie en búsqueda de alimento. Uno de estos métodos es el Particle Swarm Optimization (PSO) propuesto inicialmente por Kennedy y Eberhart a mediados de los años noventa [58]. El PSO se fundamenta en el comportamiento reflejado en los grupos de aves, abejas y peces los cuales ajustan sus movimientos para evadir depredadores o buscar una fuente de alimento.

El PSO ha demostrado ser altamente competitivo para resolver un gran variedad de problemas de optimización, además comparado con otros métodos no determinísticos, es considerado robusto y eficiente en términos del número de evaluaciones de la función objetivo. Por otro lado, puede ser implementado con relativa rapidez.

En el algoritmo PSO múltiples candidatos para la solución coexisten y colaboran simultáneamente. Cada solución es denominada como partícula, y está asociada a una posición y velocidad en el espacio multidimensional de diseño. Una partícula *vuela* en el espacio del problema buscando una posición óptima. Durante el proceso de exploración cada partícula ajusta su velocidad y posición a partir de la propia experiencia que ha adquirido como de la experiencia que han adquirido las demás partículas. La experiencia de la partícula es construida memorizando la mejor posición encontrada. Como todas las partículas recuerdan la mejor posición encontrada durante la exploración, el PSO posee una memoria. De esta manera el PSO combina el sistema de búsqueda local (a través de la experiencia propia) con el método de búsqueda global (a través de la experiencia de los vecinos), intentando balancear la exploración y explotación. A diferencia de otros algoritmos evolutivos, no se emplean operadores de cruce o mutación.

En el PSO, la velocidad  $V_i^d$  y la posición  $X_i^d$  de la  $d$ -ésima dimensión de la  $i$ -ésima partícula es ajustada como:

$$V_i^d \leftarrow V_i^d + c_1 \times rand1_i^d \times (pbest_i^d - X_i^d) + c_2 \times rand2_i^d \times (gbest^d - X_i^d) \quad (3.2)$$

$$X_i^d \leftarrow X_i^d + V_i^d \quad (3.3)$$

Donde  $X_i$  y  $V_i$  representan respectivamente la posición y velocidad de la partícula  $i$ ,  $pbest_i$  representa la mejor posición dentro del espacio de búsqueda alguna vez visitada por la partícula  $i$ ,  $gbest$  es la mejor posición dentro del enjambre encontrada hasta el momento, los coeficientes de aceleración  $c_1$  y  $c_2$  indican el grado de confianza en la mejor posición encontrada por la partícula individual (parámetro cognitivo) y la de todo el enjambre (parámetro social) respectivamente,  $rand1$  y  $rand2$  son dos números aleatorios distribuidos uniformemente en el intervalo  $[0 - 1]$ .

Muchos investigadores han realizado diferentes modificaciones al algoritmo con el fin de mejorar su desempeño en diferentes aspectos. En este sentido se encuentran dos enfoques muy reconocidos. El primero es el de Clerc [59] quien presenta el uso de

un coeficiente de constricción necesario para asegurar la convergencia del algoritmo. Dicho coeficiente se incorporó en el PSO como medio para controlar la magnitud de las velocidades, evitando de esta manera la *explosión del enjambre* que era perjudicial para la convergencia en las primeras versiones del PSO. En esta versión del PSO el ajuste de la velocidad se realiza de acuerdo con las siguientes ecuaciones,

$$V_i^d \leftarrow K \left[ V_i^d + c_1 \times rand1_i^d \times (pbest_i^d - X_i^d) + c_2 \times rand2_i^d \times (gbest^d - X_i^d) \right] \quad (3.4)$$

$$K = \frac{2}{2 - \varphi - \sqrt{\varphi^2 - 4\varphi}}, \quad \text{donde} \quad \varphi = c_1 + c_2, \quad \varphi > 4 \quad (3.5)$$

Por otro lado Shi y Eberhart [60] introdujeron un parámetro adicional de peso de inercia  $w$ , el cual principalmente hace que el ajuste de la velocidad dependa de la experiencia previa adquirida por el enjambre a través del  $pbest$  y  $gbest$ , de esta manera el ajuste de la velocidad se realiza mediante,

$$V_i^d \leftarrow w \times V_i^d + c_1 \times rand1_i^d \times (pbest_i^d - X_i^d) + c_2 \times rand2_i^d \times (gbest^d - X_i^d) \quad (3.6)$$

De la misma manera, en los últimos años, bastantes variantes del PSO han sido propuestas, esto debido a que se ha demostrado que el PSO es bastante sensible a la configuración de los parámetros tanto del peso de inercia ( $w$ ) como de los coeficientes de aceleración ( $c_1, c_2$ ). El principio del algoritmo PSO es representado en la figura 3.4.

A continuación se presenta un enfoque propuesto en la literatura, que por sus características de estabilidad, se constituye en un fuerte candidato para usos en D. D.

## PSO BASADO EN EL APRENDIZAJE AUTO-ADAPTATIVO

Un enfoque innovador del PSO es el algoritmo de optimización con enjambre de partículas basado en el aprendizaje auto-adaptativo “*self-adaptive learning based particle swarm optimization*” (SL-PSO) el cual fue propuesto por Wang et al. [61]. La idea básicamente se centra en el ajuste adecuado del operador de velocidad, el cual se realiza de acuerdo a la experiencia adquirida por el PSO durante el proceso de optimización.

### Estrategias de ajuste de velocidad

A continuación se realiza una descripción de las estrategias para el ajuste de la velocidad que emplea el SL-PSO.

El primer ajuste de velocidad se denomina estrategia de ajuste de velocidad basada en la diferencia “*difference-based velocity updating strategy*” (DbV). Este ajuste se fundamenta principalmente en la metodología que emplea la evolución diferencial, la cual realiza el proceso de optimización a partir de la diferencia de información entre los individuos. La DbV además de utilizar la diferencia de información para recombinar

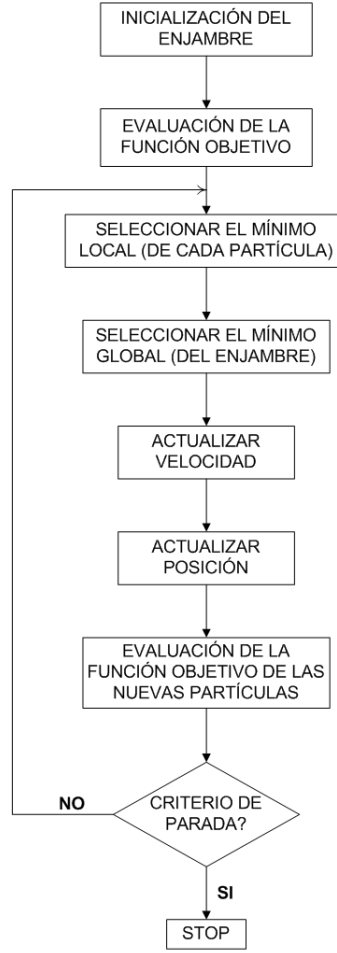


Figura 3.4: Diagrama de flujo del algoritmo PSO

el vector de velocidad, emplea el *pbest* para atraer y guiar la dirección de vuelo de la partícula. En la estrategia DbV las siguientes ecuaciones son usadas para el ajuste de la velocidad:

$$Viad_i^d \leftarrow X_k^d - X_j^d \quad (3.7)$$

$$c = N(0,5, 0,2) \quad (3.8)$$

$$V_i^d \leftarrow c \times Viad_i^d + c \times (pbest_i^d - X_i^d) \quad (3.9)$$

Donde  $X_k^d$  y  $X_j^d$  son las  $d$ -ésimas componentes de dos partículas seleccionadas aleatoriamente dentro de la población,  $Viad_i^d$  es el vector de diferencia, el parámetro  $c$

está definido por  $N(0,5, 0,2)$ , el cual representa un número aleatorio generado de acuerdo a una distribución Gausiana con media de 0,5 y desviación estándar de 0,2.

En segunda instancia el ajuste de la velocidad es la estrategia de aprendizaje global propuesto por Liang et al. [3]. Esta estrategia ha mostrado mejor desempeño en el PSO en problemas multimodales. En el PSO de aprendizaje global “*comprehensive learning PSO*” (CLPSO) la velocidad de cada componente de cada partícula permite ser influenciada por el  $pbest$  de otras partículas. Esta estrategia fomenta la diversidad del enjambre, y por lo tanto la capacidad de resolver problemas multimodales. En CLPSO el ajuste de la velocidad esta dado por:

$$V_i^d \leftarrow w \times V_i^d + c \times rand_i^d \times (pbest_{f_i(d)}^d - X_i^d) \quad (3.10)$$

Donde  $f_i = [f_i(1), f_i(2), \dots, f_i(D)]$  define cuales  $pbest$  de todas las partículas son seleccionadas para ajustar la partícula  $i$ . Entonces el  $pbest_{f_i(d)}^d$  puede ser la correspondiente dimensión del  $pbest$  de alguna otra partícula, incluyendo su propio  $pbest$ , y la decisión depende de la probabilidad  $Pc$ , definida como la probabilidad de aprendizaje, la cual podría tener diferentes valores para diferentes partículas. De esta manera para la partícula  $i$  la probabilidad  $Pc$  teniendo en cuenta el numero de partículas ( $ps$ ) se puede calcular a partir de:

$$Pc_i = 0,5 + 0,45 * \frac{\left(\exp\left(\frac{10(i-1)}{ps-1}\right) - 1\right)}{\exp(10) - 1} \quad (3.11)$$

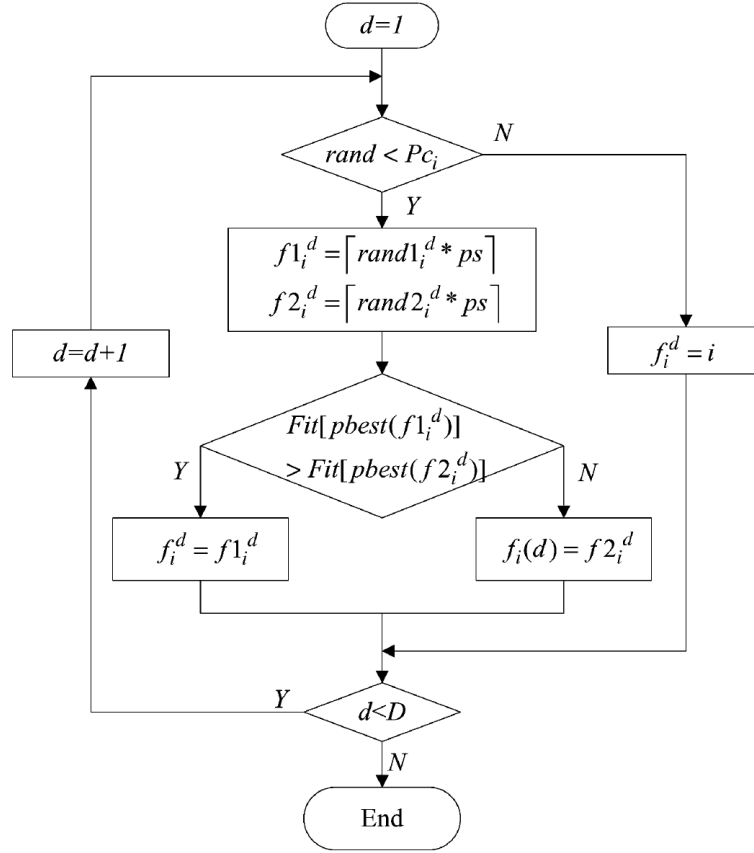
Para cada dimensión de la partícula  $i$  se genera un número aleatorio. Si este número aleatorio es tan grande como  $Pc_i$ , la correspondiente dimensión aprenderá de su propio  $pbest$ , de lo contrario aprenderá del  $pbest$  de otra partícula, la cual sera seleccionada como se describe a continuación:

1. Inicialmente se seleccionan dos partículas dentro de la población, excluyendo la partícula a la que se le realizara el ajuste de la velocidad.
2. Se comparan los valores del fitness de los  $pbest$  de estas dos partículas y se selecciona el mejor.
3. Por ultimo se usa el  $pbest$  seleccionado como ejemplar para aprender de esa dimensión.

Por otro lado el coeficiente  $c$  está definido como 1,49445, y el peso de inercia varía según las iteraciones de acuerdo con:

$$w(k) = \frac{(w_0 - 0,2)(max\_ite - k)}{max\_ite} + 0,2 \quad y \quad w_0 = 0,9 \quad (3.12)$$

Sin embargo se ha demostrado que el rango de búsqueda del CLPSO es muy grande en la mayoría de los problema, por lo tanto Wang et al. [61] desarrollaron una nueva


 Figura 3.5: selección de  $pbest_{f_i(d)}^d$  [3]

variante llamada PSO-CL-pbest que posee una región de búsqueda mas pequeño. En esta variante el ajuste de la velocidad esta dado por:

$$V_i^d \leftarrow w \times V_i^d + 0,5 \times c \times rand_i \times (pbest_{f_i(d)}^d - X_i^d + pbest_i^d - X_i^d) \quad (3.13)$$

Comparado con el CLPSO el PSO-CL-pbest utiliza  $rand_i$  en vez de  $rand_i^d$ , lo cual implica que un número aleatorio fijo es usado para ajustar todos los componentes de un vector de velocidad.

### Estrategia de aprendizaje auto-adaptativo

El principal propósito de emplear la estrategia de aprendizaje auto-adaptativo es incrementar las probabilidades de seleccionar una estrategia adecuada para diferentes problemas, basado en su previo desempeño.

En el SL-PSO, para cada estrategia, se asigna una probabilidad  $proSTR_i$  en cada ejecución, para así determinar la probabilidad de que la  $i$ -ésima estrategia sea adoptada



## TÉCNICAS DE OPTIMIZACIÓN PARA DETECCIÓN DE DAÑOS

para ajustar cada partícula. Inicialmente se asignan probabilidades iguales para todas las estrategias, es decir,  $proSTR_i = \frac{1}{3}$ , para  $i = 1, \dots, 3$ , y se realiza una acumulación para las tres estrategias  $S_i = 0$ . En cada generación las partículas son clasificadas basadas en su mejor valor fitness. Además para cada partícula es asignado un peso de acuerdo con:

$$\omega_i = \frac{\log(ps - i + 1)}{\log(1) + \dots + \log(ps)} \quad i = 1, \dots, ps \quad (3.14)$$

Finalmente el peso de las partículas es acumulado asociado con la estrategia de ajuste. Después de un número fijo de generaciones  $G_s$  (periodo de aprendizaje), la siguiente regla es usada para ajustar la probabilidad de ejecución de la  $i$ -ésima estrategia de ajuste:

$$proSTR'_i = (1 - \alpha) proSTR_i + \alpha \frac{S_i}{G_s} \quad (3.15)$$

$$proSTR_i = \frac{proSTR'_i}{proSTR'_1 + \dots + proSTR'_4} \quad (3.16)$$

Donde  $proSTR'_i$  es una probabilidad de ejecución temporal,  $\alpha$  es una tasa de aprendizaje la cual es usada para controlar la velocidad de aprendizaje de las probabilidades de ejecución durante el proceso de optimización. El principio del SL-PSO es representado en el Algoritmo 1.

## 3.2. EVALUACIÓN DE METAHEURÍSTICAS

En esta sección se realiza la descripción tanto de las funciones de prueba utilizadas para validar los algoritmos como de las métricas empleadas para determinar su respectivo desempeño. Además, se presentan los resultados obtenidos junto con un análisis estadístico del comportamiento de cada uno de los algoritmos, que fueron descritos en la sección anterior. Por último, se realizará una comparación del desempeño de cada uno de los algoritmos, y así, de esta manera poder determinar el de mejor desempeño.

### 3.2.1. FUNCIONES DE PRUEBA

Como ya se ha comentado en las secciones anteriores, las metaheurísticas son técnicas aproximadas no deterministas. Esto implica que diferentes ejecuciones del mismo algoritmo sobre un problema dado no implica que se encuentre la misma solución. Esta propiedad característica de las metaheurísticas supone un problema importante para los investigadores a la hora de evaluar sus resultados y, por tanto, a la hora de comparar su algoritmo con otros algoritmos existentes.

Los problemas seleccionados para evaluar el desempeño de las técnicas anteriormente descritas, fueron de dos clases, funciones de prueba para la validación de algoritmos

---

**Algoritmo 1** Pseudocódigo SL-PSO [61]

---

```

1: Step (0) Initialization
2: Randomly initialize the positions of all particles  $x = (x_1, \dots, x_{ps})$  of de size  $ps$ 
3: Initialize the velocity  $(v_1, v_2, \dots, v_{ps})$ 
4: Set generation  $t = 0$ 
5: Evaluate the fitness values  $F = (fit_1, fit_2, \dots, fit_{ps})$  of  $X$ 
6: Set  $X$  to be  $pbest = (pbest_1, \dots, pbest_{ps})$  for each particle
7: Set the particle with best fitness to be  $gbest$ 
8: Set probabilities of using 3 strategies  $proSTR_i = 1/3, \quad i = 1, \dots, 3$ 
9: Set the learning period  $G_s = 100$ , learning coefficient  $\alpha = 1/6$ 
10: Set accumulators of 3 strategies  $S_i = 0, \quad i = 1, \dots, 3$ 
11: Set weight  $w_i = \frac{\log(ps-i+1)}{\log(1)+\dots+\log(ps)}, \quad i = 1, \dots, ps$ 
12: Step (1) Reproduction and updating loop
13: for  $i = 1 : ps$  do
14:   Select the  $jth$  strategy by roulette wheel selection based on  $proSTR$  for the  $ith$ 
      particle
15:   Updating the velocity  $v_i$  and position  $x_i$  using the  $jth$  strategy
16:   Evaluate the fitness value  $fit_i$  of the new particle  $x_i$ 
17:   if  $x_i$  is better than  $pbest_i$  then
18:     Set  $x_i$  to be  $pbest_i$ 
19:   end if
20:   if  $x_i$  is better than  $gbest$  then
21:     Set  $x_i$  to be  $gbest$ 
22:   end if
23: end for
24: Sort the fitness value  $F$  in descending order  $F' = (fit'_1, \dots, fit'_{np})$ 
25: for  $i = 1 : ps$  do
26:   Find out the  $jth$  strategy that generates the particle with  $fit'_i$ 
27:    $S_j = S_j + w_i$ 
28: end for
29: if  $t = G_s$  then
30:   for  $j = 1 : 3$  do
31:      $proSTR'_j = (1 - \alpha) proSTR_j + \alpha S_j / G_s$ 
32:      $S_j = 0$ 
33:   end for
34:   for  $j = 1 : 3$  do
35:      $proSTR_j = proSTR'_j / (proSTR'_1 + \dots + proSTR'_3)$ 
36:   end for
37: end if
38:  $t = t + 1$ 
39: Step (2) if termination condition is not met, go to Step (1), otherwise end PSO

```

---

## TÉCNICAS DE OPTIMIZACIÓN PARA DETECCIÓN DE DAÑOS

estocásticos de optimización global y funciones de prueba para la validación de algoritmos para la optimización restringida.

### Caso global

Las funciones consideradas para el caso de optimización global fueron tomadas de la literatura especializada y se encuentran en [62, 11]. La principal razón de la elección de este conjunto de funciones, reside en que además de ser utilizadas comúnmente para validar algoritmos de optimización global, este conjunto de funciones reúne características deseables para probar la robustez y efectividad de los algoritmos. Las funciones seleccionadas, incluye funciones unimodales y funciones multimodales para así poder estudiar el desempeño de los algoritmos. La tabla 3.1 muestra las características más relevantes de cada función y en el apéndice A se describen en forma completa. Cabe resaltar que todos los problemas seleccionados son de minimización.

Tabla 3.1: Funciones de prueba para optimización continua sin restricciones

Función	Dimensiones	Espacio de búsqueda	$f_{opt}$
Colville	4	$[-10, 10]$	0
Dixon	10	$[-10, 10]$	0
Ackley	30	$[-30, 30]$	0
Sphere	30	$[-2, 56, 5, 12]$	0
Griewank	30	$[-600, 600]$	0
Rosenbrock	30	$[-10, 10]$	0
Alpine	30	$[-10, 10]$	0
Schwefel	30	$[-10, 10]$	$-418,9829n$
Rastrigin	30	$[-500, 500]$	0

Estas funciones pueden dividirse en dos categorías, cada una con diferente complejidad: funciones unimodales, las cuales son relativamente sencillas de optimizar pero la complejidad se incrementa a medida que aumenta la dimensionalidad del problema; funciones multimodales donde éstas cuentan con múltiples óptimos locales los cuales se incrementan exponencialmente cuando aumenta la dimensionalidad del problema y son los problemas más complejos de resolver para muchos de los algoritmos de optimización.

Las funciones unimodales se utilizan para evaluar el grado de convergencia del algoritmo, mientras que las funciones multimodales se consideran para analizar la habilidad del algoritmo para escapar de óptimos locales y alcanzar el óptimo global.

### Caso restringido

Un problema de optimización restringido se puede formular como sigue: Encontrar los valores de las  $n$  variables  $[x_1, x_2, \dots, x_n]$ , denotadas por el vector  $\bar{x}$ , que satisfacen  $m$  condiciones de desigualdad y  $p$  condiciones de igualdad (restricciones) y optimizan

## TÉCNICAS DE OPTIMIZACIÓN PARA DETECCIÓN DE DAÑOS

(minimizan o maximizan) la función objetivo  $f(\bar{x})$ . Generalmente, el problema de maximizar  $f(\bar{x})$  es equivalente al de minimizar  $-f(\bar{x})$ , (ver figura 3.6). Un problema de optimización con restricciones puede ser escrito como:

$$\text{Min} \quad f(\bar{x}), \bar{x} = [x_1, x_2, \dots, x_n]^T \quad (3.17)$$

Sujeto a

$$g_i(\bar{x}) \leq 0 \quad i = 1, \dots, m \quad (3.18)$$

$$h_j(\bar{x}) = 0 \quad j = 1, \dots, p \quad (3.19)$$

Donde  $\bar{x} = [x_1, x_2, \dots, x_n]^T$  es el vector solución,  $f(\bar{x})$  es la función objetivo, y  $g_i(\bar{x})$  y  $h_j(\bar{x})$  son las restricciones de desigualdad e igualdad respectivamente, donde estas ultimas son usualmente transformadas en restricciones de desigualdad de la forma siguiente[63]:

$$|h_i| - \epsilon \leq 0 \quad i = 1, \dots, p \quad (3.20)$$

Donde  $\epsilon$  es la tolerancia permitida (un valor muy pequeño). Esto permite transformar las restricciones de igualdad a restricciones de desigualdad. Este tipo de transformación se utiliza, ya que, los algoritmos evolutivos tienden a manejar más fácilmente las restricciones de desigualdad que las restricciones de igualdad [12].

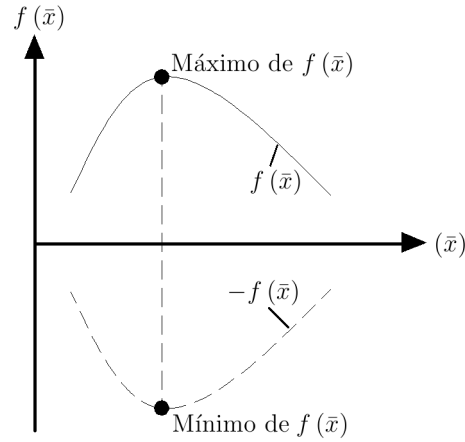


Figura 3.6: EL mínimo de la función  $f(\bar{x})$  es igual al maximo de la función  $-f(\bar{x})$ .

En la figura 3.7 se muestra un ejemplo en el que se ilustran un problema de optimización con restricciones, la región factible y la región infactible. Las restricciones correspondientes a las superficies sobre las cuales un punto en el límite cae son llamadas restricciones activas. Se dice que una restricción, en el punto  $\bar{x}$ , es:

## TÉCNICAS DE OPTIMIZACIÓN PARA DETECCIÓN DE DAÑOS

**Satisfecha**  $\iff g_i(\bar{x}) \leq 0$  o  $h_j(\bar{x}) = 0$

**Activa**  $\iff g_i(\bar{x}) = 0$  o  $h_j(\bar{x}) = 0$

**Inactiva**  $\iff g_i(\bar{x}) < 0$

**Violada**  $\iff g_i(\bar{x}) > 0$  o  $h_j(\bar{x}) \neq 0$

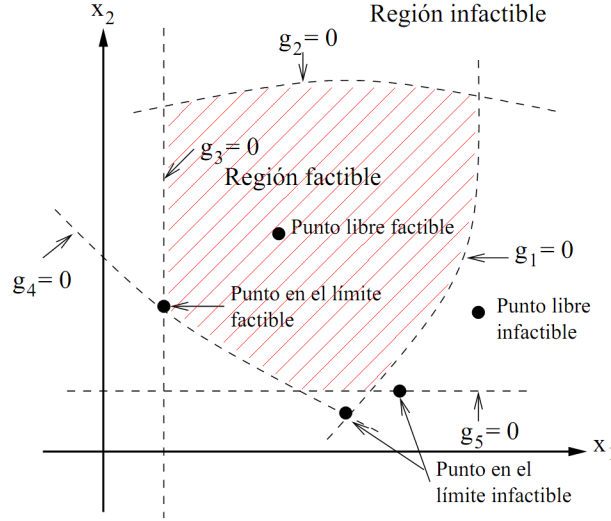


Figura 3.7: Ejemplo hipotético para un problema con dos variables de diseño y cinco restricciones de desigualdad.

Para el caso de problemas con restricciones se seleccionaron las funciones de referencias de Michalewicz [64] los cuales fueron extendidos por Runarsson [63]. En este conjunto de funciones de prueba se incluyen problemas con restricciones de desigualdad y de igualdad, espacios de búsqueda de alta y baja dimensionalidad, espacios convexos y no convexos, además de zonas factibles disyuntas y/o muy pequeñas. En la tabla 3.2, se resumen las características de las trece funciones de prueba adoptadas para evaluar las respectivas técnicas mencionadas.

$LI$  indica el número de desigualdades lineales,  $NI$  corresponde al número de desigualdades no lineales,  $LE$  es el número de igualdades lineales,  $NE$  es el número de igualdades no lineales,  $n$  indica el número de variables de decisión y  $\rho$  representa una estimación del tamaño de la región factible con respecto a todo el espacio de búsqueda.  $\rho$  se obtiene generando un millón de soluciones aleatorias. y evaluando qué porcentaje de ellas son factibles.

### Manejo de restricciones

Muchos de los problemas de optimización que existen en el mundo real están sujetos a un conjunto de restricciones. Las técnicas de programación matemática existentes

Tabla 3.2: Funciones de prueba para optimización continua con restricciones

Problema	$n$	Función	$\rho$	LI	NI	LE	NE
$g01$	13	cuadrática	0,0003 %	9	0	0	0
$g02$	20	no lineal	99,9973 %	1	1	0	0
$g03$	10	no lineal	0,0026 %	0	0	0	1
$g04$	5	cuadrática	27,0079 %	0	6	0	0
$g05$	4	no lineal	0,0000 %	2	0	0	3
$g06$	2	no lineal	0,0057 %	0	2	0	0
$g07$	10	cuadrática	0,0000 %	3	5	0	0
$g08$	2	no lineal	0,8581 %	0	2	0	0
$g09$	7	no lineal	0,5199 %	0	4	0	0
$g10$	8	lineal	0,0020 %	3	3	0	0
$g11$	2	cuadrática	0,0973 %	0	0	0	1
$g12$	3	cuadrática	4,7697 %	0	9 <sup>3</sup>	0	0
$g13$	5	no lineal	0,0000 %	0	0	1	2

para la resolución de los mismos cuentan con varias limitaciones, por ejemplo cuando los problemas cuentan con algunas de las siguientes características: la función objetivo y/o las restricciones son no diferenciables o discontinuas, cuando la región factible es disyunta o cuando la función objetivo y/o las restricciones no pueden ser expresadas de manera algebraica [12].

Sin embargo, los Algoritmos Evolutivos (AEs) han podido resolver gran variedad de problemas que cuentan con las características antes mencionadas. A pesar de que los AEs en su forma original son técnicas para optimización sin restricciones. De ahí la necesidad de implementar técnicas que permitan incorporar la información pertinente sobre la violación de restricciones en la función objetivo, y así poder incorporar mecanismos a los AEs que les permitan resolver problemas con restricciones.

Las funciones de penalización son uno de los mecanismos que se utiliza más comúnmente para manejar restricciones en los AEs. La idea de esta técnica es transformar un problema con restricciones en uno sin restricciones al agregar a la función objetivo los factores de penalización tanto para restricciones de igualdad como para las de desigualdad y, de esta forma, si las restricciones son violadas, entonces el valor de la función objetivo se ve afectada por un costo denominado *la penalización*.

El uso de esta técnica de penalización tiene una seria desventaja: el resultado depende de los factores de penalización y estos son difíciles de determinar. Se debe cuidar que los factores de penalización estén equilibrados con respecto a los valores de la función objetivo y además se debe determinar qué tanto se quiere penalizar a una solución infactible. Si se eligen valores muy pequeños, el algoritmo sólo considerará el valor de la función objetivo, pues no penalizará suficientemente a las soluciones infactibles, y por lo tanto puede no llegar a la zona factible. Si se eligen valores muy grandes, el algoritmo llegará a la zona factible, pero no podrá oscilar entre la zona factible y la infactible lo

cual impedirá llegar al óptimo en los casos en que éste se encuentre en la frontera entre estas dos zonas.

En el presente trabajo se empleará un control de la violación de las restricciones por medio de una función de penalización la cual será aplicada para el SL-PSO, la cual para todas las restricciones violadas, tiene la siguiente forma.

$$F_i(\bar{x}) = f_i + Q_i \quad (3.21)$$

Donde:

$$Q_i = c \times \sum_{i=1}^n \frac{g_i(\bar{x})}{\max[g_i(\bar{x})]_h} \quad (3.22)$$

En esta expresión  $F_i(\bar{x})$  es la función objetivo a optimizar,  $g_i(\bar{x})$  y  $[g_i(\bar{x})]_h$  son funciones de las restricciones y  $c$  es una constante positiva llamada *factor de penalización*, la cual es definida por el usuario.

El mecanismo esta basado en [12] donde se diseño un esquema de penalización para elegir los líderes del enjambre de partículas mediante el uso de reglas. Las reglas para seleccionar entre dos individuos son muy simples: si se comparan dos partículas factibles, se escoge a al que tenga mejor función objetivo. Si se compara una partícula factible con una infactible, se elige la partícula factible. Si se compara dos partículas infactibles, se elegirá a la partícula que viole menos las restricciones del problema.

El proceso de penalizar consiste en normalizar la violación de las restricciones de las partículas. El proceso de penalización de una partícula se muestra en el Algoritmo 2 y se ilustra el funcionamiento del mismo en la figura 3.8.

---

**Algoritmo 2** Penalización de una partícula  $\bar{x}_k$

---

```

1:  $SUM \leftarrow 0$ 
2: for  $i = 1 : 1 : Nr$  do
3:   if  $g_i(\bar{x}) > 0$  then
4:      $SUM \leftarrow SUM + \frac{g_i(\bar{x})}{\max[g_i(\bar{x})]_h}$ 
5:   end if
6: end for
7:  $F = aptitud + SUM * c$ 
```

---

Como se puede observar en el Algoritmo 2,  $SUM$  es el valor de la normalización de las restricciones violadas,  $Nr$  representa el número de restricciones del problema,  $g_i(\bar{x})$  corresponde al valor de la restricción  $i$  de una partícula, por otro lado  $\max[g_i(\bar{x})]_h$  es el valor de máxima violación de la restricción  $i$  dentro del enjambre de partículas, es decir,  $h = 1, 2, \dots, \#particulas$ .

### 3.2.2. ANÁLISIS DE RESULTADOS

En esta sección se muestran los resultados obtenidos con los algoritmos descritos anteriormente, los cuales fueron empleados para problemas de optimización con y sin

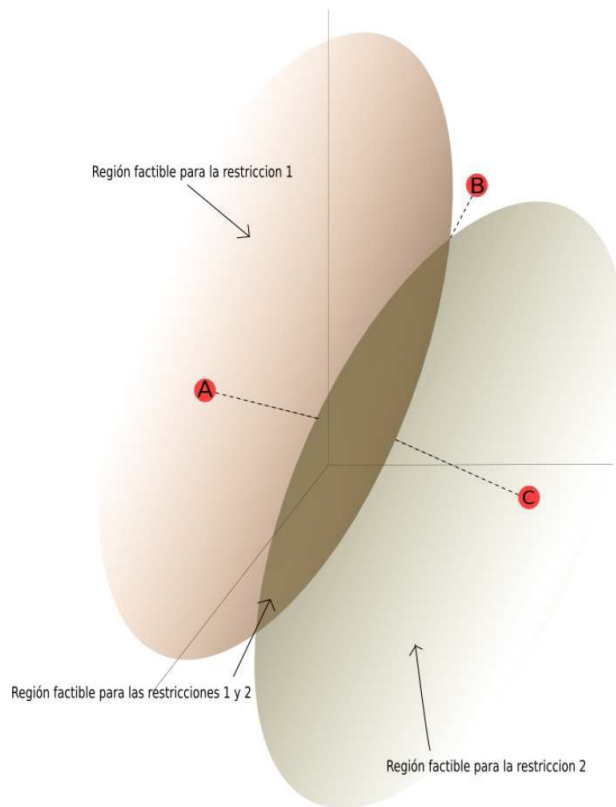


Figura 3.8: Esquema de manejo de restricciones.

restricciones. Para poder evaluar el desempeño final de cada uno de los algoritmos descritos anteriormente, tanto para el caso global como para el restringido, se realizarán 30 ejecuciones independientes. Una vez realizadas las ejecuciones para una determinada función, obtenemos 30 soluciones, las cuales conformarán nuestro conjunto de datos experimentales. Posteriormente, se aplica un análisis estadístico al conjunto de datos experimentales considerando las métricas usadas en [65], las cuales son:

- Mejor valor encontrado.
- Peor valor encontrado.
- Promedio.
- Mediana.
- Desviación estándar (Desv. Est.).



## TÉCNICAS DE OPTIMIZACIÓN PARA DETECCIÓN DE DAÑOS

### Caso global

Para la implementación de las metaheurísticas en las funciones de pruebas para el caso de optimización global, el autor realizó la programación en Matlab® de cada uno de los algoritmos seleccionados, los cuales son: algoritmo genético, simulated annealing, PSO y SL-PSO.

En el apéndice B, se presenta en una forma detallado cada uno los códigos desarrollados en MatLab para los algoritmos.

Los resultados obtenidos de los algoritmos se pueden observar en las tablas 3.3, 3.4, 3.5 y 3.6.

Tabla 3.3: Resultados AG

Función	Optimo	Mejor	Promedio	Mediana	Peor	Desv. Est.
Colville	0	175,2150	1509,2434	1348,0297	3806,2987	992,4602
Dixon	0	587,3253	2537,9976	2404,6060	5366,1192	1267,3579
Ackley	0	19,9900	20,388	20,3504	20,6123	0,1568
Sphere	0	2044,9661	2467,9141	2470,7835	2871,2878	224,4514
Griewank	0	460,6125	559,6665	567,7702	637,9534	43,9858
Rosenbrock	0	1733271,8646	2588206,0076	2710471,8739	3390286,7487	446675,4942
Alpine	0	52,1899	60,0830	60,6544	66,3654	3,4345
Schwefel	-12569,487	-3694,1885	-2613,9838	-2532,6532	-1891,3816	424,9562
Rastrigin	0	296,2764	407,1171	423,1890	455,6048	37,8595

De acuerdo a los resultados presentados en la tabla 3.3, se puede observar que el AG no obtiene buenos resultados en cada una de las funciones de prueba, pues no logro encontrar el óptimo en ninguna de las nueve funciones utilizadas. Cabe mencionar que el algoritmo implementado corresponde a una de las versiones propuestas inicialmente. Entonces se puede concluir que en el AG es necesario modificar algunos parámetros o complementarlo con otra técnica, para así mejorar su desempeño.

Tabla 3.4: Resultados SA

Función	Optimo	Mejor	Promedio	Mediana	Peor	Desv. Est.
Colville	0	0,1629	5,4413	5,1270	13,2736	2,9197
Dixon	0	16,3138	88,7372	84,4834	187,5232	36,2830
Ackley	0	18,3955	18,9247	18,9302	19,2781	0,2431
Sphere	0	62,1780	83,7783	85,1154	98,7493	8,2034
Griewank	0	244,0307	292,5196	295,2217	336,0386	23,2161
Rosenbrock	0	1528,4747	2111,3170	2131,1779	2457,9455	213,8727
Alpine	0	30,6058	36,8376	37,1812	40,1675	2,3113
Schwefel	-12569,487	-5534,2078	-5049,7527	-5005,5930	-4561,3664	249,2718
Rastrigin	0	242,65	301,7235	308,6174	321,4103	17,1704

A pesar de que el SA también corresponde a una de las versiones iniciales, presenta mejores resultados que el AG en la mayoría de las funciones, sin embargo, no logró determinar el óptimo en ninguna de las nueve funciones utilizadas.

## TÉCNICAS DE OPTIMIZACIÓN PARA DETECCIÓN DE DAÑOS

Tabla 3.5: Resultados PSO

Función	Optimo	Mejor	Promedio	Mediana	Peor	Desv. Est.
Colville	0	$1,7749E - 31$	$1,3377E - 28$	$5,9627E - 29$	$5,2376E - 28$	$1,4624E - 28$
Dixon	0	0	0,3677	0,5007	0,5011	0,2244
Ackley	0	$6,2172E - 15$	$3,4841E - 09$	$6,2172E - 15$	$1,0452E - 07$	$1,908E - 08$
Sphere	0	$1,0778E - 162$	$2,2856E - 14$	$1,2377E - 45$	$5,9267E - 13$	$1,0833E - 13$
Griewank	0	0	0,0174	0,0074	0,0829	0,0230
Rosenbrock	0	8,3009	36,6870	22,5352	88,7966	26,9963
Alpine	0	$6,5780E - 10$	0,0210	0,0024	0,1338	0,0388
Schwefel	-12569,487	-9805,8955	-8572,1468	-8687,5440	-7040,7378	585,5009
Rastrigin	0	1,9899	16,0259	13,4319	48,8239	12,7960

En los resultados presentados por el PSO en la tabla 3.5 se puede observar la superioridad con respecto al AG y el SA, dado que logró determinar el óptimo en seis de las nueve funciones de pruebas.

Tabla 3.6: Resultados SL-PSO

Función	Optimo	Mejor	Promedio	Mediana	Peor	Desv. Est.
Colville	0	0	0	0	0	0
Dixon	0	0	0	0	0	0
Ackley	0	$2,6645E - 15$	$1,0131E - 02$	$2,6645E - 15$	$2,6732E - 01$	$4,8904E - 02$
Sphere	0	0	0	0	0	0
Griewank	0	0	0	0	0	0
Rosenbrock	0	0	$4,1633E - 29$	$2,4202E - 29$	$2,0218E - 28$	$5,0706E - 29$
Alpine	0	$4,0696E - 04$	$1,0048E - 03$	$8,8614E - 04$	$3,9660E - 03$	$6,2521E - 04$
Schwefel	-12569,487	-12448,6746	-12006,0960	-11969,8860	-11436,4851	235,6394
Rastrigin	0	0	0	0	0	0

A continuación en la tabla 3.7, se realiza la comparación en terminos de la mejor solución encontrada por cada uno de los cuatro algoritmos estudiados.

Tabla 3.7: Comparación de las mejores soluciones encontradas por el AG, SA, PSO y S-LPSO

Función	Optimo	AG	SA	PSO	SL-PSO
Colville	0	175,22	0,16	0	0
Dixon	0	587,33	16,31	0	0
Ackley	0	19,99	18,40	0	0
Sphere	0	2044,97	62,18	0	0
Griewank	0	460,61	244,03	0	0
Rosenbrock	0	1733271,86	1528,47	8,30	0
Alpine	0	52,19	30,61	0	0
Schwefel	-12569,49	-3694,19	-5534,21	-9805,90	-12448,67
Rastrigin	0	296,28	242,65	1,99	0

## TÉCNICAS DE OPTIMIZACIÓN PARA DETECCIÓN DE DAÑOS

De acuerdo con los resultados presentados, se puede observar que el algoritmo que arroja mejores resultados para el caso de funciones continuas de optimización global es el SL-PSO, el cual posee un porcentaje mayor de éxito muy evidente con respecto al AG y el SA, con relación a funciones unimodales y multimodales, y con el PSO con respecto a funciones multimodales.

### Caso restringido

En esta sección se realizará un estudio comparativo entre los algoritmos AG, SA, PSO y SL-PSO, pero cabe resaltar, que el algoritmo SL-PSO fue el único algoritmo que fue realizado e implementado por el autor, mientras que para el caso del AG, SA, PSO básico, los resultados fueron recopilados de estudios reportados en la literatura internacional.

Para el caso particular del SL-PSO los parámetros seleccionados fueron: 100 partículas con un número fijo de 30000 iteraciones y los demás parámetros fueron tomados de [61].

En el Apendice B, se encuentra el código del SL-PSO, el cual fue realizado en MatLab, donde se especifica cada uno de los aspectos que se tomaron en cuenta para la implementación de este algoritmo en el caso de optimización con restricciones.

A continuación, en la tabla 3.8, se pueden observar los resultados obtenidos de la implementación del algoritmo SL-PSO, en los problemas de referencia que fueron descritos en las secciones anteriores.

Tabla 3.8: Resultados SL-PSO

Función	Optimo	Mejor	Promedio	Mediana	Peor	Desv. Est.
$g01$	-15	-15,0000000	-15,0000000	-15,0000000	-15,0000000	$1,3416E - 13$
$g02$	0,803619	0,8035774	0,8027947	0,8033406	0,7973566	$1,5375E - 03$
$g03$	1	1,0050008	1,0046303	1,00477771	1,0031768	$4,1653E - 04$
$g04$	-30665,539	-30665,5386718	-30665,5386718	-30665,5386718	-30665,5386718	$1,1101E - 11$
$g05$	5126,4981	5126,5118791	11617,6646264	5302,7278407	98977,3189511	$1,8653E + 04$
$g06$	-6961,81388	-6961,8138756	-6961,8138756	-6961,8138756	-6961,8138756	$1,2532E - 10$
$g07$	24,3062091	24,3081926	24,3724689	24,3685206	24,4378505	$3,0348E - 02$
$g08$	0,0958250	0,0958250	0,0958250	0,0958250	0,0958250	$2,8230E - 17$
$g09$	680,6300573	680,6300574	680,6315671	680,6300574	680,6361601	$2,1555E - 03$
$g10$	7049,3307	7057,1589396	7119,2396768	7108,7366037	7243,8147382	$4,6642E + 01$
$g11$	0,75	0,7499000	0,7499174	0,7499053	0,7500263	$2,7459E - 05$
$g12$	1	1,0000000	1,0000000	1,0000000	1,0000000	0
$g13$	0,0539498	0,547183	0,1290860	0,0966317	0,3692778	$8,2899E - 02$

En los resultados presentados en la tabla 3.8, se puede observar que el algoritmo SL-PSO posee la capacidad de solucionar problemas con restricciones, sin embargo cabe resaltar la dificultad en la solución de los problemas  $g05$  que corresponde a una función no lineal y  $g10$  que es una función lineal.

A continuación en la tabla 3.9, se realiza la comparación en términos de la mejor solución encontrada por SL-PSO con el AG, SA y PSO, respectivamente.

## TÉCNICAS DE OPTIMIZACIÓN PARA DETECCIÓN DE DAÑOS

Tabla 3.9: Comparación de las mejores soluciones encontradas por el SL-PSO con el AG [10], SA [11] y PSO [12].

Función	Óptimo	SL-PSO	AG	SA	PSO
$g01$	-15	-15,0000000	-15,00	-14,999105	-15,000000
$g02$	0,803619	0,8035774	0,803600	0,7549125	0,803432
$g03$	1	1,0050008	1,00	1,0000015	1,004720
$g04$	-30665,539	-30665,5386718	-30665,539	-30665,5380	-30665,500000
$g05$	5126,4981	5126,5118791	5126,498	5126,4981	5126,640000
$g06$	-6961,81388	-6961,8138756	-6961,814	-6961,81388	-6961,810000
$g07$	24,3062091	24,3081926	24,312	24,310571	24,351100
$g08$	0,0958250	0,0958250	0,095825	0,095825	0,095825
$g09$	680,6300573	680,6300574	680,630	680,63008	680,638000
$g10$	7049,3307	7057,1589396	7049,499	7059,86350	7057,590000
$g11$	0,75	0,7499000	0,7500	0,7499990	0,749999
$g12$	1	1,0000000	1,000	1,00000000	1,000000
$g13$	0,0539498	0,0547183	0,053964	0,0539498	0,068665

De acuerdo con los resultados presentados anteriormente, se puede evidenciar la capacidad de los algoritmos para la solución de este tipo de problemas de optimización, sin embargo, dado que el SL-PSO tuvo un mejor desempeño en los problemas de optimización global, este algoritmo será el empleado en el presente trabajo.

---

## METODOLOGÍA PROPUESTA

---

En este capítulo se presenta la metodología propuesta para la solución del problema de identificación de daños. El principal componente del procedimiento de detección de daños presentado, se fundamenta en el cálculo del índice de daño y el uso del algoritmo de optimización SL-PSO, el cual fue descrito en la sección anterior.

El método de detección empleado identificará daños en forma global basado en los cambios de la frecuencias y formas modales, permitiendo mejorar la estimación de los daños presentes en la estructura.

A continuación se presenta el procedimiento empleado en la metodología desarrollada para la identificación de daños:

1. Obtención de parámetros modales, tanto del estado sin daño como para los casos de daño estudiados, a partir de la información experimental de la estructura.
2. Realización del modelo numérico.
3. Formulación del problema de optimización a partir de los parámetros dinámicos disponibles.
4. Ajuste del modelo numérico empleando el SL-PSO.
5. Identificación de los casos de daño disponibles a partir del modelo ajustado, la información experimental y el SL-PSO.
6. Calculo del índice de daño.

Una vez descrito el procedimiento general empleado en la metodología, se procede a la descripción detallada de cada uno de sus componentes. Así, inicialmente se describirá el proceso para la obtención de los parámetros dinámicos del modelo a estudiar. Posteriormente, se presentará el modelo de elementos finitos empleado para la identificación. Luego, se formulará el problema de optimización, donde se establecerá la función objetivo en terminos de la definición de los parámetros de daño, a la cual se le

realizará un proceso de minimización a partir de la diferencia entre los datos experimentales de la estructura real y los datos del modelo numérico mediante el algoritmo SL-PSO. Dado que esta investigación se desarrolla a partir de información experimental, después de realizado el modelo numérico, se explicará lo pertinente a su respectivo ajuste. Finalmente se describirá la formulación del índice de daño, el cual fue propuesto por Wang et al. [13].

### 4.1. OBTENCIÓN DE PARÁMETROS MODALES A PARTIR DE LA INFORMACIÓN EXPERIMENTAL DE LA ESTRUCTURA

Generalmente, el proceso de la caracterización de las propiedades dinámicas (frecuencias naturales, formas modales y amortiguamiento) de una estructura a partir de los datos adquiridos de su respuesta vibracional, se realiza por medio del análisis modal experimental (AME).

Cabe resaltar que la planificación y ejecución de un ensayo modal constituyen tanto una ciencia como un arte. Los libros de Ewins [66] y Maia et al. [67] presentan los lineamiento generales referentes a la elaboración de un ensayo modal.

En términos generales, el AME se puede dividir en: el procesamiento de señales y la estimación de los parámetros modales.

#### 4.1.1. PROCESAMIENTO DE SEÑALES

El procesamiento de señales, se encarga del análisis de las señales eléctricas, dado que, en la excitación de una estructura, su movimiento físico es capturado por los transductores, los cuales son colocados en el sistema durante los ensayos dinámicos y son los encargados de transformar la magnitud física de movimiento registrado en un voltaje que ingresa a la cadena de medición. Los transductores comunes para la medición de las vibraciones incluyen acelerómetros piezoeléctricos, que comúnmente están compuestos de materiales tales como cuarzo o cerámica, así, cuando un sensor piezoeléctrico experimenta un cambio en la carga, como la fuerza de compresión, una carga eléctrica es generada [68].

Generalmente, las variables de respuesta del análisis de la señal se puede presentar en el dominio del tiempo o en el dominio de la frecuencia.

La principal ventaja de las técnicas de identificación en el dominio del tiempo es que se evitan las distorsiones producidas durante la transformación de las señales al dominio de la frecuencia por los efectos de la digitalización. El mayor inconveniente es la necesidad de invertir matrices con datos experimentales de gran tamaño. Además, la Función de Respuesta a Pulso (FRP) no puede utilizarse directamente en la calibración del modelo analítico sino que requieren una transformación al modelo modal o al dominio de la frecuencia [69].

## METODOLOGÍA PROPUESTA

Sin embargo, el principal interés de los modelos experimentales de respuesta es en el dominio de la frecuencia dado que permiten la correlación directa con el modelo analítico para efectos de la calibración [69].

Una de las principales preocupaciones, es la conversión de señales analógicas a sus correspondientes valores digitales. La otra preocupación es la transformación de los datos digitales del dominio del tiempo al dominio de frecuencia usando el algoritmo de la transformada discreta de Fourier.

Dado que, el algoritmo de la transformada de Fourier implica datos discretos durante un período de tiempo limitado, los errores de procesamiento de señales digitales, tales como “*aliasing*” y “*leakage*” se pueden presentar. Además, las interferencias debido al ruido eléctrico en las señales del transductor también pueden crear errores. En el procesamiento de la señal, estos errores son reducidos tanto como sea posible mediante el uso de diferentes técnicas para obtener un mejor espectro de resultados [70].

El “*aliasing*” es un fenómeno que se produce cuando la tasa de muestreo es menor que dos veces la frecuencia más alta en los datos. Es un resultado de la incapacidad de la transformada de Fourier para decidir qué frecuencias están dentro y cuales están fuera de la banda de análisis [70]. Así, el “*aliasing*” produce una representación distorsionada de los datos, lo que podría dar lugar a la interpretación errónea de las frecuencias en el análisis de vibración.

Adicionalmente durante el proceso de transformación rápida de Fourier, se pueden presentar el “*leakage*”, el cual, es la distorsión no deseada causada por el truncamiento artificial de datos muestreados. Una herramienta común utilizada para superar el problema del “*leakage*” es de ventanas. Hay algunas funciones que desarrollan las ventanas, tales como uniformes, Hamming, Hanning, forzada y ventanas exponenciales [70].

Generalmente, las pruebas de impacto son el método de caracterización dinámica mas popular hoy en día, dado que dichas mediciones se pueden realizar de manera rápida y económica. En la figura 4.1, se puede obtener una descripción general de la prueba de impacto.

Sin embargo, como a todas las estructuras no se les pueden realizar pruebas de impacto, debido a su limitado rango de frecuencia o su baja densidad de energía en un amplio espectro, ya que puede que la fuerza de impacto no sea suficiente para excitar adecuadamente los modos de interés. Por lo tanto, las mediciones de FRF se deben hacer proporcionando excitación artificial con uno o más agitadores, que se adjuntan a la estructura. Los tipos comunes de excitadores son agitadores electrodinámicos e hidráulicos. Una prueba típica agitador se representa en la Figura 4.2.

Para el caso de la estructura de referencia, los datos experimentales fueron generados por tres tipos de excitación [8]:

**Prueba con martillo de impacto** : Para cada configuración, la estructura se golpeó tres veces en cada dirección por el martillo en una ubicación fija.

**Vibración ambiental** : La duración de los datos registrados fue de 300s para cada configuración.

## METODOLOGÍA PROPUESTA

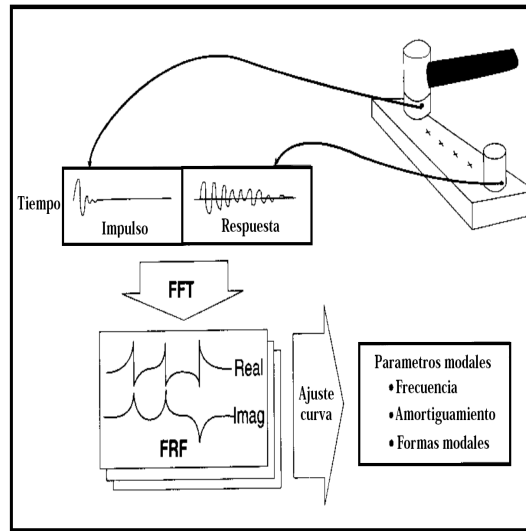


Figura 4.1: Configuración de la prueba de Impacto [4]

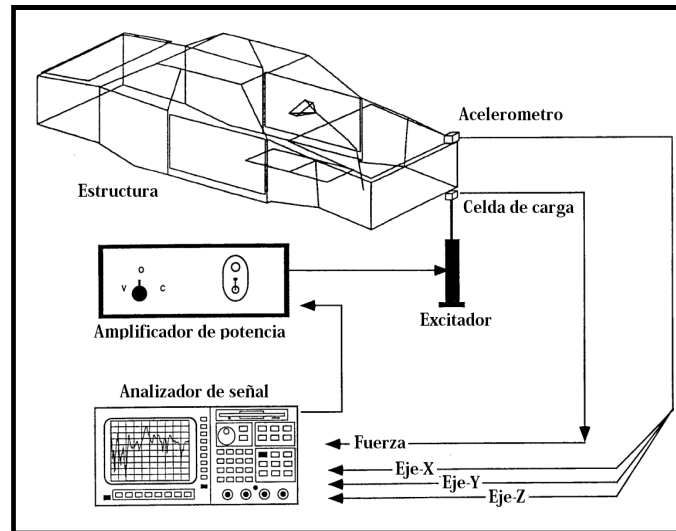


Figura 4.2: Configuración de la prueba de Excitador [4]

**Prueba con shaker electrodinámico :** El agitador se colocó más o menos en el centro de uno de los cuatro vanos en la parte superior de la estructura.

### 4.1.2. ESTIMACIÓN DE LOS PARÁMETROS MODALES

En general, un sistemas de parámetros modales (frecuencias naturales, formas modales y amortiguamiento) se pueden estimar a través de tres enfoques diferentes. En los sistemas analíticos, los parámetros modales se determinan a partir de las matrices del sistema mediante la formulación de la solución propia del sistema. En el análisis modal



## METODOLOGÍA PROPUESTA

experimental (AME), los parámetros modales se extraen ya sea desde las FRF en el dominio de la frecuencia, o de la función de respuesta de impulso (FRI) en el dominio del tiempo. Los tres diferentes métodos de estimación de los parámetros modales se ilustran en la figura 4.3.

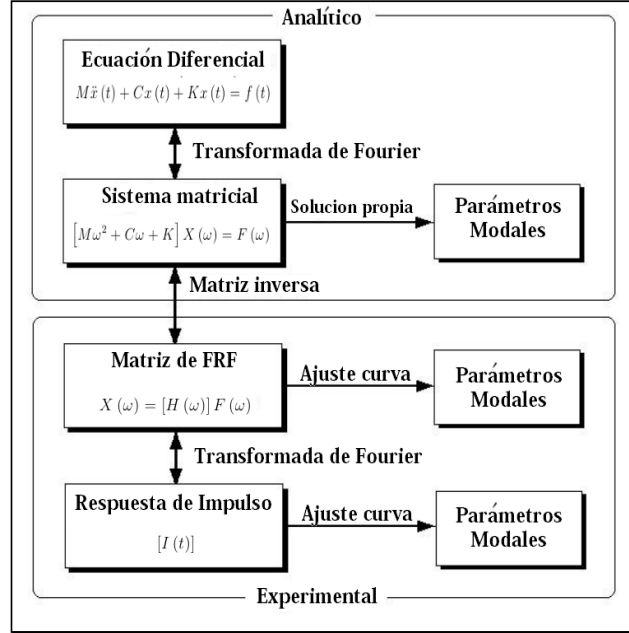


Figura 4.3: Fuentes de parámetros modales [4]

En el AME, los parámetros modales se calculados principalmente en el dominio de la frecuencia por medio de las FRF y técnicas de ajuste de curvas. En general, el ajuste de curvas es un proceso de hacer coincidir un modelo matemático para un conjunto de puntos de datos medidos. Esto se hace mediante la minimización del error entre la función analítica y de los datos medidos de entrada-salida[4].

Dentro de los estudios realizados para la identificación de los parámetros modales de la estructura de referencia de la IASC-ASCE, Ventura et al [2], emplearon el software computacional Artemis©Extractor Versión 3.2 para identificar las frecuencias naturales y formas modales de dicha estructura. Los datos fueron analizados utilizando tanto el “*Frequency Domain Decomposition*” (FDD) en el dominio de la frecuencia y el “*Stochastic Subspace Identification*” (SSI) en el dominio del tiempo.

En el estudio llevado a cabo por Caicedo [7], se realizó la implementación de la *Natural Excitation Technique* (NExT) la cual utiliza el registro de las pruebas de vibración ambiental, dado que permiten la identificación modal sin conocer las fuerzas de excitación de la estructura. Por otro lado aplicaron el *Eigensystem Realization Algorithm* (ERA), el cual es utilizado para obtener los parámetros modales de la estructura de los registros de vibración libre.

## 4.2. MODELO NUMÉRICO ADAPTADO

Como ya se mencionó anteriormente, el problema de identificación de daños se puede abordar mediante dos enfoques:

El primero es la utilización de la señal de respuesta del sistema estructural para resolver el problema de la detección de daño, por lo tanto, solamente se emplean los datos experimentales. De esta manera, inicialmente se establecen un conjunto de características para definir la condición normal a la estructura para utilizarlos como referencia, y durante el posterior proceso de monitoreo a la estructura, las nuevas características se estiman y se realiza una comparación con las de referencia, y en el momento de la generación de cualquier desviación significativa, un algoritmo indicaría la presencia de la anomalía.

El segundo enfoque, hace referencia al uso de modelos numéricos para realizar la estimación de daños dentro del sistema estructural. Este enfoque es muy atractivo, dado que hoy en día se pueden generar modelos de estructuras muy complejas a un costo relativamente bajo, además, este tipo de estudios permiten el análisis de fenómenos de interés que no pueden medirse directamente. Adicionalmente, un aspecto importante a resaltar, es la capacidad de los métodos empleados con este enfoque para llegar hasta el nivel 3 de identificación de daños, siempre y cuando se pueda obtener gran información de la estructura monitoreada.

En el presente trabajo se empleará un método de identificación que utiliza un modelo numérico para la detección de daños, por lo tanto, es necesario realizar un análisis modal numérico, dado que este nos proporciona las propiedades dinámicas más importantes de la estructura en vibración libre, que son las frecuencias naturales y sus correspondientes modos de vibración. De esta manera, la ecuación que relaciona las características dinámicas, con las propiedades físicas de la estructura, sin considerar el amortiguamiento está dada por:

$$[K] - \omega^2 [M] \{\phi\} = \{0\} \quad (4.1)$$

Donde  $[K]$  y  $[M]$  corresponden a las matrices de rigidez y de masa del sistema estructural. Mientras que  $\omega$  y  $\phi$  son las frecuencias y los modos de vibración por determinar. La ecuación 4.1 corresponde a un sistema de ecuaciones simultáneas homogéneo, el cual, por definición solo tiene solución no trivial si el determinante de la matriz es igual a cero, es decir:

$$|[K] - \omega^2 [M]| = 0 \quad (4.2)$$

Por lo tanto, para resolver la ecuación 4.2 y poder determinar las características dinámicas del sistema, se requiere conocer la matriz de rigidez y masa del modelo.

Con relación a la estructura de referencia, antes de la realización de la fase experimental, el grupo SHM formuló una fase inicial que consistió en un análisis numérico de la estructura, por lo cual se generaron dos modelos numéricos programados en MatLab®. El primero fue un modelo de 12 grados de libertad el cual fue suministrado para la

## METODOLOGÍA PROPUESTA

identificación, y por otro lado se realizó un modelo de 120 grados de libertad que se realizó con el fin de simular numéricamente la respuesta de la estructura experimental.

Dado que se tiene acceso a dichos modelos numéricos, en el presente trabajo se realizara la adaptación del modelo de 120 grados de libertad para ser implementado como modelo de identificación, el cual se encuentra en el paquete de nombre Datagen, que fue desarrollado por el grupo de salud estructural de la ASCE [44], dicho paquete consiste en 4 guiones (o códigos fuente) y un diagrama de simulación numérica, los cuales fueron programados en el software computacional MatLab®<sup>®</sup>, una descripción de estos se transcribe en la tabla 4.1.

Tabla 4.1: Descripción de los códigos fuente del programa Datagen	
Nombre función MatLab®	Descripción
datagen2.m	Función principal del paquete, el cual ayuda al usuario a introducir de forma interactiva los parámetros de entrada necesarios para el análisis.
calmodel2.m	Sub-función de datagen2.m, la cual formula la matriz de rigidez y de masa la estructura.
calresp2.m	Sub-función de datagen2.m, la cual realiza el cálculo de la respuesta de aceleración en el dominio del tiempo (aceleraciones).
draw3d2.m	Función usada para trazar la estructura después de llamar a datagen2.m.
NJintegrator	Modelo de Simulik para realizar la integración con el algoritmo Nigham-Jennings.

De acuerdo con la tabla 4.1 la función *calmodel2.m* es la que se encarga de la formulación de la matriz de rigidez y de masa del sistema, que de acuerdo con la ecuación 4.2, son necesarias para obtener la propiedades dinámicas del modelo analítico. Por tal motivo, dicha función fue adaptada para ser empleada en el presente trabajo.

Cabe resaltar, que la estructura de referencia, fue modelada por medio de elementos finitos. Dicho método, consiste en la división de la estructura principal en pequeñas subestructuras o elementos finitos unidos por nodos, donde cada nodo aporta un conjunto de grados de libertad al sistema completo. Los tipos de elementos utilizados son de geometría sencilla lo que facilita el manejo matemático del problema. Para obtener la respuesta total del sistema completo basta con ensamblar los elementos con nodos en común, es importante notar que los nodos imponen las condiciones de borde de cada elemento y dos elementos que compartan un mismo nodo compartirán las mismas condiciones de borde impuestas por el nodo. Aunque esta técnica se remonta a varias

## METODOLOGÍA PROPUESTA

décadas atrás, gracias al desarrollo de la computación ha podido ser aplicada a problemas cada vez más complejos que requieren una gran capacidad de procesamiento de datos.

Para la construcción del modelo de elementos finitos, se utilizó la viga como la división elemental (ver figura 4.4), la cual corresponde a un elemento recto de sección transversal uniforme, con un nodo en cada extremo, capaz de resistir fuerza axial, momento a flexión, momento torsor y fuerza cortante. De acuerdo al problema que se abordó, cada elemento de viga en que fue dividida la estructura tiene doce grados de libertad como se muestra en la figura 4.4.

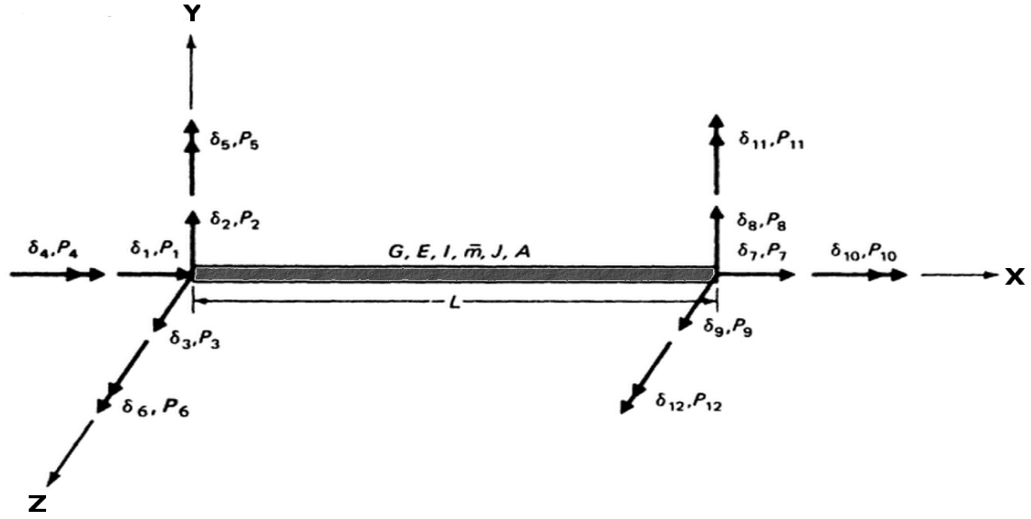


Figura 4.4: Elemento tipo viga de un pórtico espacial [5]

La matriz de rigidez asociada al elemento tipo viga de un pórtico espacial se obtiene por la combinación de la matriz de rigidez para efectos axiales con la matriz para efectos de torsión y la matriz para efectos de flexión. La matriz de rigidez para efectos de flexión aplica dos veces, para la considerar los efectos de flexión en los dos planos principales de la sección de la viga. Combinando estas matrices es forma apropiada, se obtiene la matriz de rigidez para un elemento de viga de pórtico espacial, la cual es:

$$\begin{bmatrix}
 \frac{EA}{L} & 0 & 0 & 0 & 0 & 0 & -\frac{EA}{L} & 0 & 0 & 0 & 0 & 0 \\
 0 & \frac{12EI_z}{L^3} & 0 & 0 & 0 & \frac{6EI_z}{L^2} & 0 & -\frac{12EI_z}{L^3} & 0 & 0 & 0 & \frac{6EI_z}{L^2} \\
 0 & 0 & \frac{12EI_y}{L^3} & 0 & -\frac{6EI_y}{L^2} & 0 & 0 & 0 & -\frac{12EI_y}{L^3} & 0 & -\frac{6EI_y}{L^2} & 0 \\
 0 & 0 & 0 & \frac{GJ}{L} & 0 & 0 & 0 & 0 & 0 & -\frac{GJ}{L} & 0 & 0 \\
 0 & 0 & -\frac{6EI_y}{L^2} & 0 & \frac{4EI_y}{L} & 0 & 0 & 0 & \frac{6EI_y}{L^2} & 0 & \frac{2EI_y}{L} & 0 \\
 0 & \frac{6EI_z}{L^2} & 0 & 0 & 0 & \frac{4EI_z}{L} & 0 & -\frac{6EI_z}{L^2} & 0 & 0 & 0 & \frac{2EI_z}{L} \\
 -\frac{EA}{L} & 0 & 0 & 0 & 0 & 0 & \frac{EA}{L} & 0 & 0 & 0 & 0 & 0 \\
 0 & -\frac{12EI_z}{L^3} & 0 & 0 & 0 & -\frac{6EI_z}{L^2} & 0 & \frac{12EI_z}{L^3} & 0 & 0 & 0 & -\frac{6EI_z}{L^2} \\
 0 & 0 & -\frac{12EI_y}{L^3} & 0 & \frac{6EI_y}{L^2} & 0 & 0 & 0 & \frac{12EI_y}{L^3} & 0 & \frac{6EI_y}{L^2} & 0 \\
 0 & 0 & 0 & -\frac{GJ}{L} & 0 & 0 & 0 & 0 & 0 & \frac{GJ}{L} & 0 & 0 \\
 0 & 0 & -\frac{6EI_y}{L^2} & 0 & \frac{2EI_y}{L} & 0 & 0 & 0 & \frac{6EI_y}{L^2} & 0 & \frac{4EI_y}{L} & 0 \\
 0 & -\frac{6EI_z}{L^2} & 0 & 0 & 0 & \frac{2EI_z}{L} & 0 & -\frac{6EI_z}{L^2} & 0 & 0 & 0 & \frac{4EI_z}{L}
 \end{bmatrix} \quad (4.3)$$

## METODOLOGÍA PROPUESTA

Donde  $I_y$  y  $I_z$  son, respectivamente, los momentos de inercia de la sección de la viga con respecto a los planos principales en la direcciones Y y Z.  $L$ ,  $A$  y  $J$  son respectivamente, la longitud, el área de la sección transversal y la constante torsional del elemento de la viga.

Con respecto a la masa, se consideró la formulación de la matriz de masa concentrada, la cual para un elemento uniforme de un pórtico espacial, es simplemente una matriz diagonal, en la cual los coeficientes correspondientes a los desplazamientos lineales y de torsión son iguales a la mitad del total de la inercia del segmento de viga, mientras que los coeficientes correspondientes a las rotaciones de flexión se suponen iguales a cero. Así, la matriz de masa concentrada para una viga de masa uniforme distribuida  $\bar{m}$  por unidad de longitud puede escribirse como:

$$[M] = \frac{\bar{m}L}{2} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{I_0}{A} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{I_0}{A} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (4.4)$$

Donde  $I_0$  es el momento polar de inercia y  $A$  en el área de la sección transversal de la viga.

Dado que en el modelo numérico se realizará una reducción del tamaño, se deben identificar aquellos grados de libertad que se desean condensar o reducir como grados de libertad dependientes o secundarios y expresarlos en función de los grados de libertad seleccionados como independientes o primarios. La relación entre los grados de libertad primarios y secundarios se puede establecer mediante técnicas de condensación estática.

Los grados de libertad que serán reducidos o condensados, son designados por las primeras  $s$  coordenadas y los grados de libertad que quedan son designados por las últimas  $p$  coordenadas. Con este arreglo de coordenadas, la ecuación de rigidez de la estructura, puede ser escrita como:

$$\begin{bmatrix} [K_{ss}] & [K_{sp}] \\ [K_{ps}] & [K_{pp}] \end{bmatrix} \begin{Bmatrix} \{y_s\} \\ \{y_p\} \end{Bmatrix} = \begin{Bmatrix} \{0\} \\ \{F_p\} \end{Bmatrix} \quad (4.5)$$

Donde  $\{y_s\}$ , es el vector de desplazamiento que corresponde a los grados  $s$  de libertad por reducir, e  $\{y_p\}$  es el vector correspondiente a los  $p$  grados de libertad restantes, que fueron seleccionados como independientes. En la ecuación 4.5 se ha supuesto que no existen fuerzas externas aplicadas a los  $s$  grados de libertad secundarios. Realizando las operaciones indicadas en la ecuación 4.5 se obtiene:

## METODOLOGÍA PROPUESTA

$$[K_{ss}] \{y_s\} + [K_{sp}] \{y_p\} = \{0\} \quad (4.6)$$

$$[K_{ps}] \{y_s\} + [K_{pp}] \{y_p\} = \{F_p\} \quad (4.7)$$

La ecuación 4.6 es equivalente a tener

$$\{y_s\} = [\overline{T}] \{y_p\} \quad (4.8)$$

Donde  $[\overline{T}]$  es la matriz de transformación dada por:

$$[\overline{T}] = -[K_{ss}]^{-1} [K_{sp}] \quad (4.9)$$

Empleando la ecuación 4.8 y la ecuación 4.9, se obtiene la ecuación reducida de rigidez, que relaciona las fuerzas con los desplazamientos, para las coordenadas primarias, es decir:

$$[\overline{K}] \{y_p\} = [F_p] \quad (4.10)$$

Adicionalmente, la ecuación 4.8, que expresa la relación estática entre las coordenada secundarias  $\{y_s\}$  y las coordenadas primarias  $\{y_p\}$ , puede ser escrita, usando la identidad  $\{y_p\} = [I] \{y_p\}$ , como:

$$\begin{bmatrix} \{y_s\} \\ \{y_p\} \end{bmatrix} = \begin{bmatrix} [\overline{T}] \\ [I] \end{bmatrix} \{y_p\} \quad (4.11)$$

O

$$\{y\} = [T] \{y_p\} \quad (4.12)$$

Aplicando la ecuación 4.12 en la ecuación 4.5 y premultiplicando por  $[T]^T$  se demuestra que la matriz reducida de rigidez  $[\overline{K}]$  puede expresarse como la transformación de la matriz de rigidez del sistema  $[K]$ :

$$[\overline{K}] = [T]^T [K] [T] \quad (4.13)$$

Por otro lado, un aspecto importante para resaltar dentro de las estructuras de acero, es que las uniones atornilladas son populares, debido a la economía y la simplicidad en la fabricación. A pesar de esto, el procedimiento convencional para el análisis estructural asume que las conexiones viga-columna son completamente rígidas. Sin embargo, en realidad, la mayoría de las conexiones de las estructuras son semi-rígidas. El comportamiento de los nodos hace una importante contribución al comportamiento de toda la estructura. Entre los diversos modelos de deformación para las conexiones semi-rígidas, el más utilizado es la deformación de rotación causada por un momento a flexión. La descripción básica del comportamiento a flexión es la relación momento-rotación, el cual puede ser representado como una rigidez a flexión del nodo.

## METODOLOGÍA PROPUESTA

La conexión semi-rígida se modela como resortes rotacionales e extremos de una viga, como se muestra en la figura 4.5. La rigidez rotacional ( $k_{r1}$ ) del resorte en el extremo izquierdo de la viga se define como la relación entre el momento aplicado ( $M_1$ ) y el ángulo de deslizamiento del resorte ( $\phi_1$ ).  $\theta_1^*$  es la rotación del resorte en el extremo izquierdo (nodo externo) con respecto al eje de la viga deformada, y  $\bar{\theta}_1$  es la rotación del resorte del extremo derecho (nodo interno) con respecto al eje de la viga sin deformar.

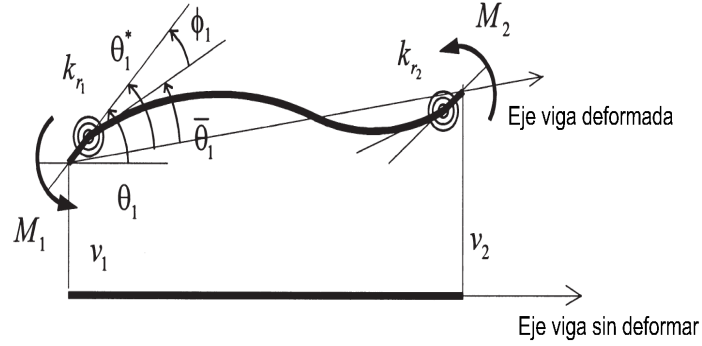


Figura 4.5: Deformada de una viga con conexiones semi-rígidas [6]

La matriz de rigidez de un elemento de viga con conexiones semi-rígidas en los extremos puede ser obtenida como [6, 71].

$$\begin{bmatrix} \frac{EA}{L} & 0 & 0 & 0 & 0 & 0 & -\frac{EA}{L} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{12EI_z}{L^3} \frac{f_1}{f_7} & 0 & 0 & 0 & \frac{6EI_z}{L^2} \frac{f_2}{f_7} & 0 & -\frac{12EI_z}{L^3} \frac{f_1}{f_7} & 0 & 0 & 0 & \frac{6EI_z}{L^2} \frac{f_6}{f_7} \\ 0 & 0 & \frac{12EI_y}{L^3} \frac{f_1}{f_7} & 0 & -\frac{6EI_y}{L^2} \frac{f_2}{f_7} & 0 & 0 & 0 & -\frac{12EI_y}{L^3} \frac{f_1}{f_7} & 0 & -\frac{6EI_y}{L^2} \frac{f_6}{f_7} & 0 \\ 0 & 0 & 0 & \frac{GJ}{L} & 0 & 0 & 0 & 0 & 0 & -\frac{GJ}{L} & 0 & 0 \\ 0 & 0 & -\frac{6EI_y}{L^2} \frac{f_2}{f_7} & 0 & \frac{4EI_y}{L} \frac{f_3}{f_7} & 0 & 0 & 0 & \frac{6EI_y}{L^2} \frac{f_2}{f_7} & 0 & \frac{2EI_y}{L} \frac{f_5}{f_7} & 0 \\ 0 & \frac{6EI_z}{L^2} \frac{f_2}{f_7} & 0 & 0 & 0 & \frac{4EI_z}{L} \frac{f_3}{f_7} & 0 & -\frac{6EI_z}{L^2} \frac{f_2}{f_7} & 0 & 0 & 0 & \frac{2EI_z}{L} \frac{f_5}{f_7} \\ -\frac{EA}{L} & 0 & 0 & 0 & 0 & 0 & \frac{EA}{L} & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{12EI_z}{L^3} \frac{f_1}{f_7} & 0 & 0 & 0 & -\frac{6EI_z}{L^2} \frac{f_2}{f_7} & 0 & \frac{12EI_z}{L^3} \frac{f_1}{f_7} & 0 & 0 & 0 & -\frac{6EI_z}{L^2} \frac{f_6}{f_7} \\ 0 & 0 & -\frac{12EI_y}{L^3} \frac{f_1}{f_7} & 0 & \frac{6EI_y}{L^2} \frac{f_2}{f_7} & 0 & 0 & 0 & \frac{12EI_y}{L^3} \frac{f_1}{f_7} & 0 & \frac{6EI_y}{L^2} \frac{f_6}{f_7} & 0 \\ 0 & 0 & 0 & -\frac{GJ}{L} & 0 & 0 & 0 & 0 & 0 & \frac{GJ}{L} & 0 & 0 \\ 0 & 0 & -\frac{6EI_y}{L^2} \frac{f_6}{f_7} & 0 & \frac{2EI_y}{L} \frac{f_5}{f_7} & 0 & 0 & 0 & \frac{6EI_y}{L^2} \frac{f_6}{f_7} & 0 & \frac{4EI_y}{L} \frac{f_4}{f_7} & 0 \\ 0 & -\frac{6EI_z}{L^2} \frac{f_6}{f_7} & 0 & 0 & 0 & \frac{2EI_z}{L} \frac{f_5}{f_7} & 0 & -\frac{6EI_z}{L^2} \frac{f_6}{f_7} & 0 & 0 & 0 & \frac{4EI_z}{L} \frac{f_4}{f_7} \end{bmatrix}$$

Donde los parámetros  $f_i$  son definidos usando los factores de fijación del nodo ( $\gamma_1$  y  $\gamma_2$ ) como [6]:

$$\begin{aligned} f_1 &= \gamma_1 + \gamma_2 + \gamma_1\gamma_2 \\ f_2 &= \gamma_1(2 + \gamma_2) \\ f_3 &= 3\gamma_1 \\ f_4 &= 3\gamma_2 \\ f_5 &= 3\gamma_1\gamma_2 \\ f_6 &= \gamma_2(2 + \gamma_1) \\ f_7 &= 4 - \gamma_1\gamma_2 \end{aligned} \tag{4.14}$$

## METODOLOGÍA PROPUESTA

Además, los factores de fijación  $\gamma_1$  y  $\gamma_2$  son definidos usando la rigidez rotacional de los extremos de los elementos como:

$$\gamma_1 = \frac{1}{1 + \frac{3EI}{Lk_{r1}}}, \gamma_2 = \frac{1}{1 + \frac{3EI}{Lk_{r2}}} \quad \text{donde} \quad 0 \leq \gamma_1 \text{ y } \gamma_2 \leq 1,0 \quad (4.15)$$

Generalmente, las características estáticas y dinámicas de las estructuras con conexiones semi-rígidas se examinan en conjunto con la fijación de la articulación y los resultados se comparan con aquellos basados en el factor de multiplicación  $\mu$  [6], que es una rigidez en las articulaciones, la cual esta definida como:

$$\mu = \frac{k_r}{EI/L} \quad (4.16)$$

Donde  $k_r$  es la rigidez de rotación de la conexión viga-columna, y termino  $EI/L$  es la rigidez a flexión de una viga. La relación entre el factor de multiplicación y el factor de fijación se puede obtener fácilmente como:

$$\gamma = \frac{1}{1 + \frac{3EI}{Lk_{r1}}} = \frac{\mu}{\mu + 3} \quad (4.17)$$

El Eurocodigo [72], tiene definido los rangos de los valores de los factores de fijación y de multiplicación para el caso de la consideración de una conexión semi-rígida, tal información se muestra en la tabla 4.2

Tabla 4.2: Clasificación de las conexiones [6]

	Conexión articulada	Conexión semi-rígida	Conexión rígida
Factor de multiplicación	$0 \leq \mu \leq 0,5$	$0,5 \leq \mu \leq 25$	$25 < \mu < \infty$
Factor de fijación	$0 \leq \gamma \leq 0,143$	$0,143 \leq \gamma \leq 0,891$	$0,891 < \gamma \leq 1$

### 4.3. FORMULACIÓN DEL PROBLEMA DE OPTIMIZACIÓN

Como es de conocimiento general, las características modales de una estructura estan descrita por la ecuación del valor propio como,[17, 73]:

$$(K - \lambda_i M) \phi_i = 0 \quad (4.18)$$

Donde  $K (n \times n)$  y  $M (n \times n)$  corresponden a las matrices de rigidez y de masa de la estructura respectivamente,  $\lambda_i$  es el  $i$ -ésimo valor propio, y  $\phi_i$  corresponde a la  $i$ -ésima forma modal (vector propio) de la estructura sin daño con  $n$  grados de libertad.



## METODOLOGÍA PROPUESTA

En la mayoría de los casos observados en la literatura, el daño presenta cambios en la matriz de rigidez en la estructura de una magnitud  $\delta K$ . Además, si se considera que la presencia de daño no produce un cambio en la matriz de masa, el cambio en la matriz de rigidez produce un cambio  $\delta\lambda_i$  en el  $i$ -ésimo valor propio, y un cambio  $\delta\phi_i$  en la  $i$ -ésima forma modal [14, 17, 31, 73]. De esta manera la ecuación del valor propio de la estructura con daño puede ser escrita como [73]:

$$[K + \delta K - (\lambda_i + \delta\lambda_i) M] (\phi_i + \delta\phi_i) = 0 \quad (4.19)$$

Pre-multiplicando la ecuación 4.19 por  $\phi_i^T$ , expandiendo, reorganizando y aplicando las condiciones de ortogonalidad para las formas modales, se obtiene:

$$\delta\lambda_i = \frac{(\phi_i^T \delta K \phi_i + \phi_i^T \delta K \delta\phi_i)}{(1 + \phi_i^T M \delta\phi_i)} \quad (4.20)$$

Cabe resaltar que en la mayoría de las investigaciones, todos los términos asociados con el cambio de las formas modales de la ecuación 4.20 han sido omitidos, debido a que las mediciones experimentales de las formas modales requieren gran cantidad de sensores [14, 17]. Por lo tanto el cambio del  $i$ -ésimo valor propio excluyendo los cambios en las formas puede escribirse como:

$$\delta\lambda_i = \phi_i^T \delta K \phi_i \quad (4.21)$$

Al respecto es conveniente destacar que, la omisión del termino  $\delta\phi$  (cambio de las formas modales) tiene unos efectos en la precisión de la ecuación 4.21 en el proceso de detección de daños, sin embargo, este enfoque ha sido implementado en algunas de las investigaciones relacionadas con este tipo de problemas [73]. De esta manera, en el presente trabajo se realizará la determinación del cambio de frecuencias a partir de la variación de algunos parámetros relacionados con la rigidez del sistema estructural.

Como se indicó en la ecuación 4.21, la variación en las frecuencias naturales se puede determinar a partir de cambios producidos en la rigidez del sistema, de esta forma se puede realizar la diferencia entre los valores medidos experimentalmente con los calculados por el modelo numérico, para así determinar la desviación de las  $n$  frecuencias medidas con respecto a la numéricas, para tal fin, se seleccionaron las siguientes funciones [74, 75, 76], para poner a prueba la capacidad de cada una para desarrollar el problema abordado.

$$\Delta\lambda_1 = \sum_{i=1}^N \left| 1 - \frac{\omega_e^i}{\omega_n^i} \right| \quad (4.22)$$

$$\Delta\lambda_2 = \sqrt{\sum_{i=1}^N (\omega_e^i - \omega_n^i)^2} \quad (4.23)$$

$$\Delta\lambda_3 = \sum_{i=1}^N \left( \frac{\omega_e^i - \omega_n^i}{\omega_e^i} \right) \quad (4.24)$$

## METODOLOGÍA PROPUESTA

Donde  $\omega_e^i$  y  $\omega_n^i$  hace referencia a la *i-ésima* frecuencia experimental y numerica respectivamente. Sin embargo, para la determinación de los parámetros seleccionados para la identificación es necesario realizar la minimización de la función seleccionada, es decir:

$$Min(\Delta\lambda) \quad (4.25)$$

Este problema de optimización es resuelto en MatLab® mediante el uso del algoritmo SL-PSO, el cual está descrito completamente en el capítulo 3.

### 4.4. AJUSTE DEL MODELO NUMÉRICO EMPLEANDO EL SL-PSO

Una vez realizado el modelo numérico del comportamiento de la estructura, es de vital importancia determinar la validez de dicho modelo, comprobando si los resultados obtenidos representan el comportamiento real de la estructura, el cual se calcula a partir de ensayos experimentales. Para esto, el medio de validación del modelo de elementos finitos resulta de la comparación de un número reducido de propiedades dinámicas predichas analíticamente, con sus correspondientes valores obtenidos mediante la experimentación.

En la figura 4.6 se muestra de forma general el procedimiento realizado para el ajuste del modelo numérico.

En las secciones anteriores se explicó la forma de abordar un análisis modal experimental de una estructura, el cual se realiza por medio de las mediciones realizadas al sistema, y de acuerdo al procedimiento seleccionado se determinan las características dinámicas. Cabe resaltar que para la estructura de referencia se han realizado numerosos estudios en este campo, de manera que, en este estudio los parámetros dinámicos fueron recolectados de la literatura tal como se muestra en el capítulo 2.

Por otro lado, una vez formulado el modelo numérico del sistema, se puede realizar un análisis modal analítico para extraer los parámetros dinámicos, y posteriormente realizar una correlación entre los datos experimentales y los del modelo numérico, esta correlación se realiza de acuerdo a la formulación del problema de optimización que se describió en la sección anterior, por lo tanto debe minimizarse según el enfoque realizado en este trabajo. Una vez determinada la correlación inicial, mediante el algoritmo SL-PSO se hace un proceso iterativo para determinar el mejor ajuste del modelo numérico con la información experimental.

Para este ajuste inicialmente se especifican las variables de interés. En este sentido, la mayoría de las investigaciones suelen variar el módulo elasticidad como la única variable que interviene en el comportamiento del modelo numérico. Sin embargo existen otras variables que pueden influir significativamente en su comportamiento, como la longitud efectiva de los elementos y la rigidez de los nodos.

## METODOLOGÍA PROPUESTA

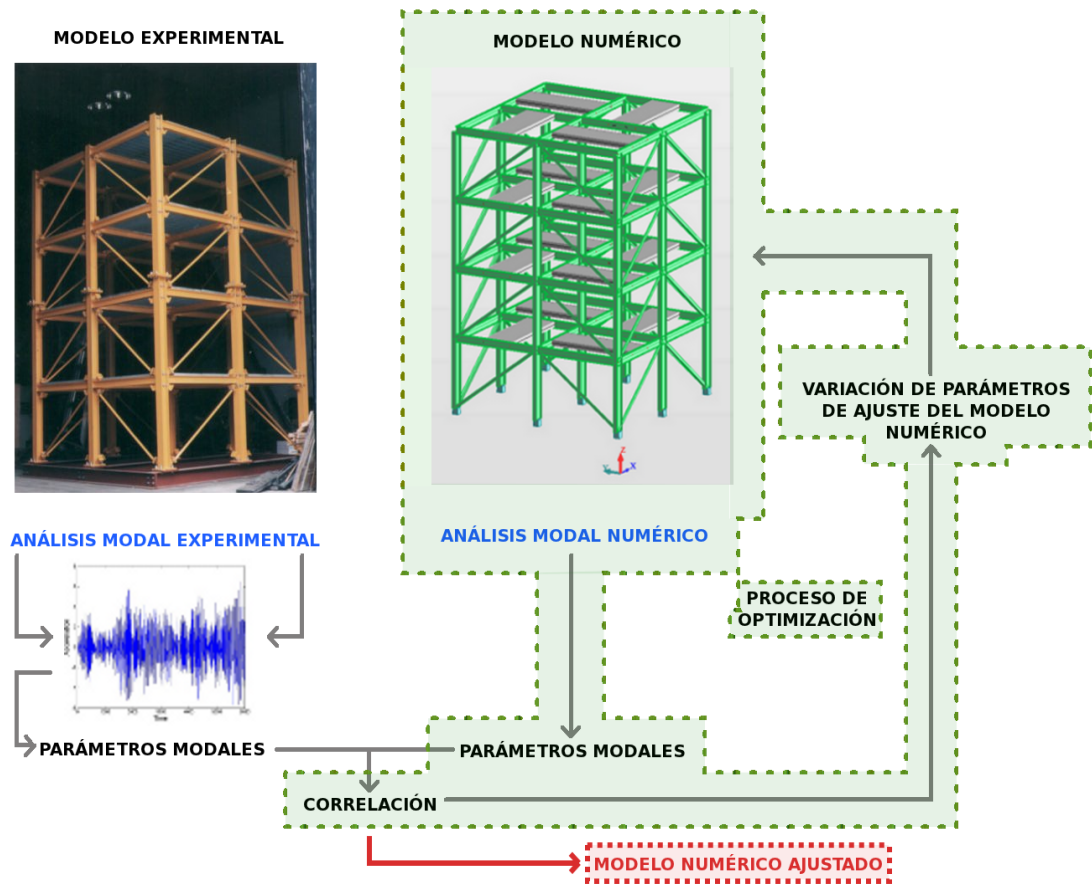


Figura 4.6: Esquema general del ajuste del modelo numérico

De acuerdo con lo anterior, en el presente trabajo se realizarán tres ajustes diferentes del modelo numérico, el primero será un ajuste basado en la sola la variación del módulo de elasticidad partiendo de la suposición de que la estructura posee nodos rígidos. En el segundo ajuste, se considerará, la estructura con nodos semi-rígidos e igualmente solo el módulo de elasticidad como variable, y finalmente se realizará un ajuste tomando más variables, así, además del módulo de elasticidad, se considerarán las variables correspondientes a la longitud efectiva de los elementos y la rigidez de los nodos.

#### 4.5. IDENTIFICACIÓN DE LOS CASOS DE DAÑO DISPONIBLE A PARTIR DEL MODELO AJUSTADO, LA INFORMACIÓN EXPERIMENTAL Y EL SL-PSO

Como se mencionó en el capítulo 2, en la formulación del problema de referencia fueron simulados algunos casos de daño, cuyo objetivo es que sean empelados en diversas técnicas para así ver la capacidad de dicha técnica para solucionar correctamente este tipo de problemas. Así, en este trabajo se realizará la implementación del SL-PSO y el índice de daño para la respectiva identificación. Los casos de daño estudiados se encuentran ilustrados desde la figura 4.7 a la figura 4.12, en donde se presentan de forma gráfica cada uno de los daños que se estudiarán. Para cada configuración de daños se presentan dos figuras, una de ellas corresponde al modelo tridimensional y en rojo se muestran los elementos que fueron removidos, y la segunda gráfica corresponde al plano donde se removió dicho elemento.

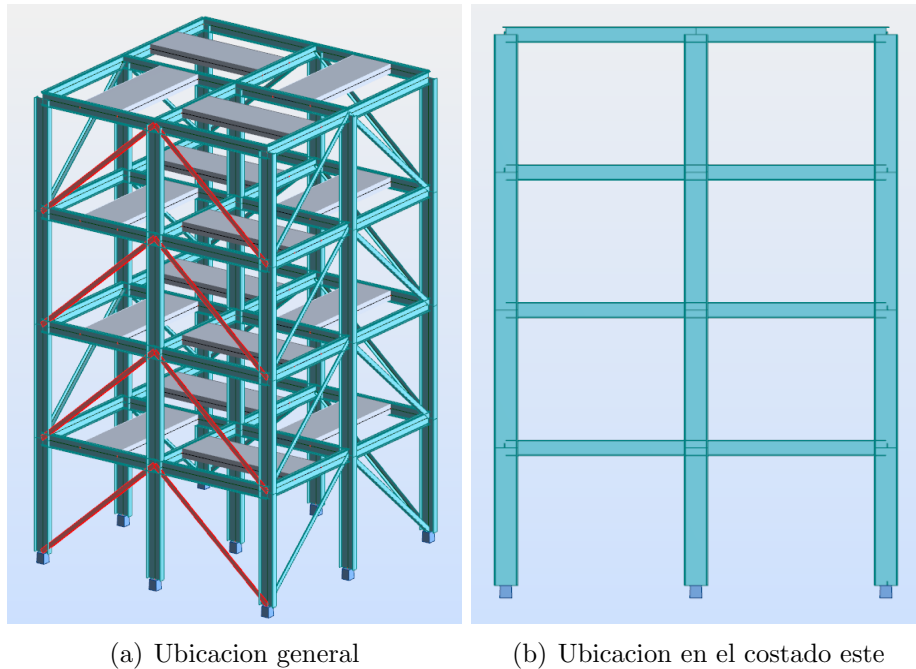


Figura 4.7: Configuración II

La identificación realizada en el presente trabajo se basa en el ajuste de modelos, es decir, que para la identificación se realizará un ajuste, con la diferencia que en este caso, solo se consideraran los elementos tipo rigidizador como variables de búsqueda y esta será de forma discreta, es decir, que podrá tomar el valor de 1 para considerar que el elemento se encuentra en la estructura y el valor de cero para representar la ausencia

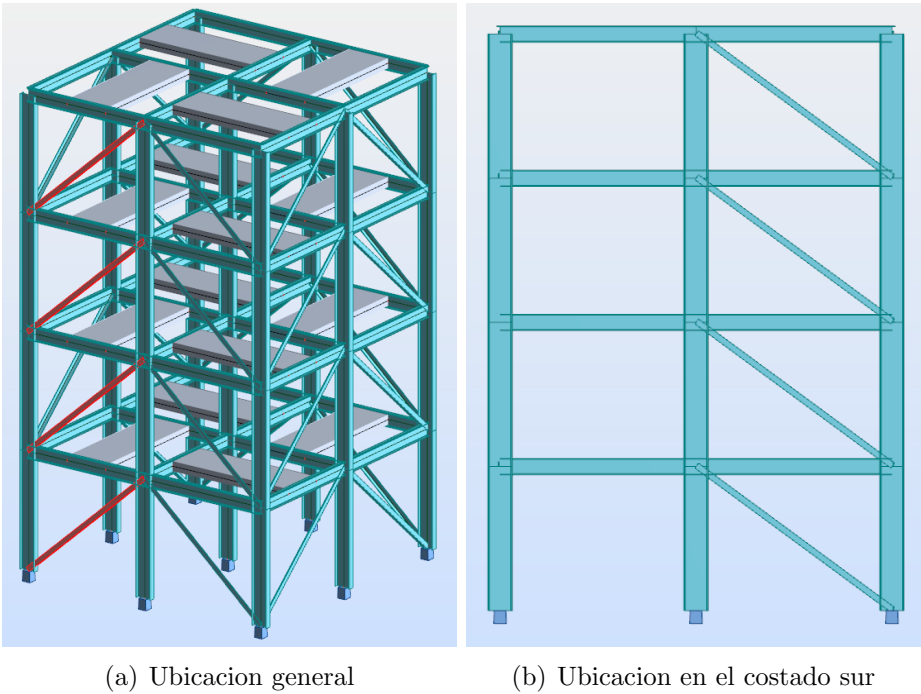


Figura 4.8: Configuración III

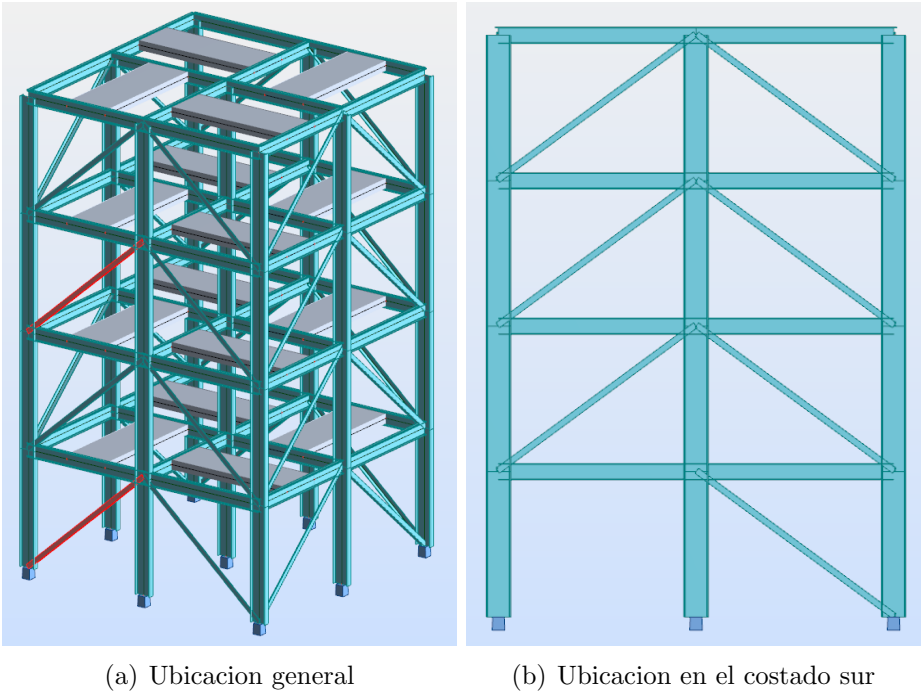


Figura 4.9: Configuración IV

METODOLOGÍA PROPUESTA

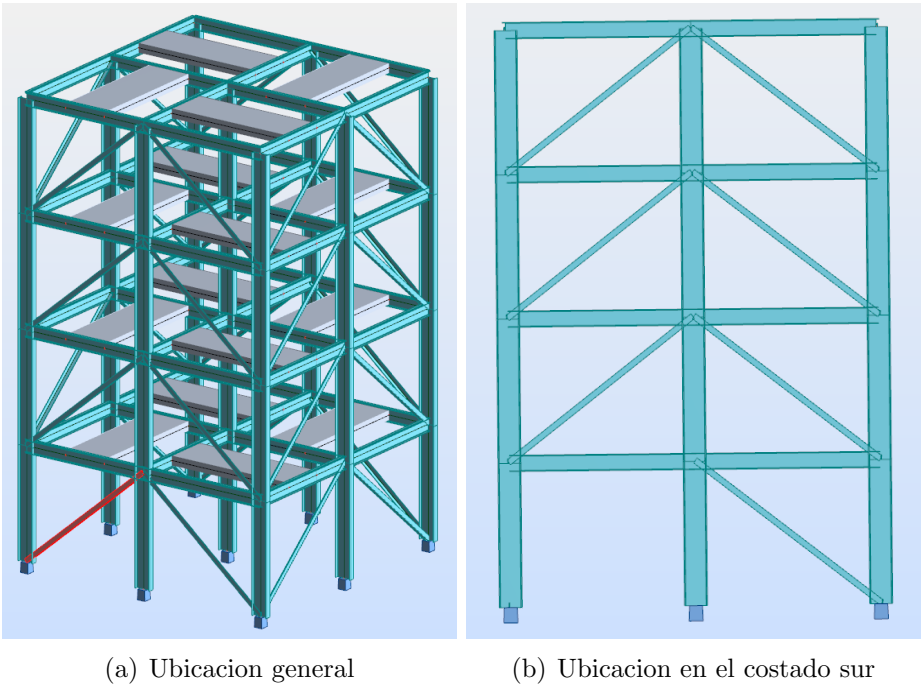


Figura 4.10: Configuración V

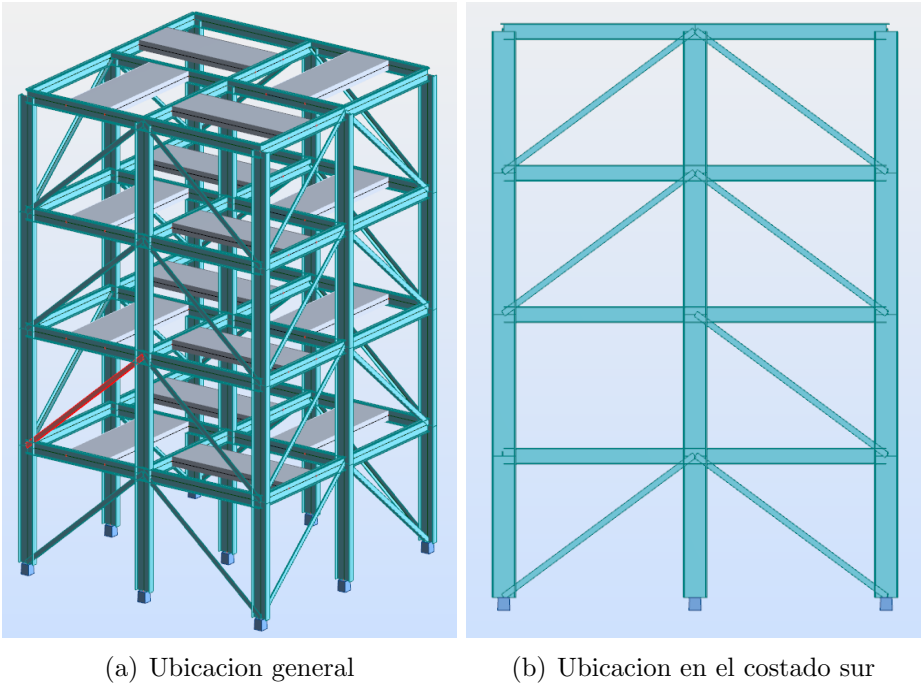


Figura 4.11: Configuración VI

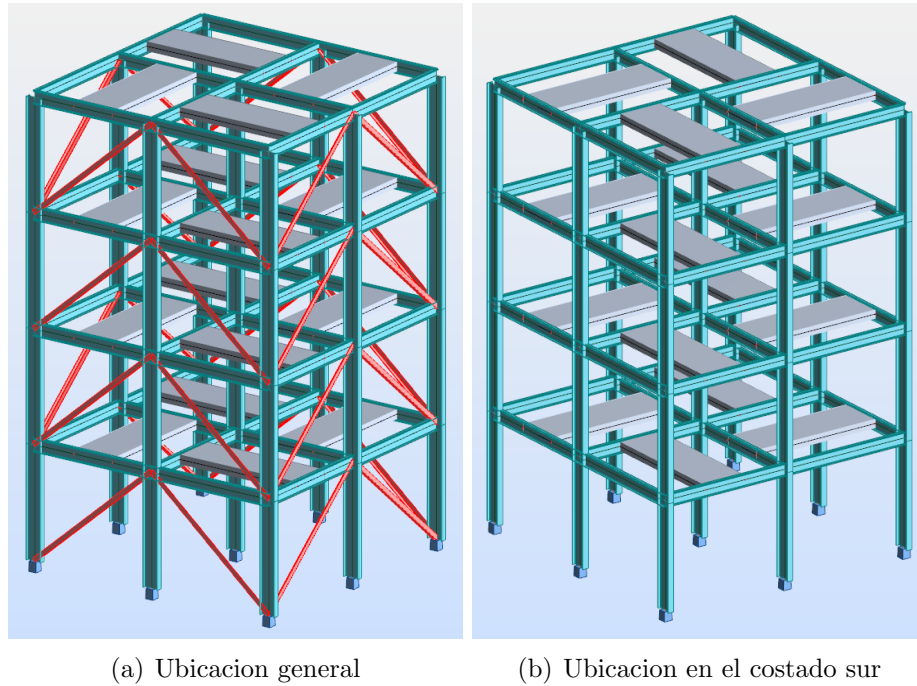


Figura 4.12: Configuración VII

de dicho elemento en el sistema. Al realizar este enfoque se seleccionó el área de los elementos como variables a optimizar, el área fue seleccionada de acuerdo a la tipología de daño.

## 4.6. MÉTODO DE ÍNDICE DE DAÑO

El método del índice de daño de piso “*Story Damage Index*” (SDI) fue propuesto por Wang et al. [13] para la evaluación de daños en edificios. Este método está expresado en función de las frecuencias, formas modales y la masa presente en cada piso. Sin embargo, los autores presentan un índice aproximado denominado “*Approximate Story Damage Index*” (ASDI) el cual no depende de la información de la masa.

A continuación se presenta el fundamento teórico del índice de daño, el cual está completamente descrito en [13]

### 4.6.1. TEORÍA DE LA IDENTIFICACIÓN DE DAÑOS USANDO EL MÉTODO DE ÍNDICE DE DAÑO

Como se mencionó anteriormente, este método utiliza un indicador basado en las frecuencias, modos de vibración y la masa de cada piso para la respectiva identificación de daños. Para la formulación de este método, los autores consideraron un edificio de cortante de  $N$ , donde el  $l$ -ésimo piso tiene una masa  $m_l$ , rigidez  $k_l$  y amortiguamiento



## METODOLOGÍA PROPUESTA

$c_l$ , y la ecuación de movimiento lineal de la estructura bajo una aceleración  $\ddot{u}_g(t)$  en la base puede ser escrita como [13]:

$$M\ddot{x}(t) + C\dot{x}(t) + Kx(t) = M\ddot{u}_g(t) \quad (4.26)$$

Donde  $M$ ,  $C$  y  $K$  son las matrices  $N \times N$  de masa amortiguamiento y rigidez respectivamente, las cuales de acuerdo con [13], están definidas como:

$$\begin{bmatrix} m_1 & 0 & \cdots & \cdots & 0 \\ 0 & m_2 & & & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & & & m_l & 0 \\ \vdots & & & & \ddots & \vdots \\ 0 & & & & & m_{N-1} & 0 \\ 0 & \cdots & \cdots & 0 & m_N \end{bmatrix} \quad (4.27)$$

$$\begin{bmatrix} c_1 + c_2 & -c_2 & 0 & \cdots & 0 \\ -c_2 & c_2 + c_3 & -c_3 & & \\ 0 & \ddots & \ddots & \ddots & \vdots \\ & & -c_l & c_l + c_{l+1} & -c_{l-1} \\ \vdots & & & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & & & -c_{N-1} & c_{N-1} + c_N & -c_N \\ & & & & \cdots & -c_N & c_N \end{bmatrix} \quad (4.28)$$

$$\begin{bmatrix} k_1 + k_2 & -k_2 & 0 & \cdots & 0 \\ -k_2 & k_2 + k_3 & -k_3 & & \\ 0 & \ddots & \ddots & \ddots & \vdots \\ & & -k_l & k_l + k_{l+1} & -k_{l-1} \\ \vdots & & & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & & & -k_{N-1} & k_{N-1} + k_N & -k_N \\ & & & & \cdots & -k_N & k_N \end{bmatrix} \quad (4.29)$$

De la ecuación 4.26,  $x(t)$  representa el vector  $N \times 1$  de los desplazamientos de cada piso con respecto al movimiento en la base en el tiempo  $t$ . Asumiendo que  $\lambda_j$  y  $\phi_j$  corresponden al  $j$ -ésimo valor y vector propio  $N \times 1$  diferente de cero de la estructura, la ecuación característica del sistema puede ser escrita como:

$$(\lambda_j^2 M + \lambda_j C + K) \phi_j = 0 \quad (4.30)$$

De acuerdo con [13], la ecuación 4.30 puede ser reescrita en el espacio de estados de la forma:



## METODOLOGÍA PROPUESTA

$$\begin{bmatrix} -M^{-1}C & -M^{-1}K \\ I & 0 \end{bmatrix} \begin{Bmatrix} \lambda_j \phi_j \\ \phi_j \end{Bmatrix} = \lambda_j \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \begin{Bmatrix} \lambda_j \phi_j \\ \phi_j \end{Bmatrix} \quad (4.31)$$

Donde  $j$ -ésimo valor propio puede ser representado por un par de complejos conjugados en términos de la  $j$ -ésima frecuencia modal  $\omega_j$  y la razón de amortiguamiento  $\xi_j$  como:

$$(\lambda_j)_{1,2} = \omega_j C_{m1j}; \quad C_{m1j} = \xi_j \pm i\sqrt{1 - \xi_j^2} \quad (4.32)$$

Con su valor elevado al cuadrado

$$(\lambda_j^2)_{1,2} = -\omega_j^2 C_{m2j}; \quad C_{m2j} = (1 - 2\xi_j^2) \pm i2\xi_j\sqrt{1 - \xi_j^2} \quad (4.33)$$

Donde  $i = \sqrt{-1}$ ,  $C_{m1j}$ ,  $C_{m2j}$  son funciones de la razón de amortiguamiento  $\xi_j$  de la  $j$ -ésima forma modal. Cuando una estructura no tiene amortiguamiento,  $C_{m1j}$  y  $C_{m2j}$  se reducen a  $i$  y  $1$  respectivamente,  $\lambda_j^2$  se convierte en  $-\omega_j^2$  y la ecuación 4.30 se reduce a:

$$\begin{bmatrix} a_{1,1} & a_{1,2} & 0 & \cdots & 0 \\ a_{2,1} & a_{2,2} & a_{2,3} & & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots \\ & & a_{l,l-1} & a_{l,l} & a_{l,l+1} \\ \vdots & & & \ddots & \ddots & \ddots & \vdots \\ & & & a_{N-1,N-2} & a_{N-1,N-1} & a_{N-1,N} \\ 0 & \cdots & & \cdots & a_{N,N-1} & a_{N,N} \end{bmatrix} \begin{Bmatrix} \phi_{1j} \\ \phi_{2j} \\ \vdots \\ \phi_{lj} \\ \vdots \\ \phi_{(N-1)j} \\ \phi_{Nj} \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \\ 0 \end{Bmatrix} \quad (4.34)$$

Donde,

$$a_{l,l} = \begin{cases} (k_l + k_{l+1}) - \omega_j^2 m_l & \text{for } l = 1, 2, \dots, N-1 \\ k_N - \omega_j^2 m_N & \text{for } l = N \end{cases} \quad (4.35)$$

$$a_{l,l+1} = -k_{l+1} \text{ for } l = 1, 2, \dots, N-1 \quad (4.36)$$

$$a_{l,l-1} = -k_l \text{ for } l = 2, 3, \dots, N \quad (4.37)$$

Según [13], resolviendo la ecuación 4.34 de la ultima a la primera columna, una expresión general para la rigidez del  $l$ -ésimo piso se puede expresar como:

$$k_l = \omega_j^2 \sum_{n=l}^N \frac{m_n \phi_{nj}}{\Delta \phi_{lj}} \quad (4.38)$$

Donde,

## METODOLOGÍA PROPUESTA

$$\Delta\phi_{lj} = \begin{cases} \phi_{lj} - \phi_{(l-1)j} & \text{for } l = 2, 3, \dots, N \\ \phi_{lj} & \text{for } l = 1 \end{cases} \quad (4.39)$$

Como se muestra en [13], para interpretar la ecuación 4.38 desde el punto de vista físico, se puede imaginar un edificio vibrando en el  $j$ -ésimo modo. En esta condición, la frecuencia de vibración de la estructura es justo la  $j$ -ésima frecuencia modal  $\omega_j$ , y el desplazamiento en el  $l$ -ésimo piso es la  $l$ -ésima fila de la  $j$ -ésima forma modal  $\phi_{lj}$ . Por lo tanto, la aceleración del  $l$ -ésimo piso es igual a  $\omega_j^2 \phi_{lj}$ . Además, la fuerza inercial del  $l$ -ésimo piso debería ser la masa multiplicada por la aceleración,  $m_l \omega_j^2 \phi_{lj}$ . La resultante de la fuerza aplicada en el  $j$ -ésimo piso, es la sumatoria de las fuerzas inerciales arriba del piso  $\sum_{n=l}^N \omega_j^2 m_n \phi_{nj}$ , y la  $l$ -ésima deriva de piso es igual al desplazamiento relativo entre el  $l$ -ésimo y el  $l$ -ésimo-1 piso,  $\phi_{lj} - \phi_{(l-1)j}$ , entonces es evidente que la rigidez del  $l$ -ésimo piso, puede ser obtenida dividiendo la fuerza resultante por la deriva del piso y la ecuación 4.38 puede ser resuelta fácilmente.

### Rigidez de piso considerando amortiguamiento

La ecuación 4.30 es la forma general de la relación entre los parámetros físicos y modales de un sistema estructural amortiguado sin importar si su amortiguamiento es proporcional o no proporcional. Siguiendo el procedimiento descrito en [13], y sustituyendo la ecuación 4.27, 4.28 y 4.29, en 4.30, los coeficientes de la ecuación 4.34 quedan de la siguiente forma:

$$a_{l,l} = \begin{cases} \lambda_j^2 m_l + \lambda_j c_l + \lambda_j c_{l+1} + k_l + k_{l+1} & \text{for } l = 1, 2, \dots, N-1 \\ \lambda_j^2 m_l + \lambda_j c_l + k_l & \text{for } l = N \end{cases} \quad (4.40)$$

$$a_{l,l+1} = -\lambda_j c_{l+1} - k_{l+1} \text{ for } l = 1, 2, \dots, N-1 \quad (4.41)$$

$$a_{l,l-1} = -\lambda_j c_{l+1} - k_{l+1} \text{ for } l = 2, 3, \dots, N \quad (4.42)$$

La ultima fila de la ecuación 4.34 puede ser escrita como

$$a_{N,N-1} \phi_{(N-1)j} + a_{N,N} \phi_{Nj} = 0 \quad (4.43)$$

Sustituyendo las ecuaciones 4.40, 4.41 y 4.42 en la ecuación 4.43 obtenemos,

$$(-c_N \lambda_j - k_N) \phi_{(N-1)j} + (m_N \lambda_j^2 + c_N \lambda_j + k_N) \phi_{Nj} = 0 \quad (4.44)$$

resolviendo la ecuación 4.34 de la ultima a la primera fila, obtenemos la rigidez del  $l$ th piso como

$$k_l = \lambda_j^2 \sum_{n=l}^N \frac{m_n \phi_{nj}}{\Delta \phi_{lj}} - c_l \lambda_j \quad (4.45)$$

## METODOLOGÍA PROPUESTA

ademas  $\lambda_j$  y  $\lambda_j^2$  pueden ser representadas con las ecuaciones 4.32 y 4.33, la ecuación se convierte en

$$k_l = -\omega_j^2 \sum_{n=l}^N \frac{m_n \phi_{nj}}{\Delta \phi_{lj}} - C_{m2j} - c_l \omega_j C_{m1j} \quad (4.46)$$

Se puede observar que la ecuación 4.46 converge a la ecuación 4.38 para el caso de una estructura sin amortiguamiento. Ademas el radio de amortiguamiento modal de estructuras de edificios es bastante pequeña. Por tal razón la ecuación 4.38 es adecuada para el calculo de la rigidez de piso.

### Definición del índice de daño por piso

En [13], la definición del daño de un edificio se puede establecer como la reducción en el porcentaje de rigidez de piso antes y después del daño. De acuerdo con [13], y basados en la ecuación 4.38, el índice de daño del  $l$ -ésimo piso, denominado  $SDI_l$ , puede ser expresado como

$$SDI_l = 1 - \frac{k_l^*}{k_l} = 1 - \frac{\omega_j^{*2} \sum_{i=l}^N \frac{m_i \phi_{ij}^*}{m_l \Delta \phi_{lj}^*}}{\omega_j^2 \sum_{i=l}^N \frac{m_i \phi_{ij}}{m_l \Delta \phi_{lj}}} \quad (4.47)$$

Donde el asterisco (\*) denota el estado de daño. El valor del  $SDI_l$  es mostrado entre 0 (sin daño) y 1 (colapso), el cual es un camino conveniente para expresar el grado de daño del piso. Para la mayoría de los edificios, la distribución de masa por piso es generalmente uniforme, por lo tanto el valor aproximado de  $SDI_l$ , representado como  $ASDI_l$  puede ser escrito como [13]:

$$ASDI_l = 1 - \frac{\omega_j^{*2} \sum_{i=l}^N \frac{\phi_{ij}^*}{\Delta \phi_{lj}^*}}{\omega_j^2 \sum_{i=l}^N \frac{\phi_{ij}}{\Delta \phi_{lj}}} \quad (4.48)$$

Obviamente el valor de  $ASDI_l$  puede ser fácilmente obtenido basado solamente en un conjunto de parámetros modales ( $\omega_j$  y  $\phi_j$ ). Esto es una ventaja para aplicaciones practicas, debido a que los parámetros modales son usualmente evaluados mediante las técnicas de identificación de sistemas, en los cuales solamente los primeros parámetros modales pueden ser identificados debido la contaminación del ruido en las mediciones [13].

Este método ha sido empleado en la presente investigación junto con el modelo numérico reducido, dada su facilidad de interpretación y la cantidad de información dinámica disponible para este estudio.

Ya descrita la metodologia, se procede a la implementacion, la cual se presenta en el siguiente capitulo, donde se realiza la identificación de danos en el pórtico de referencia propuesto por la IASC-ASCE.

---

## RESULTADOS

---

En este capítulo se presentan los resultados obtenidos de la implementación del algoritmo SL-PSO al problema de detección de daños en la estructura de referencia de la ASCE.

Inicialmente se presentan los resultados del ajuste realizado a la estructura sin daño y posteriormente se realiza la identificación de cada uno de los casos de daños establecidos. Para tal fin, se seleccionaron 3 diferentes funciones objetivo, para así poder determinar el comportamiento de problema bajo estudio, a partir de varios enfoques empleados. Dichas funciones fueron descritas anteriormente en la metodología, y por comodidad se muestran aquí nuevamente:

$$FO_1 = \sum_{i=1}^N \left| 1 - \frac{\omega_e^i}{\omega_n^i} \right| \quad (5.1)$$

$$FO_2 = \sqrt{\sum_{i=1}^N (\omega_e^i - \omega_n^i)^2} \quad (5.2)$$

$$FO_3 = \sum_{i=1}^N \left( \frac{\omega_e^i - \omega_n^i}{\omega_e^i} \right) \quad (5.3)$$

Donde  $\omega_e^i$  y  $\omega_n^i$  hace referencia a la i-ésima frecuencia experimental y numérica respectivamente. A continuación se presentan la descripción general de los resultados obtenidos.

### 5.1. AJUSTE DEL MODELO NUMÉRICO

De acuerdo con lo planteado en la metodología, el ajuste del modelo numérico se realizó mediante tres estrategias diferentes. La primera consiste en la consideración del modelo con conexiones totalmente rígidas y la única variable para ser ajustar es el módulo de elasticidad del material (E), la cual es una variable que está relacionada con la rigidez de la estructura. El segundo enfoque empleado, es el ajuste del modelo con

## RESULTADOS

conexiones semi-rígidas, donde se consideró el valor intermedio del rango de valores para este tipo de conexiones los cuales se encuentran establecidas en el Eurocodigo [72], en este caso el módulo de elasticidad es la variable a ajustar. Por ultimo, se optó por realizar el ajuste del modelo de forma mas general, en el cual los parámetros a ajustar son el módulo de elasticidad, la longitud efectiva de los elementos y la rigidez de las conexiones. Este ultimo enfoque fue implementado según las recomendaciones de [9].

Para cada caso de ajuste se realizaron 10 ejecuciones del algoritmo de optimización por cada función objetivo. Para el proceso de optimización del algoritmo SL-PSO se seleccionó un total de 50 partículas y un número máximo de 500 iteraciones. A continuación se presenta el mejor resultado encontrado en las 10 ejecuciones realizadas por el algoritmo para cada caso de ajuste del modelo numérico.

### 5.1.1. AJUSTE DEL MODELO RÍGIDO PARA LA CONDICIÓN SIN DAÑO

Este tipo de ajuste del modelo numérico consiste en la variación del modulo de elasticidad de cada uno de los elementos que componen la estructura, que para este caso en particular son:

Tabla 5.1: Elementos que componen la estructura

Tipo	Cantidad
Columna	36
Viga	48
Rigidizador	32
<b>Total</b>	116

De acuerdo a lo anterior, al variar el módulo de elasticidad de cada elemento que compone la estructura, el problema a optimizar es de 116 dimensiones. Cabe resaltar que este tipo de ajuste es muy común dada su facilidad de implementación. Inicialmente la variación del módulo de elasticidad se realizó dentro del intervalo  $[0 - 1]$ , que de forma general para el elemento  $i$  se puede expresar como:

$$E_{ai} = \alpha_i E_i \quad (5.4)$$

Donde,  $E_{ai}$  corresponde al módulo de elasticidad ajustado del elemto  $i$ ,  $\alpha_i$  es el factor de ajuste del elemeto  $i$  y  $E_i$  representa el módulo de elasticidad inicial del elemento  $i$ . En este caso,  $\alpha_i$  corresponde a la variable a determinar, ya que de acuerdo a la formulación es la que indica el grado de variación del módulo de elasticidad de cada uno de los elementos de la estructura.

Los resultados obtenidos de este ajuste, se pueden observar en las graficas 5.1 y 5.2, las cuales corresponde al mejor ajuste encontrado de las 10 ejecuciones indepedientes realizadas por cada una de las tres funciones seleccionadas.

RESULTADOS

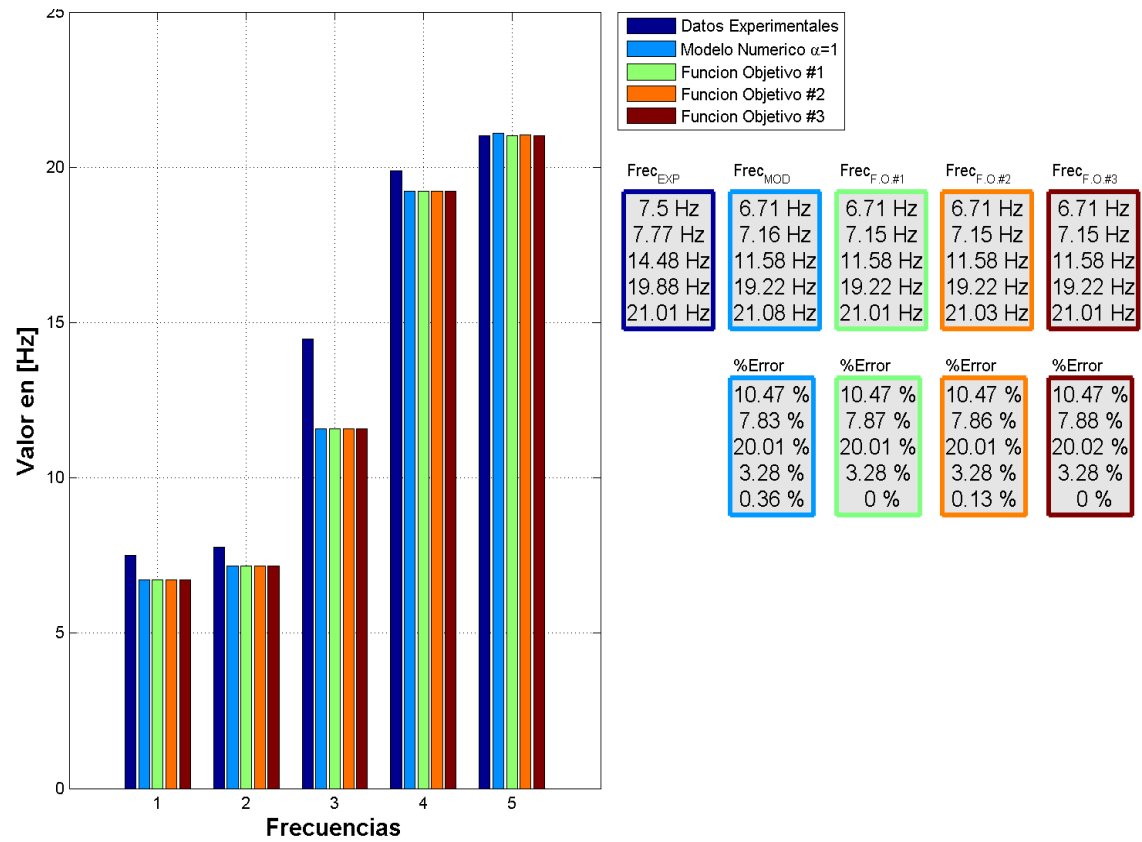


Figura 5.1: Frecuencias ajustadas modelo rígido 116D E[0-1]

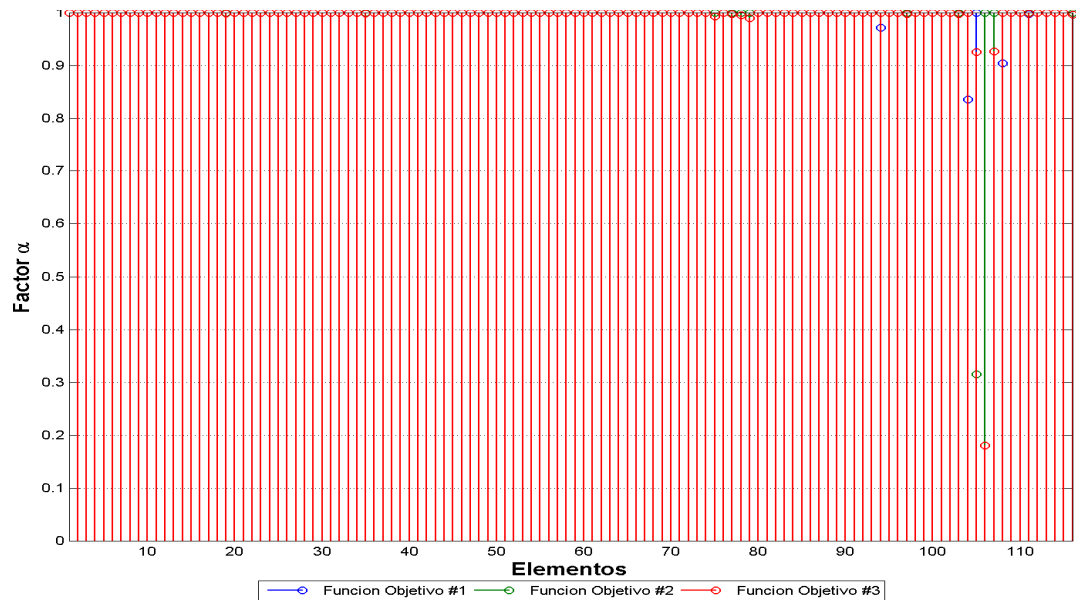


Figura 5.2: Factores de ajuste modelo rígido 116D E[0-1]

## RESULTADOS

En la grafica 5.1 se presentan las 5 primeras frecuencias del modelo experimental, el modelo numérico sin ajuste y del modelo ajustado con cada una de las tres funciones objetivo, adicionalmente se presenta el error relativo de cada una de las frecuencias con respecto al modelo experimental, y en la figura 5.2 se presentan los respectivos factores de ajuste del módulo de elasticidad de cada uno de los elementos de la estructura.

Como se puede observar en la figura 5.1, las dos primeras frecuencias ajustadas presentan un error relativo de 10,47 % y 7,83 %, la tercera es la de mayor error con 20,01 %, mientras que la cuarta y quinta frecuencia presentan un error de 3,28 % y 0,13 %, con respecto a los factores de ajuste, la mayoría se mantuvo en 1, sin embargo, el menor factor ajustado fueron de: 0.8 (función objetivo 1) , 0.3 (función objetivo 2) y 0.3 (función objetivo 3). Al analizar los resultados, se puede apreciar que no hubo una variación significativa con respecto al modelo numérico inicial.

Dado que este tipo de ajuste no presentó buenos resultados, se procedio a realizar otro enfoque, el cual consiste en reducir el espacio de búsqueda al intervalo  $[0.9-1.1]$ , que básicamente permite una variación de un 20 % en el módulo de elasticidad, lo que representa un esquema mas real. Los resultados de este enfoque fueron:

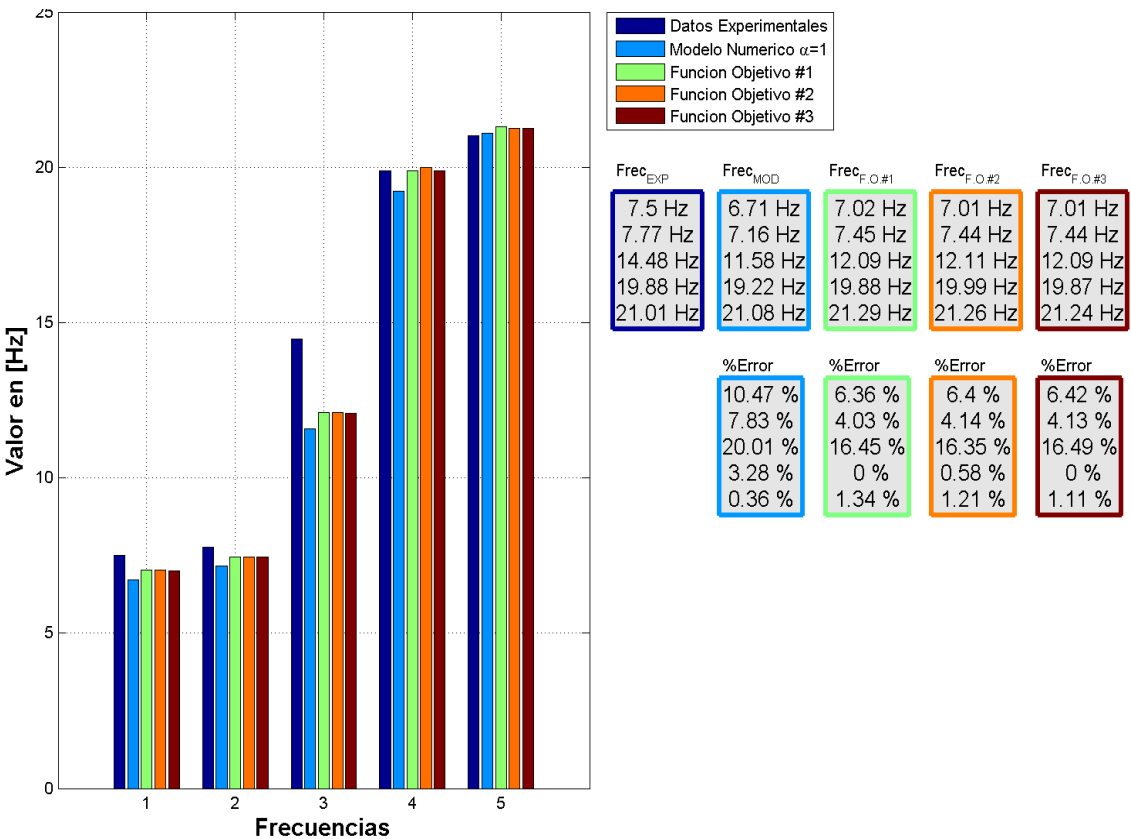


Figura 5.3: Frecuencias ajustadas modelo rígido 116D E $[0.9-1.1]$

De acuerdo a la información presentada en la figura 5.3, los errores de las frecuencias ajustadas lograron una pequeña reducción ya que en el mejor de los casos, la primera

## RESULTADOS

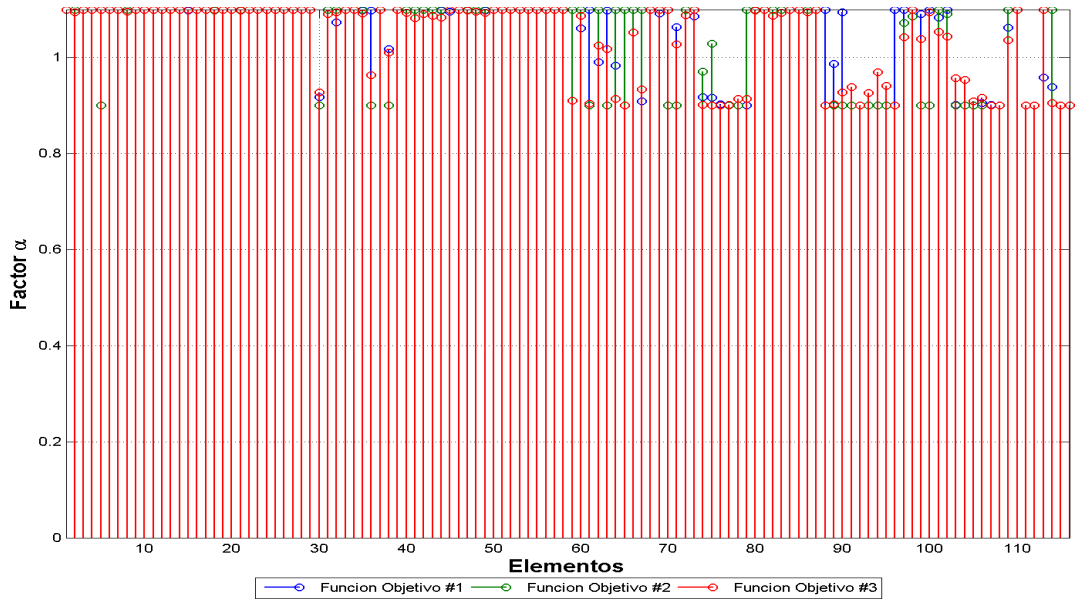


Figura 5.4: Factores de ajuste modelo rígido 116D E[0.9-1.1]

frecuencia se reduce de 10,47 % a 6,36 %, la segunda frecuencia pasa de un error de 7,83 % a 4,03 %, la tercera logra una reducción de 20,01 % a 16,35 %, mientras que la cuarta y la quinta siguen presentando errores bajos de 0 % y 1,1 % respectivamente. Con respecto a los factores de ajuste de la figura 5.4 se puede observar que cada una de las funciones implementadas, en algunos casos se asignó diferentes factores para el mismo elemento.

Con respecto al ajuste del modelo rígido se puede concluir que este tipo de ajuste no logró determinar una adecuada descripción de la estructura de referencia. Por tal motivo a continuación se presenta el ajuste del modelo de la estructura considerando conexiones semi-rígidas.

### 5.1.2. AJUSTE DEL MODELO SEMI-RÍGIDO

El ajuste presentado en esta sección se basa en consideración de la unión viga-columna como una conexión semi-rígida, la cual es muy común en las estructuras de acero.

Básicamente el ajuste se efectuó igual al enfoque del modelo rígido, es decir, se realizó la variación del módulo de elasticidad a partir del factor de ajuste  $\alpha$ , de cada uno de los 116 elementos que componen el pórtico. Sin embargo, la consideración de la conexión semi-rígida, se realizó de acuerdo a la tabla 5.2, en la cual se presenta una clasificación de la conexión a partir del factor de multiplicación  $\mu$  y el factor defijación  $\gamma$ , los cuales modifican la matriz de rigidez de la estructura. En la sección 4.2 se realizó una descripción detallada de este tipo de enfoque.

Como se pretende simular una conexión semi-rígida para el ajuste del modelo



## RESULTADOS

Tabla 5.2: Clasificación de la conexiones [6]

	Conexión articulada	Conexión semi-rígida	Conexión rígida
Factor de multiplicación	$0 \leq \mu \leq 0,5$	$0,5 \leq \mu \leq 25$	$25 < \mu < \infty$
Factor de fijación	$0 \leq \gamma \leq 0,143$	$0,143 \leq \gamma \leq 0,891$	$0,891 < \gamma \leq 1$

numérico del pórtico de referencia, el valor estimado para la la unión viga-columna es el valor medio para una conexión semi-rígida, lo cual corresponde a un factor de multiplicación de 12,75, cabe resaltar que este factor se mantendrá constante durante el proceso de ajuste.

Los resultados obtenidos de este ajuste del modelo con conexiones semi-rígidas, se pueden observar en la grafica 5.5, en la cual se encuentra el mejor ajuste encontrado del las 10 ejecuciones indepedientes realizadas por cada una de las tres funciones seleccionadas. En la grafica se presentan las primeras frecuencias del modelo experimental, el modelo numérico sin ajuste y del modelo ajustado con cada una de las tres funciones objetivo, adicionalmente se presenta el error relativo de cada una de las frecuencias con respecto al modelo experimental, y en la figura 5.6 se presentan los respectivos factores de ajuste del modulo de elasticidad de cada uno de los elementos de la estructura.

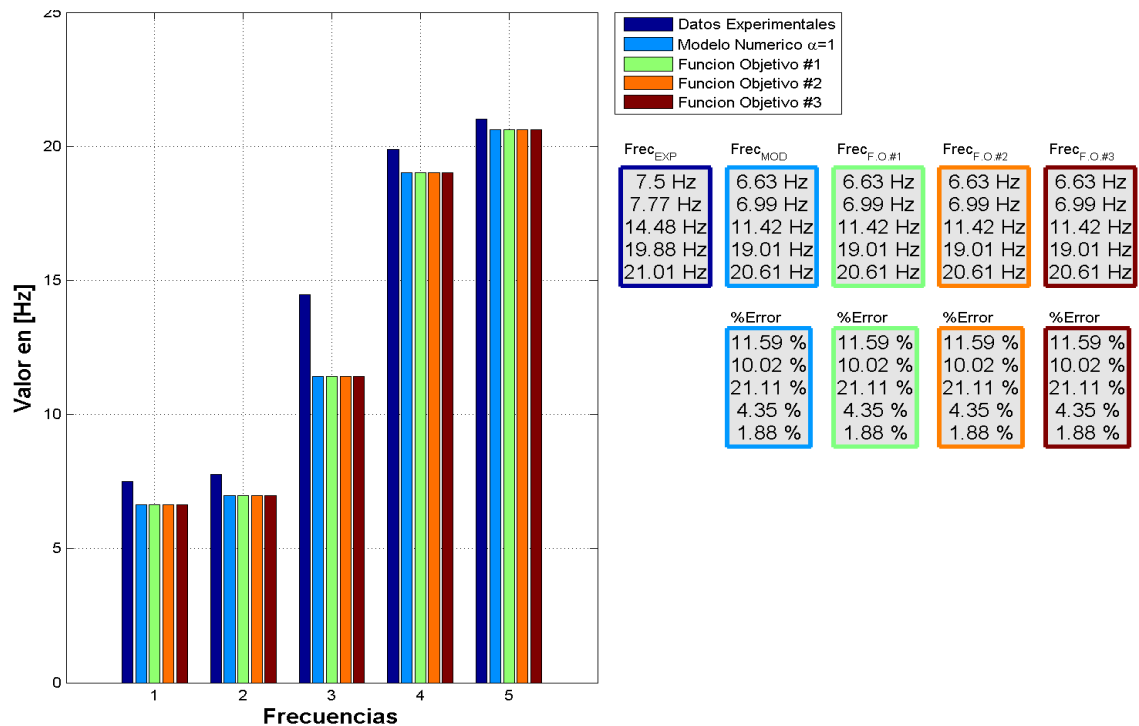


Figura 5.5: Frecuencias ajustadas modelo semi-rígido 116D E[0-1]

De acuerdo a la figura 5.5, las dos primeras frecuencias ajustadas presentan un error

## RESULTADOS

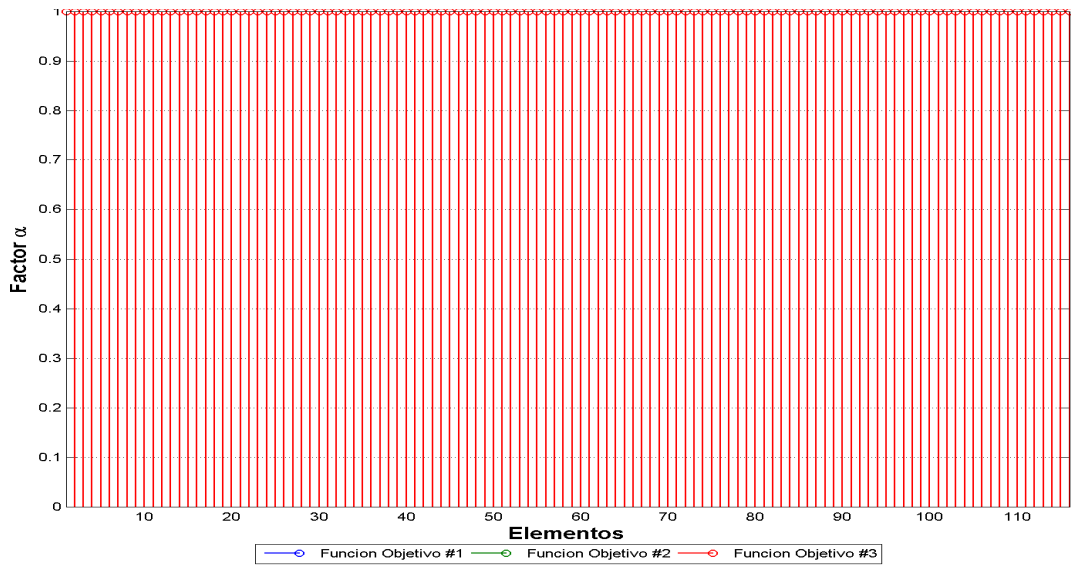


Figura 5.6: Factores de ajuste modelo semi-rígido 116D E[0-1]

relativo de 11,59 % y 10,02 %, la tercera es la de mayor error con 21,11 %, mientras que la cuarta y quinta frecuencia presentan un error de 4,35 % y 1,88 %, con respecto a los factores de ajuste, todos se fueron de 1, es decir, que no se logró determinar un mejor comportamiento de la estructura con respecto al modelo numérico inicial. De la misma forma que el ajuste del modelo rígido se realizó el enfoque de ajuste, el cual consiste en reducir el espacio de búsqueda al intervalo  $[0,9 - 1,1]$ .

De acuerdo a la información presentada en la figura 5.7, los errores de las frecuencias ajustadas lograron una pequeña reducción ya que en el mejor de los casos, la primera frecuencia se reduce de 11,59 % a 7,31 %, la segunda frecuencia pasa de un error de 10,02 % a 5,99 %, la tercera logra una reducción de 21,11 % a 17,44 %, mientras que la cuarta y la quinta siguen presentando errores bajos de 0 %. Con respecto a los factores de ajuste presentados en la figura 5.8 se puede observar que para cada una de las funciones implementadas, en algunos casos se asignó diferentes factores para el mismo elemento.

De acuerdo con ajuste del modelo rígido se puede concluir que el modelo con conexiones semi-rígidas no logro mejorar la identificación con respecto a los datos experimentales. Así se procede a la implementacion del ajuste de un modelo mas general.

### 5.1.3. AJUSTE DEL MODELO GENERAL

Finalmente, se realizó un ajuste del modelo numérico de forma más general, dado que los resultados de los anteriores enfoques no lograron ser adecuados para la representación del modelo numérico de la estructura.

En este enfoque, la idea básica consiste en la adición de variables que podrian influenciar en el comportamiento de la estrucutra como son el módulo de elasticidad,

## RESULTADOS

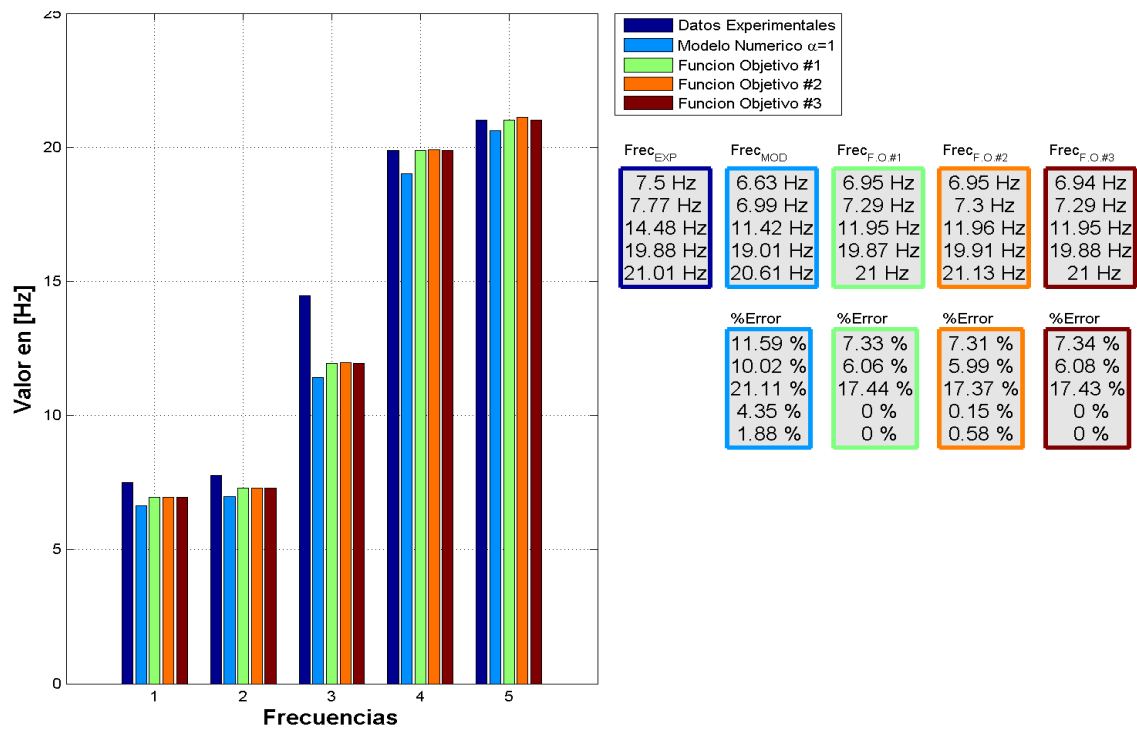


Figura 5.7: Frecuencias ajustadas ajuste modelo semi-rígido 116D E[0.9-1.1]

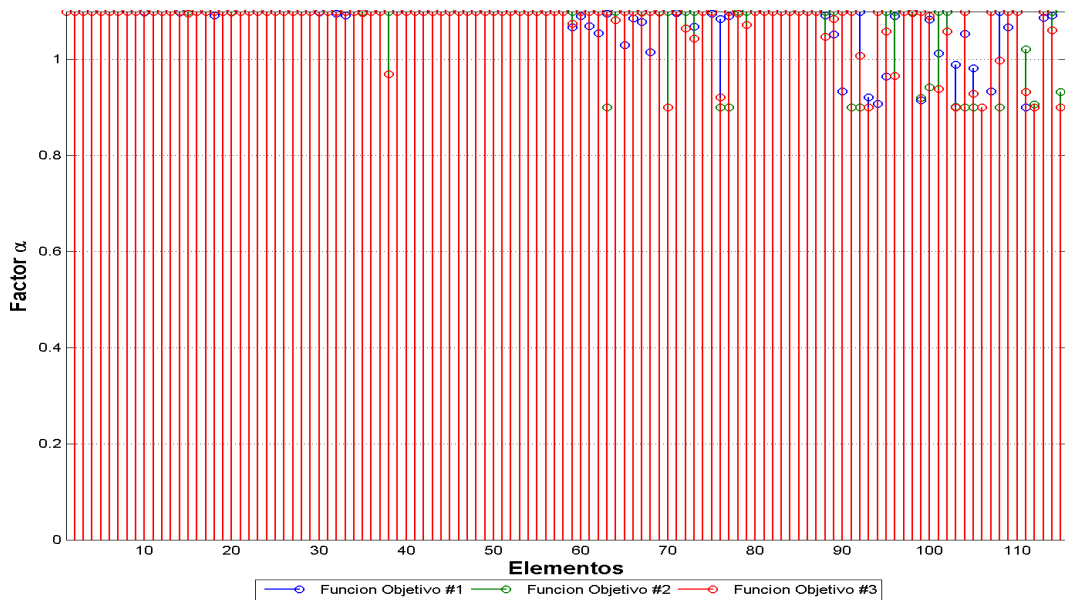


Figura 5.8: Factores de ajuste modelo semi-rígido 116D E[0.9-1.1]

la longitud efectiva de los elementos y la rigidez de las conexiones. Para este tipo de ajuste se realizaron dos enfoques. En el primero se consideró un modelo de 17 variables para ser ajustado. El segundo consiste en un modelo de 6 variables.

## RESULTADOS

A continuación se muestran los resultados obtenidos para cada enfoque establecido.

### Modelo de 17 Dimensiones

Inicialmente el ajuste realizado para el modelo numérico consistía en un problema de 116 dimensiones, que correspondían a los elementos que componen la estructura, sin embargo, para el caso estudiado en esta sección se estableció el tamaño del problema a optimizar de 17 dimensiones, dado que se realizó una agrupación de variables a ser ajustadas, las cuales se pueden observar en la tabla 5.3, en donde se establecen cada una de las variables a ajustar como su intervalo de búsqueda.

Tabla 5.3: Variables para el ajuste modelo semi-rígido 17D

Módulo de elasticidad $E[0,9 - 1,1]$		
$\mathbf{X}(1) \Rightarrow$ Columnas Piso1	$\mathbf{X}(5) \Rightarrow$ Vigas Piso1	$\mathbf{X}(9) \Rightarrow$ Rigidizadores Piso1
$\mathbf{X}(2) \Rightarrow$ Columnas Piso2	$\mathbf{X}(6) \Rightarrow$ Vigas Piso2	$\mathbf{X}(10) \Rightarrow$ Rigidizadores Piso2
$\mathbf{X}(3) \Rightarrow$ Columnas Piso3	$\mathbf{X}(7) \Rightarrow$ Vigas Piso3	$\mathbf{X}(11) \Rightarrow$ Rigidizadores Piso3
$\mathbf{X}(4) \Rightarrow$ Columnas Piso4	$\mathbf{X}(8) \Rightarrow$ Vigas Piso4	$\mathbf{X}(12) \Rightarrow$ Rigidizadores Piso4
Longitud efectiva $L[0,9 - 1,0]$		
$\mathbf{X}(13) \Rightarrow$ Longitud efectiva Columnas		
$\mathbf{X}(14) \Rightarrow$ Longitud efectiva Vigas		
$\mathbf{X}(15) \Rightarrow$ Longitud efectiva Rigidizadores		
Flexibilidad en los nodos $F[6 - 24]$		
$\mathbf{X}(16) \Rightarrow$ Flexibilidad nodos dirección fuerte		
$\mathbf{X}(17) \Rightarrow$ Flexibilidad nodos dirección debil		

Los resultados obtenidos de este ajuste, se pueden observar en la grafica 5.9, y en la tabla 5.4 se presentan los factores de ajuste para cada dimensión de cada función objetivo.

Como se pueden observar en la grafica 5.9, se presentan las primeras frecuencias del modelo experimental, el modelo numérico sin ajuste y del modelo numérico ajustado con cada una de las tres funciones objetivo, adicionalmente se presenta el error relativo de cada una de las frecuencias con respecto al modelo experimental. Los errores de las frecuencias ajustadas lograron una significativa reducción ya que en el mejor de los casos, la primera frecuencia se reduce de 11,609 % a 2,33 %, la segunda frecuencia pasa de un error de 10,05 % a 0 %, la tercera logra una reducción de 21,13 % a 12,66 %, mientras que la cuarta y la quinta siguen presentando errores bajos de 1,45 % y 3,17 % respectivamente. Con respecto a los factores de ajuste se puede observar que en forma general hubo una semejanza en el ajuste.

De acuerdo con el ajuste del modelo rígido y el modelo con conexiones semi-rígidas

## RESULTADOS

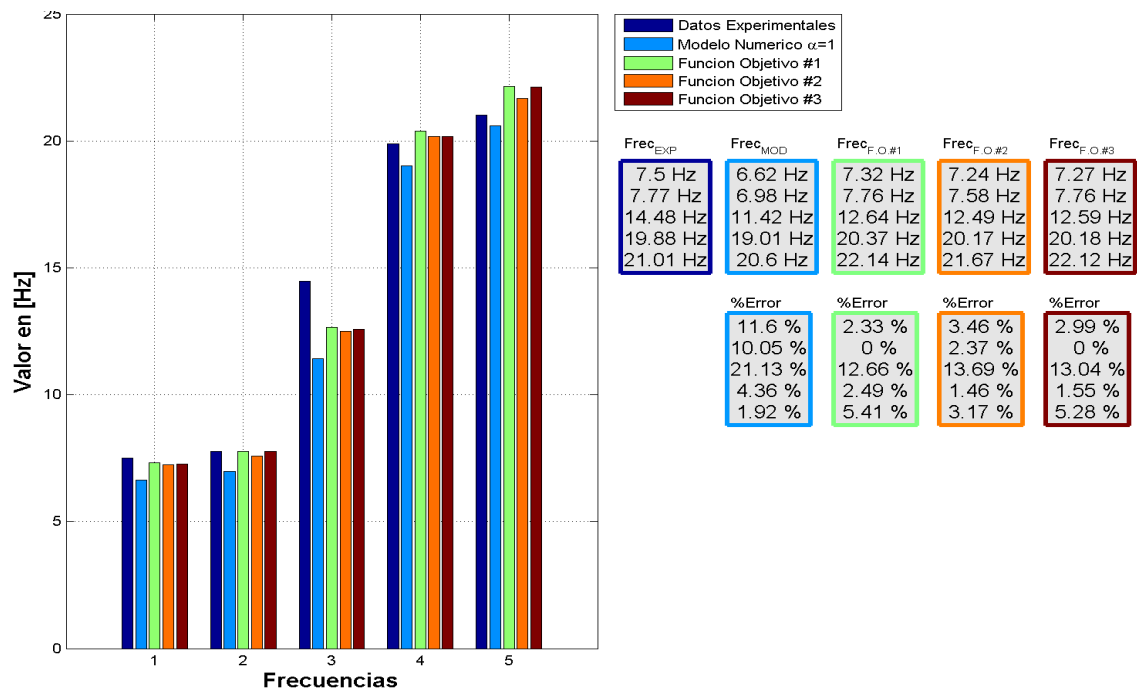


Figura 5.9: Ajuste modelo semi-rígido 17D E[0.9-1.1] L[0.9-1.1]  $\mu$ [6-24]

Tabla 5.4: Factores ajuste modelo semi-rígido 17D

Función	X(1)	X(2)	X(3)	X(4)	X(5)	X(6)
Sin Ajuste	1,00	1,00	1,00	1,00	1,00	1,00
$Fobj_1$	1,10	0,90	0,90	0,90	1,10	1,10
$Fobj_2$	0,90	0,90	0,90	0,90	1,10	1,10
$Fobj_3$	1,09	0,90	0,90	0,90	1,10	1,10

Función	X(7)	X(8)	X(9)	X(10)	X(11)	X(12)
Sin Ajuste	1,00	1,00	1,00	1,00	1,00	1,00
$Fobj_1$	0,90	0,90	1,10	1,10	1,10	0,90
$Fobj_2$	0,93	0,90	1,10	1,10	1,10	0,90
$Fobj_3$	0,90	0,90	1,10	1,10	1,04	0,90

Función	X(13)	X(14)	X(15)	X(16)	X(17)
Sin Ajuste	1,00	1,00	1,00	12,50	12,50
$Fobj_1$	0,90	0,90	0,94	24,00	15,53
$Fobj_2$	0,90	0,90	0,94	24,00	9,63
$Fobj_3$	0,90	0,90	0,94	24,00	23,73

se logró mejorar la identificación con respecto a los datos experimentales. Sin embargo, a continuación se presenta el ajuste del modelo considerando solamente 6 dimensiones.

## RESULTADOS

### Modelo de 6 Dimensiones

Para el caso del modelo de 6 dimensiones, se consideró el módulo de elasticidad como una sola variable a ajustar, es decir, que la variación de dicho módulo es el mismo para cada elemento que compone la estructura, esto debido a que se considera que el material de toda la estructura es el mismo por tal razón se presume que el módulo de elasticidad debe de variar en la misma magnitud para todos los elementos que componen el pórtico. Las variables a ser ajustadas, se pueden observar en la tabla 5.5, en donde se establecen cada una de las características de cada variable.

Tabla 5.5: Variables de ajuste para el ajuste modelo semi-rígido 6D

Variabes	Descripción	Limites
$X(1)$	Módulo de elasticidad (Columnas, Vigas, Rigidizadores)	$[0,9 - 1,1]$
$X(2)$	Longitud efectiva de columnas	$[0,9 - 1,0]$
$X(3)$	Longitud efectiva de vigas	$[0,9 - 1,0]$
$X(4)$	Longitud efectiva de rigidizadores	$[0,9 - 1,0]$
$X(5)$	Flexibilidad nodos dirección fuerte	$[6 - 24]$
$X(6)$	Flexibilidad nodos dirección debil	$[6 - 24]$

En la grafica 5.10 se encunetran registrados los resultados del ajuste del modelo de 6 dimensiones. En dicha grafica se presentan las primeras frecuencias del modelo experimental, el modelo numérico sin ajuste y del modelo ajustado con cada una de las tres funciones objetivo, adicionalmente se presenta el error relativo de cada una de las frecuencias con respecto al modelo experimental.

En los resultados obtenidos, los errores de las frecuencias ajustadas lograron una significativa reducción con respecto a los modelos de 116 dimensiones, y muy similar al modelo de 17 dimensiones, ya que en el mejor de los casos, la primera frecuencia se reduce de 11,609 % a 0,02 %, la segunda frecuencia pasa de un error de 10,05 % a 0 %, la tercera logar una reducción de 21,13 % a 11,56 %, mientras que la cuarta y la quinta siguen presentando errores de 2,70 % y 4,12 % respectivamente.

Tabla 5.6: Factores ajuste modelo semi-rígido 6D

Función	$X(1)$	$X(2)$	$X(3)$	$X(4)$	$X(5)$	$X(6)$
Sin Ajuste	1,00	1,00	1,00	1,00	12,50	12,50
$Fobj_1$	1,10	0,90	0,90	0,91	24,00	6,00
$Fobj_2$	1,10	0,92	0,90	1,00	24,00	6,00
$Fobj_3$	1,10	0,90	0,90	0,98	24,00	24,00

Con relación al ajuste del modelo de 17 dimensiones se obtuvo una similitud en la identificación con respecto a los datos experimentales.

## RESULTADOS

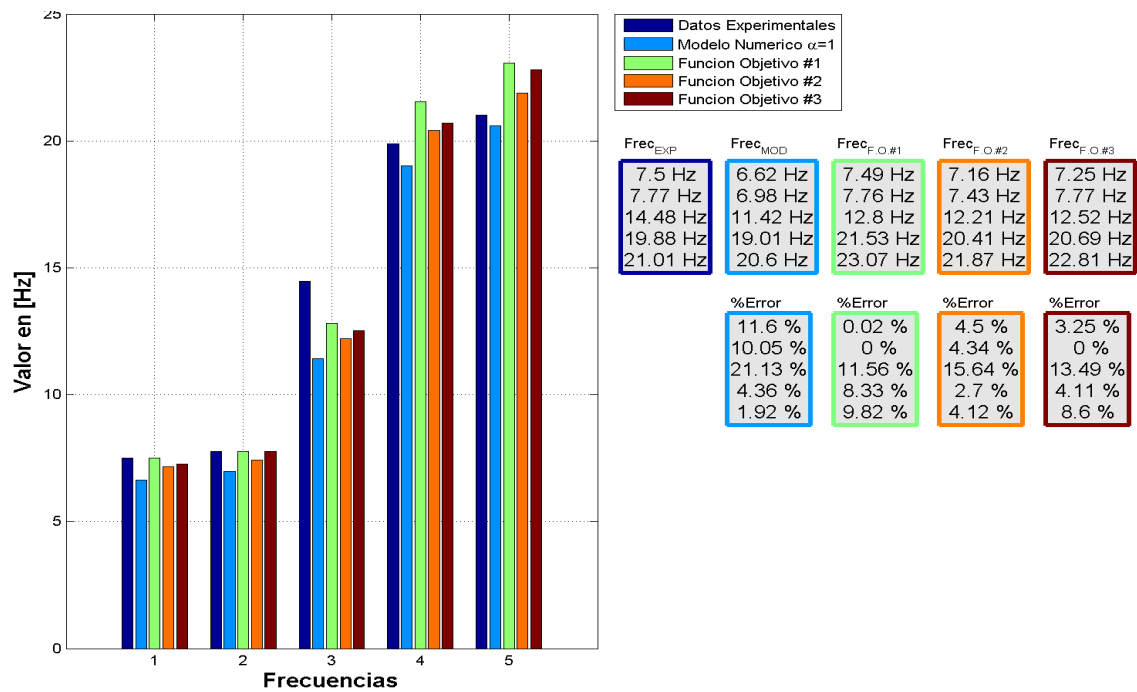


Figura 5.10: Ajuste modelo semi-rígido 6D E[0.9-1.1] L[0.9-1.1]  $\mu$ [6-24]

Por último, en la tabla 5.7 se pueden observar los resultados del algoritmo SL-PSO, en términos del valor de la función objetivo y el tiempo de ejecución, tanto para cada una de las funciones empleadas como para cada uno de los ajustes realizados. En términos generales se puede apreciar que el tiempo promedio en realizar el ajuste es muy similar e independiente de las variables que intervienen, esto es debido al tamaño del problema físico es el mismo, es decir, que la estructura presenta 216 grados de libertad y el ensamble de las matrices tanto de rigidez como de masa para el cálculo de la función objetivo posee un costo computacional. En cuanto a la calidad de la función objetivo, se puede observar que las que presentan los mejores resultados son, la función objetivo número uno y número tres con valores de ajuste muy similares.

Tabla 5.7: Resultados modelos ajustados

MODELO AJUSTE	FO1		FO2		FO3	
	tiempo (segundos)	mejor	tiempo (segundos)	mejor	tiempo (segundos)	mejor
116D C.R. E[0-1]	5142,607	0,487	4827,848	3,133	4116,465	0,417
116D C.R. E[0,9-1,1]	4073,531	0,320	4864,040	2,454	4194,820	0,281
116D C.S-R. E[0-1]	4515,234	0,575	4146,676	3,408	4264,464	0,490
116D C.S-R E[0,9-1,1]	4841,209	0,355	4865,565	2,620	4297,160	0,309
17D	4852,949	0,245	4906,220	2,136	4286,313	0,229
6D	4928,138	0,297	4767,642	2,529	4318,063	0,295

## RESULTADOS

Finalmente, del proceso de ajuste del modelo numérico a partir de la información experimental se puede observar que los modelos de 116 dimensiones que tenían un único parámetro de ajuste (módulo de elasticidad) no presentaron un buen ajuste, sin embargo los modelos de 17 y 6 dimensiones que presentaban varios parámetros de ajuste lograron una mejor identificación del modelo numérico, por tal motivo en este trabajo se seleccionó el ajuste del modelo de 6 dimensiones como el modelo que representa de forma más realista la estructura sin daño de acuerdo a la información experimental.

### 5.2. IDENTIFICACIÓN DE LOS CASOS DE DAÑOS

Una vez ajustado el modelo numérico sin daño, se procede a la identificación de los casos de daños considerados para la estructura de referencia, cuya descripción detallada se encuentra en la sección 4.5.

Cabe resaltar, que la estimación del daño para el problema estudiado, se puede llegar hasta el nivel 1 (Determinación de que el daño esta presente dentro de la estructura), esto debido a la información con la que se dispone, dado que las frecuencias de vibración son propiedades globales de la estructura.

Teniendo en cuenta lo anterior, la estimación de la presencia de daño en la estructura se realizó mediante el indice denominado “*Aproximate Story Damage Index*” (ASDI) el cual esta definido para el  $l$ -ésimo piso como [13]:

$$ASDI_l = 1 - \frac{\omega_j^{*2} \sum_{i=l}^N \frac{\phi_{ij}^*}{\Delta \phi_{lj}^*}}{\omega_j^2 \sum_{i=l}^N \frac{\phi_{ij}}{\Delta \phi_{lj}}} \quad (5.5)$$

Donde  $\omega_j$  representa la frecuencia natural en la dirección  $j$ ,  $\phi_{ij}$  corresponde al componente del modo de vibración del piso  $i$  en la dirección  $j$ ,  $\Delta \phi_{lj}$  es la deriva del piso  $l$  en la dirección  $j$ , y el asterisco (\*) denota el estado de daño. El valor del  $ASDI_l$  se encuentra entre 0 (sin daño) y 1 (colapso), el cual es una forma conveniente para expresar el grado de daño del piso.

De acuerdo a la ecuación 5.5, para calcular el indice de daño, es necesario determinar tanto las frecuencias como los modos de vibración de la estructura. Para tal fin, se realizó el proceso de optimización tomando como partida el modelo ajustado en la sección anterior y por medio del SL-PSO, se efectuó la identificación del modelo a partir de la información experimental disponible. En este caso en particular se consideró el modelo con un único parámetro para ser ajustado que corresponde al área de la sección transversal de los rigidizadores (A), para un tamaño de 32 variables, que corresponde a los elementos tipo rigidizador que posee la estructura.

Como se pudo establecer en la seccion 2.2 los casos de daño fueron simulados mediante la ausencia de uno o algunos rigidizadores, por lo tanto el proceso de optimización realizado fue discreto, es decir, que el parámetro que representa el daño toma el valor



## RESULTADOS

de 0 cuando se considera la perdida del elemento y el valor de 1 cuando el elemento no es retirado de la estructura.

Para cada caso de identificación, se realizaron 10 ejecuciones del algoritmo de optimización SL-PSO, mediante la función:

$$\Delta\lambda_1 = \sum_{i=1}^N \left| 1 - \frac{\omega_e^i}{\omega_n^i} \right| \quad (5.6)$$

La cual se seleccionó debido a que presentó el mejor desempeño en el ajuste del modelo numérico sin daño. En la ecuación 5.6,  $\omega_e^i$  y  $\omega_n^i$  hace referencia a la i-ésima frecuencia experimental y numérica respectivamente.

Para realizar el proceso de optimización del algoritmo SL-PSO, se seleccionó un total de 50 partículas y se definió un número máximo de 500 iteraciones por cada ejecución del algoritmo. A continuación en la tabla 5.8, se presenta el mejor resultado encontrado en las 10 ejecuciones realizadas por el algoritmo para cada caso de identificación de daño estudiado:

Tabla 5.8: Frecuencias ajustadas para los casos de daño

Frecuencia [Hz] en la dirección debil			
Configuración	Ajustada	Experimental	%Error
I	7.50	7.50	0.00
II	7.44	7.73	3.75
III	7.44	7.65	2.74
IV	7.44	7.60	2.10
V	7.45	7.56	1.45
VI	5.94	5.96	0.16
VII	2.55	2.64	3.40
Frecuencia [Hz] en la dirección fuerte			
Configuración	Ajustada	Experimental	%Error
I	7.77	7.77	0.00
II	5.72	5.56	2.87
III	6.66	6.65	0.15
IV	7.19	7.35	2.18
V	7.26	7.45	2.55
VI	7.77	7.79	0.25
VII	3.74	3.63	3.03

En la tabla 5.8, se presentan los resultados obtenidos, correspondientes a la primera frecuencia en la dirección **debil** y la primera frecuencia en dirección **fuerte**, sin embargo cabe resaltar que la configuración I corresponde al modelo sin daño, por lo tanto las frecuencias mostradas para dicha configuración corresponde a las del modelo ajustado

## RESULTADOS

en la sección anterior. Adicionalmente, se presenta la frecuencia que fue identificada experimentalmente junto con el error relativo con respecto a la identificada mediante el modelo numérico empleando el SL-PSO. De los resultados obtenidos se puede apreciar que los errores presentados están entre el 0,15 % y el 3,75 %.

Adicionalmente, en la tabla 5.9 se encuentran registrados los modos de vibración asociados a las dos primeras frecuencias ajustadas del modelo numérico. Para este caso, se presentan la componente principal de cada modo, por cada piso de la estructura de referencia y en cada dirección, para cada una de las configuraciones de daños establecidas.

Tabla 5.9: Modos de vibración							
Modo en la dirección débil							
	Conf-I	Conf-II	Conf-III	Conf-IV	Conf-V	Conf-VI	Conf-VII
$\phi_4$	1,00	1,00	1,00	1,00	1,00	1,00	1,00
$\phi_3$	0,87	0,84	0,84	0,84	0,84	0,92	0,86
$\phi_2$	0,63	0,60	0,60	0,60	0,61	0,69	0,60
$\phi_1$	0,31	0,30	0,30	0,30	0,30	0,27	0,26
Modo en la dirección fuerte							
	Conf-I	Conf-II	Conf-III	Conf-IV	Conf-V	Conf-VI	Conf-VII
$\phi_4$	1,00	1,00	1,00	1,00	1,00	1,00	1,00
$\phi_3$	0,85	0,85	0,82	0,81	0,84	0,85	0,81
$\phi_2$	0,60	0,61	0,53	0,54	0,57	0,60	0,52
$\phi_1$	0,27	0,32	0,22	0,25	0,27	0,27	0,19

Una vez establecidos tanto las frecuencias como los modos de vibración, se procede a la estimación del índice de daño, a partir de la ecuación 5.5:

Con respecto a los resultados que se encuentran registrados en la tabla 5.10, se puede observar que para las configuraciones:

- II Ausencia de todos los rigidizadores en el costado este.
- III Ausencia de rigidizadores de la mitad sur del costado este.
- IV Ausencia de rigidizadores en el primer y cuarto piso en la mitad sur del costado este.
- V Ausencia de rigidizadores en el primer piso en la mitad sur del costado este.
- VII Ausencia de todos los rigidizadores.

Se logró obtener una buena identificación tanto en el sentido como en la estimación de la pérdida de rigidez, sin embargo, para estos casos en el piso 4 en la dirección débil, se obtuvo un índice entre 0.20 y 0.22 que se puede considerar como una falsa

## RESULTADOS

Tabla 5.10: Índices de daño

$ASDI_{debil}$						
	Conf-II	Conf-III	Conf-IV	Conf-V	Conf-VI	Conf-VII
P4	0,22	0,20	0,21	0,20	0,03	0,81
P3	0,01	0,01	0,01	0,00	0,30	0,81
P2	0,00	0,00	0,00	0,00	0,50	0,81
P1	0,00	0,00	0,00	0,00	0,27	0,77

$ASDI_{fuerte}$						
	Conf-II	Conf-III	Conf-IV	Conf-V	Conf-VI	Conf-VII
P4	0,48	0,40	0,35	0,22	0,00	0,82
P3	0,42	0,37	0,20	0,18	0,00	0,80
P2	0,37	0,23	0,05	0,05	0,00	0,78
P1	0,54	0,18	0,14	0,14	0,00	0,70

indicación en la presencia del daño. Por otro lado, en el sentido fuerte se logró obtener índices que oscilan entre 0.37 y 0.54 para el caso II, en el cual se considera una buena estimación del daño. El caso de daño III fue identificado adecuadamente ya que los factores son un poco menores al de caso II, lo cual es evidente de esperarse. Para el caso IV se logró identificar pérdida de rigidez en el piso 1 de 0.14 y en el piso 4 de 0.35 sin embargo en el piso 3 hay una falsa indicación de 0.20, por otro lado, en la configuración V se estimó el índice del piso 1 en 0.14, sin embargo, el índice correspondiente para el piso 3 y 4 generó una falsa indicación de daño de 0.18 y 0.22 respectivamente, finalmente para el caso de daño VII, que hace referencia al daño más severo, se obtuvo un factor de daño entre 0.70 y 0.82 para ambas direcciones lo que se puede considerar como una buena estimación de daño. Particularmente, para el caso de identificación VI que corresponde a la ausencia de rigidizadores en el segundo piso del costado oeste, no se logró determinar una adecuada estimación del daño ni en dirección ni la ubicación del piso.

En la tabla 5.11 encuentran registrados los resultados del algoritmo SL-PSO, tanto en términos del valor de la función objetivo como en el tiempo promedio en realizar una ejecución de 500 iteraciones, para cada uno de los siete casos de daño identificado. Al observar el tiempo promedio de ejecución del algoritmo, se puede apreciar que este es muy similar independiente del caso de daño estudiado, tardando aproximadamente 1.10 horas por ejecución. Por otro lado, en cuanto a la calidad de la función objetivo obtenida, se puede observar que en cada uno de los siete patrones de daño, se lograron obtener buenos resultados, con un valor de la función objetivo que oscila entre 0.205 y 0.632.

A continuación se presentan los resultados obtenidos por el estudio realizado por [13], donde se empleó el índice de daño para identificar las configuraciones de daño II, V y VII, de la estructura de referencia propuesta por la IASC-ASCE.

## RESULTADOS

Tabla 5.11: Resultados casos de daño

Patron de daño	tiempo (segundos)	Mejor
II	3823.076	0.357
III	3762.202	0.205
IV	3840.298	0.204
V	3843.978	0.220
VI	3657.015	0.216
VII	3768.376	0.632

Tabla 5.12: Indices de daño [13]

$ASDI_x$			
	Conf-II	Conf-V	Conf-VII
P4	-0,01	0,10	0,89
P3	0,04	-0,06	0,88
P2	-0,18	-0,08	0,89
P1	-0,01	-0,09	0,86

$ASDI_y$			
	Conf-II	Conf-V	Conf-VII
P4	0,42	-0,10	0,82
P3	0,63	0,04	0,78
P2	0,49	0,02	0,81
P1	0,57	0,21	0,78

Al observar los índices obtenidos por [13] en los cuales hubo una buena identificación en los casos de daño estudiados, se puede apreciar que para la configuración II hay una similitud en los índices de los 4 pisos en la dirección fuerte (dirección y), mientras que para la dirección debil (dirección x) hay una diferencia considerable en el piso 4, lo cual representa una falsa identificación del daño. Por otro lado, para el caso de daño V que representa un daño leve, se presentan diferencias considerables en el índice de los pisos 3 y 4 en la dirección fuerte (dirección y), mientras que para la dirección debil (dirección x), hay una discrepancia en el piso 4. Sin embargo, para los trabajos comparados, en la configuración de daño VII, se logró una adecuada estimación de los índices de daño para cada uno de los 4 pisos, dado que esta configuración es más severa que las dos configuraciones mencionadas anteriormente.

---

## CONCLUSIONES

---

- Se presentó un panorama general del estado del arte en la investigación sobre la detección e identificación de daños en estructuras civiles. Aunque esta es una disciplina que se ha venido desarrollando durante los últimos veinte años, las necesidades por cubrir aún son muchas (en instrumentación, software de extracción de datos, desarrollo de modelos numéricos), principalmente en países como Colombia, que a pesar de contar con un gran implementación en el campo de la Ingeniería Sísmica, el gremio profesional especializado aún no incorpora dentro de sus prácticas las técnicas de D.D.
- Se realizó una descripción de la estructura de referencia, la cual consta de un pórtico de acero tridimensional desarrollado en la Universidad de la Columbia Británica. Las frecuencias naturales para cada uno de los casos de daños estudiados, fueron determinadas a partir de trabajos reportados en la literatura internacional [7, 8, 9, 2] relacionados a dicha estructura. Adicionalmente se obtuvo la rutina de matlab denominada Datagen, que es un programa que simula la respuesta dinámica de la estructura de referencia, por medio de un modelo de 120 grados de libertad, del cual se modificó la flexibilidad incluyendo las variables a optimizar, para lo cual se colocó como “*esclavo*” del SL-PSO para realizar el proceso de optimización.
- Las cuatro técnicas de optimización seleccionadas (SA, AG, PSO y SLPSO) han sido ampliamente usadas en el campo de la investigación. Cada una de dichas técnicas fue implementada en problemas de optimización continua con y sin restricciones, esto con el fin de determinar las fortalezas y debilidades de cada técnica estudiada. Finalmente el SL-PSO fue seleccionado dado que para el caso de optimización continua presentó el mejor desempeño en cada una de las funciones de prueba, donde se obtuvo una desviación estándar máxima de  $6.25e-4$  para 8 de las 9 funciones de prueba, sin embargo para la función Schwefel el SL-PSO presentó una desviación de 235,63 frente a 585.50 del PSO, 249 del SA y 424.95 del AG. Para el caso restringido el algoritmo SL-PSO presentó buen desempeño para 11 de las 13 funciones de prueba, sin embargo en relación con el PSO, AG y SA

## CONCLUSIONES

reportados en la literatura se evidencio la capacidad del SL-PSO de abordar esta tipología de problemas al igual que las demás técnicas con las que se comparó.

- La metodología propuesta para la identificación de daños en pórticos de acero en 3D combina ideas y conceptos- de la dinámica estructural para llevar a cabo la identificación. A pesar de que en la literatura se pueden encontrar procedimientos y técnicas que parten del supuesto que poseen suficiente información de las estructuras, es importante seguir investigando en técnicas que obtengan un mayor provecho de la información parcial que realmente se puede llegar a obtener.
- El ajuste del modelo numérico de la estructura de referencia se realizó por medio del SL-PSO, para el cual se seleccionaron 3 funciones objetivo y se tuvo en cuenta diferentes parámetros de ajuste. De los resultados, se estableció que la función que mejor desempeño presentó, logró realizar un ajuste de las dos primeras frecuencias con un error del 0.02 % y el 0 % respectivamente. Por otro lado el mejor ajuste realizado fue el enfoque que contenía 3 parámetros de ajuste (Longitud efectiva, módulo elasticidad y flexibilidad en nodos). A partir de lo anterior, se puede establecer que es importante tener en cuenta la mayor cantidad de parámetros que pueden llegar a influir en el comportamiento de la estructura.
- La metodología propuesta fue implementada en la estructura de referencia para 6 patrones de daño, que consistían en la ausencia de uno a varios rigidizadores dentro de la estructura. Para cada caso de daño se determinó un índice de daño por piso y por dirección de la estructura. Para el caso de la dirección débil, las configuraciones II, III, IV y V en los tres primeros pisos se logró determinar un índice de 0.00 (sin daño) mientras que para el cuarto piso se determinó un índice de 0.20, la configuración VI presentaron un índice promedio por piso de 0.4 y por último La configuración VII presento un índice promedio de 0.8 (daño severo). Para la dirección fuerte, la configuración II III IV y V, presentaron índices desde 0.05 a 0.54, la configuración VII presento un índice de 0.00 en cada piso y por último La configuración VII presento un índice promedio de 0.78 (daño severo). Con respecto a la identificación del dirección del daño se consideró adecuada excepto para la configuración VI. Finalmente se realizó una comparación con un trabajo de la literatura internacional, donde se pudo establecer que para casos de daños leves se logró determinar la dirección del daño, mas no la magnitud, mientras para casos de daños severos se llegó a estimar la dirección y la perdida de rigidez. Finalmente la metodolgia presentada se puede ubicar en el nivel 1 de idetificación de daños según la clasificación de Rytter [20].

---

## TRABAJOS FUTUROS

---

Desde el punto de vista del que presenta este trabajo, las líneas de investigación que pueden seguirse a raíz de lo desarrollado en este estudio se enumeran a continuación:

- Realizar de estudios de instrumentación en estructuras civiles.
- Desarrollar metodologías de extracción de datos, para la caracterización de estructuras reales.
- Elaborar programas para el modelamiento numérico de estructuras a partir de librerías optimizadas y técnicas en paralelo.

---

## BIBLIOGRAFÍA

---

- [1] Dyke S.J., Bernal D., Beck J.L., and Ventura C. An experimental benchmark problem in structural health monitoring. In *Third International Workshop on Structural Health Monitoring*, 2001.
- [2] Ventura C.E., LORD J.F., Turek M., Sereci A. M., Radulescu D., and Radulescu C. Experimental studies and remote monitoring of iasc-asce benchmark test frame. In *Proceeding of the XXI International Modal Analysis Conference*, 2003.
- [3] Liang J.J., Qin A.K., Suganthanand P.N., and Baskar S. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Transactions on Evolutionary Computation*, 10:281–295, 2006.
- [4] Schwarz J. and Richardson M. Experimental modal analysis. Vibrant Technology, Inc. 1999.
- [5] Paz Mario. *Dinámica estructural: teoría y cálculo*. Editorial Reverté S. A., 2002.
- [6] Yun Chung-Bang, Yi Jin-Hak, and Bahng Eun Young. Joint damage assessment of framed structures using a neural networks technique. *Engineering Structures*, 23:425–435, 2001.
- [7] Caicedo Juan Martin. *Structural health monitoring of flexible civil structures*. PhD thesis, Saint Louis: Washington University. Department of Civil Engineering, 2003.
- [8] Ching J. and Beck J.L. Bayesian analysis of the phase ii iasc-asce structural health monitoring experimental benchmark data. *Journal of Engineering Mechanics*, 130:1233–1244, 2004.
- [9] Wu J.R. and Li Q.S. Structural parameter identification and damage detection for a steel structure using a two-stage finite element model updating method. *Journal of Constructional Steel Research*, 62:231–299, 2006.
- [10] Li Hong, Jiao Yong-Chang, and Wang Yuping. Integrating the simplified interpolation into the genetic algorithm for constrained pptimization problems. *Springer-Verlag Berlin Heidelberg*, 3801:247–254, 2005.



## BIBLIOGRAFÍA

- [11] Hedar A. *Studies on metaheuristics for continuous global optimization problems*. PhD thesis, Kioto: Kioto University, 2004.
- [12] Toscano Pulido Gregorio and Coello Coello Carlos. A constraint-handling mechanism for particle swarm optimization. *Evolutionary Computation*, 4:1396–1403, 2004.
- [13] Wang Jer-Fu, Lin Chi-Chang, and Yen Shih-Min. A story damage index of seismically-excited buildings based on modal frequency and mode shape. *Engineering Structures*, 29:2143–2157, 2006.
- [14] Doebling S.W., Farrar C.R., Prime M.B., and Shevitz D.W. Damage identification and health monitoring of structural and mechanical systems from changes in their vibration characteristics: a literature review. Technical report, Los Alamos National Laboratory, 1996.
- [15] Farrar C.R. and Worden K. An introduction to structural health monitoring. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 365:303–315, 2007.
- [16] Friswell M.I., Penny J.E.T., and Wilson D.A. Using vibration data and statistical measures to locate damage in structures. *Modal Analysis: The International Journal of Analytical and Experimental Modal Analysis*, 9:239–254, 1994.
- [17] Carden Peter E. and Fanning Paul. Vibration based condition monitoring: A review. *Sage Publications*, 3:355–377, 2004.
- [18] Reynders Edwin and De Roeck Guido. A local flexibility method for vibration-based damage localization and quantification. *Journal of Sound and Vibration*, 329:2367–2383, 2010.
- [19] Friswell M.I. and Penny J.E.T. Crack modelling for structural health monitoring. *Structural Health Monitoring: An International Journal*, 1:139–148, 2002.
- [20] Rytter A. *Vibration based inspection of civil engineering structures*. PhD thesis, Denmark: Aalborg University. Department of Building Technology and Structural Engineering, 1993.
- [21] Vélez William, Gómez Daniel, and Thomson Peter. Ajuste de modelos de elementos finitos. *DYNA*, 76:177–189, 2009.
- [22] Zapico Valle J.L., Alonso Cambor R., and Gonzalez Martinez M.P. and Garcia Dieguez M. A new method for finite element model updating in structural dynamics. *Mechanical Systems and Signal Processing*, 24:2137–2159, 2010.
- [23] Li Hong-Nan, Li Dong-Sheng, and Song Gang-Bing. Recent applications of fiber optic sensors to health monitoring in civil engineering. *Engineering Structures*, 26:1647–1657, 2004.

## BIBLIOGRAFÍA

- [24] Friswell M.I. and Inman D.J. Sensor validation for smart structures. *Journal of Intelligent Material Systems and Structures*, 10:973–982, 1999.
- [25] Posenato Daniele, Kripakaran Prakash, Inaudi Daniele, and Smith Ian. Methodologies for model-free data interpretation of civil engineering structures. *Computers and Structures*, 88:467–482, 2010.
- [26] Friswell M.I. Damage identification using inverse methods. *Special Issue of the Royal Society Philosophical Transactions on Structural Health Monitoring and Damage Prognosis*, 365:393–410, 2007.
- [27] Friswell M.I., Mottershead J.E., and Ahmadian H. Finite element model updating using experimental test data: parameterisation and regularisation. *Transactions of the Royal Society of London, Series A*, 359:169–186, 2001.
- [28] Zhu Hong-ping, He Bo, and Chen Xiao-qiang. Detection of structural damage through changes in frequency. *Wuhan University Journal of Natural Sciences*, 10:1069–1073, 2005.
- [29] Gómez Bernardo. *Detección de fallas en estructuras civiles parcialmente instrumentadas a través de observadores de estado*. PhD thesis, México: Instituto Politécnico Nacional. Departamento de Control Automático, 2005.
- [30] Friswell M.I. and Mottershead J.E. *Finite element model updating in structural dynamics*. Kluwer Academic Publishers, 1995.
- [31] Maeck Johan. *Damage Assessment of Civil Engineering Structures by Vibration Monitoring*. PhD thesis, Belgium: Katholieke Universiteit Leuven, Faculteit Toegepaste Wetenschappen, Departement Burgerlijke Bouwkunde, 2003.
- [32] Begambre Oscar. *Deteção de dano a partir da resposta dinâmica da estrutura: Estudo analítico com aplicação a estruturas do tipo viga*. Master’s thesis, São Paulo: Universidade de São Paulo. Escola de Engenharia de São Carlos, 2004.
- [33] Ratcliffe Colin P. A frequency and curvature based experimental method for locating damage in structures. *Journal of Vibration and Acoustics*, 122:324–329, 2000.
- [34] Yang Q.W. and Liu J.K. Structural damage identification based on residual force vector. *Journal of sound and vibration*, 305:298–307, 2007.
- [35] Chellkni Giuseppe, De Roeck Guido, Nardini Luca, and Salvatore Walter. Damage analysis of a steel-concrete composite frame by finite element model updating. *Journal of Constructional Steel Research*, 66:398–411, 2010.
- [36] Dréo Johann, Pérowski Alain, Siarry Patrick, and Taillard Eric. *Metaheuristics for Hard Optimization Methods and Case Studies*. Springer Berlin Heidelberg New York, 2006.

## BIBLIOGRAFÍA

- [37] Villalba Jesús Daniel and Laier José Elías. Detección de daño estructural por algoritmos genéticos: una comparación de diferentes tipos de codificación de individuos. *Ingeniería y desarrollo*, 27:169–186, 2010.
- [38] Begambre O. and Laier J.E. A hybrid particle swarm optimization-simplex algorithm (psos) for structural damage identification. *Advances in Engineering Software*, 40:883–891, 2009.
- [39] Na Chaekuk, Kim Sun-Pil, and Kwak Hyo-Gyoung. Structural damage evaluation using genetic algorithm. *Journal of Sound and Vibration*, 330:2772–2783, 2011.
- [40] Nobahari M. and Seyedpoor S.M. Structural damage detection using an efficient correlation-based index and a modified genetic algorithm. *Mathematical and Computer Modelling*, 53:1798–1809, 2011.
- [41] Meruane V. and Heylen W. An hybrid real genetic algorithm to detect structural damage using modal properties. *Mechanical Systems and Signal Processing*, 25:1559–1573, 2011.
- [42] Begambre Oscar, Correa Rodrigo, and Laier José. Detección daño estructural empleando el vector de fuerza residual modificado y el algoritmo simulated annealing (sa). *Revista Ingenierías Universidad de Medellín*, 9:25–35, 2009.
- [43] He Rong-Song and Hwang Shun-Fa. Damage detection by an adaptive real-parameter simulated annealing genetic algorithm. *Computers and Structures*, 84:2231–2243, 2006.
- [44] Dyke S.J., Caicedo J.M., and Johnson E.A. Monitoring of a benchmark structure for damage identification. In *Proceedings of the 14th ASCE Engineering Mechanics Conference*, 2000.
- [45] Dyke S.J., Bernal D., Beck J.L., and Ventura C. Experimental phase ii of the structural health monitoring benchmark problem. In *Proceedings of the 16th Engineering Mechanics Conference*, 2003.
- [46] Caicedo Juan Martin. Two structural health monitoring strategies based on global acceleration responses: development, implementation, and verifivation. Master’s thesis, Saint Louis: Washington University. Department of Civil Engineering, 2001.
- [47] Bernal D., Dyke S.J., Lam H.F., and Beck J.L. Phase ii of the asce benchmark study in shm. In *Proceedings of the 15th ASCE Engineering Mechanics Conference*, 2002.
- [48] Caicedo Juan and Dyke Shirley. Solution for the second phase of the analytical shm berchmark problem. In *15th ASCE Engineering Mechanics Conference*, 2002.

## BIBLIOGRAFÍA

- [49] Barroso Lusiana and Rodriguez Ramses. Application of the damage index method to phase ii of the analytical shm benchmark problem. In *15th ASCE Engineering Mechanics Conference*, 2002.
- [50] Ching J. and Beck J.L. Two-stage bayesian structural health monitoring approach for phase ii asce benchmark studies. 2003.
- [51] Preston Robin. Lms-based method for damage detection applied to phase ii of the structural health monitoring benchmark problem. Master’s thesis, Texas Tech University, 2006.
- [52] Dharap Prasad. *Real-time structural damage detection using interaction matrix formulation and observers*. PhD thesis, Houston: Rice University, 2006.
- [53] Calderón Leandro. Determinación de propiedades dinámicas del pórtico metálico en 3d de referencia del asce: enfoque numérico. Tesis de pregrado. Universidad Industrial de Santander, 2011.
- [54] Goldberg D. E. *Genetic algorithms in search, optimization and machine learning*. Addison-Wesley Publishing Company, 1989.
- [55] Holland J.H. *Adaptation in natural and artificial systems*. MIT Press, 1992.
- [56] Kirkpatrick S., Gelatt C.D., and Vecchi M.P. Optimization by simulated annealing. *Science*, 220:671–680, 1983.
- [57] Metropolis N., Rosenbluth A., Rosenbluth M., Teller A., and Teller E. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092, 1953.
- [58] Kennedy James and Eberhart Russell. Particle swarm optimization. *Proceedings of the IEEE International Conference on Neural*, 4:1942–1948, 1995.
- [59] Clerc M. The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. *Proceedings of the Congress on Evolutionary Computation*, 3:1951–1957, 1999.
- [60] Shi Hi Yuhui and Eberhart Russell. Particle swarm optimization: developments, applications and resources. *Congress on Evolutionary Computation*, 1:81–86, 2001.
- [61] Wang Yu, Li Bin, Weise Thomas, Wang Jianyu, Yuan Bo, and Tian Qiongjie. Self-adaptive learning based particle swarm optimization. *Information Sciences*, 181:4515–458, 2011.
- [62] Montaz M., Khompatraporn C., and Zabinsky Z.B. A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems. *Journal of Global Optimization*, 31:635–672, 2005.

## BIBLIOGRAFÍA

- [63] T.P. Runarsson and X. Yao. Stochastic ranking for constrained evolutionary optimization. *IEEE Transactions on Evolutionary Computation*, 4:284–294, 2000.
- [64] Michalewicz Z. and Schoenauer M. Evolutionary algorithms for constrained parameter optimization problems. *Evolutionary Computation*, 4:1–32, 1996.
- [65] Mallipeddi R. and Suganthan P.N. Problems definitions and evaluation criteria for the cec 2010 competition on constrained real-parameter optimization. Technical report, Nanyang Technological University, 2010.
- [66] Ewings D. J. *Modal testing: theory and practice*. Research studies press ltd., 1984.
- [67] Maia M. and Silva J. *Theoretical and experimental modal analysis*. Research studies press ltd., 1997.
- [68] Dackermann Ulrike. *Vibration-based damage identification methods for civil engineering structures using artificial neural networks*. PhD thesis, Faculty of Engineering and Information Technology. University of Technology Sydney, 2010.
- [69] Ceballos Marcelo. *Análisis modal experimental aplicado a la calibración de modelos de sistemas con Interacción suelo-estructura*. PhD thesis, Facultad de ciencias exactas, físicas y naturales. Universidad nacional de cordoba, 2004.
- [70] Allemang Randall. Vibrations: Analytical and experimental modal analysis. Technical report, University of Cincinnati. Department of Mechanical, Industrial and Nuclear Engineering, 1999.
- [71] Katkhuda Hasan, Dwairi Hazim, and Shatarat Nasim. System identification of steel framed structures with semi-rigid connections. *Structural Engineering and Mechanics*, 34:351–366, 2010.
- [72] *EN 1993-1-8: Eurocode 3: Design of steel structures - Part 1-8: Design of joints*.
- [73] Esfandiari A., Bakhtiari-Nejad F., and Rahai A. Theoretical and experimental structural damage diagnosis method using natural frequencies through and improved sensitivity equation. *Mechanical Sciences*, 10:10–15, 2013.
- [74] Sotelo Fredy and Acavedo Javier. Identificación de daño en pórticos 2d empleando información modal y el método particle swarm optimization (pso). Tesis de pregrado. Universidad Industrial de Santander, 2009.
- [75] Perea Ricardo, Ruiz Antonio, and Manzano Carlos. Performance assessment of multicriteria damage identification genetic algorithms. *Computers and Structures*, 87:120–127, 2009.
- [76] Au F.T.K., Cheng Y.S., Tham L.G., and Bai Z.Z. Structural damage detection based on a micro-genetic algorithm using incomplete and noisy modal test data. *Journal of Sound and Vibration*, 259:1081–1094, 2003.

---

## FUNCIONES DE PRUEBA

---

### A.1. FUNCIONES OPTIMIZACIÓN CONTINUA SIN RESTRICCIONES

#### Función Colville

Definición

$$f_1(x) = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2 + (x_3 - 1)^2 + 90(x_3^2 - x_4)^2 \\ + 10,1((x_2 - 1)^2 + (x_4 - 1)^2) + 19,8(x_2 - 1)(x_4 - 1)$$

Espacio de búsqueda

$$-10 \leq x_i \leq 10, i = 1, \dots, 4$$

Minimo global

$$x^* = (1; 1; 1; 1) \quad f_1(x^*) = 0$$

#### Función Dixon

Definición

$$f_2(x) = (1 - x_1)^2 + (1 - x_{10})^2 + \sum_{i=1}^9 (x_i^2 - x_{i+1})^2$$

Espacio de búsqueda

$$-10 \leq x_i \leq 10, i = 1, \dots, 10$$

Minimo global

$$x^* = (1, \dots, 1) \quad f_2(x^*) = 0$$

## FUNCIONES DE PRUEBA

### Función Ackley

Definición

$$f_3(x) = 20 + \exp(1) - 20\exp\left(-\frac{1}{5}\sqrt{\left(\frac{1}{n}\sum_{i=1}^n x_i^2\right)}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^n \cos(2\pi x_i)\right)$$

Espacio de búsqueda

$$-30 \leq x_i \leq 30, i = 1, \dots, n$$

Minimo global

$$x^* = (0, \dots, 0) \quad f_3(x^*) = 0$$

### Función Sphere

Definición

$$f_4(x) = \sum_{i=1}^n x_i^2$$

Espacio de búsqueda

$$-2,56 \leq x_i \leq 5,12, i = 1, \dots, n$$

Minimo global

$$x^* = (0, \dots, 0) \quad f_4(x^*) = 0$$

### Función Griewank

Definición

$$f_5(x) = 1 + \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right)$$

Espacio de búsqueda

$$-600 \leq x_i \leq 600, i = 1, \dots, n$$

Minimo global

$$x^* = (0, \dots, 0) \quad f_5(x^*) = 0$$

### Función Rosenbrock

Definición

$$f_6(x) = \sum_{i=1}^{n-1} \left[ (1 - x_i)^2 + 100 (x_i^2 - x_{i+1})^2 \right]$$

Espacio de búsqueda

$$-10 \leq x_i \leq 10, i = 1, \dots, n$$

Minimo global

$$x^* = (1, \dots, 1) \quad f_6(x^*) = 0$$

## FUNCIONES DE PRUEBA

### Función Alpine

Definición

$$f_7(x) = \sum_{i=1}^n |x_i \sin(x_i) + 0,1x_i|$$

Espacio de búsqueda

$$-10 \leq x_i \leq 10, i = 1, \dots, n$$

Minimo global

$$x^* = (0, \dots, 0) \quad f_7(x^*) = 0$$

### Función Schwefel

Definición

$$f_8(x) = -\sum_{i=1}^n x_i \sin\left(\sqrt{|x_i|}\right)$$

Espacio de búsqueda

$$-500 \leq x_i \leq 500, i = 1, \dots, n$$

Minimo global

$$x^* = (1, \dots, 1) \quad f_8(x^*) = 418,9829n$$

### Función Rastrigin

Definición

$$f_9(x) = 10n \sum_{i=1}^n \left[ x_i^2 - 10 \cos(2\pi x_i) \right]$$

Espacio de búsqueda

$$-2,56 \leq x_i \leq 5,12, i = 1, \dots, n$$

Minimo global

$$x^* = (0, \dots, 0) \quad f_9(x^*) = 0$$



## A.2. FUNCIONES OPTIMIZACIÓN CONTINUA CON RESTRICCIONES

### Problema g01

Definición

$$f_1(x) = 5 \sum_{i=1}^4 x_i - 5 \sum_{i=1}^4 x_i^2 - \sum_{i=5}^{13} x_i$$

Sujeto a

$$g_1(x) = 2x_1 + 2x_2 + x_{10} + x_{11} - 10 \leq 0$$

$$g_2(x) = 2x_1 + 2x_3 + x_{10} + x_{12} - 10 \leq 0$$

$$g_3(x) = 2x_2 + 2x_3 + x_{11} + x_{12} - 10 \leq 0$$

$$g_4(x) = -8x_1 + x_{10} \leq 0$$

$$g_5(x) = -8x_2 + x_{11} \leq 0$$

$$g_6(x) = -8x_3 + x_{12} \leq 0$$

$$g_7(x) = -2x_4 - x_5 + x_{10} \leq 0$$

$$g_8(x) = -2x_6 - x_7 + x_{11} \leq 0$$

$$g_9(x) = -2x_8 - x_9 + x_{12} \leq 0$$

Espacio de búsqueda

$$0 \leq x_i \leq 1, i = 1, \dots, 9$$

$$0 \leq x_i \leq 100, i = 10, 11, 12$$

$$0 \leq x_{13} \leq 1, i = 1, \dots, 9$$

Minimo global

$$x^* = (1; 1; 1; 1; 1; 1; 1; 1; 1; 3; 3; 3; 1) \quad f_1(x^*) = 15$$

### Problema g02

Definición

$$f_2(x) = - \left| \frac{\sum_{i=1}^4 \cos^4(x_i) - 2 \prod_{i=1}^n \cos^2(x_i)}{\sqrt{\sum_{i=1}^n i x_i^2}} \right|$$

Sujeto a

$$g_1(x) = 0,75 - \prod_{i=1}^n x_i \leq 0$$

## FUNCIONES DE PRUEBA

$$g_2(x) = \sum_{i=1}^n -7,5n \leq 0$$

Espacio de búsqueda

$$0 \leq x_i \leq 10, i = 1, \dots, 20$$

Minimo global

$$f_1(x^*) = -0,803619$$

### Problema g03

Definición

$$f_3(x) = -(\sqrt{n})^n \prod_{i=1}^n x_i$$

Sujeto a

$$h(x) = \sum_{i=1}^n x_i^2 - 1 = 0$$

Espacio de búsqueda

$$0 \leq x_i \leq 1, i = 1, \dots, 10$$

Minimo global

$$x^* = 1/\sqrt{n} (i = 1, 2, \dots, n) \quad f_1(x^*) = 1$$

### Problema g04

Definición

$$f_4(x) = 5,3578547x_3^3 + 0,8356891x_1x_5 + 37,293239x_1 - 40792,141$$

Sujeto a

$$g_1(x) = 85,334407 + 0,0056858x_2x_5 + 0,0006262x_1x_4 - 0,0022053x_3x_5 - 92 \leq 0$$

$$g_2(x) = -85,334407 - 0,0056858x_2x_5 - 0,0006262x_1x_4 + 0,0022053x_3x_5 \leq 0$$

$$g_3(x) = 80,51249 + 0,0071317x_2x_5 + 0,0029955x_1x_2 + 0,0021813x_3^2 - 110 \leq 0$$

$$g_4(x) = -80,51249 - 0,0071317x_2x_5 - 0,0029955x_1x_2 - 0,0021813x_3^2 + 90 \leq 0$$

$$g_5(x) = 9,300961 + 0,0047026x_3x_5 + 0,0012547x_1x_3 + 0,0019085x_3x_4 - 25 \leq 0$$

$$g_6(x) = -9,300961 - 0,0047026x_3x_5 - 0,0012547x_1x_3 - 0,0019085x_3x_4 + 20 \leq 0$$

Espacio de búsqueda

$$78 \leq x_1 \leq 102$$

$$33 \leq x_2 \leq 45$$

$$27 \leq x_i \leq 45, i = 3, 4, 5$$

Minimo global

$$x^* = (78; 33; 29,995256025682; 45; 36,775812905788) \quad f_1(x^*) = -30665,539$$

## FUNCIONES DE PRUEBA

### Problema g05

Definición

$$f_5(x) = 3x_1 + 0,000001x_1^3 + 2x_2 + (0,000002/3)x_2^3$$

Sujeto a

$$g_1(x) = -x_4 + x_3 - 0,55 \leq 0$$

$$g_2(x) = -x_3 + x_4 - 0,55 \leq 0$$

$$h_3(x) = 1000\sin(-x_3 - 0,25) + 1000\sin(-x_4 - 0,25) + 894,8 - x_1 = 0$$

$$h_4(x) = 1000\sin(x_3 - 0,25) + 1000\sin(x_3 - x_4 - 0,25) + 894,8 - x_2 = 0$$

$$h_5(x) = 1000\sin(x_4 - 0,25) + 1000\sin(x_4 - x(3, :) - 0,25) + 1294,8 = 0$$

Espacio de búsqueda

$$0 \leq x_1 \leq 1200$$

$$0 \leq x_2 \leq 1200$$

$$-0,55 \leq x_3 \leq 0,55$$

$$-0,55 \leq x_4 \leq 0,55$$

Minimo global

$$x^* = (679,94453; 1026,067; 0,1188764; -0,3962336) \quad f_1(x^*) = 5126,4981$$

### Problema g06

Definición

$$f_6(x) = (x_1 - 10)^3 + (x_2 - 20)^3$$

Sujeto a

$$g_1(x) = -(x_1 - 5)^2 - (x_2 - 5)^2 + 100 \leq 0$$

$$g_1(x) = (x_1 - 6)^2 + (x_2 - 5)^2 - 82,81 \leq 0$$

Espacio de búsqueda

$$0 \leq x_1 \leq 100$$

$$0 \leq x_2 \leq 100$$

Minimo global

$$x^* = (14,095; 0,84296) \quad f_1(x^*) = -6961,81381$$

## FUNCIONES DE PRUEBA

### Problema g07

Definición

$$f_6(x) = x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + (x_3 - 10)^2 + 4(x_4 - 5)^2 + (x_5 - 3)^2 \\ + 2(x_6 - 1)^2 + 5x_7^2 + 7(x_8 - 11)^2 + 2(x_9 - 10)^2 + (x_{10} - 7)^2 + 45$$

Sujeto a

$$g_1(x) = -105 + 4x_1 + 5x_2 - 3x_7 + 9x_8 \leq 0$$

$$g_2(x) = 10x_1 - 8x_2 - 17x_7 + 2x_8 \leq 0$$

$$g_3(x) = -8x_1 + 2x_2 + 5x_9 - 2x_{10} - 12 \leq 0$$

$$g_4(x) = 3(x_1 - 2)^2 + 4(x_2 - 3)^2 + 2x_7^2 - 7x_4 - 12 \leq 0$$

$$g_5(x) = 5x_1^2 + 8x_2 + (x_3 - 6)^2 - 2x_4 - 40 \leq 0$$

$$g_6(x) = x_1^2 + 2(x_2 - 2)^2 - 2x_1x_2 + 14x_5 - 6x_6 \leq 0$$

$$g_7(x) = 0,5(x_1 - 8)^2 + 2(x_2 - 4)^2 + 3x_5^2 - x_6 - 30 \leq 0$$

$$g_8(x) = -3x_1 + 6x_2 + 12(x_9 - 8)^2 - 7x_{10} \leq 0$$

Espacio de búsqueda

$$-10 \leq x_i \leq 10, i = 3, \dots, 10$$

Minimo global

$$x^* = (2,171996; 2,363686; 8,773926; 5,095984; 0,9906548; 1,430574; 1,321644; \\ 9,828726; 8,0280092; 8,375927) \quad f_1(x^*) = 24,3062091$$

### Problema g08

Definición

$$f_8(x) = -\frac{\sin(2\pi x_1)}{x_1^3(x_1 + x_2)}$$

Sujeto a

$$g_1(x) = x_1^2 - x_2 + 1$$

$$g_2(x) = 1 - x_1 + (x_2 - 4)^2$$

Espacio de búsqueda

$$0 \leq x_1 \leq 10$$

$$0 \leq x_2 \leq 10$$

Minimo global

$$x^* = (1,2279713; 4,2453733) \quad f_1(x^*) = 0,095825$$

## FUNCIONES DE PRUEBA

### Problema g09

Definición

$$f_9(x) = (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 + 10x_5^6 + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7$$

Sujeto a

$$g_1(x) = -127 + 2x_1^2 + 3x_2^4 + x_3 + 4x_4^2 + 5x_4 \leq 0$$

$$g_2(x) = -282 + 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 \leq 0$$

$$g_3(x) = -196 + 23x_1 + x_2^2 + 6x_6^2 - 8x_7 \leq 0$$

$$g_4(x) = 4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7 \leq 0$$

Espacio de búsqueda

$$-10 \leq x_i \leq 10, i = 7, \dots, 10$$

Minimo global

$$x^* = (2,330499; 1,951372; -0,4775414; 4,365726; -0,6244870; 1,038131; 1,594227)$$

$$f_1(x^*) = 680,6300573$$

### Problema g10

Definición

$$f_{10}(x) = x_1 + x_2 + x_3$$

Sujeto a

$$g_1(x) = -1 + 0,0025(x_4 + x_6) \leq 0$$

$$g_2(x) = -1 + 0,0025(x_5 + x_7 - x_4) \leq 0$$

$$g_3(x) = -1 + 0,01(x_8 - x_5) \leq 0$$

$$g_4(x) = -x_1x_6 + 833,33252x_4 + 100x_1 - 83333,33 \leq 0$$

$$g_5(x) = -x_2 * x_7 + 1250x_5 + x_2x_4 - 1250x_4 \leq 0$$

$$g_6(x) = -x_3x_8 + 1250000 + x_3x_5 - 2500x_5 \leq 0$$

Espacio de búsqueda

$$100 \leq x_1 \leq 10000$$

$$1000 \leq x_i \leq 10000, i = 2, 3$$

$$10 \leq x_i \leq 1000, i = 4, \dots, 8$$

Minimo global

$$x^* = (579,3167; 1359,943; 5110,071; 182,0174; 295,5985; 217,9799; 286,4162; 395,5979)$$

$$f_1(x^*) = 7049,3307$$

## FUNCIONES DE PRUEBA

### Problema g11

Definición

$$f_{11}(x) = x_1^2 + (x_2 - 1)^2$$

Sujeto a

$$g_x(x) = x_2 - x_1^2 = 0$$

Espacio de búsqueda

$$-1 \leq x_1 \leq 1$$

$$-1 \leq x_2 \leq 1$$

Minimo global

$$x^* = (\pm\sqrt{2}; 1/2) \quad f_1(x^*) = 0,75$$

### Problema g12

Definición

$$f_{12}(x) = \frac{-\left(100 - (x_1 - 5)^2 - (x_2 - 5)^2 - (x_3 - 5)^2\right)}{100}$$

Sujeto a

$$g_1(x) = (x_1 - p)^2 + (x_2 - q)^2 + (x_3 - r)^2 - 0,0625 \leq 0$$

Espacio de búsqueda

$$0 \leq x_i \leq 1, i = 1, 2, 3 \quad y \quad p, q, r = 1, 2, \dots, 9$$

Minimo global

$$x^* = (5; 5; 5) \quad f_1(x^*) = 1$$

### Problema g13

Definición

$$f_{12}(x) = e^{x_1 x_2 x_3 x_4 x_5}$$

Sujeto a

$$h_1(x) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 - 10 = 0$$

$$h_2(x) = x_2 x_3 - 5 x_4 x_5 = 0$$

$$h_3(x) = x_1^3 + x_2^3 + 1 = 0$$

Espacio de búsqueda

$$-2,3 \leq x_i \leq 2,3, i = 1, 2$$

$$-3,2 \leq x_i \leq 3,2, i = 3, 4, 5$$

Minimo global

$$x^* = (-1,717143; 1,595709; 1,827247; -0,7636413; -0,763645) \quad f_1(x^*) = 0,0539498$$

---

## ALGORITMOS DE OPTIMIZACIÓN

---

### B.1. ALGORITMOS PARA OPTIMIZACIÓN CONTINUA

#### B.1.1. PSO

```

1  %%
2  clc;clear all;format compact;
3
4  %%
5  disp('*****')
6  disp('*****')
7  disp('***** PSO *****')
8  disp('*****')
9  disp('*****')
10
11  %% NUMERO DE PRUEBAS A REALIZAR
12  ni=input('Ingrese el numero de pruebas a realizar = ');
13
14  for q=1:1:ni
15
16      %% DATOS DE ENTRADA
17      fun=input('Ingrese la funcion a evaluar = ');
18      ps=input('Ingrese el # de particulas = ');
19      E=input('Desviacion Standar admisible = ');
20      rd=input('Numero de datos para determinar la Desviacion Standar = ');
21      mite=input('Numero maximo de iteraciones = ');
22
23      %% LIMITES DE LAS FUNCIONES
24      dim=[4 10 30 30 30 30 30 30 5];lim=[-10 10;-10 10;-30 30;-20 20;...
25          -600 600;-10 10;-10 10;-500 500;-5.12 5.12];
26
27

```

## ALGORITMOS DE OPTIMIZACIÓN

```

28     d=dim(fun);           %-----DIMENSIONES
29
30     a=lim(fun,1); %-----LIMITE INFERIOR
31     b=lim(fun,2); %-----LIMITE SUPERIOR
32
33     %% PARAMETROS DEL ALGORITMO
34     p=abs((b-a))/5; %-----PASO MAXIMO PERMITIDO PARA LA ...
        VELOCIDAD
35     c1=2; %-----COEFICIENTE DE CONFIANZA LOCAL
36     c2=2; %-----COEFICIENTE DE CONFIANZA GLOBAL
37     w=1; %-----PESO DE INERCIA
38
39     %% CONTADOR DE TIEMPO
40     tic;
41
42     %% DISTRIBUCION DE LA VELOCIDAD(vi) Y LA POSICION(xi) INICIAL
43     [xi,vi]=dis_inic(ps,d,a,b,p);
44
45     %% CALCULO DE LA FUNCION OBJETIVO
46     [fobj]=fittnes(fun,xi,d,ps);
47
48     %% SELECCION DE LOS MEJORES LOCALES
49     pbest=fobj;
50     coord_pbest=xi;
51
52     %% ELECCION DEL MEJOR GLOBAL
53     gbest=min(pbest);
54     [coord_gbest]=coord_cgbest(ps,gbest,pbest,coord_pbest);
55
56     %% PARAMEROS PARA EL CONTROL DE LA EJECUCION DEL ALGORITMO\
57     e=0;
58     des=10;
59
60     %% EJECUCION DEL ALGORITMO
61     while e<mite && des>E
62         e=e+1;
63         x0=xi;
64         v0=vi;
65         vi=[];
66
67         %% ALCULO VELOCIDAD vi
68         r1=rand(ps,d);
69         r2=rand(ps,d);
70
71         for k=1:1:ps
72             vi(k,:)=w*v0(k,:)+c1*r1(k,:).*(coord_pbest(k,:)-x0(k,:))+...
73                 c2*r2(k,:).*(coord_gbest-x0(k,:));
74         end
75
76         %% CHEQUEO DE LA VELOCIDAD
77         [vi]=check_v(vi,p);
78

```



## ALGORITMOS DE OPTIMIZACIÓN

```

79         %% CALCULO DE xi A PARTIR DE vi
80         xi=x0+vi;
81
82         %% CHEQUEO DE LA POSICION
83         [xi]=check_x(xi,a,b);
84
85         %% CALCULO FUNCION OBJETIVO
86         [fobj]=fittnes(fun,xi,d,ps);
87
88         %% SELECCION gbest,pbest,coord_pbest Y coord_gbest
89         [gbest,pbest,coord_pbest]=select_pbest(ps,pbest,fobj,xi,coord_pbest);
90         [coord_gbest]=coord_cgbest(ps,gbest,pbest,coord_pbest);
91
92         %% REDUCCION LINEAL DE w
93         w=w*0.98;
94
95         %% ALMACENAMIENTO DE DATOS PARA DETERMINAR EL CRITERIO DE PARADA
96         ds(e)=gbest;
97
98         %% CALCULO DEL CRITERIO DE PARADA
99         if e>rd
100             des=std(ds(1,e-rd:e));
101         end
102     end
103
104     %% ALMACENAMIENTO DE LOS RESULTADOS
105     tabla(q,1)=toc; %-----tiempo
106     tabla(q,2)=e; %-----iteraciones
107     tabla(q,3)=gbest; %-----mejor global
108     CGlobal(q,:)=coord_gbest; %---puntos mejor global
109     Convergencia(:,q)=ds; %-----convergencia
110 end
111
112 resultados(1)=mean(tabla(:,1)); %-----tiempo promedio
113 resultados(2)=mean(tabla(:,2)); %-----promedio de iteraciones
114 resultados(3)=max(tabla(:,3)); %-----peor gbest
115 resultados(4)=min(tabla(:,3)); %-----mejor gbest
116 resultados(5)=mean(tabla(:,3)); %-----promedio gbest
117 resultados(7)=median(tabla(:,3)); %-----mediana gbest
118 resultados(6)=std(tabla(:,3)); %-----desviacion estanadr gbest
119
120
121 switch fun
122     case 1 %-----FUNCION COLVILLE (4D)
123         save('COLVILLE','tabla','CGlobal','Convergencia','resultados')
124     case 2 %-----FUNCION DIXON (10D)
125         save('DIXON','tabla','CGlobal','Convergencia','resultados')
126     case 3 %-----FUNCION ACKLEY (ND)
127         save('ACKLEY','tabla','CGlobal','Convergencia','resultados')
128     case 4 %-----FUNCION DE SPHERE (ND)
129         save('SPHERE','tabla','CGlobal','Convergencia','resultados')
130     case 5 %-----FUNCION GRIEWANK (ND)

```

## ALGORITMOS DE OPTIMIZACIÓN

```

131         save('GRIEWANK','tabla','CGlobal','Convergencia','resultados')
132     case 6 %-----FUNCION ROSENBROCK (ND)
133         save('ROSENBROCK','tabla','CGlobal','Convergencia','resultados')
134     case 7 %-----FUNCION ALPINE (ND)
135         save('ALPINE','tabla','CGlobal','Convergencia','resultados')
136     case 8 %-----FUNCION SCHWEFEL (ND)
137         save('SCHWEFEL','tabla','CGlobal','Convergencia','resultados')
138     case 9 %-----FUNCION RASTRIGIN (ND)
139         save('RASTRIGIN','tabla','CGlobal','Convergencia','resultados')
140 end
141
142 clear tabla CGlobal Convergencia resultados

```

```

1 function [vi]=check_v(vi,p)
2
3 c=(-p≤vi) & (vi≤p); %ones=estan dentro del intervalo
4 m=vi.*c;
5 pd=not(c);
6 t=pd.*vi; %ones=estén por fuera del intervalo
7 e1=-p*(t≤-p);
8 e2=p*(t≥p);
9 vi=m+e1+e2;
10
11 % for k=1:1:ps
12 %     for h=1:1:d
13 %         if vi(k,h)>p
14 %             vi(k,h)=p;
15 %         elseif vi(k,h)<-p
16 %             vi(k,h)=-p;
17 %         end
18 %     end
19 % end

```

```

1 function [xi]=check_x(xi,a,b)
2
3 c=(a≤xi) & (xi≤b); %ones=estan dentro del intervalo
4 m=xi.*c;
5 p=not(c);
6 t=p.*xi; %ones=estén por fuera del intervalo
7 e1=a*(t≤a);
8 e2=b*(t≥b);
9 xi=m+e1+e2;
10
11 %x1=(xf<b) & (xf>a)
12 %xf2=xf.*x1
13 %xln=not(x1)
14 %xt=xln.*xf
15 %x1=a*(xt<0)
16 %x2=b*(xt>0)

```

## ALGORITMOS DE OPTIMIZACIÓN

```

17 %xf3=xf2+x1+x2
18
19 % %for k=1:1:ps
20 %     for h=1:1:d
21 %         if xi(k,h)>b
22 %             xi(k,h)=b;
23 %         elseif xi(k,h)<a
24 %             xi(k,h)=a;
25 %         end
26 %     end
27 % end

1 function [coord_gbest]=coord_cgbest(ps,gbest,pbest,coord_pbest)
2 for k=1:1:ps
3     if gbest==pbest(k,1)
4         coord_gbest=coord_pbest(k,:);
5         break
6     end
7 end
8 end

1 function [xi,vi]=dis_inic(ps,d,a,b,p)
2
3 xi=a+rand(ps,d)*(b-a);
4 vi=a+rand(ps,d)*(b-a);
5
6 [vi]=check_v(vi,p);

1 function [fobj]=fittnes(fun,xi,d,ps)
2
3 switch fun
4
5     case 1 %-----FUNCION COLVILLE (4D)
6
7         fobj=100*(xi(:,1).^2-xi(:,2)).^2+(xi(:,1)-1).^2+...
8             (xi(:,3)-1).^2+90*(xi(:,3).^2-xi(:,4)).^2 ...
9             +10.1*((xi(:,2)-1).^2+(xi(:,4)-1).^2)+...
10            19.8*(xi(:,2)-1).*(xi(:,4)-1);
11
12     case 2 %-----FUNCION DIXON (10D)
13
14         fobj=(1-xi(:,1)).^2+(1-xi(:,10)).^2+(sum((xi(:,1:9)).*xi(:,1:9))'-...
15             xi(:,2:10)').^2))';
16
17     case 3 %-----FUNCION ACKLEY (ND)
18
19         fobj=(20+exp(1)-20*exp(-1/5*sqrt((1/d)*sum(xi'.*xi'))))-...
20             exp((1/d)*sum(cos((2*pi()*xi'))))';

```

## ALGORITMOS DE OPTIMIZACIÓN

```

21
22     case 4 %-----FUNCION DE SPHERE (ND)
23
24         fobj=(sum(xi' .*xi'))';
25
26     case 5 %-----FUNCION GRIEWANK (ND)
27
28         xj=ones(ps,d);
29
30         for j=1:1:d
31             xj(:,j)=xj(:,j)*j;
32         end
33
34         fobj=(1+1/4000*(sum(xi' .*xi'))'-(prod(cos(xi'./sqrt(xj')))))';
35
36     case 6 %-----FUNCION ROSENBROCK (ND)
37
38         fobj=(sum((1-xi(:,1:d-1))'.^2+100*(xi(:,1:d-1)' .*xi(:,1:d-1)'.^2)...
39             xi(:,2:d)')'.^2))';
40
41     case 7 %-----FUNCION ALPINE (ND)
42
43         fobj=sum(abs(xi' .*sin(xi')+0.1*xi'))';
44
45     case 8 %-----FUNCION SCHWEFEL (ND)
46
47         fobj=((xi' .*sin(sqrt(abs(xi))))')';
48         fobj=-sum(fobj)';
49
50     case 9 %-----FUNCION RASTRIGIN (ND)
51
52         fobj=(xi'.^2-10*cos(2*pi()*xi))';
53         fobj=10*d+(sum(fobj))';
54
55
56 end
57 end

1  function ...
2      [gbest,pbest,coord_pbest]=select_pbest(ps,pbest,fobj,xi,coord_pbest)
3  for k=1:1:ps
4      if fobj(k,1)<pbest(k,1)
5          pbest(k,1)=fobj(k,1);
6          coord_pbest(k,:)=xi(k,:);
7      else
8          pbest(k,1)=pbest(k,1);
9          coord_pbest(k,:)=coord_pbest(k,:);
10     end
11 end
12 gbest=min(pbest);

```

## ALGORITMOS DE OPTIMIZACIÓN

### B.1.2. SL-PSO

```
1  %%
2  clc;clear all;format compact;
3
4  %%
5  disp('*****')
6  disp('*****')
7  disp('***** SL-PSO *****')
8  disp('*****')
9  disp('*****')
10
11  %% NUMERO DE PRUEBAS A REALIZAR
12  ni=input('Ingrese el numero de pruebas a realizar = ');
13
14  for q=1:1:ni
15
16      %% DATOS DE ENTRADA
17      fun=input('Ingrese la funcion a evaluar = ');
18      ps=input('Ingrese el # de particulas = ');
19      E=input('Desviacion Standar admisible = ');
20      rd=input('Numero de datos para determinar la Desviacion Standar = ');
21      max_ite=input('Numero maximo de iteraciones = ');
22
23      %% LIMITES DE LAS FUNCIONES Y DIMENSIONES
24      dim=[4 10 30 30 30 30 30 30 5];
25      lim=[-10 10;-10 10;-30 30;-20 20;-600 600;-10 10;-10 10;-500 ...
26          500;-5.12...
27          5.12];
28      d=dim(fun); %-----DIMENSIONES
29
30      a=lim(fun,1); %-----LIMITE INFERIOR
31      b=lim(fun,2); %-----LIMITE SUPERIOR
32
33      %% PARAMETROS DEL ALGORITMO
34      p=abs((b-a)); %-----PASO MAXIMO PERMITIDO ...
35          PARA LA VELOCIDAD
36      c3=1.49445; %-----COEFICIENTE ESTRATEGIA ...
37          N-3 y N-4 (SELECCIONADO DEL CLPSO ORIGINAL)
38      w0=0.9; %-----PESO DE INERCIA ...
39          ESTRATEGIA N-3 Y N-4
40
41      Gs=100; %-----PERIODO DE APRENDIZAJE
42      alpha=1/6; %-----COEFICIENTE DE APRENDIZAJE
43      proSTR(1:3)=1/3; %-----ASIGNACION DE ...
44          PROBABILIDAD INICIAL PARA CADA ESTRTEGIA DE ACTUALIZACION DE V
45      S(1:3)=0; %-----ACUMULADOR DE LAS CUATRO ...
46          ESTRATEGIAS
```

## ALGORITMOS DE OPTIMIZACIÓN

```

43     sumlog=0;
44     for k=1:1:ps
45         sumlog=log(k)+sumlog;
46     end
47     w=(1:ps);
48     w=ps-w+1;
49     w=log(w) ./sumlog;
50
51     Pci=(1:ps);
52     Pci=0.05+0.45*((exp((10.*(Pci-1))/(ps-1))-1)/(exp(10)-1));
53
54     %% COANTADOR DE TIEMPO
55     tic;
56
57     %% DISTRIBUCION DE LA VELOCIDAD(vi) Y LA POSICION(xi) INICIAL
58     [xi,vi]=dis_inic(ps,d,a,b,p);
59
60     %% ASIGNACION DE LA ESTRATEGIA PARA v
61     strategy_pbest=round(rand(ps,1).*(3-1)+1);
62
63     %% CALCULO DE LA FUNCION OBJETIVO
64     [fobj]=fittnes(fun,xi,d,ps);
65
66     %% SELECCION DE LOS MEJORES LOCALES
67     pbest=fobj;
68     coord_pbest=xi;
69
70     %% ELECCION DEL MEJOR GLOBAL
71     gbest=min(pbest);
72     [coord_gbest]=coord_cgbest(ps,gbest,pbest,coord_pbest);
73
74     %% PARAMEROS PARA EL CONTROL DE LA EJECUCION DEL ALGORITMO
75     e=0; %-----CONTADOR
76     des=10; %-----DESVIACION ESTANDAR
77     driver=0; %-----CONTROLADOR PARA EL PROCESO ...
78         DE OPTIMIZACION
79
80     %% EJECUCION DEL ALGORITMO
81     while e<max_ite && des>E
82         %% DETERMINACION DE LA VELOCIDAD
83         e=e+1;
84         x0=xi;
85         v0=vi;
86
87         if e>(Gs*driver)
88             na=rand();
89             driver=driver+1;
90         end
91
92         if na<proSTR(1)
93             %% ACTUALIZACION DE v POR MEDIO DE LA ESTRATEGIA 1
94             [vi,strategy]=calc_DbV(ps,x0,coord_pbest);

```

## ALGORITMOS DE OPTIMIZACIÓN

```

94
95     elseif na<(proSTR(1)+proSTR(2))
96         %% ACTUALIZACION DE v POR MEDIO DE LA ESTRATEGIA 3
97         [vi,strategy]=calc_CLPSO(ps,d,coord_pbest,v0,x0,c3,w0,Pci,...
98             e,max_ite,pbest);
99
100    elseif na<(proSTR(1)+proSTR(2)+proSTR(3))
101        %% ACTUALIZACION DE v POR MEDIO DE LA ESTRATEGIA 4
102        [vi,strategy]=calc_PSO_CL_pbest(ps,d,coord_pbest,v0,x0,c3,...
103            w0,Pci,e,max_ite,pbest);
104    end
105
106    %% CHEQUEO DE LA VELOCIDAD
107    [vi]=check_v(vi,p);
108
109    %% CALCULO DE xi A PARTIR DE vi
110    xi=x0+vi;
111
112    %% CHEQUEO DE LA POSICION
113    [xi]=check_x(xi,a,b);
114
115    %% CALCULO FUNCION OBJETIVO
116    [fobj]=fittnes(fun,xi,d,ps);
117
118    %% SELECCION gbest,pbest,S,coord_pbest Y coord_gbest
119    [gbest,pbest,S,coord_pbest,strategy_pbest]=select(ps,pbest,...
120        fobj,strategy,strategy_pbest,xi,coord_pbest,S,w,e,Gs,driver);
121    [coord_gbest]=coord_cgbest(ps,gbest,pbest,coord_pbest);
122
123    %% ASIGNACION DE LA PROBABILIDAD PARA LA SELECCION DE LA ...
124    VELOCIDAD
125    if e==(Gs*driver)
126        proSTRi=(1-alpha)*proSTR+alpha*(S./Gs);
127        S(1:3)=0;
128        sum_proSTRi=sum(proSTRi);
129        proSTR=proSTRi./sum_proSTRi;
130    end
131
132    %% ALMACENAMIENTO DE DATOS PARA DETERMINAR EL CRITERIO ...
133    DE PARADA
134    ds(e)=gbest;
135
136    %% CALCULO DEL CRITERIO DE PARADA
137    if e>rd
138        des=std(ds(1,e-rd:e));
139    end
140
141    end
142
143    %% ALMACENAMIENTO DE LOS RESULTADOS
144    tabla(q,1)=toc; %-----tiempo
145    tabla(q,2)=e; %-----iteraciones

```

## ALGORITMOS DE OPTIMIZACIÓN

```

144         tabla(q,3)=gbest;%-----mejor global
145         CGlobal(q,:)=coord_gbest;%---puntos mejor global
146         Convergencia(:,q)=ds;%-----convergencia
147
148     end
149     resultados(1)=mean(tabla(:,1));%-----tiempo promedio
150     resultados(2)=mean(tabla(:,2));%-----promedio de iteraciones
151     resultados(3)=max(tabla(:,3));%-----peor gbest
152     resultados(4)=min(tabla(:,3));%-----mejor gbest
153     resultados(5)=mean(tabla(:,3));%-----promedio gbest
154     resultados(6)=std(tabla(:,3));%-----desviacion estandar gbest
155     resultados(7)=median(tabla(:,3));%-----medians gbest
156
157
158     switch fun
159     case 1 %-----FUNCION COLVILLE (4D)
160         save('COLVILLE','tabla','CGlobal','Convergencia','resultados')
161     case 2 %-----FUNCION DIXON (10D)
162         save('DIXON','tabla','CGlobal','Convergencia','resultados')
163     case 3 %-----FUNCION ACKLEY (ND)
164         save('ACKLEY','tabla','CGlobal','Convergencia','resultados')
165     case 4 %-----FUNCION DE SPHERE (ND)
166         save('SPHERE','tabla','CGlobal','Convergencia','resultados')
167     case 5 %-----FUNCION GRIEWANK (ND)
168         save('GRIEWANK','tabla','CGlobal','Convergencia','resultados')
169     case 6 %-----FUNCION ROSENBROCK (ND)
170         save('ROSENBROCK','tabla','CGlobal','Convergencia','resultados')
171     case 7 %-----FUNCION ALPINE (ND)
172         save('ALPINE','tabla','CGlobal','Convergencia','resultados')
173     case 8 %-----FUNCION SCHWEFEL (ND)
174         save('SCHWEFEL','tabla','CGlobal','Convergencia','resultados')
175     case 9 %-----FUNCION RASTRIGIN (ND)
176         save('RASTRIGIN','tabla','CGlobal','Convergencia','resultados')
177     end
178
179     clear tabla CGlobal Convergencia resultados

1 function [vi,strategy]=calc_CLPSO(ps,d,coord_pbest,v0,x0,c3,w0,Pci,e,...
2     max_ite,pbest)
3     vi=[];
4     w=(w0-0.2)*(max_ite-e)/(max_ite)+0.2;
5     for i=1:1:ps
6         for j=1:1:d
7             na=rand;
8             if na<Pci(i)
9                 fd1=0;
10                while fd1==0
11                    xk=round(rand()*(ps-1)+1);
12                    if xk==i
13                        fd1=0;

```



## ALGORITMOS DE OPTIMIZACIÓN

```

14         else
15             fd1=1;
16         end
17     end
18
19     fd2=0;
20     while fd2==0
21         xj=round(rand()*(ps-1)+1);
22         if xj==i
23             fd2=0;
24         else
25             fd2=1;
26         end
27     end
28
29     if pbest(xk,1)<pbest(xj,1)
30         pbestFD(i,j)=coord_pbest(xk,j);
31     else
32         pbestFD(i,j)=coord_pbest(xj,j);
33     end
34 else
35     pbestFD(i,j)=coord_pbest(i,j);
36 end
37 end
38 end
39
40 r1=rand(ps,d);
41 vi=w*v0+c3*r1.*(pbestFD-x0);
42
43 strategy=2;

1 function [vi,strategy]=calc_DbV(ps,x0,coord_pbest)
2 cl=normrnd(0.5,0.2); %-----COEFICIENTE ESTRATEGIA N-1
3 vi=[];
4 for k=1:1:ps
5     xk=round(rand()*(ps-1)+1);
6     xj=round(rand()*(ps-1)+1);
7     Viad=x0(xk,:)-x0(xj,:);
8     vi(k,:)=cl.*Viad+cl.*(coord_pbest(k,:)-x0(k,:));
9 end
10
11 strategy=1;

1 function ...
2     [vi,strategy]=calc_PSO-CL-pbest(ps,d,coord_pbest,v0,x0,c3,w0,Pci,...
3     e,max_ite,pbest)
4 vi=[];
5 vi=[];
6 w=(w0-0.2)*(max_ite-e)/(max_ite)+0.2;

```

## ALGORITMOS DE OPTIMIZACIÓN

```

6  for i=1:1:ps
7      for j=1:1:d
8          na=rand;
9          if na<Pci(i)
10             fd1=0;
11             while fd1==0
12                 xk=round(rand()*(ps-1)+1);
13                 if xk==i
14                     fd1=0;
15                 else
16                     fd1=1;
17                 end
18             end
19
20             fd2=0;
21             while fd2==0
22                 xj=round(rand()*(ps-1)+1);
23                 if xj==i
24                     fd2=0;
25                 else
26                     fd2=1;
27                 end
28             end
29
30             if pbest(xk,1)<pbest(xj,1)
31                 pbestFD(i,j)=coord_pbest(xk,j);
32             else
33                 pbestFD(i,j)=coord_pbest(xj,j);
34             end
35         else
36             pbestFD(i,j)=coord_pbest(i,j);
37         end
38     end
39 end
40
41 r1=rand(ps,1);
42
43 for h=1:1:d
44     r2(:,h)=r1(:,1);
45 end
46
47 vi=w*v0+0.5*c3*r2.*(pbestFD-x0+coord_pbest-x0());
48
49 strategy=3;

1  function [vi]=check_v(vi,p)
2
3  c=(-p≤vi) & (vi≤p);
4  m=vi.*c;
5  pd=not(c);

```

## ALGORITMOS DE OPTIMIZACIÓN

```
6 t=pd.*vi;
7 e1=-p*(t<=-p);
8 e2=p*(t>=p);
9 vi=m+e1+e2;
```

```
1 function [xi]=check_x(xi,a,b)
2
3 c=(a<=xi) & (xi<=b);
4 m=xi.*c;
5 p=not(c);
6 t=p.*xi;
7 e1=a*(t<=a);
8 e2=b*(t>=b);
9 xi=m+e1+e2;
```

```
1 function [coord_gbest]=coord_cgbest(ps,gbest,pbest,coord_pbest)
2 for k=1:1:ps
3     if gbest==pbest(k,1)
4         coord_gbest=coord_pbest(k,:);
5         break
6     end
7 end
8 end
```

```
1 function [xi,vi]=dis_inic(ps,d,a,b,p)
2
3 xi=a+rand(ps,d)*(b-a);
4 vi=a+rand(ps,d)*(b-a);
5
6 [vi]=check_v(vi,p);
```

```
1 function [fobj]=fittnes(fun,xi,d,ps)
2
3 switch fun
4
5     case 1 %-----FUNCION COLVILLE (4D)
6
7         fobj=100*(xi(:,1).^2-xi(:,2)).^2+(xi(:,1)-1).^2+(xi(:,3)-1).^2+...
8             90*(xi(:,3).^2-xi(:,4)).^2+10.1*((xi(:,2)-1).^2+...
9             (xi(:,4)-1).^2)+19.8*(xi(:,2)-1).*(xi(:,4)-1);
10
11     case 2 %-----FUNCION DIXON (10D)
12
13         fobj=(1-xi(:,1)).^2+(1-xi(:,10)).^2+(sum((xi(:,1:9)).*xi(:,1:9)).^2+...
14             xi(:,2:10)).^2));
15
16     case 3 %-----FUNCION ACKLEY (ND)
```

## ALGORITMOS DE OPTIMIZACIÓN

```

17
18         fobj=(20+exp(1)-20*exp(-1/5*sqrt((1/d)*sum(xi'.*xi')))-...
19             exp((1/d)*sum(cos((2*pi()*xi))))');
20
21     case 4 %-----FUNCION DE SPHERE (ND)
22
23         fobj=(sum(xi'.*xi'))';
24
25     case 5 %-----FUNCION GRIEWANK (ND)
26
27         xj=ones(ps,d);
28
29         for j=1:1:d
30             xj(:,j)=xj(:,j)*j;
31         end
32
33         fobj=(1+1/4000*(sum(xi'.*xi'))'-(prod(cos(xi'./sqrt(xj))))')';
34
35     case 6 %-----FUNCION ROSENBROCK (ND)
36
37         fobj=(sum((1-xi(:,1:d-1))'.^2+100*(xi(:,1:d-1)'.*xi(:,1:d-1))-...
38             xi(:,2:d)')'.^2))';
39
40     case 7 %-----FUNCION ALPINE (ND)
41
42         fobj=sum(abs(xi'.*sin(xi')+0.1*xi'))';
43
44     case 8 %-----FUNCION SCHWEFEL (ND)
45
46         fobj=((xi.*sin(sqrt(abs(xi))))')';
47         fobj=-sum(fobj)';
48
49     case 9 %-----FUNCION RASTRIGIN (ND)
50
51         fobj=(xi.^2-10*cos(2*pi()*xi))';
52         fobj=10*d+(sum(fobj))';
53
54
55 end
56 end

1 function [organizer]=organizer_pbest(ps,pbest,strategy)
2 disp('Organizacion de pbest'),pause
3 ps;
4 pbest;
5 strategy;
6 organizer=sort(pbest);
7 strategy_pbest=[];
8 for k=1:1:ps
9     for h=1:1:ps

```

## ALGORITMOS DE OPTIMIZACIÓN

```

10     if pbest(h,1)==organizer(h,1)
11         strategy_pbest(k,1)=strategy(k,1);
12         break
13     end
14     end
15 end
16 strategy_pbest;

1 function[gbest,pbest,S,coord_pbest,strategy_pbest]=select(ps,pbest,fobj,...
2     strategy,strategy_pbest,xi,coord_pbest,S,w,e,Gs,driver)
3 %% SELECCION DE LOS MEJORES LOCALES
4 for k=1:1:ps
5     if fobj(k,1)<pbest(k,1)
6         pbest(k,1)=fobj(k,1);
7         coord_pbest(k,:)=xi(k,:);
8         strategy_pbest(k,1)=strategy;
9     else
10         pbest(k,1)=pbest(k,1);
11         coord_pbest(k,:)=coord_pbest(k,:);
12         strategy_pbest(k,1)=strategy_pbest(k,1);
13     end
14 end
15
16 %% SELECCION DEL MEJOR GLOBAL
17 gbest=min(pbest);
18
19 %% ASIGNACION DEL PESO PARA LA ASIGNACION DE PROBABILIDAD
20 if e==(Gs*driver)
21     organizer=sort(pbest);
22     strategy_S=[];
23     for k=1:1:ps
24         for h=1:1:ps
25             if organizer(k,1)==pbest(h,1)
26                 strategy_S(k,1)=strategy_pbest(h,1);
27                 S(strategy_S(k,1))=S(strategy_S(k,1))+w(k);
28                 break
29             end
30         end
31     end
32 end
33
34 end

1 function ...
2     [gbest,pbest,coord_pbest]=select_pbest(ps,pbest,fobj,xi,coord_pbest)
3 for k=1:1:ps
4     if fobj(k,1)<pbest(k,1)
5         pbest(k,1)=fobj(k,1);
6         coord_pbest(k,:)=xi(k,:);

```

## ALGORITMOS DE OPTIMIZACIÓN

```
6         else
7             pbest(k,1)=pbest(k,1);
8             coord_pbest(k,:)=coord_pbest(k,:);
9         end
10     end
11     gbest=min(pbest);
```

### B.1.3. AG

```
1  clc;
2  clear all;
3  format compact;
4
5
6  ni=input('Ingrese el numero de pruebas a realizar = ');
7
8  for q=1:1:ni
9
10     %% DATOS GENERALES
11     fun=input('Ingrese la funcion a evaluar = ');
12     M=input('Ingrese el numero de individuos = '); %----NUMERO DE ...
        INDIVIDUOS !DEBE SER UN NUMERO PAR!
13     NN=input('Numero maximo de iteraciones = '); %-----NUMERO MAXIMO ...
        DE ITERACIONES
14     Pc=input('Ingrese la probabilidad de cruce = '); %----PROBABILIDAD ...
        DE CRUCE
15     Pm=input('Ingrese la probabilidad de mutacion = '); %_PROBABILIDAD ...
        DE MUTACION
16     toler=input('Ingrese la tolerancia para el criterio de parada = ...
        '); %----TOLERANCIA PARA EL CRITERIO DE PARADA
17     JC=input('Numero de datos para determinar la Desviacion Standar = ...
        '); %_NUMERO DE DATOS PARA CALCULAR LA DESVIACION ESTANDAR
18     a=0.2; %-----VALOR PARA CRUCE BLX-A
19
20     %% DEFINICION DE DIMENSIONES Y LIMITES DE LAS FUNCIONES
21     dim=[4 10 30 30 30 30 30 30 30];
22     lim=[-10 10;-10 10;-30 30;-20 20;-600 600;-10 10;-10 10;-500 500;...
        -5.12 5.12];
23
24
25     N=dim(fun); %-----DIMENSIONES
26     XL=lim(fun,1); %---LIMITE INFERIOR
27     XU=lim(fun,2); %---LIMITE SUPERIOR
28
29     %% GENERACION DE INDIVIDUOS ALEATORIOS
30     X=XL+(XU-XL)*rand(M,N);
31
32     ite=1;
33     des=10;
34     tic
35
```

## ALGORITMOS DE OPTIMIZACIÓN

```

36     while NN>ite
37         %% EVALUAR FUNCION OBJETIVO
38         F=my_fun(X,N,fun,M);
39
40         %% SELECCION POR RULETA
41         S=sum(F);
42         PROB(:,1)=F(:,1)/S;
43         PROB_ACUM=0.;
44         PROB_ACUM(1,1)=PROB(1,1);
45
46         for k=2:1:M
47             PROB_ACUM(k,1)=PROB(k,1)+PROB_ACUM(k-1,1);
48         end
49
50         %% GENERACION DE NUMEROS ALEATORIOS
51         SR(:,1)=rand(M,1);
52
53         %% ASIGNACION DE NUEVOS PADRES
54         XS=father(PROB_ACUM,SR,X,M);
55
56         %% CRUCE
57         XH=cruce(M,Pc,N,XS,a);
58
59         %% MUTACION
60         XM=mutacion(M,Pm,XL,XU,N,XH);
61         FM=my_fun(XM,N,fun,M);
62
63         %% ELITISMO
64         Xopt= elitismo(F,FM,M,N,XU,XL,X,XM);
65         Fopt(1,1)=-my_fun2(Xopt,N,fun)+100000;
66
67         %% ALMACENAR LOS MEJORES DE CADA ITERACION
68         Fmej(1,ite)=Fopt;
69         ite=ite+1;
70
71     end
72
73     %% ALMACENAMIENTO DE LOS RESULTADOS
74     tabla(q,1)=toc; %-----tiempo
75     tabla(q,2)=ite; %-----iteraciones
76     tabla(q,3)=Fopt; %-----mejor global
77     CGlobal(q,:)=Xopt; %-----puntos mejor global
78     Convergencia(:,q)=Fmej; %----convergencia
79
80 end
81
82 resultados(1)=mean(tabla(:,1)); %-----tiempo promedio
83 resultados(2)=mean(tabla(:,2)); %-----promedio de iteraciones
84 resultados(3)=max(tabla(:,3)); %-----peor gbest
85 resultados(4)=min(tabla(:,3)); %-----mejor gbest
86 resultados(5)=mean(tabla(:,3)); %-----promedio gbest
87 resultados(6)=std(tabla(:,3)); %-----desviacion estandr gbest

```

## ALGORITMOS DE OPTIMIZACIÓN

```

88 resultados(7)=median(tabla(:,3)); %----medians gbest
89
90
91 switch fun
92     case 1 %-----FUNCION COLVILLE (4D)
93         save('COLVILLE','tabla','CGlobal','Convergencia','resultados')
94     case 2 %-----FUNCION DIXON (10D)
95         save('DIXON','tabla','CGlobal','Convergencia','resultados')
96     case 3 %-----FUNCION ACKLEY (ND)
97         save('ACKLEY','tabla','CGlobal','Convergencia','resultados')
98     case 4 %-----FUNCION DE SPHERE (ND)
99         save('SPHERE','tabla','CGlobal','Convergencia','resultados')
100    case 5 %-----FUNCION GRIEWANK (ND)
101        save('GRIEWANK','tabla','CGlobal','Convergencia','resultados')
102    case 6 %-----FUNCION ROSENBROCK (ND)
103        save('ROSENBROCK','tabla','CGlobal','Convergencia','resultados')
104    case 7 %-----FUNCION ALPINE (ND)
105        save('ALPINE','tabla','CGlobal','Convergencia','resultados')
106    case 8 %-----FUNCION SCHWEFEL (ND)
107        save('SCHWEFEL','tabla','CGlobal','Convergencia','resultados')
108    case 9 %-----FUNCION RASTRIGIN (ND)
109        save('RASTRIGIN','tabla','CGlobal','Convergencia','resultados')
110 end
111
112 clear tabla CGlobal Convergencia resultados

```

```

1 function [XH] = cruce(M,Pc,N,XS,a)
2
3 CR(:,1)=rand(M/2,1);
4
5 for k=1:1:M/2
6     if CR(k,1)<Pc
7         for l=1:1:N
8             Imin(k,l)=min(XS(2*k-1,l),XS(2*k,l));
9             Imax(k,l)=max(XS(2*k-1,l),XS(2*k,l));
10            r=Imax(k,l)-Imin(k,l);
11            XH(k,l)=Imin(k,l)-r*a+rand()*(Imax(k,l)+r*a-Imin(k,l)-r*a);
12        end
13    else
14        XH(k,:)=XS(2*k-1,:);
15    end
16 end
17
18 for k=1:1:M/2
19     if CR(k,1)<Pc
20         for l=1:1:N
21             Imin(k+M/2,l)=min(XS(2*k-1,l),XS(2*k,l));
22             Imax(k+M/2,l)=max(XS(2*k-1,l),XS(2*k,l));
23             r=Imax(k+M/2,l)-Imin(k+M/2,l);
24             XH(k+M/2,l)=Imin(k+M/2,l)-r*a+rand()*(Imax(k+M/2,l)+r*a-...

```



## ALGORITMOS DE OPTIMIZACIÓN

```

25             Imin(k+M/2,1)-r*a);
26         end
27     else
28         XH(k+M/2,1)=XS(2*k,1);
29     end
30 end
31
32 end

```

```

1 function [Xopt] = elitismo(F,FM,M,N,XU,XL,X,XM)
2
3 Fmax=max(F);
4 Fmin=min(FM);
5
6 for k=1:1:M
7     if F(k,1)==Fmax;
8         Xmax=X(k,:);
9     end
10 end
11
12 for k=1:1:M
13     if FM(k,1)==Fmin;
14         X2(k,:)=Xmax(1,:);
15     else
16         X2(k,:)=XM(k,:);
17     end
18 end
19
20 X=X2;
21
22 for k=1:1:M
23     for h=1:1:N
24         if X(k,h)>XU
25             X(k,h)=XL+rand()*(XU-XL);
26         end
27         if X(k,h)<XL
28             X(k,h)=XL+rand()*(XU-XL);
29         end
30     end
31 end
32
33 Xopt=Xmax;
34
35 end

```

```

1 function [XS] = father(PROB_ACUM,SR,X,M)
2
3 for k=1:1:M
4     if SR(k,1)≤PROB_ACUM(1,1)

```

## ALGORITMOS DE OPTIMIZACIÓN

```

5         XS(k,:) = X(1,:);
6     end
7     for h=2:1:M
8         if PROB_ACUM(h-1,1) < SR(k,1) && SR(k,1) ≤ PROB_ACUM(h,1);
9             XS(k,:) = X(h,:);
10        end
11    end
12 end
13
14 end

```

```

1 function [XM] = mutacion(M,Pm,XL,XU,N,XH)
2
3 MR(:,1) = rand(M,1);
4
5 for k=1:1:M
6     if MR(k,1) < Pm
7         XM(k,:) = XL + (XU-XL) * rand(1,N);
8     else
9         XM(k,:) = XH(k,:);
10    end
11 end
12
13 end

```

```

1 function [F] = my_fun(X,N,fun,ps)
2
3 xi=X;
4 d=N;
5
6 switch fun
7
8     case 1 %-----FUNCION COLVILLE (4D)
9
10        fobj=100*(xi(:,1).^2-xi(:,2)).^2+(xi(:,1)-1).^2+(xi(:,3)-1).^2+...
11            90*(xi(:,3).^2-xi(:,4)).^2+10.1*((xi(:,2)-1).^2+...
12            (xi(:,4)-1).^2)+19.8*(xi(:,2)-1).*(xi(:,4)-1);
13
14     case 2 %-----FUNCION DIXON (10D)
15
16        fobj=(1-xi(:,1)).^2+(1-xi(:,10)).^2+(sum((xi(:,1:9)).*xi(:,1:9)).*...
17            xi(:,2:10)).^2));
18
19     case 3 %-----FUNCION ACKLEY (ND)
20
21        fobj=(20+exp(1)-20*exp(-1/5*sqrt((1/d)*sum(xi'.*xi'))))-...
22            exp((1/d)*sum(cos((2*pi()*xi)))));
23
24     case 4 %-----FUNCION DE SPHERE (ND)

```

## ALGORITMOS DE OPTIMIZACIÓN

```

25         fobj=(sum(xi' .*xi'))';
26
27
28     case 5 %-----FUNCION GRIEWANK (ND)
29
30         xj=ones(ps,d);
31
32         for j=1:1:d
33             xj(:,j)=xj(:,j)*j;
34         end
35
36         fobj=(1+1/4000*(sum(xi' .*xi'))'-(prod(cos(xi'./sqrt(xj')))))';
37
38     case 6 %-----FUNCION ROSENBROCK (ND)
39
40         fobj=(sum((1-xi(:,1:d-1)).^2+100*(xi(:,1:d-1)' .*xi(:,1:d-1)'. ...
41             xi(:,2:d)')'.^2))';
42
43     case 7 %-----FUNCION ALPINE (ND)
44
45         fobj=sum(abs(xi' .*sin(xi')+0.1*xi'))';
46
47     case 8 %-----FUNCION SCHWEFEL (ND)
48
49         fobj=((xi.*sin(sqrt(abs(xi))))')';
50         fobj=-sum(fobj)';
51
52     case 9 %-----FUNCION RASTRIGIN (ND)
53
54         fobj=(xi.^2-10*cos(2*pi()*xi))';
55         fobj=10*d+sum(fobj)';
56 end
57
58 F=100000-fobj;
59
60 end

1 function [F] = my_fun2(X,N,fun)
2
3 xi=X;
4 d=N;
5 ps=1;
6
7 switch fun
8
9     case 1 %-----FUNCION COLVILLE (4D)
10
11         fobj=100*(xi(:,1).^2-xi(:,2)).^2+(xi(:,1)-1).^2+(xi(:,3)-1).^2+...
12             90*(xi(:,3).^2-xi(:,4)).^2+10.1*((xi(:,2)-1).^2+...
13             (xi(:,4)-1).^2)+19.8*(xi(:,2)-1).*(xi(:,4)-1);

```

## ALGORITMOS DE OPTIMIZACIÓN

```

14
15     case 2 %-----FUNCION DIXON (10D)
16
17         fobj=(1-xi(:,1)).^2+(1-xi(:,10)).^2+(sum((xi(:,1:9)).*xi(:,1:9)).*...
18             -xi(:,2:10)).^2)';
19
20     case 3 %-----FUNCION ACKLEY (ND)
21
22         fobj=(20+exp(1)-20*exp(-1/5*sqrt((1/d)*sum(xi'.*xi')))-...
23             exp((1/d)*sum(cos((2*pi()*xi')))))';
24
25     case 4 %-----FUNCION DE SPHERE (ND)
26
27         fobj=(sum(xi'.*xi'))';
28
29     case 5 %-----FUNCION GRIEWANK (ND)
30
31         xj=ones(ps,d);
32
33         for j=1:1:d
34             xj(:,j)=xj(:,j)*j;
35         end
36
37         fobj=(1+1/4000*(sum(xi'.*xi'))'-(prod(cos(xi'./sqrt(xj')))))';
38
39     case 6 %-----FUNCION ROSENBROCK (ND)
40
41         fobj=(sum((1-xi(:,1:d-1)).^2+100*(xi(:,1:d-1)).*xi(:,1:d-1))-...
42             xi(:,2:d)).^2)';
43
44     case 7 %-----FUNCION ALPINE (ND)
45
46         fobj=sum(abs(xi'.*sin(xi')+0.1*xi'))';
47
48     case 8 %-----FUNCION SCHWEFEL (ND)
49
50         fobj=((xi.*sin(sqrt(abs(xi))))')';
51         fobj=-sum(fobj)';
52
53     case 9 %-----FUNCION RASTRIGIN (ND)
54
55         fobj=(xi.^2-10*cos(2*pi()*xi))';
56         fobj=10*d+sum(fobj)';
57 end
58
59 F=100000-fobj;
60
61 end

```

## ALGORITMOS DE OPTIMIZACIÓN

### B.1.4. SA

## B.2. ALGORITMOS PARA OPTIMIZACIÓN CON RESTRICCIONES

### B.2.1. SL-PSO

```
1  %%
2  clc;clear all;format compact;
3
4  %%
5  disp('*****')
6  disp('*****')
7  disp('***** SL-PSO *****')
8  disp('*****')
9  disp('*****')
10 disp('-----')
11 disp('FUNCIONES A EVALUAR ')
12 disp('PROBLEMA g01..... (1) ') %OPTIMO=-15
13 disp('PROBLEMA g02..... (2) ') %OPTIMO=0.803619
14 disp('PROBLEMA g03..... (3) ') %OPTIMO=1
15 disp('PROBLEMA g04..... (4) ') %OPTIMO=-30665.539
16 disp('PROBLEMA g05..... (5) ') %OPTIMO=5126.4981
17 disp('PROBLEMA g06..... (6) ') %OPTIMO=-6961.81388
18 disp('PROBLEMA g07..... (7) ') %OPTIMO=24.3062091
19 disp('PROBLEMA g08..... (8) ') %OPTIMO=0.095825
20 disp('PROBLEMA g09..... (9) ') %OPTIMO=680.6300573
21 disp('PROBLEMA g10..... (10) ') %OPTIMO=7049.25
22 disp('PROBLEMA g11..... (11) ') %OPTIMO=0.75
23 disp('PROBLEMA g12..... (12) ') %OPTIMO=1
24 disp('PROBLEMA g13..... (13) ') %OPTIMO=0.0539498
25 disp('-----')
26
27
28 %% NUMERO DE PRUEBAS A REALIZAR
29 ni=input('Ingrese el numero de pruebas a realizar = ');
30
31 for q=1:1:ni
32     %% DATOS DE ENTRADA
33     fun=input('Ingrese la funcion a evaluar = ');
34     ps=input('Ingrese el # de particulas = ');
35     E=input('Desviacion Standar admisible = ');
36     rd=input('Numero de datos para determinar la Desviacion Standar = ');
37     max_ite=input('Numero maximo de iteraciones = ');
38
39     disp(['PRUEBA ',num2str(q),' DEL PROBLEMA g0',num2str(fun)])
40
41     %% LIMITES DE LAS FUNCIONES
```

## ALGORITMOS DE OPTIMIZACIÓN

```

42     [d,lim]=limit(fun);
43
44     %% PARAMETROS DEL ALGORITMO
45     p=abs((lim(2,:)-lim(1,:))./5;%-----PASO MAXIMO PERMITIDO ...
        PARA LA VELOCIDAD
46     c3=1.49445;%-----COEFICIENTE ESTRATEGIA ...
        N-3 y N-4 (SELECCIONADO DEL CLPSO ORIGINAL)
47     w0=0.9;%-----PESO DE INERCIA ...
        ESTRATEGIA N-3 Y N-4
48
49     Gs=100;%-----PERIODO DE APRENDIZAJE
50     alpha=1/6;%-----COEFICIENTE DE APRENDIZAJE
51     proSTR(1:3)=1/3;%-----ASIGNACION DE ...
        PROBABILIDAD INICIAL PARA CADA ESTRTEGIA DE ACTUALIZACION DE V
52     S(1:3)=0;%-----ACUMULADOR DE LAS CUATRO ...
        ESTRATEGIAS
53
54     sumlog=0;
55     for k=1:1:ps
56         sumlog=log(k)+sumlog;
57     end
58     w=(1:ps);
59     w=ps-w+1;
60     w=log(w)./sumlog;
61
62     Pci=(1:ps);
63     Pci=0.05+0.45*((exp((10.*(Pci-1))/(ps-1))-1)/(exp(10)-1));
64
65     %% COANTADOR DE TIEMPO
66     tic;
67
68     %% DISTRIBUCION DE LA VELOCIDAD(vi) Y LA POSICION(xi) INICIAL
69     [xi,vi]=dis_inic(ps,d,lim,p);
70
71     %% ASIGNACION DE LA ESTRATEGIA PARA v
72     strategy_pbest=round(rand(ps,1).*(3-1)+1);
73
74     %% CALCULO DE LA FUNCION OBJETIVO
75     [C,fobj]=fittnes(fun,xi,ps,d);
76
77     %% SELECCION DE LOS MEJORES LOCALES
78     pbest=fobj;
79     coord_pbest=xi;
80
81     %% ELECCION DEL MEJOR GLOBAL
82     [gbest pos]=min(pbest(:,1));
83     coord_gbest=coord_pbest(pos,:);
84
85     %% PARAMEROS PARA EL CONTROL DE LA EJECUCION DEL ALGORITMO
86     e=0;%-----CONTADOR
87     des=10;%-----DESVIACION ESTANDAR
88     driver=0;%-----CONTROLADOR PARA EL PROCESO ...

```

## ALGORITMOS DE OPTIMIZACIÓN

```

DE OPTIMIZACION
89
90 %% EJECUCION DEL ALGORITMO
91 while e<max_ite && des>E
92     %% DETERMINACION DE LA VELOCIDAD
93     e=e+1;
94     x0=xi;
95     v0=vi;
96
97     if e>(Gs*driver)
98         na=rand();
99         driver=driver+1;
100    end
101
102    if na<proSTR(1)
103        %% ACTUALIZACION DE v POR MEDIO DE LA ESTRATEGIA 1
104        [vi,strategy]=calc_DbV(ps,x0,coord_pbest);
105
106    elseif na<(proSTR(1)+proSTR(2))
107        %% ACTUALIZACION DE v POR MEDIO DE LA ESTRATEGIA 3
108        [vi,strategy]=calc_CLPSO(ps,d,coord_pbest,v0,x0,c3,w0,...
109            Pci,e,max_ite,pbest);
110
111    elseif na<(proSTR(1)+proSTR(2)+proSTR(3))
112        %% ACTUALIZACION DE v POR MEDIO DE LA ESTRATEGIA 4
113        [vi,strategy]=calc_PSO_CL_pbest(ps,d,coord_pbest,v0,x0,...
114            c3,w0,Pci,e,max_ite,pbest);
115    end
116
117    %% CHEQUEO DE LA VELOCIDAD
118    [vi]=check_v(vi,p,d);
119
120    %% CALCULO DE xi A PARTIR DE vi
121    xi=x0+vi;
122
123    %% CHEQUEO DE LA POSICION
124    [xi]=check_x(xi,lim,d);
125
126    %% CALCULO FUNCION OBJETIVO
127    [C,fobj]=fittnes(fun,xi,ps,d);
128
129    %% SELECCION gbest,pbest,S,coord_pbest Y coord_gbest
130    [gbest,pbest,S,coord_pbest,strategy_pbest,pos]=select(ps,...
131        pbest,fobj,strategy,strategy_pbest,xi,coord_pbest,S,w,e,...
132        Gs,driver);
133    coord_gbest=coord_pbest(pos,:);
134
135    %% ASIGNACION DE LA PROBABILIDAD PARA LA SELECCION DE LA ...
136    VELOCIDAD
137    if e==(Gs*driver)
138        proSTRi=(1-alpha)*proSTR+alpha*(S./Gs);
139        S(1:3)=0;

```

## ALGORITMOS DE OPTIMIZACIÓN

```

139         sum_proSTri=sum(proSTri);
140         proSTR=proSTri./sum_proSTri;
141     end
142
143     prob(:,e)=proSTR;
144
145     %% ALMACENAMIENTO DE DATOS PARA DETERMINAR EL CRITERIO ...
146     DE PARADA
147     ds(e)=gbest;
148
149     %% CALCULO DEL CRITERIO DE PARADA
150     if e>rd
151         des=std(ds(1,e-rd:e));
152     end
153
154     end
155
156     %% ALMACENAMIENTO DE LOS RESULTADOS
157     tabla(q,1)=toc; %-----tiempo
158     tabla(q,2)=e; %-----iteraciones
159     tabla(q,3)=gbest; %-----mejor global
160     CGlobal(q,:)=coord_gbest; %---puntos mejor global
161     Convergencia(:,q)=ds; %-----convergencia
162     probabilidad(:,q)=prob'; %----probabilidad de cada estrategia
163
164 end
165
166 %% RESULTAODS
167 resultados(1)=mean(tabla(:,1)); %-----tiempo promedio
168 resultados(2)=mean(tabla(:,2)); %-----promedio de iteraciones
169 resultados(3)=max(tabla(:,3)); %-----peor gbest
170 resultados(4)=min(tabla(:,3)); %-----mejor gbest
171 resultados(5)=mean(tabla(:,3)); %-----promedio gbest
172 resultados(6)=std(tabla(:,3)); %-----desviacion estandar gbest
173 resultados(7)=median(tabla(:,3)); %-----mediana gbest
174
175 switch fun
176 case 1 %-----FUNCION g01
177     save('g01','tabla','CGlobal','Convergencia','resultados','probabilidad')
178 case 2 %-----FUNCION g02
179     save('g02','tabla','CGlobal','Convergencia','resultados','probabilidad')
180 case 3 %-----FUNCION g03
181     save('g03','tabla','CGlobal','Convergencia','resultados','probabilidad')
182 case 4 %-----FUNCION g04
183     save('g04','tabla','CGlobal','Convergencia','resultados','probabilidad')
184 case 5 %-----FUNCION g05
185     save('g05','tabla','CGlobal','Convergencia','resultados','probabilidad')
186 case 6 %-----FUNCION g06
187     save('g06','tabla','CGlobal','Convergencia','resultados','probabilidad')
188 case 7 %-----FUNCION g07
189     save('g07','tabla','CGlobal','Convergencia','resultados','probabilidad')

```



## ALGORITMOS DE OPTIMIZACIÓN

```
190     case 8%-----FUNCION g08
191         save('g08','tabla','CGlobal','Convergencia','resultados','probabilidad')
192     case 9%-----FUNCION g09
193         save('g09','tabla','CGlobal','Convergencia','resultados','probabilidad')
194     case 10%-----FUNCION g10
195         save('g10','tabla','CGlobal','Convergencia','resultados','probabilidad')
196     case 11%-----FUNCION g11
197         save('g11','tabla','CGlobal','Convergencia','resultados','probabilidad')
198     case 12%-----FUNCION g12
199         save('g12','tabla','CGlobal','Convergencia','resultados','probabilidad')
200     case 13%-----FUNCION g13
201         save('g13','tabla','CGlobal','Convergencia','resultados','probabilidad')
202 end
203 clear resultados tabla CGlobal Convergencia probabilidad
```

```
1 function [vi,strategy]=calc_CLPSO(ps,d,coord_pbest,v0,x0,c3,w0,Pci,e,...
2     max_ite,pbest)
3 vi=[];
4 w=(w0-0.2)*(max_ite-e)/(max_ite)+0.2;
5 for i=1:1:ps
6     for j=1:1:d
7         na=rand;
8         if na<Pci(i)
9             fd1=0;
10            while fd1==0
11                xk=round(rand()*(ps-1)+1);
12                if xk==i
13                    fd1=0;
14                else
15                    fd1=1;
16                end
17            end
18
19            fd2=0;
20            while fd2==0
21                xj=round(rand()*(ps-1)+1);
22                if xj==i
23                    fd2=0;
24                else
25                    fd2=1;
26                end
27            end
28
29            if pbest(xk,1)<pbest(xj,1)
30                pbestFD(i,j)=coord_pbest(xk,j);
31            else
32                pbestFD(i,j)=coord_pbest(xj,j);
33            end
34        else
35            pbestFD(i,j)=coord_pbest(i,j);
```

## ALGORITMOS DE OPTIMIZACIÓN

```

36         end
37     end
38 end
39
40 r1=rand(ps,d);
41 vi=w*v0+c3*r1.*(pbestFD-x0);
42
43 strategy=2;

1 function [vi,strategy]=calc_DbV(ps,x0,coord_pbest)
2 c1=normrnd(0.5,0.2); %-----COEFICIENTE ESTRATEGIA N-1
3 vi=[];
4 for k=1:1:ps
5     xk=round(rand()*(ps-1)+1);
6     xj=round(rand()*(ps-1)+1);
7     Viad=x0(xk,:)-x0(xj,:);
8     vi(k,:)=c1.*Viad+c1.*(coord_pbest(k,:)-x0(k,:));
9 end
10
11 strategy=1;

1 function ...
2     [vi,strategy]=calc_PSO_CL_pbest(ps,d,coord_pbest,v0,x0,c3,w0,Pci,...
3     e,max_ite,pbest)
4 vi=[];
5 w=(w0-0.2)*(max_ite-e)/(max_ite)+0.2;
6 for i=1:1:ps
7     for j=1:1:d
8         na=rand;
9         if na<Pci(i)
10             fd1=0;
11             while fd1==0
12                 xk=round(rand()*(ps-1)+1);
13                 if xk==i
14                     fd1=0;
15                 else
16                     fd1=1;
17                 end
18             end
19
20             fd2=0;
21             while fd2==0
22                 xj=round(rand()*(ps-1)+1);
23                 if xj==i
24                     fd2=0;
25                 else
26                     fd2=1;
27             end

```

## ALGORITMOS DE OPTIMIZACIÓN

```

28         end
29
30         if pbest(xk,1)<pbest(xj,1)
31             pbestFD(i,j)=coord_pbest(xk,j);
32         else
33             pbestFD(i,j)=coord_pbest(xj,j);
34         end
35     else
36         pbestFD(i,j)=coord_pbest(i,j);
37     end
38 end
39 end
40
41 r1=rand(ps,1);
42
43 for h=1:1:d
44     r2(:,h)=r1(:,1);
45 end
46
47 vi=w*v0+0.5*c3*r2.*(pbestFD-x0+coord_pbest-x0());
48
49 strategy=3;

```

```

1 function [vi]=check_v(vi,p,d)
2
3 for i=1:1:d
4     c=(-p(i)≤vi(:,i)) & (vi(:,i)≤p(i));
5     m=vi(:,i).*c;
6     pd=not(c);
7     t=pd.*vi(:,i);
8     e1=-p(i).*(t≤-p(i));
9     e2=p(i).*(t≥p(i));
10    vi(:,i)=m+e1+e2;
11 end
12
13 end

```

```

1 function [xi]=check_x(xi,lim,d)
2
3 for i=1:1:d
4     c=(lim(1,i)≤xi(:,i)) & (xi(:,i)≤lim(2,i));
5     m=xi(:,i).*c;
6     pd=not(c);
7     t=pd.*xi(:,i);
8     e1=pd.*(lim(1,i).*(t≤lim(1,i)));
9     e2=pd.*(lim(2,i).*(t≥lim(2,i)));
10    xi(:,i)=m+e1+e2;
11 end

```

## ALGORITMOS DE OPTIMIZACIÓN

```

1 function [fobj]=constraint(fobj1,C)
2 [nr ps]=size(C);
3 SUM=zeros(ps,1);
4
5 aptitud=fobj1;
6 for i=1:1:ps
7     for j=1:1:nr
8         if C(j,i)>0
9             SUM(i,1)=SUM(i,1)+C(j,i)/max(C(j,:));
10        end
11    end
12 end
13 fobj2=aptitud+SUM*1e10;
14 fobj(:,1)=fobj2;
15 fobj(:,2)=fobj1;
16 end

```

```

1 function [xi,vi]=dis_inic(ps,d,lim,p)
2 xi=[];
3 vi=[];
4 for i=1:1:ps
5     xi(i,:)=lim(1,:)+rand(1,d).*(lim(2,:)-lim(1,:));
6     vi(i,:)=lim(1,:)+rand(1,d).*(lim(2,:)-lim(1,:));
7 end
8
9 [vi]=check_v(vi,p,d);
10
11 end

```

```

1 function [C,fobj]=fittnes(fun,xi,ps,d)
2 C=[];
3 x=xi';
4 switch fun
5     case 1
6         fobj=(5*sum(x(1:4,:))-5*sum(x(1:4,:).*x(1:4,:))-sum(x(5:13,:)))';
7         C(1,:)=2*x(1,:)+2*x(2,:)+x(10,:)+x(11,:)-10;
8         C(2,:)=2*x(1,:)+2*x(3,:)+x(10,:)+x(12,:)-10;
9         C(3,:)=2*x(2,:)+2*x(3,:)+x(11,:)+x(12,:)-10;
10        C(4,:)= -8*x(1,:)+x(10,:);
11        C(5,:)= -8*x(2,:)+x(11,:);
12        C(6,:)= -8*x(3,:)+x(12,:);
13        C(7,:)= -2*x(4,:)-x(5,:)+x(10,:);
14        C(8,:)= -2*x(6,:)-x(7,:)+x(11,:);
15        C(9,:)= -2*x(8,:)-x(9,:)+x(12,:);
16    case 2
17        j=[];
18        for i=1:1:ps
19            j(:,i)=1:1:d;
20        end

```

## ALGORITMOS DE OPTIMIZACIÓN

```

21     fobj=(-abs((sum((cos(x)).^4)-2*prod((cos(x)).^2))./sqrt...
22         (sum(j.*(x.*x))))))';
23     C(1,:)=0.75-prod(x);
24     C(2,:)=sum(x)-7.5*d;
25 case 3
26     fobj=(-((sqrt(d)^d)*prod(x)))';
27     C(1,:)=abs((sum(x.*x)-1))-1e-3;
28 case 4
29     fobj=(5.3578547*x(3,:).^2+0.8356891*x(1,:).*x(5,:)+37.293239*x(1,:) ...
30         -40792.141)';
31     C(1,:)=85.334407+0.0056858*x(2,:).*x(5,:)+0.0006262*x(1,:).*x(4,)-...
32         0.0022053*x(3,:).*x(5,)-92;
33     C(2,:)=85.334407-0.0056858*x(2,:).*x(5,)-0.0006262*x(1,:).*x(4,)+...
34         0.0022053*x(3,:).*x(5,);
35     C(3,:)=80.51249+0.0071317*x(2,:).*x(5,)+0.0029955*x(1,:).*x(2,)+...
36         0.0021813*x(3,:).^2-110;
37     C(4,:)=80.51249-0.0071317*x(2,:).*x(5,)-0.0029955*x(1,:).*x(2,)-...
38         0.0021813*x(3,:).^2+90;
39     C(5,:)=9.300961+0.0047026*x(3,:).*x(5,)+0.0012547*x(1,:).*x(3,)+...
40         0.0019085*x(3,:).*x(4,)-25;
41     C(6,:)=9.300961-0.0047026*x(3,:).*x(5,)-0.0012547*x(1,:).*x(3,)-...
42         0.0019085*x(3,:).*x(4,)+20;
43 case 5
44     fobj=(3*x(1,)+0.000001*x(1,).^3+2*x(2,)+(0.000002/3)*x(2,).^3)';
45     C(1,:)=x(3,)-x(4,)-0.55;
46     C(2,:)=x(4,)-x(3,)-0.55;
47     C(3,:)=abs(1000*sin(-x(3,)-0.25)+1000*sin(-x(4,)-0.25)+894.8-...
48         x(1,))-1e-3;
49     C(4,:)=abs(1000*sin(x(3,)-0.25)+1000*sin(x(3,)-x(4,)-0.25)+...
50         894.8-x(2,))-1e-3;
51     C(5,:)=abs(1000*sin(x(4,)-0.25)+1000*sin(x(4,)-x(3,)-0.25)+...
52         1294.8)-1e-3;
53 case 6
54     fobj=((x(1,)-10).^3+(x(2,)-20).^3)';
55     C(1,:)=-(x(1,)-5).^2-(x(2,)-5).^2+100;
56     C(2,:)=(x(1,)-6).^2+(x(2,)-5).^2-82.81;
57 case 7
58     fobj=(x(1,).^2+x(2,).^2+x(1,).*x(2,)-14*x(1,)-16*x(2,)+...
59         (x(3,)-10).^2+4*(x(4,)-5).^2+(x(5,)-3).^2+2*(x(6,)-1).^2+...
60         5*x(7,).^2+...
61         +7*(x(8,)-11).^2+2*(x(9,)-10).^2+(x(10,)-7).^2+45)';
62     C(1,:)=105+4*x(1,)+5*x(2,)-3*x(7,)+9*x(8,);
63     C(2,:)=10*x(1,)-8*x(2,)-17*x(7,)+2*x(8,);
64     C(3,:)=8*x(1,)+2*x(2,)+5*x(9,)-2*x(10,)-12;
65     C(4,:)=3*(x(1,)-2).^2+4*(x(2,)-3).^2+2*x(3,).^2-7*x(4,)-120;
66     C(5,:)=5*x(1,).^2+8*x(2,)+(x(3,)-6).^2-2*x(4,)-40;
67     C(6,:)=x(1,).^2+2*(x(2,)-2).^2-2*x(1,).*x(2,)+14*x(5,)-6*x(6,);
68     C(7,:)=0.5*(x(1,)-8).^2+2*(x(2,)-4).^2+3*x(5,).^2-x(6,)-30;
69     C(8,:)=3*x(1,)+6*x(2,)+12*(x(9,)-8).^2-7*x(10,);
70 case 8
71     fobj=(((-(sin(2*pi*x(1,))))).^3).*sin(2*pi*x(2,)))./...
72         ((x(1,).^3)*(x(1,)+x(2,)))));

```

## ALGORITMOS DE OPTIMIZACIÓN

```

73         C(1,:) = x(1,:).^2 - x(2,:) + 1;
74         C(2,:) = 1 - x(1,:) + (x(2,:) - 4).^2;
75     case 9
76         fobj = ((x(1,:) - 10).^2 + 5*(x(2,:) - 12).^2 + x(3,:).^4 + 3*(x(4,:) - 11).^2 + ...
77             10*x(5,:).^6 + 7*x(6,:).^2 + x(7,:).^4 - 4*x(6,:).*x(7,:) - 10*...
78             x(6,:) - 8*x(7,:))';
79         C(1,:) = -127 + 2*x(1,:).^2 + 3*x(2,:).^4 + x(3,:) + 4*x(4,:).^2 + 5*x(5,:);
80         C(2,:) = -282 + 7*x(1,:) + 3*x(2,:) + 10*x(3,:).^2 + x(4,:) - x(5,:);
81         C(3,:) = -196 + 23*x(1,:) + x(2,:).^2 + 6*x(6,:).^2 - 8*x(7,:);
82         C(4,:) = 4*x(1,:).^2 + x(2,:).^2 - 3*x(1,:).*x(2,:) + 2*x(3,:).^2 + ...
83             5*x(6,:) - 11*x(7,:);
84     case 10
85         fobj = (x(1,:) + x(2,:) + x(3,:))';
86         C(1,:) = -1 + 0.0025*(x(4,:) + x(6,:));
87         C(2,:) = -1 + 0.0025*(x(5,:) + x(7,:) - x(4,:));
88         C(3,:) = -1 + 0.01*(x(8,:) - x(5,:));
89         C(4,:) = -x(1,:).*x(6,:) + 833.33252*x(4,:) + 100*x(1,) - 83333.333;
90         C(5,:) = -x(2,:).*x(7,:) + 1250*x(5,:) + x(2,:).*x(4,) - 1250*x(4,:);
91         C(6,:) = -x(3,:).*x(8,) + 1250000 + x(3,:).*x(5,) - 2500*x(5,:);
92     case 11
93         fobj = (x(1,:).^2 + (x(2,) - 1).^2)';
94         C(1,:) = abs(x(2,) - x(1,:).^2) - 1e-4;
95     case 12
96         fobj = -(100 - (x(1,) - 5).^2 - (x(2,) - 5).^2 - (x(3,) - 5).^2) ./ 100)';
97
98         re = 0;
99         for p = 1:1:9
100             for q = 1:1:9
101                 for r = 1:1:9
102                     re = re + 1;
103                     D(re,:) = (x(1,) - p).^2 + (x(2,) - q).^2 + (x(3,) - r) - 0.0625;
104                 end
105             end
106         end
107         C = min(D);
108
109     case 13
110         fobj = (exp(x(1,:).*x(2,:).*x(3,:).*x(4,:).*x(5,:)))';
111         C(1,:) = abs(x(1,:).^2 + x(2,:).^2 + x(3,:).^2 + x(4,:).^2 + x(5,:).^2 - 10) - 0.001;
112         C(2,:) = abs(x(2,:).*x(3,) - 5*x(4,:).*x(5,:)) - 0.001;
113         C(3,:) = abs(x(1,:).^3 + x(2,:).^3 + 1) - 0.001;
114 end
115
116 [fobj] = constraint(fobj, C);
117
118 end

1 function [d, lim] = limit(fun)
2 switch fun
3     case 1

```

## ALGORITMOS DE OPTIMIZACIÓN

```
4         d=13;
5         lim=zeros(2,d);
6         lim(2,:)=1;
7         lim(2,10:12)=100;
8     case 2
9         d=20;
10        lim=zeros(2,d);
11        lim(2,:)=10;
12    case 3
13        d=10;
14        lim=zeros(2,d);
15        lim(2,:)=1;
16    case 4
17        d=5;
18        lim=zeros(2,d);
19        lim(1,1)=78;
20        lim(2,1)=102;
21        lim(1,2)=33;
22        lim(2,2)=45;
23        lim(1,3:5)=27;
24        lim(2,3:5)=45;
25    case 5
26        d=4;
27        lim=zeros(2,d);
28        lim(1,1:2)=0;
29        lim(1,3:4)=-0.55;
30        lim(2,1:2)=1200;
31        lim(2,3:4)=0.55;
32    case 6
33        d=2;
34        lim=zeros(2,d);
35        lim(1,1)=13;
36        lim(2,:)=100;
37    case 7
38        d=10;
39        lim=zeros(2,d);
40        lim(1,:)=-10;
41        lim(2,:)=10;
42    case 8
43        d=2;
44        lim=zeros(2,d);
45        lim(2,:)=10;
46    case 9
47        d=7;
48        lim=zeros(2,d);
49        lim(1,:)=-10;
50        lim(2,:)=10;
51    case 10
52        d=8;
53        lim=zeros(2,d);
54        lim(1,1)=100;
55        lim(2,1:3)=10000;
```

## ALGORITMOS DE OPTIMIZACIÓN

```
56         lim(1,2:3)=1000;
57         lim(1,4:8)=10;
58         lim(2,4:8)=1000;
59     case 11
60         d=2;
61         lim=zeros(2,d);
62         lim(1,:)=−1;
63         lim(2,:)=1;
64     case 12
65         d=3;
66         lim=zeros(2,d);
67         lim(2,:)=10;
68     case 13
69         d=5;
70         lim=zeros(2,d);
71         lim(1,1:2)=−2.3;
72         lim(2,1:2)=2.3;
73         lim(1,3:5)=−3.2;
74         lim(2,3:5)=3.2;
75 end
76 end
```

```
1 function [organizer]=organizer_pbest(ps,pbest,strategy)
2 organizer=sort(pbest);
3 strategy_pbest=[];
4 for k=1:1:ps
5     for h=1:1:ps
6         if pbest(h,1)==organizer(h,1)
7             strategy_pbest(k,1)=strategy(k,1);
8             break
9         end
10    end
11 end
12 strategy_pbest;
```

```
1 function[gbest,pbest,S,coord_pbest,strategy_pbest,pos]=select(ps,pbest,...
2     fobj,strategy,strategy_pbest,xi,coord_pbest,S,w,e,Gs,driver)
3 %% SELECCION DE LOS MEJORES LOCALES
4 for k=1:1:ps
5     if fobj(k,1)<pbest(k,1)
6         pbest(k,:)=fobj(k,:);
7         coord_pbest(k,:)=xi(k,:);
8         strategy_pbest(k,1)=strategy;
9     else
10        pbest(k,:)=pbest(k,:);
11        coord_pbest(k,:)=coord_pbest(k,:);
12        strategy_pbest(k,1)=strategy_pbest(k,1);
13    end
14 end
```



## ALGORITMOS DE OPTIMIZACIÓN

```
15
16 %% SELECCION DEL MEJOR GLOBAL
17 [gbest pos]=min(pbest(:,1));
18
19 %% ASIGNACION DEL PESO PARA LA ASIGNACION DE PROBABILIDAD
20 if e==(Gs*driver)
21     organizer=sort(pbest(:,1));
22     strategy_S=[];
23     for k=1:1:ps
24         for h=1:1:ps
25             if organizer(k,1)==pbest(h,1)
26                 strategy_S(k,1)=strategy_pbest(h,1);
27                 S(strategy_S(k,1))=S(strategy_S(k,1))+w(k);
28                 break
29             end
30         end
31     end
32 end
33
34 end
```

---

## ALGORITMO PARA LA DETECCIÓN DE DAÑOS

---

```

1  %% FUNCIONES OBJETIVO CON FRECUENCIAS DE VIBRACION
2  clear all;clc;
3
4  %%
5  % ...
6  %
7  % Damage detection for the Benchmark Problem of the ASCE using SL-PSO
8  %
9  % Fredy Sotelo 29-enero-2013
10 % ...
11
12
13 %% FRECUENCIAS DEL MODELO EXPERIMENTAL
14 % FRECUENCIAS VIBRACION AMBIENTAL ref[54]
15 % filas =====> Numero de modos
16 % columnas ==> Patron de dano
17 Fr_ME=[7.50 7.77 14.48 19.88 21.01;
18         7.73 5.66 12.74 20.13 15.02;
19         7.65 6.65 13.48 20.02 18.91;
20         7.60 7.35 13.97 20.12 19.71;
21         7.56 7.45 14.03 19.89 20.63;
22         5.96 7.79 13.19 19.87 20.99;
23         2.64 3.63 4.34 8.49 12.05]';
24
25 COMP=size(Fr_ME);
26
27 NFr=COMP(1); % =====> Numero de frecuencias de vibracion (5)
28 Pd=COMP(2); % =====> Patrones de dano (6)
29

```

## ALGORITMO PARA LA DETECCIÓN DE DAÑOS

```

30
31 %% DATOS INICIALES DE ENTRADA
32 NFA=5; %      ==> Numero de frecuencias para la deteccion
33 ni=10; %      ==> Numero de pruebas a realizar
34 ps=50; %      ==> Numero de particulas
35 max_ite=500; %==> Numero maximo de iteraciones
36 d=32; %      ==> Dimensiones del problema
37
38 for fun=1:1 %-----CONTADOR FUNCIONES OBJETIVO
39
40     for dd=4:7 %---CONTADOR PATRON DE DANO
41
42         Damage=Fr_ME(1:NFA,dd);
43
44         for q=1:1:ni
45             clc;
46             str1=['funcion objetivo numero ', num2str(fun)];
47             str2=['Patron de dano numero ', num2str(dd)];
48             str3=['Prueba numero ', num2str(q)];
49
50             %% LIMITES DEL ESPACIO DE BUSQUEDA
51             lim=zeros(2,d); % ==> VECTOR PARA ESTABLECER LOS LIMITES ...
                    PARA CADA VARIABLE
52             lim(1,:)=0.0001; % ==> LIMITE INFERIOR
53             lim(2,:)=1; % ==> LIMITE SUPERIOR
54
55             %% PARAMETROS DEL ALGORITMO
56             p=round(abs((lim(2,:)-lim(1,:)))/1); %-----PASO ...
                    MAXIMO PERMITIDO PARA LA VELOCIDAD
57             c3=1.49445; %-----COEFICIENTE ...
                    ESTRATEGIA No3 y No4 (SELECCIONADO DEL CLPSO ORIGINAL)
58             w0=0.9; %-----PESO DE INERCIA ...
                    ESTRATEGIA N-3 Y N-4
59
60             Gs=100; %-----PERIODO DE ...
                    APRENDIZAJE
61             alpha=1/6; %-----COEFICIENTE DE ...
                    APRENDIZAJE
62             proSTR(1:3)=1/3; %-----ASIGNACION DE ...
                    PROBABILIDAD INICIAL PARA CADA ESTRTEGIA DE ...
                    ACTUALIZACION DE V
63             S(1:3)=0; %-----ACUMULADOR DE LAS ...
                    CUATRO ESTRATEGIAS
64
65             sumlog=0;
66             for k=1:1:ps
67                 sumlog=log(k)+sumlog;
68             end
69             w=(1:ps);
70             w=ps-w+1;
71             w=log(w)/sumlog;
72

```

## ALGORITMO PARA LA DETECCIÓN DE DAÑOS

```

73     Pci=(1:ps);
74     Pci=0.05+0.45*((exp((10.*(Pci-1))/(ps-1))-1)/(exp(10)-1));
75
76     %% COANTADOR DE TIEMPO
77     tic;
78
79     %% DISTRIBUCION DE LA VELOCIDAD(vi) Y LA POSICION(xi) INICIAL
80     [xi,vi]=dis_inic(ps,d,lim,p);
81
82     %% ASIGNACION DE LA ESTRATEGIA PARA v
83     strategy_pbest=round(rand(ps,1).*(3-1)+1);
84
85     %% CALCULO DE LA FUNCION OBJETIVO
86     [fobj]=fittnes1(NFA,xi,ps,Damage,fun);
87
88     %% SELECCION DE LOS MEJORES LOCALES
89     pbest=fobj;
90     coord_pbest=xi;
91
92     %% ELECCION DEL MEJOR GLOBAL
93     gbest=min(pbest);
94     [coord_gbest]=coord_cgbest(ps,gbest,pbest,coord_pbest);
95
96     %% PARAMEROS PARA EL CONTROL DE LA EJECUCION DEL ALGORITMO
97     e=0; %-----CONTADOR
98     des=10; %-----DESVIACION ESTANDAR
99     driver=0; %-----CONTROLADOR PARA EL ...
        PROCESO DE OPTIMIZACION
100
101     %% EJECUCION DEL ALGORITMO
102     while e<max_ite
103
104         clc;
105         disp('*****')
106         disp('*****')
107         disp('***** SL-PSO ...
            *****')
108         disp('*****')
109         disp('*****')
110         disp(' ')
111         e=e+1;
112         str4=['Iteracion numero ', num2str(e)];
113         str5=['Funcion objetivo ', num2str(gbest)];
114         disp(str1)
115         disp(str2)
116         disp(str3)
117         disp(str4)
118         disp(str5)
119
120         if gbest<=1e-5
121             break
122         end

```

## ALGORITMO PARA LA DETECCIÓN DE DAÑOS

```

123
124         x0=xi;
125         v0=vi;
126
127         if e>(Gs*driver)
128             na=rand();
129             driver=driver+1;
130         end
131
132         %% DETERMINACION DE LA VELOCIDAD
133
134         if na<proSTR(1)
135             %% ACTUALIZACION DE v POR MEDIO DE LA ESTRATEGIA 1
136             [vi,strategy]=calc_DbV(ps,x0,coord_pbest);
137
138         elseif na<(proSTR(1)+proSTR(2))
139             %% ACTUALIZACION DE v POR MEDIO DE LA ESTRATEGIA 3
140             [vi,strategy]=calc_CLPSO(ps,d,coord_pbest,v0,x0,c3,...
141                 w0,Pci,e,max_ite,pbest);
142
143         elseif na<(proSTR(1)+proSTR(2)+proSTR(3))
144             %% ACTUALIZACION DE v POR MEDIO DE LA ESTRATEGIA 4
145             [vi,strategy]=calc_PSO_CL_pbest(ps,d,coord_pbest,v0,...
146                 x0,c3,w0,Pci,e,max_ite,pbest);
147         end
148
149         %% CHEQUEO DE LA VELOCIDAD
150         [vi]=check_v(vi,p,d);
151
152         %% CALCULO DE xi A PARTIR DE vi
153         xi=x0+vi;
154
155         %% CHEQUEO DE LA POSICION
156         [xi]=check_x(xi,lim,d);
157
158         %% CALCULO FUNCION OBJETIVO
159         [fobj]=fittnes1(NFA,xi,ps,Damage,fun);
160
161         %% SELECCION gbest,pbest,S,coord_pbest Y coord_gbest
162         [gbest,pbest,S,coord_pbest,strategy_pbest]=select(ps,...
163             pbest,fobj,strategy,strategy_pbest,xi,coord_pbest,S,...
164             w,e,Gs,driver);
165         [coord_gbest]=coord_cgbest(ps,gbest,pbest,coord_pbest);
166
167         %% ASIGNACION DE LA PROBABILIDAD PARA LA SELECCION DE ...
168             LA VELOCIDAD
169         if e==(Gs*driver)
170             proSTRi=(1-alpha)*proSTR+alpha*(S./Gs);
171             S(1:3)=0;
172             sum_proSTRi=sum(proSTRi);
173             proSTR=proSTRi./sum_proSTRi;
174         end

```

## ALGORITMO PARA LA DETECCIÓN DE DAÑOS

```

174
175         Fobjetivo(e,1,q)=gbest;
176     end
177
178     %% ALMACENAMIENTO DE LOS RESULTADOS
179     tabla(q,1)=toc;%-----tiempo
180     tabla(q,2)=e;%-----iteraciones
181     tabla(q,3)=gbest;%-----mejor global
182     CGlobal(q,:)=coord_gbest;%---puntos mejor global
183
184 end
185
186 resultados(1)=mean(tabla(:,1));%-----tiempo promedio
187 resultados(2)=mean(tabla(:,2));%-----promedio de iteraciones
188 resultados(3)=max(tabla(:,3));%-----peor gbest
189 resultados(4)=min(tabla(:,3));%-----mejor gbest
190 resultados(5)=mean(tabla(:,3));%-----promedio gbest
191 resultados(6)=std(tabla(:,3));%-----desviacion estandar gbest
192 resultados(7)=median(tabla(:,3));%----mediana gbest
193
194 save(['PDANO-',num2str(dd),'_FOBJ-',num2str(fun),'_PAR-',...
195     num2str(ps),'_ITE-',num2str(e),'_NFA-',num2str(NFA),...
196     '_DIM-',num2str(d)],'Fobjetivo','tabla','CGlobal','resultados')
197
198 clear tabla CGlobal resultados
199 end
200
201 end

1 % =====
2 % cal_model2: System stiffness & mass matrix calculation
3 %         for PHASE II of the Benchmark Problem of the
4 %         ASCE Task Group on Structural Health Monitoring
5 % =====
6 %
7 % [K,M,T,node,elem,prop]=cal_model_damage;
8 %
9 % =====
10 %
11 % EXAMPLE ==> [K,M] = cal_model_damage;
12 %
13 % =====
14 % by Paul Lam <paullam@ust.hk>, 12-Oct-2001 (modified from cal_model2.m)
15 % Modificado Fredy Sotelo 05-Mayo-2013
16 % =====
17
18 function [K,M,T,node,elem,prop]=cal_model_damage_MSR_01(theta_damage)
19
20 load('AJUSTE');
21

```

## ALGORITMO PARA LA DETECCIÓN DE DAÑOS

```

22
23 column_index_1 = [ 1:9]; % ==> COLUMNAS PISO1
24 column_index_2 = [30:38]; % ==> COLUMNAS PISO2
25 column_index_3 = [59:67]; % ==> COLUMNAS PISO3
26 column_index_4 = [88:96]; % ==> COLUMNAS PISO4
27 beam_index_1 = [10:21]; % ==> VIGAS PISO1
28 beam_index_2 = [39:50]; % ==> VIGAS PISO2
29 beam_index_3 = [68:79]; % ==> VIGAS PISO3
30 beam_index_4 = [97:108]; % ==> VIGAS PISO4
31 brace_index_1 = [22:29]; % ==> RIGIDIZADORES PISO1
32 brace_index_2 = [51:58]; % ==> RIGIDIZADORES PISO2
33 brace_index_3 = [80:87]; % ==> RIGIDIZADORES PISO3
34 brace_index_4 = [109:116]; % ==> RIGIDIZADORES PISO4
35
36 theta(column_index_1,:)=X(1,1);
37 theta(column_index_2,:)=X(1,2);
38 theta(column_index_3,:)=X(1,3);
39 theta(column_index_4,:)=X(1,4);
40 theta(beam_index_1,:)=X(1,5);
41 theta(beam_index_2,:)=X(1,6);
42 theta(beam_index_3,:)=X(1,7);
43 theta(beam_index_4,:)=X(1,8);
44 theta(brace_index_1,:)=X(1,9);
45 theta(brace_index_2,:)=X(1,10);
46 theta(brace_index_3,:)=X(1,11);
47 theta(brace_index_4,:)=X(1,12);
48
49 length_factor_column=X(1,13);
50 length_factor_beam=X(1,14);
51 length_factor_brace=X(1,15);
52
53 factor_node_1=X(1,16);
54 factor_node_2=X(1,17);
55
56
57 % ***** lump_id = 0 : use consistent mass matrix
58 % ***** otherwise : use lump mass matrix
59 lump_id=1;
60
61 % ***** call internal function get_nep for structure information *****
62 % ***** nep stands for node, elem and prop ...
63 [node,elem,prop,L,H]=get_nep;
64
65
66 % ***** theta scale the young's modulus of each member
67 % ***** elem(:,5) is connection flexibility at node i
68 % ***** elem(:,6) is connection flexibility at node j
69 % ***** scale the Young's modulus of the elements according to theta ...
    *****
70 brace_index = [22:29 51:58 80:87 109:116];
71 prop(:,5)=prop(:,5).*theta;
72 prop(brace_index,4)=prop(brace_index,4).*theta_damage';

```

## ALGORITMO PARA LA DETECCIÓN DE DAÑOS

```

73 prop(brace_index,6)=prop(brace_index,6).*theta_damage';
74
75 % ***** factor for connection flexibility *****
76 beam_index_1 = [10:15 39:44 68:73 97:102];
77 beam_index_2 = [16:21 45:50 74:79 103:108];
78 elem(beam_index_1,5)=factor_node_1;
79 elem(beam_index_1,6)=factor_node_1;
80 elem(beam_index_2,5)=factor_node_2;
81 elem(beam_index_2,6)=factor_node_2;
82
83
84 % ***** define the reference point for each story *****
85 pts=[
86     L   L   H
87     L   L  2*H
88     L   L  3*H
89     L   L  4*H
90 ];
91
92 % ***** call internal function cal_T to calculate the transformation ...
93 % ***** for the consideration of rigid-floor effect ...
94 % *****
95 T=cal_T(node,pts);
96
97 % ***** call frame3d to calculate the system stiffness and mass ...
98 % ***** (before transformation) ...
99 % *****
100 [SysK,SysM]=frame3d(node,elem,prop,lump_id,length_factor_column,...
101     length_factor_beam,length_factor_brace);
102
103 % ***** transform the system stiffness and mass matrix *****
104 K=T'*SysK*T;
105 M=T'*SysM*T;
106
107 % ***** add floor masses to the transformed system mass matrix *****
108 amval=[1000 1000 1000 750]*ones(1,4);
109
110 dx=[L L -L -L]/2;
111 dy=[L -L -L L]/2;
112
113 dr=(dx.^2+dy.^2).^0.5;
114
115 % ***** addmatrix (12 by 12) is calculated for the consideration *****
116 % ***** of floor masses *****
117 for i=1:4
118     addmatrix(1+(i-1)*3,1+(i-1)*3)= sum(amval(i,:));
119     addmatrix(2+(i-1)*3,2+(i-1)*3)= sum(amval(i,:));
120     addmatrix(3+(i-1)*3,3+(i-1)*3)= sum(amval(i,)*(2*L^2/12))+...
121         sum(amval(i,).*dr.^2);
122     addmatrix(1+(i-1)*3,3+(i-1)*3)=-sum(amval(i,).*dy);

```



## ALGORITMO PARA LA DETECCIÓN DE DAÑOS

```

121     addmatrix(3+(i-1)*3,1+(i-1)*3)=-sum(amval(i,:).*dy);
122     addmatrix(2+(i-1)*3,3+(i-1)*3)= sum(amval(i,:).*dx);
123     addmatrix(3+(i-1)*3,2+(i-1)*3)= sum(amval(i,:).*dx);
124 end;
125
126 % ***** add addmatrix to the transformed system mass matrix *****
127 M(1:12,1:12)=M(1:12,1:12)+addmatrix;
128
129 % ***** remove very small elements in M *****
130 [Iindx,Jindx]=find(abs(M) < 1.e-10);
131 if isempty(Iindx)≠1
132     for i=1:length(Iindx)
133         M(Iindx(i),Jindx(i))=0;
134     end;
135 end;
136 %*****
137 % END of program cal_model!!
138 %*****
139
140 %=====
141 % get_nep: get the structural parameters NODE, ELE, and PROP.
142 %         The structure is defined in this function!!!
143 %=====
144 function [node,elem,prop,L,H]=get_nep;
145 % ***** L is the bay width, H is the story height *****
146 L=1.25; H=0.9;
147
148 % ***** node coordinates and node condition index*****
149 node=[
150     0      0      0      0 0 0 0 0 0
151     L      0      0      0 0 0 0 0 0
152     2*L    0      0      0 0 0 0 0 0
153     0      L      0      0 0 0 0 0 0
154     L      L      0      0 0 0 0 0 0
155     2*L    L      0      0 0 0 0 0 0
156     0      2*L    0      0 0 0 0 0 0
157     L      2*L    0      0 0 0 0 0 0
158     2*L    2*L    0      0 0 0 0 0 0
159     0      0      H      1 1 1 1 1 1
160     L      0      H      1 1 1 1 1 1
161     2*L    0      H      1 1 1 1 1 1
162     0      L      H      1 1 1 1 1 1
163     L      L      H      1 1 1 1 1 1
164     2*L    L      H      1 1 1 1 1 1
165     0      2*L    H      1 1 1 1 1 1
166     L      2*L    H      1 1 1 1 1 1
167     2*L    2*L    H      1 1 1 1 1 1
168     0      0      2*H    1 1 1 1 1 1
169     L      0      2*H    1 1 1 1 1 1
170     2*L    0      2*H    1 1 1 1 1 1
171     0      L      2*H    1 1 1 1 1 1
172     L      L      2*H    1 1 1 1 1 1

```

## ALGORITMO PARA LA DETECCIÓN DE DAÑOS

```

173     2*L    L    2*H    1  1  1  1  1  1
174     0     2*L   2*H    1  1  1  1  1  1
175     L     2*L   2*H    1  1  1  1  1  1
176     2*L   2*L   2*H    1  1  1  1  1  1
177     0     0     3*H    1  1  1  1  1  1
178     L     0     3*H    1  1  1  1  1  1
179     2*L   0     3*H    1  1  1  1  1  1
180     0     L     3*H    1  1  1  1  1  1
181     L     L     3*H    1  1  1  1  1  1
182     2*L   L     3*H    1  1  1  1  1  1
183     0     2*L   3*H    1  1  1  1  1  1
184     L     2*L   3*H    1  1  1  1  1  1
185     2*L   2*L   3*H    1  1  1  1  1  1
186     0     0     4*H    1  1  1  1  1  1
187     L     0     4*H    1  1  1  1  1  1
188     2*L   0     4*H    1  1  1  1  1  1
189     0     L     4*H    1  1  1  1  1  1
190     L     L     4*H    1  1  1  1  1  1
191     2*L   L     4*H    1  1  1  1  1  1
192     0     2*L   4*H    1  1  1  1  1  1
193     L     2*L   4*H    1  1  1  1  1  1
194     2*L   2*L   4*H    1  1  1  1  1  1
195 ];
196
197 % ***** element connectivity and element group *****
198 elem=[
199     1     10     4     1
200     2     11     5     2
201     3     12     6     3
202     4     13     7     4
203     5     14     8     5
204     6     15     9     6
205     7     16     4     7
206     8     17     5     8
207     9     18     6     9
208    10     11     2    10
209    11     12     3    11
210    13     14     5    12
211    14     15     6    13
212    16     17     8    14
213    17     18     9    15
214    10     13     4    16
215    13     16     7    17
216    11     14     5    18
217    14     17     8    19
218    12     15     6    20
219    15     18     9    21
220     1     11    10    22
221    11     3     12    23
222     3     15    12    24
223    15     9     18    25
224     9     17    18    26

```

## ALGORITMO PARA LA DETECCIÓN DE DAÑOS

225	17	7	16	27
226	7	13	16	28
227	13	1	10	29
228	10	19	13	30
229	11	20	14	31
230	12	21	15	32
231	13	22	16	33
232	14	23	17	34
233	15	24	18	35
234	16	25	13	36
235	17	26	14	37
236	18	27	15	38
237	19	20	11	39
238	20	21	12	40
239	22	23	14	41
240	23	24	15	42
241	25	26	17	43
242	26	27	18	44
243	19	22	13	45
244	22	25	16	46
245	20	23	14	47
246	23	26	17	48
247	21	24	15	49
248	24	27	18	50
249	10	20	19	51
250	20	12	21	52
251	12	24	21	53
252	24	18	27	54
253	18	26	27	55
254	26	16	25	56
255	16	22	25	57
256	22	10	19	58
257	19	28	22	59
258	20	29	23	60
259	21	30	24	61
260	22	31	25	62
261	23	32	26	63
262	24	33	27	64
263	25	34	22	65
264	26	35	23	66
265	27	36	24	67
266	28	29	20	68
267	29	30	21	69
268	31	32	23	70
269	32	33	24	71
270	34	35	26	72
271	35	36	27	73
272	28	31	22	74
273	31	34	25	75
274	29	32	23	76
275	32	35	26	77
276	30	33	24	78

## ALGORITMO PARA LA DETECCIÓN DE DAÑOS

```

277     33     36     27     79
278     19     29     28     80
279     29     21     30     81
280     21     33     30     82
281     33     27     36     83
282     27     35     36     84
283     35     25     34     85
284     25     31     34     86
285     31     19     28     87
286     28     37     31     88
287     29     38     32     89
288     30     39     33     90
289     31     40     34     91
290     32     41     35     92
291     33     42     36     93
292     34     43     31     94
293     35     44     32     95
294     36     45     33     96
295     37     38     29     97
296     38     39     30     98
297     40     41     32     99
298     41     42     33    100
299     43     44     35    101
300     44     45     36    102
301     37     40     31    103
302     40     43     34    104
303     38     41     32    105
304     41     44     35    106
305     39     42     33    107
306     42     45     36    108
307     28     38     37    109
308     38     30     39    110
309     30     42     39    111
310     42     36     45    112
311     36     44     45    113
312     44     34     43    114
313     34     40     43    115
314     40     28     37    116
315 ];
316
317 % ***** (1) column, (2) beam and (3) brace properties *****
318 %     Iy  —  the moment of inertia in the local major axis
319 %     Iz  —  the moment of inertia in the local minor axis
320 %     J   —  the torsional constant
321 %     A   —  the cross-sectional area
322 %     E   —  the Young's Modulus
323 %     mbar— the mass per unit length
324 %     Io  —  polar moment of inertia of the cross-sectional area A
325 %     G   —  modulus of elasticity in shear
326 Iz1 = 1.97e-6;      Iz2 = 1.22e-6;      Iy3 = 0;
327 Iy1 = 0.664e-6;     Iy2 = 0.249e-6;     Iz3 = 0;
328 J1  = 8.01e-9;      J2  = 3.82e-8;      J3  = 0;

```

## ALGORITMO PARA LA DETECCIÓN DE DAÑOS

```

329 A1      = 1.133E-3;      A2      = 1.430E-3;      A3      = 0.141E-3;
330 E1      = 2.0e11;      E2      = E1;      E3      = E1;
331 mbar1    = 7800*A1;      mbar2    = 7800*A2;      mbar3    = 7800*A3;
332 Io1      = Iy1+Iz1;      Io2      = Iy2+Iz2;      Io3      = 0;
333 G1      = E1/2/(1+0.3);  G2      = E2/2/(1+0.3);  G3      = E3/2/(1+0.3);
334
335 % ***** floor      1      2      3      4
336 column_index = [ 1:9  30:38 59:67 88:96];
337 beam_index   = [10:21 39:50 68:79 97:108];
338 brace_index  = [22:29 51:58 80:87 109:116];
339
340 % additional columns to consider connection flexibility at two ends ...
    of the member
341 % 1.e20  -> rigid connection;
342 % 1.e-20 -> pin connection;
343 elem(:,5:6) = 1.e20;
344 elem(beam_index,5:6) = 1.e20;
345
346 % ***** generation of prop
347 prop(column_index,:) = ones(length(column_index),1)*[Iz1 Iy1 J1 A1 E1 ...
348     mbar1 Io1 G1];
349 prop(beam_index, :) = ones(length(beam_index), 1)*[Iz2 Iy2 J2 A2 E2 ...
350     mbar2 Io2 G2];
351 prop(brace_index, :) = ones(length(brace_index), 1)*[Iz3 Iy3 J3 A3 E3 ...
352     mbar3 Io3 G3];
353
354 %=====
355 % cal_T: Calculate the transformation matrix T
356 %      for the consideration of the rigid-floor effect.
357 %=====
358 function T=cal_T(node,pts);
359 if nargin < 2, error('not enough input parameters ...'); end;
360 Dlnum=sum(sum(node(:,4:6)));
361 [Pnum,Pdim]=size(pts);
362 Tcount=1;
363 addnum=Pnum*3+1;
364 for i=1:Pnum
365     indx=find(node(:,3)==pts(i,3));
366     vv=[1+(i-1)*3:i*3];
367     for j=1:length(indx)
368         if node(indx(j),4)==1
369             T(Tcount,vv)=[1 0 pts(i,2)-node(indx(j),2)];
370             Tcount=Tcount+1;
371         end;
372         if node(indx(j),5)==1
373             T(Tcount,vv)=[0 1 -(pts(i,1)-node(indx(j),1))];
374             Tcount=Tcount+1;
375         end;
376         if node(indx(j),6)==1
377             T(Tcount,addnum)=1;
378             Tcount=Tcount+1; addnum=addnum+1;
379         end;

```

## ALGORITMO PARA LA DETECCIÓN DE DAÑOS

```

380         if node(indx(j),7)==1
381             T(Tcount,addnum)=1;
382             Tcount=Tcount+1; addnum=addnum+1;
383         end;
384         if node(indx(j),8)==1
385             T(Tcount,addnum)=1;
386             Tcount=Tcount+1; addnum=addnum+1;
387         end;
388         if node(indx(j),9)==1
389             T(Tcount,vv)=[0 0 1];
390             Tcount=Tcount+1;
391         end;
392     end;
393 end;
394
395 %=====
396 % frame3d: Assemble the system stiffness and mass matrix.
397 %=====
398 function ...
399     [SysK, SysM]=frame3d(node,elem,prop,lump_id,length_factor_column,...
400         length_factor_beam,length_factor_brace);
401 if nargin < 4, error('not enough input parameters ...'); end;
402 [nnum,nn]=size(node);
403 [enum,ee]=size(elem);
404 [pnum,pp]=size(prop);
405 DOFnum=sum(sum(node(:,4:9)));
406 Ks=zeros(6*nnum); Ms=zeros(6*nnum);
407 v1=[1:6]; v2=[7:12];
408 for i=1:enum
409     % ***** calculate the element length *****
410     Le=sum((node(elem(i,1),1:3)-node(elem(i,2),1:3)).^2).^0.5;
411     if Le==0.9
412         L=Le*length_factor_column;
413     elseif Le==1.25
414         L=Le*length_factor_beam;
415     else
416         L=Le*length_factor_brace;
417     end
418     % ***** calculate the element local stiffness and mass matrix *****
419     [Ke,Me]=fe3d( ...
420         prop(elem(i,4),1), ...
421         prop(elem(i,4),2), ...
422         prop(elem(i,4),3), ...
423         prop(elem(i,4),4), ...
424         L, ...
425         prop(elem(i,4),5), ...
426         prop(elem(i,4),6), ...
427         prop(elem(i,4),7), ...
428         prop(elem(i,4),8), ...
429         lump_id, ...
430         elem(i,5), ...
431         elem(i,6));

```

## ALGORITMO PARA LA DETECCIÓN DE DAÑOS

```

431 % ***** calculate the element transformation matrix *****
432 T=tf3d_geradin(node(elem(i,1),1:3),node(elem(i,2),1:3),...
433     node(elem(i,3),1:3));
434 % ***** calculate the element global stiffness and mass matrix *****
435 Kt=T'*Ke*T; Mt=T'*Me*T;
436 % ***** assemble the system stiffness matrix *****
437 DOFi=elem(i,1)*6-5; vi=[DOFi:DOFi+5];
438 DOFj=elem(i,2)*6-5; vj=[DOFj:DOFj+5];
439 Ks(vi,vi)=Kt(v1,v1)+Ks(vi,vi);
440 Ks(vj,vj)=Kt(v2,v2)+Ks(vj,vj);
441 Ks(vi,vj)=Kt(v1,v2)+Ks(vi,vj);
442 Ks(vj,vi)=Kt(v2,v1)+Ks(vj,vi);
443 % ***** assemble the system mass matrix *****
444 Ms(vi,vi)=Mt(v1,v1)+Ms(vi,vi);
445 Ms(vj,vj)=Mt(v2,v2)+Ms(vj,vj);
446 Ms(vi,vj)=Mt(v1,v2)+Ms(vi,vj);
447 Ms(vj,vi)=Mt(v2,v1)+Ms(vj,vi);
448 end;
449 % ***** extract movable dofs from Ks and Ms *****
450 Kvec=mat2vec(node(:,4:9));
451 indx=find(Kvec==1);
452 SysK=Ks(indx,indx);
453 SysM=Ms(indx,indx);
454
455 %=====
456 % fe3d: Formation of the element stiffness and mass matrix.
457 %=====
458 % For rigid frame element stiffness and mass matrix, please
459 % refer to:
460 %   Structural Dynamics Theory & Computation by Mario Paz
461 %
462 % For stiffness matrix for frame element with flexible
463 % connection, please refer to:
464 %
465 %=====
466 function [K,M]=fe3d(Iz,Iy,J,A,L,E,mbar,Io,G,lump_id,fa,fb);
467
468 if Iz == 0 | Iy == 0
469     % ***** I keep this part ONLY because Iz and/or Iy = 0 will ...
470     % ***** flexible connection formulation.
471     % ***** Therefore, this part is for braces ONLY!!
472     K11=[
473         E*A/L    0    0    0    0    0
474         0    12*E*Iz/L^3    0    0    0    ...
475         6*E*Iz/L^2
476         0    0    12*E*Iy/L^3    0    -6*E*Iy/L^2    0
477         0    0    0    G*J/L    0    0
478         0    0    -6*E*Iy/L^2    0    4*E*Iy/L    0
479         0    6*E*Iz/L^2    0    0    0    4*E*Iz/L
480     ];
481     K21=[

```

## ALGORITMO PARA LA DETECCIÓN DE DAÑOS

```

481      -E*A/L  0      0      0      0      0
482      0      -12*E*Iz/L^3  0      0      0      ...
          -6*E*Iz/L^2
483      0      0      -12*E*Iy/L^3  0      6*E*Iy/L^2  0
484      0      0      0      -G*J/L  0      0
485      0      0      -6*E*Iy/L^2  0      2*E*Iy/L  0
486      0      6*E*Iz/L^2  0      0      0      2*E*Iz/L
487  ];
488  K22=[
489      E*A/L  0      0      0      0      0
490      0      12*E*Iz/L^3  0      0      0      ...
          -6*E*Iz/L^2
491      0      0      12*E*Iy/L^3  0      6*E*Iy/L^2  0
492      0      0      0      G*J/L  0      0
493      0      0      6*E*Iy/L^2  0      4*E*Iy/L  0
494      0      -6*E*Iz/L^2  0      0      0      4*E*Iz/L
495  ];
496  K=[
497      K11      K21'
498      K21      K22
499  ];
500  else
501      % ***** for beam and column elements!! *****
502      Ya=fa/(fa+3);
503      Yb=fb/(fb+3);
504
505      f1=Ya+Yb+Ya*Yb;
506      f2=Ya*(2+Yb);
507      f3=3*Ya;
508      f4=3*Yb;
509      f5=3*Ya*Yb;
510      f6=Yb*(2+Ya);
511      f7=4-Ya*Yb;
512
513      K11=[
514          E*A/L  0      0      0      0      ...
                    0
515          0      12*E*Iz/L^3*f1/f7  0      0      0      ...
                    6*E*Iz/L^2*f2/f7
516          0      0      12*E*Iy/L^3*f1/f7  0      ...
          -6*E*Iy/L^2*f2/f7  0
517          0      0      0      G*J/L  0      ...
                    0
518          0      0      -6*E*Iy/L^2*f2/f7  0      ...
          4*E*Iy/L*f3/f7  0
519          0      6*E*Iz/L^2*f2/f7  0      0      0      ...
                    4*E*Iz/L*f3/f7
520  ];
521      K21=[
522          -E*A/L  0      0      0      0      ...
                    0
523          0      -12*E*Iz/L^3*f1/f7  0      0      0      ...

```



## ALGORITMO PARA LA DETECCIÓN DE DAÑOS

```

524         0      0       $\frac{-6*E*I_z/L^2*f2/f7}{6*E*I_y/L^2*f2/f7}$        $\frac{-12*E*I_y/L^3*f1/f7}{0}$       0      ...
525         0      0      0      0       $-G*J/L$  0 ...
526         0      0       $\frac{2*E*I_y/L*f5/f7}{6*E*I_z/L^2*f6/f7}$        $\frac{-6*E*I_y/L^2*f6/f7}{0}$       0      ...
527         0       $\frac{6*E*I_z/L^2*f6/f7}{2*E*I_z/L*f5/f7}$       0      0      0 ...
528     ];
529     K22=[
530         E*A/L      0      0      0      0      0 ...
531         0       $\frac{12*E*I_z/L^3*f1/f7}{-6*E*I_z/L^2*f6/f7}$       0      0      0 ...
532         0      0       $\frac{12*E*I_y/L^3*f1/f7}{6*E*I_y/L^2*f6/f7}$       0      ...
533         0      0      0      0       $G*J/L$  0 ...
534         0      0       $\frac{6*E*I_y/L^2*f6/f7}{4*E*I_y/L*f4/f7}$       0      ...
535         0       $\frac{-6*E*I_z/L^2*f6/f7}{4*E*I_z/L*f4/f7}$       0      0      0 ...
536     ];
537     K=[
538         K11      K21'
539         K21      K22
540     ];
541 end
542 if lump_id==0
543     % ***** consistent mass matrix *****
544     M11=[
545         140      0      0      0      0      0
546         0      156      0      0      0      22*L
547         0      0      156      0      -22*L      0
548         0      0      0       $140*I_o/A$  0      0
549         0      0      -22*L      0       $4*L^2$  0
550         0      22*L      0      0      0       $4*L^2$ 
551     ];
552     M21=[
553         70      0      0      0      0      0
554         0      54      0      0      0      13*L
555         0      0      54      0      -13*L      0
556         0      0      0       $70*I_o/A$  0      0
557         0      0      13*L      0      -3*L^2      0
558         0      -13*L      0      0      0      -3*L^2
559     ];
560     M22=[
561         140      0      0      0      0      0
562         0      156      0      0      0      -22*L
563         0      0      156      0      22*L      0
564         0      0      0       $140*I_o/A$  0      0

```

## ALGORITMO PARA LA DETECCIÓN DE DAÑOS

```

565         0         0         22*Le         0         4*L^2         0
566         0        -22*L         0         0         0         4*L^2
567     ];
568     M=[
569         M11     M21'
570         M21     M22
571     ].*(mbar*L/420);
572 else
573     % ***** lumped mass matrix *****
574     M=[1 1 1 Io/A 0 0 1 1 1 Io/A 0 0].*(mbar*L/2);
575     M=diag(M);
576 end;
577
578 %=====
579 % tf3d: Formation of the element transformation matrix.
580 %=====
581 % Please refer to:
582 %   Structural Dynamics Theory & Computation by Mario Paz
583 %=====
584 function T=tf3d_geradin(pti,ptj,ptk);
585 if nargin < 3, error('not enough input parameters ...'); end;
586 d2=ptj-pti;
587 d3=ptk-pti;
588 ex=d2./norm(d2);
589 ez=cross(d2,d3);
590 ez=ez./norm(ez);
591 ey=cross(ez,ex);
592 T1=[
593     ex(1) ex(2) ex(3)
594     ey(1) ey(2) ey(3)
595     ez(1) ez(2) ez(3)
596 ];
597 v=1:3;
598 for i=1:4
599     vv=v+(i-1)*3;
600     T(vv,vv)=T1;
601 end;
602
603 %=====
604 % mat2vec: Rearrange a matrix to a vector.
605 %=====
606 function xout=mat2vec(xin)
607 if nargin < 1, error('not enough input parameters ...'); end;
608 [nrow,ncol]=size(xin);
609 xout=[];
610 for i=1:nrow
611     xout = [xout xin(i,1:ncol)];
612 end
613
614 1 function [vi,strategy]=calc_CLPSO(ps,d,coord_pbest,v0,x0,c3,w0,Pci,...

```

## ALGORITMO PARA LA DETECCIÓN DE DAÑOS

```

2     e,max_ite,pbest)
3     vi=[];
4     w=(w0-0.2)*(max_ite-e)/(max_ite)+0.2;
5     for i=1:ps
6         for j=1:1:d
7             na=rand;
8             if na<Pci(i)
9                 fd1=0;
10                while fd1==0
11                    xk=round(rand()*(ps-1)+1);
12                    if xk==i
13                        fd1=0;
14                    else
15                        fd1=1;
16                    end
17                end
18
19                fd2=0;
20                while fd2==0
21                    xj=round(rand()*(ps-1)+1);
22                    if xj==i
23                        fd2=0;
24                    else
25                        fd2=1;
26                    end
27                end
28
29                if pbest(xk,1)<pbest(xj,1)
30                    pbestFD(i,j)=coord_pbest(xk,j);
31                else
32                    pbestFD(i,j)=coord_pbest(xj,j);
33                end
34            else
35                pbestFD(i,j)=coord_pbest(i,j);
36            end
37        end
38    end
39
40    r1=rand(ps,d);
41    vi=w*v0+c3*r1.*(pbestFD-x0);
42
43    strategy=2;

1 function [vi,strategy]=calc_DbV(ps,x0,coord_pbest)
2 c1=normrnd(0.5,0.2); %-----COEFICIENTE ESTRATEGIA N-1
3 vi=[];
4 for k=1:ps
5     xk=round(rand()*(ps-1)+1);
6     xj=round(rand()*(ps-1)+1);
7     Viad=x0(xk,:)-x0(xj,:);

```

## ALGORITMO PARA LA DETECCIÓN DE DAÑOS

```
8     vi(k,:) = c1.*Viad + c1.*(coord_pbest(k,:) - x0(k,:));
9 end
10
11 strategy = 1;

1 function [vi, strategy] = calc_PSO_CL_pbest(ps, d, coord_pbest, v0, x0, c3, w0, ...
2     Pci, e, max_ite, pbest)
3 vi = [];
4 vi = [];
5 w = (w0 - 0.2) * (max_ite - e) / (max_ite) + 0.2;
6 for i = 1:ps
7     for j = 1:1:d
8         na = rand;
9         if na < Pci(i)
10             fd1 = 0;
11             while fd1 == 0
12                 xk = round(rand() * (ps - 1) + 1);
13                 if xk == i
14                     fd1 = 0;
15                 else
16                     fd1 = 1;
17                 end
18             end
19
20             fd2 = 0;
21             while fd2 == 0
22                 xj = round(rand() * (ps - 1) + 1);
23                 if xj == i
24                     fd2 = 0;
25                 else
26                     fd2 = 1;
27                 end
28             end
29
30             if pbest(xk, 1) < pbest(xj, 1)
31                 pbestFD(i, j) = coord_pbest(xk, j);
32             else
33                 pbestFD(i, j) = coord_pbest(xj, j);
34             end
35         else
36             pbestFD(i, j) = coord_pbest(i, j);
37         end
38     end
39 end
40
41 r1 = rand(ps, 1);
42
43 for h = 1:1:d
44     r2(:, h) = r1(:, 1);
45 end
```

## ALGORITMO PARA LA DETECCIÓN DE DAÑOS

```
46
47 vi=w*v0+0.5*c3*r2.*(pbestFD-x0+coord_pbest-x0());
48
49 strategy=3;

1 function [vi]=check_v(vi,p,d)
2 vi=round(vi);
3 for i=1:1:d
4     c=(-p(i)≤vi(:,i)) & (vi(:,i)≤p(i));
5     m=vi(:,i).*c;
6     pd=not(c);
7     t=pd.*vi(:,i);
8     e1=-p(i).*(t≤-p(i));
9     e2=p(i).*(t≥p(i));
10    vi(:,i)=m+e1+e2;
11 end
12
13 end

1 function [xi]=check_x(xi,lim,d)
2
3 for i=1:1:d
4     c=(lim(1,i)≤xi(:,i)) & (xi(:,i)≤lim(2,i));
5     m=xi(:,i).*c;
6     pd=not(c);
7     t=pd.*xi(:,i);
8     e1=pd.*(lim(1,i).*(t≤lim(1,i)));
9     e2=pd.*(lim(2,i).*(t≥lim(2,i)));
10    xi(:,i)=m+e1+e2;
11 end
12
13 end

1 function [coord_gbest]=coord_cgbest(ps,gbest,pbest,coord_pbest)
2 for k=1:1:ps
3     if gbest==pbest(k,1)
4         coord_gbest=coord_pbest(k,:);
5         break
6     end
7 end
8 end

1 function [fobj]=fittnes1(NFA,xi,ps,Fr_MEx,fun)
2 %% CALCULO DE LA FUNCION OBJETIVO
3 Fr_ME=Fr_MEx(1:NFA);
4
5 for k=1:ps
```

## ALGORITMO PARA LA DETECCIÓN DE DAÑOS

```

6      %% DEFINICION DEL DANO A PARTIR DE xi (VARIACION EN E)
7
8      theta_damage=xi(k,:);
9
10     %% ENSAMBLE DELA MATRIZ DE RIGIDEZ CON LAS VARIABLES xi
11     [K,M] = cal_model_damage_MSR_01(theta_damage);
12
13     %% DETERMINACION DE LAS FRECUENCIAS
14     [MV,FR]=eig(K,M); %-----CALCULO DE LAS FRECUENCIAS ...
        Y MODOS DE VIBRACION A PARTIR DE [K] Y [M]
15     FR=diag(FR); %-----FRECUENCIAS EN (rad/seg
16     Fr_MN1=sqrt(FR) ./ (2*pi()); %-----FRECUENCIAS EN (HZ)
17     Fr_MN2=sort(Fr_MN1); %-----FRECUANCIAS ORDENADAS DE ...
        MENOR A MAYOR
18
19     Fr_MN=zeros(NFA,1);
20
21     for h=1:1:120
22         mv=zeros(120,1);
23         f1=Fr_MN2(h);
24         a=find(f1==Fr_MN1);
25         mv(:,1)=MV(:,a);
26         mv=abs(mv);
27         mv=mv(10:12);
28
29         if mv(1,1)>0.8 %=====> Modox
30             if Fr_MN(1,1)==0
31                 Fr_MN(1,1)=f1;
32                 MVI(:,1,1)=mv;
33             else
34                 Fr_MN(4,1)=f1;
35                 MVI(:,4,1)=mv;
36             end
37
38         elseif mv(2,1)>0.8 %=====> Modoy
39             if Fr_MN(2,1)==0
40                 Fr_MN(2,1)=f1;
41                 MVI(:,2,1)=mv;
42             else
43                 Fr_MN(5,1)=f1;
44                 MVI(:,5,1)=mv;
45             end
46
47         elseif mv(3,1)>0.8 %=====> Rot
48             Fr_MN(3,1)=f1;
49             MVI(:,3,1)=mv;
50         else
51             end
52
53         if sum(Fr_MN==0)==0
54             break
55         else

```

## ALGORITMO PARA LA DETECCIÓN DE DAÑOS

```

56
57     end
58
59     clear mv
60
61     end
62
63     clear theta
64
65     switch fun
66     case 1
67         %% FUNCION OBJETIVO N-1
68         A=ones(NFA,1);
69         F1=abs(A-Fr_ME./Fr_MN);
70         F2=sum(F1);
71         fobj(k,1)=F2;
72         %fobj(k,1)=sqrt(F2);
73
74     case 2
75         %% FUNCION OBJETIVO N-2
76         F1=(Fr_ME-Fr_MN).^2;
77         F2=sum(F1);
78         fobj(k,1)=sqrt(F2);
79
80     case 3
81         %% FUNCION OBJETIVO N-3
82         F1=abs((Fr_MN-Fr_ME)./Fr_ME);
83         fobj(k,1)=sum(F1);
84
85     end
86
87 end

1 function [organizer]=organizer_pbest(ps,pbest,strategy)
2 disp('Organizacion de pbest'),pause
3 ps;
4 pbest;
5 strategy;
6 organizer=sort(pbest);
7 strategy_pbest=[];
8 for k=1:1:ps
9     for h=1:1:ps
10        if pbest(h,1)==organizer(h,1)
11            strategy_pbest(k,1)=strategy(k,1);
12            break
13        end
14    end
15 end
16 strategy_pbest;

```

## ALGORITMO PARA LA DETECCIÓN DE DAÑOS

```
1  function [gbest,pbest,S,coord_pbest,strategy_pbest]=select (ps,pbest,fobj...
2      ,strategy,strategy_pbest,xi,coord_pbest,S,w,e,Gs,driver)
3  %% SELECCION DE LOS MEJORES LOCALES
4  for k=1:1:ps
5      if fobj(k,1)<pbest(k,1)
6          pbest(k,1)=fobj(k,1);
7          coord_pbest(k,:)=xi(k,:);
8          strategy_pbest(k,1)=strategy;
9      else
10         pbest(k,1)=pbest(k,1);
11         coord_pbest(k,:)=coord_pbest(k,:);
12         strategy_pbest(k,1)=strategy_pbest(k,1);
13     end
14 end
15
16 %% SELECCION DEL MEJOR GLOBAL
17 gbest=min(pbest);
18
19 %% ASIGNACION DEL PESO PARA LA ASIGNACION DE PROBABILIDAD
20 if e==(Gs*driver)
21     organizer=sort(pbest);
22     strategy_S=[];
23     for k=1:1:ps
24         for h=1:1:ps
25             if organizer(k,1)==pbest(h,1)
26                 strategy_S(k,1)=strategy_pbest(h,1);
27                 S(strategy_S(k,1))=S(strategy_S(k,1))+w(k);
28                 break
29             end
30         end
31     end
32 end
33
34 end

1  function ...
2      [gbest,pbest,coord_pbest]=select_pbest (ps,pbest,fobj,xi,coord_pbest)
3  for k=1:1:ps
4      if fobj(k,1)<pbest(k,1)
5          pbest(k,1)=fobj(k,1);
6          coord_pbest(k,:)=xi(k,:);
7      else
8          pbest(k,1)=pbest(k,1);
9          coord_pbest(k,:)=coord_pbest(k,:);
10     end
11 end
12 gbest=min(pbest);
```