

01173
12



UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO

FACULTAD DE INGENIERÍA

DIVISIÓN DE ESTUDIOS DE POSGRADO

DISEÑO ÓPTIMO EVOLUTIVO DE
ARMADURAS PLANAS

T E S I S

QUE PARA OBTENER EL GRADO DE:
MAESTRO EN INGENIERÍA (MECÁNICA)

P R E S E N T A :

FERNANDO VELÁZQUEZ VILLEGAS

DIRECTOR DE TESIS:
DR. SAÚL D. SANTILLÁN GUTIÉRREZ

TESIS CON
FALLA DE ORIGEN

CIUDAD UNIVERSITARIA

2003.



A



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

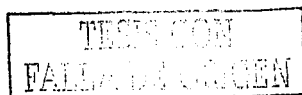
Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

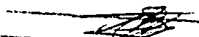
ÍNDICE

Introducción.....	1
Capítulo 1 Marco Teórico.....	4
1.1 OPTIMIZACIÓN ESTRUCTURAL.....	4
1.2 DISEÑO ÓPTIMO EVOLUTIVO (DOPE).....	5
1.3 OBJETIVOS.....	9
Capítulo 2 Computación Evolutiva.....	11
2.1 MÉTODOS DE LA COMPUTACIÓN EVOLUTIVA.....	11
2.1.a Recocido Simulado.....	12
2.1.b Programas Evolutivos.....	13
2.1.c Estrategias Evolutivas.....	14
2.1.d Programación Genética.....	14
2.1.e Programación Evolutiva.....	15
2.1.f Clasificadores Genéticos.....	16
2.1.g Algoritmos Miméticos.....	16
2.1.h Eliminación de material superfluo.....	17
2.2 ALGORITMOS GENÉTICOS.....	18
2.2.a Diferencias entre los Métodos de Búsqueda Comunes y los GA.....	19
2.2.b Mecanismos de un Algoritmo Genético Sencillo.....	21
2.2.c Efectos de los Operadores Reproducción y Crossover.....	23
2.2.d El Algoritmo Genético Sencillo.....	25
Capítulo 3 Características y Solución de Armaduras Planas.....	28
3.1 ARMADURAS PLANAS.....	28
3.1.a Cálculo del Grado de Indeterminación de Armaduras Planas.....	29
3.1.b Inestabilidad Geométrica.....	31
3.1.c Métodos de nodos y de las secciones.....	31
3.2 ANALISIS POR EL MÉTODO DEL ELEMENTO FINITO.....	32
Método del Elemento Finito aplicado a armaduras.....	33
Capítulo 4 Implementación del Algoritmo Genético al Problema de Optimización Estructural de Armaduras Planas.....	39
4.1 DEFINICIÓN DEL DOMINIO DE DISEÑO.....	39
4.2 RESTRICCIONES Y REQUERIMIENTOS.....	41
4.3 CODIFICACIÓN DEL DOMINIO DE DISEÑO.....	41
4.4 FUNCIÓN OBJETIVO.....	43
4.5 ANÁLISIS POR ELEMENTO FINITO.....	44
4.6 ALGORITMO GENÉTICO EMPLEADO.....	45
4.6.a Crossover de doble punto.....	45
4.6.b Elitismo.....	47
4.7 POBLACIÓN INICIAL.....	48
4.8 FILTROS.....	48
4.8.a Filtro por Inestabilidad Geométrica.....	48
4.8.b Filtro de Singularidad.....	49
4.8.c Filtro por Fluencia.....	49

B



4.8.d Estructuras no Rechazadas	49
4.9 CRITERIOS DE PARO	50
4.10 RESULTADOS	50
Capítulo 5 El Programa OPTI	55
5.1 DESCRIPCIÓN DEL PROGRAMA	55
5.2 INICIO DEL PROGRAMA	55
5.3 VIZUALIZAR	58
5.4 RESTRICCIONES Y MATERIAL	59
5.5 OPTIMIZACIÓN	61
Capítulo 6. Aplicación del Programa OPTI en la Determinación de Armaduras Planas Óptimas	64
6.1 CASO 1	64
6.2 CASO 2	67
6.3 CASO 3	69
6.4 CASO 4	72
6.5 CASO 5	76
6.6 CASO 6	79
Discusión de Resultados y Conclusiones	84
Perspectivas	86
Referencias	88

FIRMA: 
 FECHA: 26-VI-03
 NOMBRE: Fernando Velázquez Villegas
 contenido de mi trabajo recien
 UNAM a través de la Dirección General de Bibliotecas

TESIS CON
 FALLA DE ORIGEN

C

DEDICADA

A MIS PADRES:

Teresa Villegas Anaya y Rodolfo Velázquez Elizalde

Con profundo agradecimiento, cariño y respeto. De ustedes he aprendido que sólo el esfuerzo y la dedicación permiten alcanzar lo deseado. Les agradezco la confianza y el apoyo incondicional que siempre me brindan

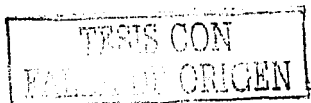
TESIS CON
FALDA DE ORIGEN

RECONOCIMIENTOS

Al Dr. Saúl Daniel Santillán Gutiérrez, director de esta tesis. Por compartir su experiencia como ingeniero para el desarrollo de este trabajo.

A todos los sinodales: Dr. Marcelo López Parra, Dr. Vicente Borja Ramírez, Dr. Jesús Manuel Dorador González, M en I. Leopoldo González González por sus valiosas observaciones a este trabajo.

Al M en I. Álvaro Ayala Álvarez y al Ing. Lázaro Morales por las facilidades prestadas durante las pruebas al programa OPTI.



AGRADECIMIENTOS

A la Universidad Nacional Autónoma de México y a la División de Estudios de Posgrado de la Facultad de Ingeniería por los conocimientos adquiridos durante mi formación como universitario.

Al Centro de Diseño y Manufactura (CDM) por abrirme sus puertas durante los estudios de maestría.

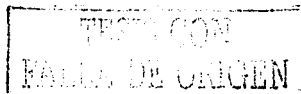
Al Consejo Nacional de Ciencia y Tecnología (CONACYT) por el apoyo brindado en mi formación como Maestro en Ingeniería.

Al personal del CDM y en especial a la Sra. Ana Orozco Jiménez.

Al grandioso grupo de amigos que siempre me ha rodeado por el privilegio de su amistad y apoyo.

A mis hermanos por todo lo que hemos compartido y por ser complemento en mi vida.

A mis padres por ser guía de mi camino y porque les debo todo lo que soy.



F

INTRODUCCIÓN

El diseño estructural es una actividad desempeñada por gente conocedora del comportamiento mecánico de los materiales y de las estructuras. El desarrollo de un diseño eficiente, por tanto, depende de la experiencia de aquellas personas llamadas diseñadores.

Varias son las metodologías que se emplean al diseñar una estructura o producto en general; sin embargo, todas conllevan actividades específicas efectuadas por un diseñador: elección de criterios, modelado y análisis de soluciones, etc.

En particular, el diseño convencional de estructuras rígidas tipo armadura involucra proponer geometrías de las mismas y efectuar su análisis. Lo segundo amerita una serie de cálculos repetitivos y complejos definidos en los métodos de nodos y secciones. Otra opción es aplicar el Análisis por Elemento Finito (FEA).

FEA es un método que ha cobrado popularidad en las últimas décadas gracias a su capacidad para solucionar problemas de ingeniería y a la aparición de equipo de cómputo cada vez más poderoso. FEA implica cálculos matemáticos complicados de desarrollar manualmente, por lo que lo indicado es el empleo de computadoras para explotar toda su riqueza.

Otros métodos computacionales que se han difundido gracias a la aparición de las computadoras son los métodos de la Computación Evolutiva. Estos métodos proponen algoritmos que imitan el comportamiento de la naturaleza a nivel genético realizando una metáfora de la evolución de las especies. El principal atractivo de estos métodos es que permiten resolver problemas de la mejor manera, es decir, determinando la solución que tenga el mayor desempeño ante el problema en cuestión. A ésta se le llama solución óptima.

En este trabajo se propone un concepto de diseño que fusiona el FEA con la Computación Evolutiva para resolver problemas de diseño estructural, dando lugar al Diseño Evolutivo Óptimo (DOPE). Dicha fusión se desarrolla en un algoritmo computacional que

suma las virtudes del FEA con las de los Algoritmos Genéticos (GA por sus siglas en inglés) para determinar la topología de armaduras planas sometidas a cargas y restricciones nodales.

En este reporte escrito se plasman las ideas que facilitan la comprensión y dan sustento a la propuesta hecha. En el Capítulo 1 se plantea el concepto DOPE como una alternativa eficiente para el diseño estructural y se le compara con el diseño convencional, mostrando las ventajas del primero. En seguida se plasman los objetivos que se plantearon en este trabajo.

El Capítulo 2 hace una revisión de las diferentes técnicas de la Computación Evolutiva haciendo énfasis en los GA ya que son parte fundamental del algoritmo propuesto.

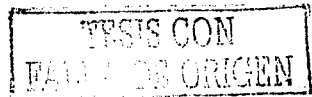
En el Capítulo 3 se encuentran algunas generalidades de las armaduras planas. Se describen brevemente sus características y los métodos de nodos y secciones para solucionarlas. La metodología del FEA para resolver armaduras planas se puede consultar en un apartado este mismo capítulo.

El algoritmo computacional que se diseña en este trabajo cuenta con particularidades propias del problema de las armaduras planas. Básicamente se trata de implementar o fusionar el FEA, los GA y las características de las armaduras planas. Esta tarea se describe en el Capítulo 4.

Resultado de aplicar el algoritmo desarrollado es el programa OPTI. La función de éste es determinar la armadura que mejor comportamiento tenga ante un sistema de cargas y restricciones aplicadas en un dominio de diseño. Una explicación del empleo del programa OPTI se encuentra a lo largo del Capítulo 5.

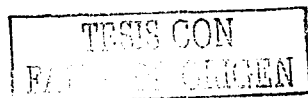
OPTI se prueba con 6 casos de estudio mostrados en el Capítulo 6. Los resultados demostrarán la factibilidad del concepto DOPE en el diseño de armaduras planas.

Para finalizar este reporte se hace una discusión de los resultados obtenidos en los casos de estudio y se enlistan algunas conclusiones al respecto.



CAPÍTULO 1

MARCO TEÓRICO



CAPÍTULO 1 MARCO TEÓRICO

1.1 OPTIMIZACIÓN ESTRUCTURAL

En los últimos años el Análisis por Elemento Finito (FEA por sus siglas en inglés) se ha convertido en una herramienta ampliamente usada por ingenieros de muchas disciplinas.

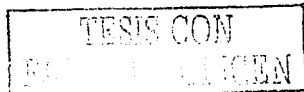
Por otra parte, la popularidad alcanzada en la práctica por la Optimización Estructural fue menor, hasta hace unos cuantos años, a pesar del extraordinario progreso de la teoría de optimización y los algoritmos asociados a ésta en las últimas décadas. La razón se encuentra, en parte, en las dificultades y la complejidad de implementar los métodos de optimización estructural (Xie y Steven, 1997).

El avance en materia de cómputo y desarrollo de equipos cada vez más poderosos y veloces ha permitido que se mire hacia los métodos de optimización para encontrar una forma eficiente de obtener soluciones de alto desempeño en el diseño estructural.

La Optimización Estructural es una fusión de diferentes áreas tales como ingeniería, matemáticas, ciencia y tecnología que tiene como objetivo alcanzar el mejor desempeño para una estructura.

Hasta hace un par de décadas, este tópico permaneció sólo en el interés de académicos debido a la complejidad matemática que implica. El desarrollo y aplicación creciente del FEA fomentó el deseo de contar con herramientas de diseño que vayan más allá de la simple inspección de la funcionalidad de un diseño dado. Al mismo tiempo dos factores promovieron el interés por los métodos de optimización. Primero, el desarrollo de equipo de cómputo poderoso y de bajo costo; segundo, las rápidas mejoras de los algoritmos empleados en la optimización de diseños en los cuales miles de variables y restricciones pueden ser manejadas, es decir, algoritmos cada vez más robustos (Xie y Steven 1997).

Con el tiempo, varios métodos demostraron ser capaces de generar soluciones óptimas a problemas estructurales de diversa índole. Ejemplos de estos son los Métodos de Búsqueda



Basados en Parámetros (Vanderplaats 1984), los criterios agrupados bajo el título "Optimality Criteria" (Rozvany 1989), el Método de Homogeneización (Bendsee 1995), entre otros.

Otros métodos y técnicas de búsqueda de soluciones con desempeño máximo (óptimo) que se desarrollaron desde hace tiempo son aquellos que tienen como base el comportamiento y mecanismos propios de las especies naturales. Estos hacen uso de la computación como herramienta de cálculo, pues el desarrollo manual de los mismos es impráctico. A estos métodos y técnicas se les agrupa en lo que se llama Computación Evolutiva¹.

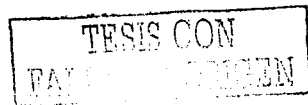
La fusión de las metodologías de la Computación Evolutiva y el diseño estructural resulta en la concepción de un nuevo concepto: el Diseño Óptimo Evolutivo (DOPE). Parte importante de éste es la utilización del FEA como herramienta evaluadora del comportamiento mecánico de estructuras. En otras palabras, se refiere a un método de optimización estructural que emplea FEA como herramienta de cálculo, y utiliza a las estrategias de la Computación Evolutiva como motor de búsqueda para generar soluciones a un problema dado (estructuras) que muestren el mejor desempeño.

1.2 DISEÑO ÓPTIMO EVOLUTIVO (DOPE)

En general, cuando se diseña un producto se cuenta con un conjunto de herramientas de análisis y una base de datos que interactúan con un módulo de evaluación controlado por los objetivos de diseño.

En el proceso creativo del diseño de un producto, la base de datos juega dos papeles principales. En primer lugar, proporciona información del medio en el cual el producto existirá y de los materiales que pueden emplearse en la fabricación del mismo. Segundo, puede contener información de todos aquellos productos que en el pasado fueron creados para resolver el problema. Esta información puede ser interesante si existe interés en: ¿qué funcionó bien y qué funcionó mal en los productos anteriores?, ¿qué puede hacer el diseñador para mejorar el producto?, ¿qué deberían hacer los usuarios para mejorar el desempeño del

¹ Ver Capítulo 2 Computación Evolutiva



producto?, etc. Sin embargo, la información en la base de datos dice muy poco del cómo debe ser la forma del producto para que sea óptimo.

Frecuentemente se emplea la información de la base de datos como fuente para generar la forma de nuevos productos considerando la de los anteriores. De esta manera, la historia de los productos antecesores se ve reflejada en la forma o geometría de un nuevo diseño en lugar de que sea resultado del análisis completo de las leyes que rigen el problema y la comprensión total de los requerimientos. Esta manera de diseñar basándose en la historia del producto puede generar de manera simple una solución adecuada y satisfactoria; sin embargo, descuida la búsqueda de aquel resultado que muestre el máximo desempeño, es decir, aquel que de manera óptima satisfaga los requerimientos de diseño.

Por otra parte, si no existe una base de datos que funcione como fuente histórica, la responsabilidad del diseño recae con mayor peso en el diseñador. En el caso del diseño estructural, debe encargarse de:

- 1) Plantear el problema de diseño
- 2) Proponer posibles soluciones
- 3) Elegir los criterios de diseño a emplear
- 4) Modelar las propuestas
- 5) Aplicar (auxiliándose de algún software) análisis por elemento finito a dichos modelos
- 6) Analizar los resultados y con base en ello elegir la mejor solución de entre las propuestas

Es posible que, con el afán de mejorar el diseño, modifique la propuesta elegida lo que implica repetir varias de las etapas arriba enunciadas. En este caso, es el diseñador quien encabeza el proceso y, en gran medida, el encontrar una solución satisfactoria depende de su experiencia.

Adicionalmente, a pesar de que la solución que obtenga sea satisfactoria, difícilmente será la solución que muestre el mejor desempeño. El desarrollar una solución óptima

implicaría un proceso prolongado de depuración. Por lo tanto, el tiempo de diseño se extendería.

Como se puede observar, en ambos casos (el que exista o no una base de datos) la solución que se obtiene con el diseño convencional no es, en general, la que mejor satisface los requerimientos de diseño.

En este sentido, el Diseño Óptimo Evolutivo supera al diseño convencional pues asegura la obtención de una solución satisfactoria que además es muy cercana a la óptima. Esto gracias a la implementación de métodos de la Computación Evolutiva como motores de búsqueda. Estos métodos realizan una búsqueda a través del espacio de posibles soluciones, eligiendo aquellas que mejor satisfacen los requerimientos. Al final, la solución que encuentra se aproxima a la solución óptima.

La comparación de ambos métodos señala claras ventajas del DOPE sobre el Diseño Convencional:

Diseño Convencional

- Gran parte del proceso es responsabilidad del diseñador
- El diseño obtenido, en general, no es óptimo
- Se analiza una cantidad limitada de propuestas a la vez
- La experiencia del diseñador es fundamental

DOPE

- El modelado del diseño no recae directamente en el diseñador
- El diseño obtenido satisface óptimamente los requerimientos
- Es posible analizar un gran número de propuestas a la vez
- El tiempo necesario en el diseño puede ser abatido

En las Figuras 1.1 y 1.2 se muestran esquemáticamente ambos métodos de diseño.

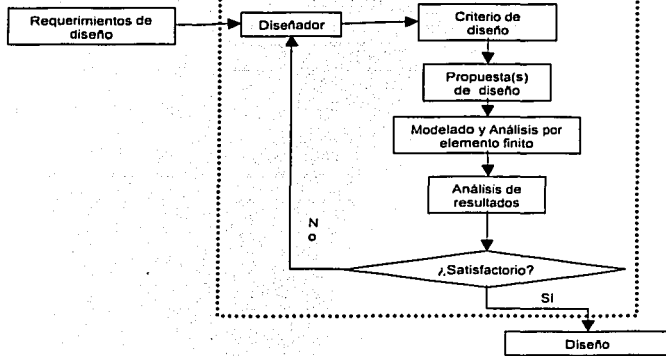


Figura 1.1 Esquema Diseño Conventional

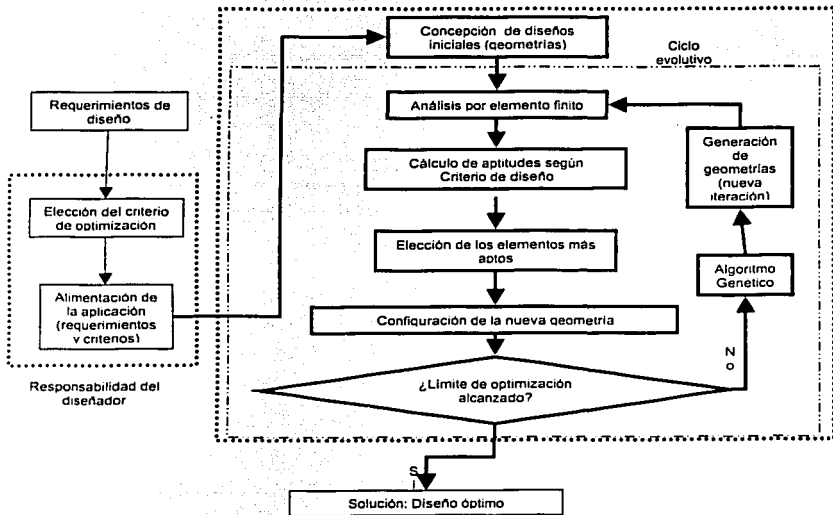
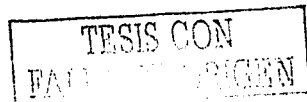


Figura 1.2 Esquema Diseño Evolutivo Óptimo (DOPE)



1.3 OBJETIVOS

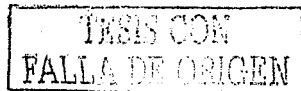
Algunos problemas de optimización estructural se han abordado de manera individual con los métodos de la Computación Evolutiva (Michalewicz 1996, Cane y Schoenauer 1996, Alexandrov 1998 y 2000, Mark 1999, Mark y Foley 1999). El objetivo general de este trabajo es el implementar el concepto DOPE al diseño estructural dentro de una aplicación computacional.

El problema que se aborda aquí es el de encontrar estructuras óptimas tipo armadura² a partir de un dominio de diseño sometido a diferentes restricciones y un sistema de carga, aplicando GA, un método de la Computación Evolutiva, como motor de búsqueda.

Los objetivos que se plantean en este trabajo son los siguientes:

- Desarrollar un algoritmo computacional con base en el concepto DOPE para la optimización de estructuras tipo armadura
- Desarrollar una aplicación computacional con base en el algoritmo desarrollado

² Ver Capítulo 3 Características y Solución de Armaduras Planas



CAPÍTULO 2

COMPUTACIÓN EVOLUTIVA

TESIS CON
FALLA DE ORIGEN

CAPÍTULO 2 COMPUTACIÓN EVOLUTIVA

En el pasado algunas herramientas matemáticas disponibles para la solución de problemas, no eran aplicadas de manera cotidiana por causa de la complejidad de cálculo que representaban. Tales son los casos del FEA y las Técnicas Evolutivas, cuya aplicación exige gran cantidad de cálculos repetitivos y complicados de realizar de manera manual.

El avance tecnológico acelerado promovió el uso de dichos métodos implementándolos dentro de algoritmos computacionales, explotando el poder y velocidad de cálculo de las computadoras.

En los últimos años la aparición de software de FEA a nivel comercial es evidente y se ha convertido en una herramienta casi indispensable en el área de diseño en ingeniería.

Adicionalmente el interés en las Técnicas Evolutivas se ha incrementado gracias a dicho avance, dando lugar a la Computación Evolutiva como un conjunto de técnicas útiles en la solución de problemas complejos.

2.1 MÉTODOS DE LA COMPUTACIÓN EVOLUTIVA

Con el término de Computación Evolutiva se engloba un conjunto de técnicas que, basándose en la imitación de varios procesos naturales que intervienen en la evolución de las especies (selección natural, mutaciones, sobrecruzamientos), tratan de resolver problemas complejos de búsqueda, optimización, aprendizaje, predicción o clasificación.

La Computación Evolutiva trata de obtener aprendizaje imitando la evolución natural (Aprendizaje a nivel de especie). Los métodos considerados en ella basan su funcionamiento en una metáfora de la evolución biológica.

Dentro de la Computación Evolutiva se encuentran los siguientes métodos:

- Algoritmos Genéticos (Genetic Algorithms)
- Recocido Simulado (Simulated Annealing)
- Programas Evolutivos (Evolving Programs)
- Estrategias Evolutivas (Evolutionary Strategies)
- Programación Genética (Genetic Programming)
- Programación Evolutiva (Evolutionary Programming)
- Sistemas de Clasificación
- Algoritmos Miméticos
- Eliminación gradual de material

Partiendo del conocimiento del funcionamiento de los GA, el resto de las técnicas, métodos o algoritmos se pueden interpretar como variaciones o mejoras de los mismos (excepto Recocido Simulado que puede considerarse como una simplificación), añadiendo complejidad o modificando ciertas características del algoritmo.

A continuación se hace una breve revisión de cada uno de estos métodos. Dado que los GA son parte sustancial de este trabajo, una revisión más amplia de ellos se encuentra al final de este capítulo.

2.1.a Recocido Simulado

Su origen está en la Física estadística. El proceso físico de recocido consiste en calentar un sólido hasta la temperatura de recocido, y seguidamente, ir enfriándolo de forma que cristalice en una estructura perfecta, sin malformaciones locales.

En el Recocido Simulado se parte de una solución válida a la que se le provocan pequeñas mutaciones aleatorias. A medida que avanza el algoritmo, éste cada vez será más exigente en cuanto a las soluciones aceptadas. En analogía con la solidificación física, se trata de tener un cuerpo (problema) a alta temperatura (admitiendo soluciones muy diversas a pesar de ofrecer malos resultados) e ir disminuyendo la temperatura (aumentando la exigencia del

algoritmo) de forma que termine por solidificarse según la forma deseada (ofreciendo el mejor resultado).

Este método se ha empleado en múltiples problemas de optimización. Algunos ejemplos de su aplicación son:

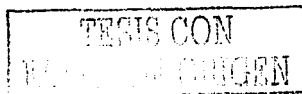
- Optimización de la recarga del núcleo de un reactor (Mahlers 2002)
- Generación de gráficos de buena apariencia empleando líneas rectas (Davidson y Harel 1996)
- Diseño conceptual (O'Sullivan 1998)

2.1.b Programas Evolutivos

Los Programas de Evolutivos son un refinamiento de los GA, propuesto por Michalewicz en 1992 (Michalewicz 1992). En un programa de evolución, la codificación binaria de 0's y 1's empleada por los GA sencillos se amplía hasta admitir cualquier tipo de alfabeto. El hecho de utilizar 0's y 1's evita ciertos problemas que los Programas de Evolución intentan resolver explícitamente, por ejemplo, el cómo se efectúa una mutación. De esta forma se consigue una codificación más cercana al problema a resolver.

Por lo general, los Programas de Evolución se han englobado dentro de los GA (como una variante), y al revés (los GA como un caso particular).

Los Programas de Evolución, también llamados Algoritmos Evolutivos, han demostrado ser útiles en problemas de ingeniería con múltiples restricciones. Los primeros trabajos de Michalewicz consisten en el diseño de una estructura de mínimo peso compuesta por 10 barras unidas rigidamente, cuya variable de diseño es el área transversal de las barras. Otro de sus trabajos fue el diseño de mínimo peso de una armadura tridimensional compuesta por 25 barras (Michalewicz 1996).



2.1.c Estrategias Evolutivas

Las estrategias evolutivas surgieron inicialmente para resolver problemas de optimización paramétrica. Con el paso del tiempo fueron incorporando procedimientos propios de la computación evolutiva, con lo que han llegado a convertirse en una disciplina más.

Las Estrategias Evolutivas Simples son el modelo más sencillo de su paradigma, y aunque no son métodos evolutivos propiamente dichos contienen en su forma más pura gran parte de las ideas que incorporan las Estrategias Evolutivas Múltiples.

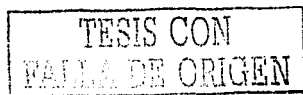
En las Estrategias Evolutivas Simples, la población está compuesta por un solo individuo y únicamente se usa el operador de mutación. Se procede a la mutación del individuo que sustituirá al anterior si es más apto que su antecesor, en caso contrario permanece el individuo anterior.

Aunque los orígenes de ambas disciplinas partieron de enfoques distintos, los sucesivos desarrollos de ambos se han ido acercando hasta el punto de que, actualmente, una Estrategia Evolutiva se puede considerar como un tipo de Algoritmo Genético y viceversa.

Un aspecto característico de las estrategias evolutivas es que las tasas de mutación de cada "gen" se codifican como parte del "genoma", por lo que también se encuentran sujetas a evolución.

Algunas de las aplicaciones de este método son las siguientes

- Desarrollo de software del juego ajedrez (Gross 1998)
- Diseño evolutivo de robots (Floreano 1998)



2.1.d Programación Genética

Así como en los GA los elementos de la cadena (genes) son un tipo de datos que puede ser bits, caracteres o enteros, en Programación Genética los elementos de la cadena (genes) son instrucciones en un lenguaje de programación. La inducción de programas mediante programación genética está muy identificada con los trabajos de John Koza (Koza 1992 y

1997) que hacía evolucionar expresiones en un lenguaje LISP. Koza fue discípulo de Holland (Holland 1975), al igual que Goldberg, y patentó varios de sus algoritmos. Entre otras cosas, los últimos trabajos de Koza ofrecen pruebas de que la programación genética puede implementarse en lenguajes basados en procedimientos como C y C++.

En la literatura se exponen algunos trabajos donde se resuelven problemas empleando Programación Genética

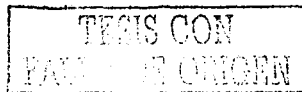
- Control de sistemas en ingeniería (Fleming and Purshousef 2001)
- Redes Neuronales mínimas (Zhang and Mühlenbein 1993)
- Control de robots basado en el aprendizaje del comportamiento (Lee and Zhang 2000)

2.1.e Programación Evolutiva

Este método fue creado por David Fogel (Fogel 1993 a y b). En la Programación Evolutiva los genes codifican un autómata finito o máquina de estados. Cada gen puede consistir en una terna del tipo "estado origen + cadena reconocida + estado destino". En definitiva, cada gen es una regla que especifica las condiciones (cadena reconocida) que se deben cumplir para llegar a un estado (estado destino) suponiendo que la máquina se encuentra en un cierto estado (estado origen).

ESTADO + SUCESO => NUEVO ESTADO

Esta regla tiene una forma imperativa. Puede resumirse en la forma "Si estoy realizando tal acción, y veo tal información, debo obligatoriamente cambiar a realizar tal acción". Si el autómata se encuentra en determinado estado, y ocurre cierto suceso, el autómata pasa a un nuevo estado. La "cadena reconocida" corresponde con los sentidos o sensores del autómata (el suceso), y los estados corresponden con las acciones o actuadores. Este planteamiento tiene grandes similitudes con una red neuronal, y es el más sencillo de programar, pero hace difícil encontrar la explicación de porqué el autómata realiza cierta acción.



Se puede interpretar la misma regla como declarativa, modificando el significado de cada elemento de la regla.

ESTADO + ACCIÓN => NUEVO ESTADO

Esta regla puede resumirse en la forma "Si veo tal información, y realizo tal acción, estimo que como consecuencia de mi acción, la próxima información será tal". En este caso, los estados corresponden con la cadena reconocida y se supone que las acciones del autómata pueden modificar los futuros valores de la cadena reconocida. Este planteamiento tiene grandes similitudes con un sistema experto. Requiere fijar un objetivo al autómata que consistirá en el estado ideal que el autómata debe observar, y permite explicar porqué el autómata realiza una acción.

Casos de estudios abordados con esta técnica son:

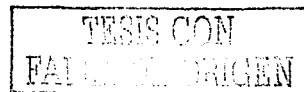
- Generación de particiones adaptativas para la compresión de imágenes fractales (Ruhl y Hartenstein 1997)
- Traductores de idiomas (Davis, M., y Dunning, T 1995, y Aljlal y Frieder 2000)

2.1.f Clasificadores Genéticos

Los Clasificadores Genéticos son algoritmos evolutivos cuya finalidad es obtener un sistema clasificador. Normalmente los clasificadores genéticos hacen evolucionar un conjunto de reglas que serán las encargadas de realizar la clasificación, y en ese caso se pueden considerar como un tipo de Programación Evolutiva.

2.1.g Algoritmos Miméticos

Son la combinación de un algoritmo genético, que se encarga de realizar la búsqueda aplicando el énfasis en la exploración (exploración más a fondo del espacio de búsqueda), con otro algoritmo especializado en la explotación (convergencia más rápida hacia una solución)



2.1.h Eliminación de material superfluo

Éste no es propiamente un método que utilice la metáfora de algún mecanismo natural. Este método propuesto por Xie y Steven (Xie y Steven 1997, Querin 1997) ha sido empleado con gran éxito para generar diseños estructurales óptimos en múltiples casos de estudio.

Consiste en eliminar aquel material que resulte innecesario o superfluo de un Dominio de Diseño sobre el cual se aplica un sistema de solicitaciones (fuerzas, vibraciones, deformaciones, etc.). Parte importante de este método es la discretización de Dominio de Diseño en elementos y la evaluación del mismo empleando FEA. Los resultados del FEA son utilizados para determinar qué elementos deben ser eliminados según un criterio de rechazo. El criterio de rechazo es una función que está compuesta por dos parámetros: Tasa de rechazo (RR) y tasa de evolución (ER). En el caso de optimización según el nivel de esfuerzos, la tasa de rechazo, en cada iteración, es comparada con la relación entre el esfuerzo de cada elemento (σ_e) en el dominio de diseño y el esfuerzo máximo (σ_{\max}) en él.

$$RR_i > \frac{\sigma_e}{\sigma_{\max}}$$

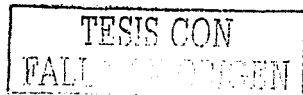
Aquellos elementos que satisfagan la inequación anterior son eliminados de la estructura. El subíndice i se refiere al número de iteración. Después de varias iteraciones es posible que ningún elemento satisfaga la ecuación anterior. En ese momento, la tasa de rechazo es aumentada por el valor de la tasa de evolución.

$$RR_{i+1} = RR_i + ER$$

El proceso continúa hasta que se satisface un criterio de terminación o convergencia. Xie y Steven dan a este método el nombre de Evolutionary Structural Optimization (ESO).

Algunos de los problemas abordados con esta metodología son los siguientes:

- Diseño del soporte para rodamientos
- Diseño de un puente con cargas móviles
- Diseño del marco de una bicicleta de carreras



2.2 ALGORITMOS GENÉTICOS

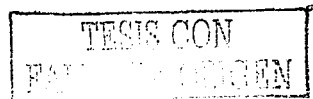
Los GA son algoritmos de búsqueda basados en los mecanismos de selección natural y genéticos naturales. Combinan el concepto de supervivencia del más apto de entre individuos artificiales (cadenas de información codificada de tal forma que representen posibles soluciones al problema) con un intercambio de información estructurado y aleatorio, para formar un algoritmo de búsqueda. En cada generación un conjunto nuevo de criaturas artificiales (cadenas) es creado usando segmentos de los más aptos (Goldberg 1989).

Aunque son algoritmos de búsqueda aleatoria, explotan eficientemente la información histórica para explorar nuevos puntos en el espacio de soluciones con expectativas de encontrar un desempeño mejor.

John Holland, en la Universidad de Michigan, fue el desarrollador de los GA (Holland 1975). Los objetivos de su investigación fueron: a) abstraer y explicar rigurosamente los procesos adaptativos de los sistemas naturales, y 2) diseñar software de sistemas artificiales que retengan los mecanismos importantes de los sistemas naturales.

El concepto principal de interés en AG es su carácter robusto, es decir, que trabajan con un balance entre eficiencia y eficacia, necesario para los individuos sobrevivan en medios diferentes. Las implicaciones de esto para los sistemas artificiales son varias. Si los sistemas artificiales pueden hacerse más robustos, los rediseños costosos pueden ser eliminados o reducidos. Si los más altos niveles de adaptación pueden alcanzarse, los sistemas existentes pueden desempeñar sus funciones de mejor forma.

Las características de auto reparación, auto guía y reproducción son reglas de los sistemas naturales o biológicos, las cuales pueden implementarse en los sistemas artificiales.



2.2.a Diferencias entre los Métodos de Búsqueda Comunes y los GA

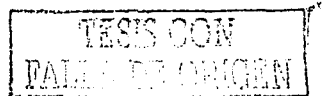
En la literatura se identifican tres tipos principales de métodos de búsqueda: basados en el cálculo, enumerativos y aleatorios.

Los métodos basados en cálculos se subdividen en dos clases principales: indirectos y directos. Los métodos indirectos buscan extremos locales resolviendo el conjunto de ecuaciones usualmente no lineales que resultan de fijar el gradiente de la función objetivo igual a cero. Lo anterior permite encontrar un posible pico (máximo) restringiendo la búsqueda a aquellos puntos con pendiente cero en todas direcciones. Los métodos directos buscan óptimos locales dando "saltos" en la función y moviéndose en la dirección relativa al gradiente local.

Los métodos basados en cálculos no son robustos. Esto se percibe en que son métodos de alcance local (encuentran los óptimos en la vecindad de un punto) y en que dependen de la existencia de al menos una derivada de la función objetivo.

Los esquemas enumerativos buscan en espacios finitos o infinitos discretizados. El algoritmo de búsqueda comienza en los valores de la función objetivo en cada punto del espacio, uno a la vez. Aunque son atractivamente simples, al final deben ser descartados pues no son robustos ya que su eficiencia es baja. En general, los espacios de búsqueda son grandes en la mayoría de los problemas de optimización como para que una búsqueda punto a punto sea adecuada y útil a la vez.

Los métodos aleatorios y los algoritmos asociados a ellos han acumulado popularidad con el paso del tiempo en aquellos campos en donde los métodos basados en cálculos y enumerativos han fracasado. Sin embargo, los buscadores y esquemas aleatorios que encuentran y almacenan a las mejores soluciones son descartados por los requerimientos de eficiencia. Para espacios de búsqueda muy grandes, los métodos aleatorios no son muy diferentes a los enumerativos.



CAPÍTULO 2. COMPUTACIÓN EVOLUTIVA

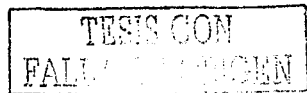
Sin embargo, existen técnicas que utilizan la aleatoriedad como una herramienta guía en la búsqueda. Los GA son un ejemplo de estas técnicas. Ellos utilizan elecciones aleatorias como una herramienta para guiar una búsqueda altamente explosiva a través de un espacio paramétrico codificado. Además, dicha búsqueda se realiza sobre una población de individuos o soluciones al mismo tiempo y no sólo con una solución a la vez. De esta forma, los GA resultan más robustos que los métodos aleatorios. Lo anterior es evidente si se revisan las siguientes diferencias entre los GA y los métodos de búsqueda tradicionales.

1. Los AG trabajan con la codificación de un conjunto paramétrico, no con los parámetros en sí
2. Los AG buscan en una población de puntos, no en un punto simple
3. Los AG emplean información valiosa (función objetivo), no derivadas o alguna otra información auxiliar
4. Los AG usan reglas de transición probabilística, no reglas determinísticas

Los AG requieren el conjunto de parámetros del problema de optimización para codificarlos como una cadena de longitud finita con base en un alfabeto finito.

En muchos métodos se explora cuidadosamente de un punto del espacio de búsqueda al siguiente usando alguna regla de transición para determinar el punto siguiente. Este método punto a punto es peligroso porque es una prescripción perfecta para localizar óptimos falsos en espacios de búsqueda multimodales (con muchos máximos). En contraste, los AG trabajan con una base de datos de puntos simultáneamente (una población de individuos artificiales), explorando muchos máximos en paralelo. Por ello la probabilidad de encontrar picos falsos se reduce.

Muchas técnicas de búsqueda requieren demasiada información auxiliar para desempeñarse adecuadamente (derivadas por ejemplo). Diferentes a estos métodos, los AG son ciegos a esa información pues no requieren de ella. Para realizar la búsqueda de los mejores individuos sólo necesitan los valores de la función objetivo asociada con las cadenas individuales. Esta característica convierte a los AG en un método más canónico que los demás esquemas de búsqueda.



No común a muchos métodos, los AG usan reglas de transición probabilísticas para guiar su búsqueda. Usan elección aleatoria como una herramienta guía al indagar sobre regiones del espacio de soluciones con mejoras atractivas.

Juntas estas cuatro diferencias contribuyen al carácter robusto de los AG y resultan ventajosas comparadas con las técnicas comunes.

2.2.b Mecanismos de un Algoritmo Genético Sencillo

Los mecanismos de un AG sencillo son sorprendentemente simples. No conllevan nada más complejo que copiar cadenas (individuos artificiales) y cambiarlas parcialmente. La simplicidad de operación y el poder de ésta, son dos de los principales atractivos de la aplicación de los AG.

Un AG sencillo que produce buenos resultados en muchos problemas prácticos se compone de tres operadores.

1. Reproducción
2. Crossover
3. Mutación

En el proceso que desarrolla el operador de reproducción las cadenas individuales son copiadas de acuerdo con el valor de sus funciones objetivo "f", también llamada función de aptitud. Esta función puede pensarse como una medida del beneficio, utilidad o bondad que se desea maximizar en el problema. El copiar cadenas con base en el valor de su aptitud significa que cadenas con mayor aptitud tienen una más elevada probabilidad de contribuir en una o más cadenas descendientes en la siguiente generación (iteración).

Este operador no es más que una versión artificial de la selección natural. En poblaciones naturales, la aptitud se determina por la capacidad de una criatura para sobrevivir de depredadores, enfermedades y otros obstáculos en su desarrollo y subsiguiente reproducción. En un conjunto de individuos artificiales, la función objetivo es quien determina si un individuo prevalece o desaparece.

En un AG los individuos seleccionados en el proceso de reproducción son copiados idénticamente y almacenados en lo que se llama "mating pool", una tentativa población nueva que posteriormente se somete a la acción de los otros operadores genéticos.

Posterior al proceso de reproducción, el operador de Crossover puede aplicarse a los miembros de la "mating pool". El operador de Crossover simple se ejecuta en dos pasos: 1) de los miembros del conjunto "mating pool" se eligen aleatoriamente parejas, 2) cada par de cadenas o individuos se somete al operador de Crossover. Éste se desarrolla de la siguiente forma: se selecciona aleatoriamente una posición "k" a lo largo de las cadenas, en el rango de 1 a l-1, donde l es la longitud de la cadena o del individuo. Dos individuos nuevos son engendrados intercambiando todos los caracteres entre las posiciones k+1 y l inclusive de ambas cadenas. La aplicación de este operador entre dos individuos depende del valor de P_c que es la probabilidad de ocurrencia del operador. En general, el valor de P_c se elige entre 0.4 y 0.8.

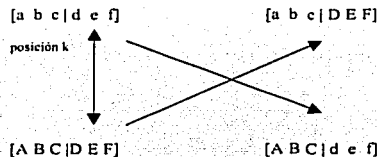


Figura 2. 1 Mecanismo de Crossover

Tanto el mecanismo de reproducción como el de Crossover son sumamente sencillos, pues no incluyen más que generación de números aleatorios, copiar cadenas e intercambiarlas

parcialmente. El énfasis combinado de la reproducción y el intercambio estructurado de información del Crossover brindan a los AG mucho de su poder de búsqueda.

La mutación juega un papel determinante, aunque secundario, en la operación de los AG. La mutación es necesaria porque ocasionalmente la reproducción y el Crossover pierden algo del material genético potencialmente útil de las cadenas de individuos, es decir, pierden diversidad en la población al no indagar sobre algunas regiones del espacio de soluciones. En los sistemas genéticos artificiales, el operador mutación protege contra una pérdida de información irrecuperable. En un AG, la mutación es la ocasional alteración aleatoria (con probabilidad pequeña P_m) del valor de una posición en una cadena. Por sí mismo, la mutación es un buscador aleatorio que trabaja a través del espacio de cadenas o individuos. Cuando se aplica con moderación, junto con la reproducción y el Crossover, la mutación previene la pérdida prematura de nociones importantes. La frecuencia de mutación para obtener buenos resultados en un estudio con AG empíricos es del orden de una mutación por cada mil posiciones intercambiadas o transferidas.

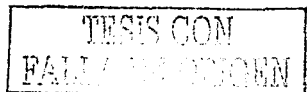


Figura 2. 2 Operador de mutación

2.2.c Efectos de los Operadores Reproducción y Crossover

Parece sorprendente que dos operadores simples y computacionalmente triviales sean tan útiles. Aun es más sorprendente que produzcan un mecanismo de búsqueda robusto. Además, ¿no es extraño que el azar juegue un papel fundamental en un proceso de búsqueda dirigida?

Hadamard (Hadamard 1949) sugiere que aunque el descubrimiento no es un resultado del azar puro, sí puede ayudar en dar dirección a la búsqueda. Por otra parte, en un mecanismo



de descubrimiento de tipo humano, el papel propio del azar es provocar la yuxtaposición de nociones o ideas diferentes.

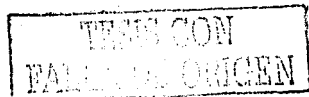
Los AG adoptan una mezcla de dirección y azar de manera que construyen eficientemente nuevas soluciones a partir de las mejores soluciones parciales de pruebas previas.

Considérese una población de "n" cadenas con base en un alfabeto apropiado, codificadas de forma que cada una es una idea o prescripción para desempeñar una tarea particular. Las subcadenas dentro de cada cadena (o idea) contienen varias nociones de lo que es importante o relevante para la tarea en sí. Visto de esta forma, la población contiene no sólo una muestra de "n" ideas; más que eso, contiene una multitud de nociones y categorías de éstas para el desempeño de la tarea.

Los AG explotan esta riqueza de información: 1) reproduciendo nociones de alta calidad de acuerdo con su desempeño (mecanismo de reproducción), y 2) cruzando estas nociones con muchas otras de alto desempeño de otras cadenas (mecanismo de Crossover). De aquí, la acción del Crossover con reproducción previa especula sobre nuevas ideas construidas con "bloques" de construcción de alto desempeño (nociones o ideas) de resultados previos. Además, el procesamiento de nociones de los AG debe operar de manera efectiva aun cuando éstas dependan de componentes característicos en formas altamente no lineales o complejas.

El intercambio de nociones para formar nuevas ideas es intuitivamente atractivo, si se piensa en términos del proceso de innovación. La innovación puede entenderse como una yuxtaposición de cosas que han trabajado bien en el pasado. En este sentido, la reproducción y el Crossover se combinan para buscar nuevas ideas potenciales engendradas en el pasado.

Los operadores de reproducción y Crossover en un AG conforman una especie de intercambio. Nociones de alto desempeño son probadas repetidamente e intercambiadas en la búsqueda del mejor desempeño.



2.2.d El Algoritmo Genético Sencillo

La aplicación de un AG sencillo a un problema de optimización es bastante simple y consta de varios pasos bien definidos (Figura 2.3).

1. Para usar un AG lo primero es codificar las variables decisivas del problema como una cadena de longitud finita, es decir, en un individuo artificial
2. Lo siguiente es seleccionar una población inicial de manera aleatoria. Para ello hay que determinar el tamaño de la población. Para cada individuo en la población se obtiene el valor de la función objetivo definida anteriormente por los requerimientos del problema.
3. Una generación o iteración del AG comienza con la reproducción. La reproducción genera la "mating pool" conjuntada por individuos artificiales esperando ser apareados
4. El Crossover se aplica en dos pasos: 1) aleatoriamente se hacen parejas de cadenas, y 2) se realiza el intercambio de información con el operador de Crossover
5. El operador de mutación es usado (con probabilidad pequeña) sobre cada una de las posiciones de las cadenas en la población
6. El resultado en este punto es una nueva población lista para ser probada. La prueba se realiza decodificando las cadenas para calcular la función objetivo de cada una.
7. Ahora se puede generar una nueva población repitiendo los pasos del 2 al 6. El proceso se detiene cuando se cumpla con algún criterio de convergencia o se llegue al máximo número de iteraciones

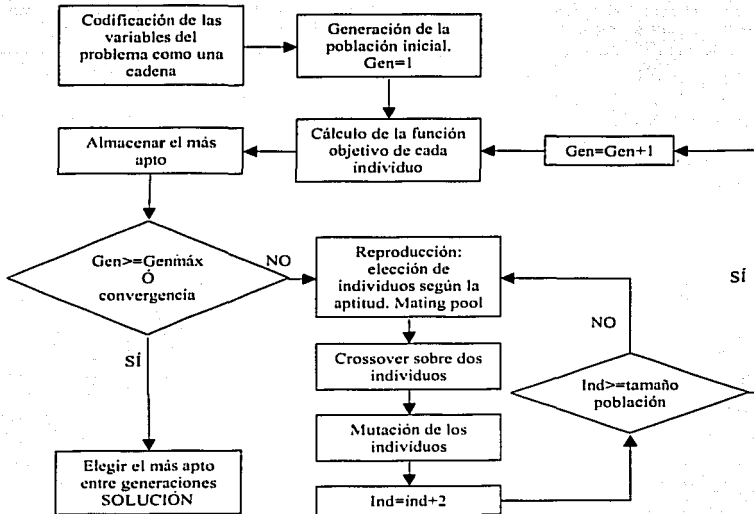


Figura 2. 3 Esquema del Algoritmo Genético Sencillo

CAPÍTULO 3

CARACTERÍSTICAS Y

SOLUCIÓN DE ARMADURAS

PLANAS

TESIS CON
FALLA DE ORIGEN

CAPÍTULO 3 CARACTERÍSTICAS Y SOLUCIÓN DE ARMADURAS PLANAS

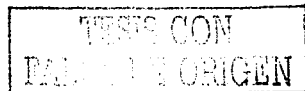
Las armaduras son estructuras rígidas que son utilizadas ampliamente. Se les puede ver en puentes, fuselaje de vehículos, entre otros. Determinar la solución de una armadura se refiere a calcular las fuerzas, esfuerzos y deformaciones de los elementos que la integran, así como los desplazamientos de las uniones entre estos.

En el presente capítulo se hace una revisión de las características trascendentes de este tipo de estructuras. También se mencionan los métodos convencionales de solución y se presenta el FEA aplicado a las armaduras planas.

3.1 ARMADURAS PLANAS

Una armadura es una estructura rígida constituida por barras unidas por articulaciones libres de fricción. Las cargas se aplican sólo sobre los nodos o articulaciones. Cada barra o elemento en una armadura se supone sólo sometido a cargas axiales. Los esfuerzos generados por las cargas axiales se les llaman esfuerzos primarios. En la realidad, es posible que aparezcan cargas de flexión en los elementos debido a que en general las uniones entre ellos no están del todo libres de fricción. A los esfuerzos generados por estas cargas se les conoce como esfuerzos secundarios. En la práctica los esfuerzos primarios resultan ser mucho mayores que los secundarios, por lo que los últimos son despreciados generalmente (Housner y Hudson 1965, González 2002)

Una armadura plana es aquella en la cual todos sus elementos se encuentran en un mismo plano.



3.1.a Cálculo del Grado de Indeterminación de Armaduras Planas

Una estructura estáticamente determinada o isostática tiene grado de indeterminación cero. Las estructuras estáticamente indeterminadas o hiperestáticas pueden tener varios grados de indeterminación. Por cada grado se requiere de una ecuación adicional de compatibilidad de deformación para calcular las reacciones.

Las armaduras pueden ser externamente o internamente indeterminadas. Son externamente indeterminadas cuando el número de reacciones de apoyo es mayor que el número de ecuaciones de equilibrio más el número de ecuaciones de condición. Si para una armadura ambos números son iguales, ésta será isostática externamente.

Para el caso de las armaduras planas existen tres ecuaciones de equilibrio (balance de fuerzas y momentos) y no se consideran ecuaciones de condición. El número mínimo de reacciones (r) es 3. Si existen más de tres reacciones se tiene una armadura hiperestática externamente. Menos de tres reacciones implican una estructura inestable.

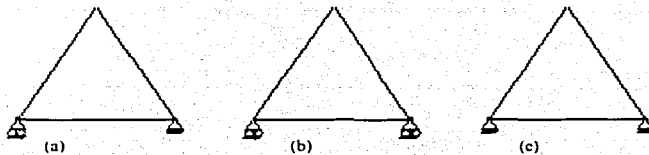
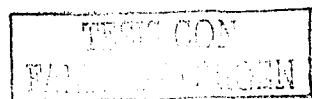


Figura 3. 1 (a) Armadura isostática $r=3$, (b) Armadura inestable $r=2$, (c) Armadura hiperestática externamente $r=4$



CAPÍTULO 3. CARACTERÍSTICAS Y SOLUCIÓN DE ARMADURAS PLANAS

- $r=3 \rightarrow$ isostática externamente
- $r>3 \rightarrow$ hiperestática externamente
- $r<3 \rightarrow$ inestable

La indeterminación interna ocurre cuando el número de elementos de una armadura es mayor que el mínimo necesario para que sea estable. Si se denomina con n el número de nodos y con e el número de elementos, la siguiente ecuación la satisfacen armaduras planas isostáticas y estables (González 2002)

$$r+e=2n \rightarrow \text{armadura estable e isostática}$$

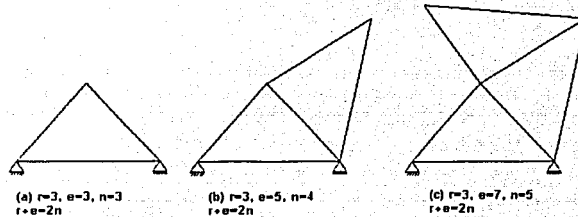
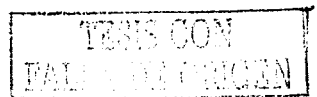


Figura 3. 2 Armaduras estáticamente determinadas

Por otra parte, el grado de indeterminación total se calcula con la relación entre $r+e$ y $2n$. Para una armadura hiperestática $r+e>2n$, la diferencia es el grado de indeterminación total. Por el contrario, si $r+e<2n$ la estructura es inestable.

La diferencia entre el grado de indeterminación total y el grado de indeterminación externo determinan la condición interna. Si ambos son iguales la armadura es internamente isostática. En cambio, si el total es mayor en una cantidad X la armadura es internamente hiperestática con grado X .



CAPÍTULO 3. CARACTERÍSTICAS Y SOLUCIÓN DE ARMADURAS PLANAS

La relación entre r , e y $2n$ establece una condición necesaria pero no suficiente para la indeterminación o la inestabilidad de una armadura. Lo anterior se explica en el siguiente apartado.

3.1.b Inestabilidad Geométrica

Existen algunas estructuras que son inestables a pesar de que al aplicar los criterios anteriores resulten estáticamente determinadas o aun indeterminadas. La inestabilidad se deriva de un número insuficiente o de una disposición inadecuada de los apoyos, o bien, de un arreglo incorrecto de partes en la estructura. En el primer caso se trata de inestabilidad geométrica externa y en el segundo de inestabilidad geométrica interna (Figura 3.3)

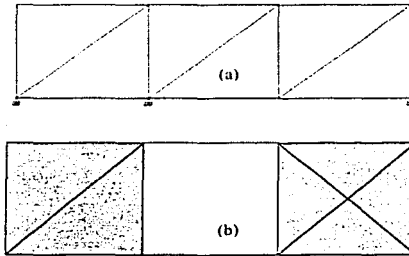


Figura 3.3 Armaduras inestables a pesar de cumplir con $r+e=2n$: (a) inestabilidad externa, (b) inestabilidad interna

3.1.c Métodos de nodos y de las secciones

Los métodos convencionales para calcular las fuerzas axiales presentes en los elementos de las armaduras son el método de nodos y el método de las secciones (Housner y Hudson 1965). A continuación se hace una breve descripción de ambos métodos.

CAPÍTULO 3. CARACTERÍSTICAS Y SOLUCIÓN DE ARMADURAS PLANAS

El método de nodos consiste en dibujar un diagrama de cuerpo libre de cada uno de los nodos de la armadura y formular las ecuaciones de equilibrio de cada diagrama. En el caso de armaduras planas el sistema se reduce a dos ecuaciones. Para la mayor parte de las armaduras es posible comenzar con un diagrama de cuerpo libre en el que sólo haya dos incógnitas.

El método de las secciones es particularmente útil para los problemas en que sólo tienen que calcularse las fuerzas existentes en una o más piezas de la armadura. Consiste en dibujar un diagrama de cuerpo libre de alguna parte de la armadura. Dicha parte se obtiene seccionando la armadura en aquellas barras de las cuales se desean determinar las fuerzas. Lo anterior da como resultado un sistema general de fuerzas, para el cual pueden formularse las ecuaciones de equilibrio.

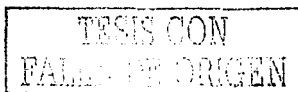
Si bien los métodos enunciados ilustran los fundamentos de la estática, ellos se vuelven tediosos al aplicarlos a grandes armaduras estructurales estáticamente indeterminadas. Además los desplazamientos nodales no se obtienen con facilidad.

Una alternativa viable para el cálculo aproximado de fuerzas, desplazamientos, esfuerzos, etc., en armaduras es el Análisis por Elemento Finito. Además es aplicable tanto a estructuras estáticamente determinadas como indeterminadas.

3.2 ANALISIS POR EL MÉTODO DEL ELEMENTO FINITO

El Método del Elemento Finito es una herramienta poderosa en la solución numérica de un amplio rango de problemas de ingeniería. Algunas de sus aplicaciones son: análisis de deformaciones y esfuerzos de estructuras, análisis de campos de flujo de calor, análisis de fluidos, análisis de flujos magnéticos, análisis de filtraciones, y otros problemas de flujo. Con los acelerados avances en la tecnología de las computadoras y los sistemas CAD, pueden modelarse problemas complejos con relativa facilidad. En una computadora pueden probarse varias configuraciones alternas antes de construir el primer prototipo.

En este método de análisis, una región compleja que define un medio continuo se discretiza en formas geométricas simples llamadas elementos finitos. Las propiedades del



CAPÍTULO 3. CARACTERÍSTICAS Y SOLUCIÓN DE ARMADURAS PLANAS

material y las relaciones gobernantes, son consideradas sobre esos elementos expresadas en términos de valores desconocidos en los bordes del elemento. Un proceso de ensamble, cuando se consideran debidamente las cargas y restricciones, da lugar a un conjunto de ecuaciones. La solución de estas ecuaciones resulta en el comportamiento aproximado del medio continuo.

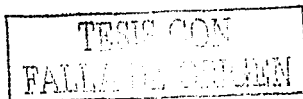
Método del Elemento Finito aplicado a armaduras

Las armaduras están conformadas por elementos que sólo están sujetos a cargas de tracción o compresión. Por lo anterior, se puede suponer a cada elemento de una armadura como un elemento unidimensional cuando se considera en el sistema coordenado local (Chandrupatla y Gelegundu 1999).

En una armadura los elementos se presentan con una configuración no unidimensional (2D ó 3D). Es necesario relacionar la orientación local de los elementos (en la cual son unidimensionales) con su orientación global dentro de un sistema de referencia. Para ello se introducen dos sistemas de coordenadas: local y global.

En la Figura 3.4 se muestra un elemento típico de una armadura plana en sistemas de coordenadas locales (X') y globales (XY). En el esquema de numeración local, los nodos del elemento se numeran como 1 y 2. Todas las cantidades en el sistema coordenado local se denominan por medio de caracteres primados ('). El sistema de coordenadas global XY está fijo y no depende de la orientación del elemento. En éste, cada nodo tiene dos grados de libertad (GDL). Un nodo cuyo número global es j , tiene asociados los grados de libertad $2j-1$ y $2j$. Los desplazamientos nodales globales asociados son U_{j-1} y U_j . En el sistema de coordenadas local los desplazamientos de los nodos 1 y 2 son, respectivamente, u'_{j-1} y u'_j . El vector de desplazamiento del elemento en el sistema de coordenadas local se denota por \underline{u}' y está dado por:

$$\underline{u}' = [u'_1, u'_2]^T$$



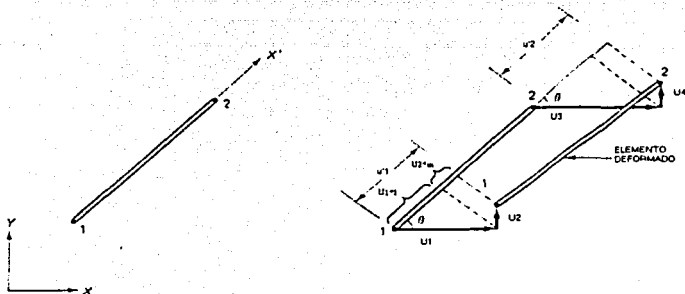


Figura 3. 4 Elemento bidimensional en los sistemas de coordenadas global y local

El vector de desplazamiento del elemento en el sistema coordenado global se denota por:

$$U = [U_1, U_2, U_3, U_4]^T$$

La relación entre \underline{u}' y \underline{U} está dada de la siguiente forma:

$$u'_1 = U_1 l + U_2 m$$

y

$$u'_2 = U_3 l + U_4 m$$

donde, l y m son los cosenos directores del vector asociado con la orientación del elemento. En forma matricial, las ecuaciones anteriores pueden escribirse como:

$$\underline{u}' = \underline{L} \underline{U}$$

CAPÍTULO 3. CARACTERÍSTICAS Y SOLUCIÓN DE ARMADURAS PLANAS

Donde la matriz de transformación L está dada por

$$L = \begin{bmatrix} 1 & m & 0 & 0 \\ 0 & 0 & 1 & m \end{bmatrix}$$

La matriz de rigidez para un elemento armadura en el sistema coordenado local es:

$$k_c = \frac{E_c A_c \epsilon}{l_c} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

donde, E_c , A_c y l_c son el módulo de elasticidad, la sección transversal la longitud del elemento.

Dicha matriz en el sistema de coordenadas globales se calcula como:

$$K = \frac{E_c A_c \epsilon}{l_c} \begin{bmatrix} l^2 & lm & -l^2 & -lm \\ lm & m^2 & -lm & -m^2 \\ -l^2 & -lm & l^2 & lm \\ -lm & -m^2 & lm & m^2 \end{bmatrix}$$

donde l y m son los cosenos directores del elemento. El problema estático en el FEA es resolver la ecuación:

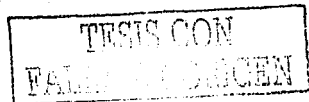
$$K U = F$$

donde, F es el vector de cargas aplicadas en los nodos del elemento. Entonces, el vector de desplazamientos nodales se calcula como:

$$U = K^{-1} F$$

Una vez calculados los desplazamientos nodales, el esfuerzo del elemento se puede calcular de la ley de Hooke:

$$\sigma_c = E_c \epsilon_c$$



CAPÍTULO 3. CARACTERÍSTICAS Y SOLUCIÓN DE ARMADURAS PLANAS

donde ϵ_e es la deformación del elemento y se define como el cociente del cambio de longitud de elemento entre su longitud inicial.

El esfuerzo de un elemento se calcula como:

$$\sigma_e = \frac{E}{l_e} [-1 \quad -m \quad 1 \quad m] U$$

En una armadura se encuentran conectados un número determinado de elementos. La conexión entre ellos está definida por la relación que existe entre los grados de libertad comunes numerados globalmente, dicha relación recibe el nombre de conectividad. La conectividad de cada elemento es diferente y se describe con el vector de conectividad (Figura 3.5 y Tabla 3.1).

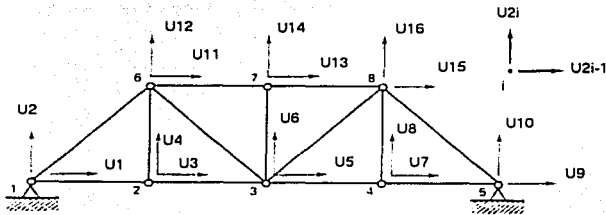


Figura 3.5 Conectividad en el caso de armaduras planas

Nodo	Conectividad
1	1,2
2	3,4
3	5,6
4	7,8
5	9,10
6	11,12
7	13,14
8	15,16
i	2i,2i-1

Tabla 3. 1 Conectividad de la armadura Figura 3.5

Resolver una armadura implica encontrar la solución de la ecuación $KU=F$, donde la matriz K y el vector F están asociados a la armadura y no sólo a un elemento. Ambos se obtienen ensamblando las matrices de rigidez y los vectores de carga de cada elemento en una matriz y un vector único. Dicho ensamble se realiza tomando en cuenta la conectividad de cada elemento (Chandrupatla y Gelegundu 1999). De esta forma, el orden de la matriz de rigidez y la longitud del vector de carga son iguales al número de grados de libertad contenidos en la armadura.

CAPÍTULO 4

IMPLEMENTACIÓN DEL

ALGORITMO GENÉTICO AL

PROBLEMA DE OPTIMIZACIÓN

ESTRUCTURAL DE

ARMADURAS PLANAS

CAPÍTULO 4 IMPLEMENTACIÓN DEL ALGORITMO GENÉTICO AL PROBLEMA DE OPTIMIZACIÓN ESTRUCTURAL DE ARMADURAS PLANAS

En este trabajo se implementa, en un programa de computadora, la metodología de los AG al problema de optimización de estructuras tipo armadura; en particular, armaduras planas. Para dicho propósito se empleó el AG simple propuesto por Goldberg (Goldberg 1989) con algunas modificaciones, fusionándolo con el FEA aplicado a armaduras planas³.

En los apartados siguientes se describe la forma como está integrado el algoritmo computacional desarrollado para la optimización de armaduras planas. Al final se presenta dicho algoritmo y un diagrama de flujo que muestra la secuencia de operaciones que realiza.

4.1 DEFINICIÓN DEL DOMINIO DE DISEÑO

La optimización de armaduras contemplada en este trabajo comienza estableciendo un dominio de diseño adecuado para el problema en cuestión. El dominio de diseño se refiere a una estructura inicial general tipo armadura compuesta por arreglos de ocho elementos y cinco nodos como el mostrado en la Figura 4.1.

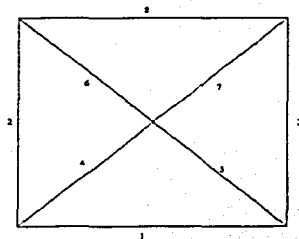
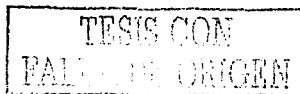


Figura 4.1 Arreglo básico del dominio de diseño

³ Ver Capítulo 3 Características y Solución de Armaduras Planas



CAPÍTULO 4. IMPLEMENTACIÓN DEL ALGORITMO GENÉTICO

El comportamiento mecánico de cada elemento es el de uno sometido solamente a cargas de tracción o compresión características de los elementos en armaduras.

Los nodos o uniones se comportan como articulaciones sin fricción. La aplicación de las cargas externas se restringe a los nodos. Lo anterior conlleva que se desprecia cualquier efecto flexionante (Housner y Hudson 1965)

El dominio de diseño está delimitado por las dimensiones base (B) y altura (H) que la estructura inicial debe tener. Estas dimensiones pueden variar en la armadura óptima al final del proceso de optimización dependiendo de las restricciones (apoyos) y requerimientos (cargas externas) propios de cada problema.

El número de arreglos básicos, elementos y nodos que conforman la estructura inicial dependen del número de elementos horizontales (b) y verticales (h) que se deseen considerar y se calculan respectivamente de la siguiente manera:

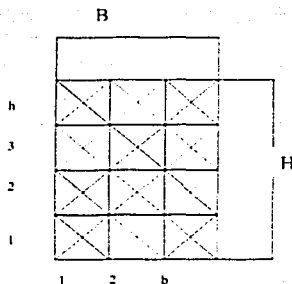


Figura 4. 2 Dimensiones del Dominio de diseño

$$\text{No. de arreglos} = bh$$

$$e = b + h + 6bh$$

$$n = l + b + h + 2bh$$

CAPÍTULO 4. IMPLEMENTACIÓN DEL ALGORITMO GENÉTICO

donde, e es el número de elementos y n el número de nodos (Figura 4.2).

4.2 RESTRICCIONES Y REQUERIMIENTOS

Las restricciones se refieren a los apoyos y su tipo empleados para fijar la armadura. Dado que se trata de un problema en dos dimensiones cada nodo tiene dos grados de libertad. Si consideramos un sistema de coordenadas cartesiano, los grados de libertad están asociados con desplazamientos en las direcciones X y Y. El número máximo de grados de libertad en la armadura es:

$$GDL=2(1+b+h+2bh)$$

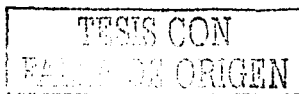
Los requerimientos son las cargas aplicadas en los nodos. El número de cargas está delimitado por el número de nodos en la estructura, sin embargo, no es común que sean iguales. Cada carga debe ser descompuesta, para efectos del FEA, en dos componentes cartesianas.

$$\underline{F} = F_x + F_y$$

4.3 CODIFICACIÓN DEL DOMINIO DE DISEÑO

En el apartado anterior se menciona que el dominio de diseño está compuesto por $b+h+6bh$ elementos. Cada elemento en el dominio ocupa una posición bien determinada y no repetida. Es decir, cada elemento puede ser representado de manera independiente dentro del dominio. Esta característica permite representar al dominio de diseño como una cadena de datos con base en un alfabeto bien definido. Para este trabajo se emplea los números identificadores de cada elemento como parte del alfabeto. Por ejemplo, la estructura mostrada en la Figura 4.3, b y h valen 2 y 3 respectivamente.

El número de elementos es, por lo tanto, $e=2+3+6(2*3)=41$. La codificación de esta estructura como una cadena de datos utilizando un alfabeto que comprende los números entre 1 y 41 inclusive es la siguiente:



ESTRUCTURA=[1 2 3 4 5 6 ... 39 40 41]

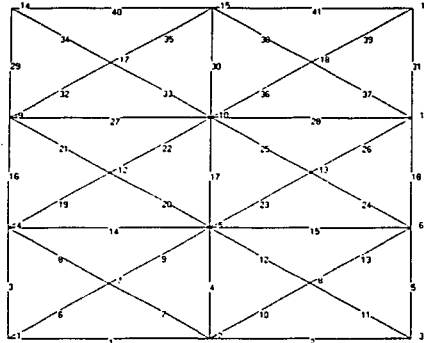


Figura 4. 3 Dominio de diseño para b=2 y h=3

En el proceso de optimización ocurrirá que varios elementos pueden desaparecer de la estructura. Para ello, se elige el negativo del número del elemento para representar su ausencia. Por ejemplo, para la armadura de la Figura 4.4 donde los elementos 1, 2, 4, 7 y 10 no se encuentran, la codificación de la misma es:

ESTRUCTURA'=[-1 -2 3 -4 5 6 -7 8 9 -10 ... 39 40 41]

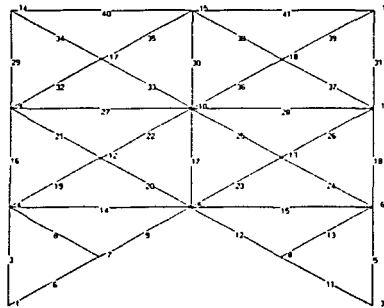


Figura 4. 4 Armadura que resulta de eliminar los elementos 1, 2, 4, 7 y 10 del dominio de la Figura 4.3

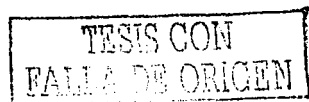
De esta manera, cualquier armadura puede ser codificada como una cadena de datos con base en un alfabeto $[-(b+h+6bh), (b+h+6bh)]$ excluyendo el cero (0).

4.4 FUNCIÓN OBJETIVO

La función objetivo o de aptitud decodifica las cadenas de datos que representan a cada armadura. Es ésta la que establece la posibilidad de que la información intrínseca en cada cadena sea empleada en el procesamiento posterior.

La definición de la aptitud es particular a cada problema, por lo que puede ser definida de muchas maneras diferentes pero siempre de acuerdo con la meta que se establezca. En este trabajo se optimizan las armaduras con base en su energía de deformación y su volumen. El criterio es que las mejores estructuras serán aquellas que presenten valores tanto de energía de deformación y volumen mínimos. Por lo anterior, la función de objetivo es:

$$f_i = \left(\frac{U_i * V_i}{U_o * V_o} \right)^{-1}$$



CAPÍTULO 4. IMPLEMENTACIÓN DEL ALGORITMO GENÉTICO

donde, f_i , U_i y V_i son la aptitud, la energía de deformación y el volumen de la armadura en cuestión; U_o y V_o son la energía de deformación y el volumen del dominio de diseño.

La energía de deformación de un elemento sometido a cargas de tracción está dada por:

$$U_j = \frac{1}{2} \int \sigma \epsilon dV = \frac{1}{2} \sigma_j \epsilon_j V_j$$

Para una armadura con e elementos la energía de deformación se define como (Chandrupatla y Belegundu 1999):

$$U = \sum U_j = \frac{1}{2} \sum \sigma_j \epsilon_j V_j$$

El volumen de un elemento se calcula multiplicando la sección transversal y la longitud. El volumen de una armadura está dado por:

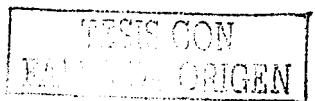
$$V = \sum V_j = \sum A_j l_j$$

La mejor armadura será la que presente el mayor valor de f_i por lo que la meta es maximizarla.

4.5 ANÁLISIS POR ELEMENTO FINITO

El cálculo de la energía de deformación de una armadura, y, consecuentemente, de la aptitud de la misma implica conocer el esfuerzo (σ_e) y la deformación (ϵ_e) de cada elemento. Este conocimiento lo brinda el FEA que se incorpora en el programa como un módulo encargado de dar el tratamiento adecuado a cada armadura y entregar como resultado los vectores de desplazamientos nodales, esfuerzos y deformaciones de cada elemento.

Durante el proceso de optimización el módulo de FEA interactúa con el Algoritmo Genético. Esta fusión consolida la parte principal del método propuesto para la optimización evolutiva de armaduras.



4.6 ALGORITMO GENÉTICO EMPLEADO

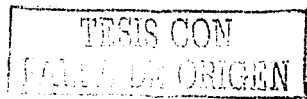
Como se mencionó anteriormente, el AG empleado es el propuesto por Goldberg con algunas modificaciones. Estas son la inclusión de un operador de Crossover de doble punto, selección de las mejores estructuras empleando un porcentaje de Elitismo. En la Figura 4.5 se muestra el esquema del Algoritmo Genético implementado.

4.6.a Crossover de doble punto

El Crossover de doble punto no es muy diferente al Crossover sencillo. La diferencia estriba en que se eligen de manera aleatoria dos puntos de cruce sobre la longitud del par de cadenas a operar en lugar de sólo uno (De Jong 1975). La longitud de las cadenas operadas es $l = b+h+6bh$, si los puntos de cruce son k_1 y k_2 , ambos en el intervalo $[1, l-1]$, las cadenas son divididas en tres segmentos definidos de la siguiente forma:

$$\begin{aligned} \text{Segmento1} &= [1, \dots, k_1] \\ \text{Segmento2} &= [k_1+1, \dots, k_2] \\ \text{Segmento3} &= [k_2+1, \dots, l] \end{aligned}$$

Por ejemplo, para las cadenas $P1=[1\ 2\ 3\ 4\ 5\ 6\ 7\ 8]$ y $P2=[-1\ 2\ -3\ 4\ 5\ -6\ 7\ 8]$, si k_1 y k_2 son 1 y 3 respectivamente, los segmentos resultantes son:



CAPÍTULO 4. IMPLEMENTACIÓN DEL ALGORITMO GENÉTICO

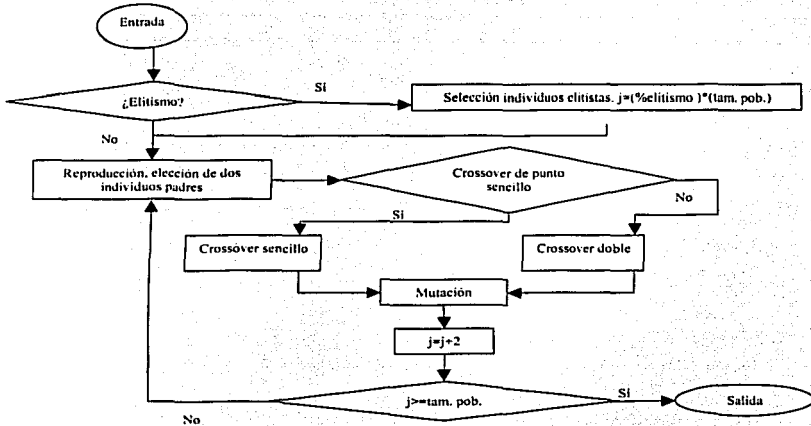


Figura 4. 5 Algoritmo Genético sencillo modificado

Segmento1_{P1} =[1]

Segmento2_{P1} =[2 3]

Segmento3_{P1} =[4 5 6 7 8]

Segmento1_{P2} =[-1]

Segmento2_{P2} =[2 -3]

Segmento3_{P2} =[4 5 -6 7 8]

El operador de Crossover de doble punto se aplica con la misma probabilidad de ocurrencia P_c que se fije para el Crossover normal. El intercambio de información entre las cadenas P1 y P2 del ejemplo al aplicar el Crossover de doble punto resulta en dos cadenas descendientes integradas como:

$H1 = [-1 \ 2 \ 3 \ 4 \ 5 \ -6 \ 7 \ 8]$

$H2 = [1 \ 2 \ -3 \ 4 \ 5 \ 6 \ 7 \ 8]$

Las armaduras correspondientes a P1, P2, H1 y H2 se muestran en la Figura 4.6.



Figura 4. 6 Las armaduras H1 y H2 son el resultado de aplicar Crossover de doble punto sobre las armaduras P1 y P2

4.6.b Elitismo

El Elitismo retiene en cada generación un número (Tamaño de población) x (%E) de individuos con base en la aptitud, es decir, copia íntegramente a aquellos individuos que poseen los mayores valores de la función objetivo de una generación a la siguiente hasta una cantidad entera menor o igual a (Tamaño de población)(%E), donde %E es el porcentaje de elitismo (De Jong 1975). Los valores de %E van de 5 al 20 porciento. Por ejemplo, si el Tamaño de población en un problema de optimización es 100 y %E es 20, se copiarán de una generación a la siguiente las 20 mejores cadenas en cada iteración.

El Elitismo se aplica en cada generación. Si en una generación aparecen individuos o cadenas que presentan mayor aptitud que los copiados por Elitismo anteriormente, en la siguiente generación cambiará la fracción elitista de la población.

El empleo de Elitismo en los AG puede contribuir a que la convergencia hacia una solución se dé con mayor rapidez.

CAPÍTULO 4. IMPLEMENTACIÓN DEL ALGORITMO GENÉTICO

Al final de este capítulo se muestra el algoritmo computacional desarrollado y en la Figura 4.7 el diagrama de flujo del mismo.

4.7 POBLACIÓN INICIAL

La configuración de los elementos de las armaduras de la población inicial se determina de manera aleatoria. Cada posición de la cadena codificada del dominio de diseño se somete a un proceso de selección aleatoria para definir su presencia en la cadena que representa a un nuevo individuo o armadura. La cadena así generada tiene que ser filtrada (ver apartado siguiente) para determinar si en realidad se trata de una estructura rígida y no de una inestable o un mecanismo.

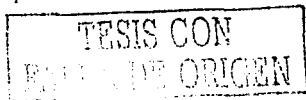
Al final de se obtiene una cantidad igual al tamaño de la población de cadenas codificadas que representan armaduras planas.

4.8 FILTROS

El uso de filtros es necesario en el problema de optimización de armaduras por varias razones. Durante el proceso de optimización pueden aparecer configuraciones de elementos que no sean estructuras tipo armadura, sino mecanismos o estructuras inestables (no rígidas). Por supuesto, tales configuraciones no pertenecen al espacio de soluciones posibles; su aparición se debe a la forma aleatoria como trabaja el AG. Para detectar y descartar dichas configuraciones se usan los siguientes filtros.

4.8.a Filtro por Inestabilidad Geométrica

La ecuación $r+e < 2n$ se satisface para armaduras inestables. En el programa se aplica un filtro basado en esta ecuación. Aquellas estructuras generadas durante el funcionamiento del programa que satisfagan dicha ecuación, son descartadas y se almacena la cadena que las representa en un archivo. Cada vez que se genera una armadura primero se verifica su existencia en el archivo. Si la armadura se encuentra en él, se rechaza y se continúa con el tratamiento de la siguiente. Esto es una forma de ahorrar tiempo de cálculo.



4.8.b Filtro de Singularidad

La relación entre r , e y $2n$ establece una condición necesaria pero no suficiente respecto a la indeterminación o la inestabilidad de una armadura. Sin embargo, para toda armadura inestable la matriz de rigidez correspondiente es singular, es decir, su determinante es nulo.

Lo anterior es la base para el filtro de singularidad. Si al diagonalizar la matriz de rigidez de una armadura y calcular su determinante éste resulta igual a cero, la armadura se rechaza y es almacenada en el archivo de armaduras rechazadas.

4.8.c Filtro por Fluencia

El módulo de elasticidad y el esfuerzo de fluencia del material que se empleó, son datos que deben ser alimentados al programa. Al aplicar el FEA se calculan los desplazamientos nodales y posteriormente los esfuerzos y deformaciones de los elementos. Si para algún elemento en la armadura el esfuerzo calculado rebasa al de fluencia del material, la armadura en cuestión es rechazada y se almacena en el archivo de armaduras rechazadas.

El resultado de la aplicación de los tres filtros anteriores es un archivo que se emplea para agilizar el desarrollo del programa y reducir el tiempo de máquina necesario para resolver un problema, ya que evita realizar cálculos innecesarios o repetitivos.

4.8.d Estructuras no Rechazadas

Es muy común que las armaduras se repitan de una generación a otra. La aplicación del FEA a una armadura ya calculada con anterioridad implica repetir cálculos y aumento innecesario del tiempo para resolver el problema. Por ello, cada vez que se calculan los parámetros de una armadura válida, estos se almacenan en un archivo de armaduras válidas. En cada iteración, se comparan las cadenas de datos de las armaduras generadas con las existentes en este archivo. Si existe coincidencia se copian los parámetros del archivo a los

CAPÍTULO 4. IMPLEMENTACIÓN DEL ALGORITMO GENÉTICO

parámetros de la cadena en la generación actual. De esta forma se evita el repetir cálculos y, por lo tanto, se reduce el tiempo de búsqueda del programa.

4.9 CRITERIOS DE PARO

Se consideran tres criterios de paro para el algoritmo:

- ▶ Máximo número de generaciones
- ▶ Pérdida de diversidad de la población
- ▶ Repetición del individuo de mayor aptitud

En el primer caso, se detiene la ejecución del algoritmo cuando se han calculado el máximo número de generaciones.

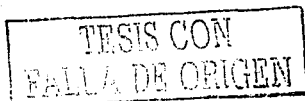
La pérdida de diversidad de la población significa que no existe diferencia sustancial entre los individuos que conforman una población en una generación dada, es decir, la población es homogénea. Para determinarlo se calcula la media y la desviación estándar de las aptitudes de la población en cada generación.

Se considera que se ha perdido la diversidad de la población si la desviación estándar es menor o igual al 1% de la media de las aptitudes. Si se da esta condición se toma al mejor individuo anterior como el óptimo del proceso.

Por último, en caso de que un individuo aparezca repetidamente como el de mayor aptitud se adopta a éste como el óptimo. El número de repeticiones establecido para que se detenga la ejecución del algoritmo es el 10% del número máximo de generaciones.

4.10 RESULTADOS

Se almacena en un archivo de texto, nombrado Reporte, la información referente a los parámetros de optimización, la mejor armadura y motivo de paro. A la par se genera otro archivo de texto que contiene los datos correspondientes a los mejores resultados obtenidos para el problema alimentado al algoritmo.



CAPÍTULO 4. IMPLEMENTACIÓN DEL ALGORITMO GENÉTICO

A continuación se encuentra el algoritmo computacional propuesto expuesto paso a paso. La Figura 4.7 representa el diagrama de flujo del mismo.

0. Inicio

1. Alimentación de los parámetros del dominio de diseño

- Longitud
- Altura
- Divisiones horizontales
- Divisiones verticales

2. Generación del dominio de diseño (malla)

- Elementos
- Nodos
- Conectividades

3. Alimentación de restricciones, cargas y propiedades

- Restricción de nodos
 - En X
 - En Y
 - Ambas direcciones
- Cargas
 - Nodo
 - Magnitud
 - Orientación
- Propiedades
 - Sección transversal de los elementos
 - Módulo de elasticidad
 - Esfuerzo de fluencia

TESIS CON
FALLA DE ORIGEN

4. Generación

- Matriz de rigidez del dominio de diseño
- Vector de cargas del dominio de diseño
- Almacenamiento de matriz de rigidez y vector de cargas

CAPÍTULO 4. IMPLEMENTACIÓN DEL ALGORITMO GENÉTICO

5. Alimentación de los parámetros de optimización
 - Tamaño de población
 - Máximo número de generaciones
 - Probabilidad de Crossover
 - Probabilidad de mutación
 - Elección entre Crossover de punto sencillo o doble
 - Elección de elitismo y porcentaje
6. Generación de geometrías iniciales (población inicial)
7. Aplicación de FEA a las geometrías
8. Cálculo de aptitudes de las geometrías iniciales
9. Elección de la mejor geometría (elección del más apto)
 - Almacenamiento de la mejor geometría
10. Verificación de criterios de paro
 - Si el máximo número de generaciones es alcanzado o algún criterio de paro se satisface:
 - Óptimo igual a la mejor geometría de 9
 - En caso contrario
 - Aplicación del GA a la población de geometrías
 - Generación de nuevas geometrías
11. Repetición de los pasos del 7 al 10

TESIS CON
FALLA DE ORIGEN

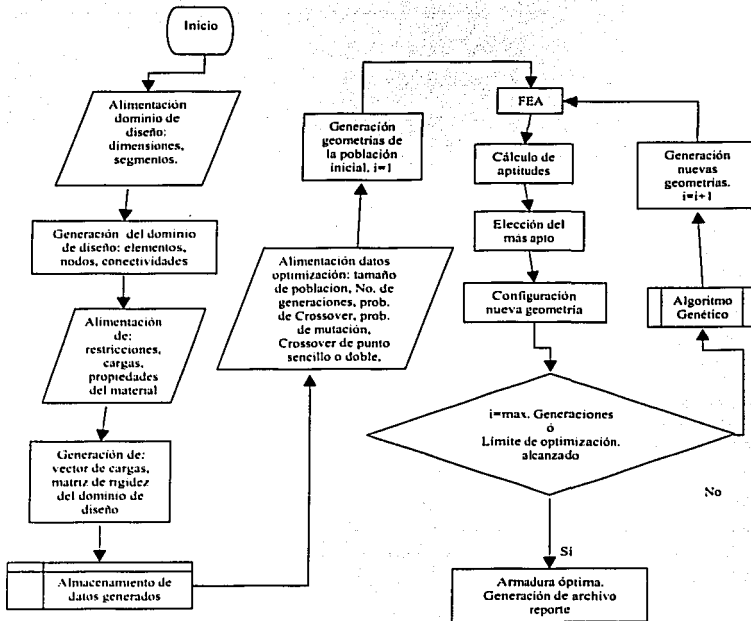


Figura 4. 7 Diagrama de flujo del algoritmo computacional desarrollado

CAPÍTULO 5

EL PROGRAMA OPTI

TESIS CON
FALLA DE ORIGEN

CAPÍTULO 5 EL PROGRAMA OPTI

En este capítulo se hace un descripción del programa OPTI. En éste se fusionan el Análisis por Elemento Finito (FEM) con los Algoritmos Genéticos (GA) para generar una herramienta práctica con base en el concepto DOPE.

5.1 DESCRIPCIÓN DEL PROGRAMA

La función del programa OPTI es determinar la configuración de una armadura que proporcione la mejor solución al problema de soportar un sistema de cargas aplicado sobre un dominio de diseño, bajo un criterio que combina los criterios de mínima energía de deformación y mínimo volumen.

OPTI es una aplicación desarrollada en Delphi versión 5. Delphi es un producto basado en un entorno de desarrollo visual y un lenguaje orientado a objetos llamado Object Pascal (Louis 2000). OPTI es una aplicación que opera en el ambiente visual de Windows.

Se requiere que el usuario alimente al programa los valores de algunos parámetros necesarios para el funcionamiento del mismo. A continuación se describen las acciones que son necesarias y posibles en el programa OPTI.

5.2 INICIO DEL PROGRAMA

El programa se inicia ejecutando el archivo OPTI.exe el cual debe ser colocado de preferencia en una carpeta independiente. Lo anterior se debe a que el programa genera un conjunto de archivos, algunos de los cuales son empleados en la ejecución del programa y el resto son archivos de texto en los que se almacena información útil para el usuario.

Al ejecutar el archivo OPTI.exe se generan los 4 archivos para almacenar información importante para el programa. En uno de ellos se almacena información para identificar armaduras inestables o mecanismos que pueden generarse durante el proceso de optimización.



CAPÍTULO 5. EL PROGRAMA OPTI

Los otros tres son archivos en los que se encuentra la información de armaduras válidas. Lo anterior se realiza para no repetir cálculos y acelerar la ejecución del programa. Tales archivos almacenan la cadena codificada, los desplazamientos nodales y los esfuerzos de los elementos de una armadura respectivamente.

A la par de la creación de los archivos mencionados aparece la ventana principal del programa, la cual se muestra en la Figura 5.2.

En la parte superior de la ventana principal se encuentra el menú principal compuesto por los submenús: Archivo, Visualizar, Restricciones y OPTI, Figura 5.1.



Figura 5.1 Menú principal del programa OPTI

El submenú Archivo se compone por dos partes: Nuevo y Cerrar. La segunda permite finalizar la aplicación. Nuevo (Figura 5.3) genera una ventana en la que el usuario debe ingresar la longitud y altura del dominio de diseño, en metros, y el número de divisiones horizontales y verticales.

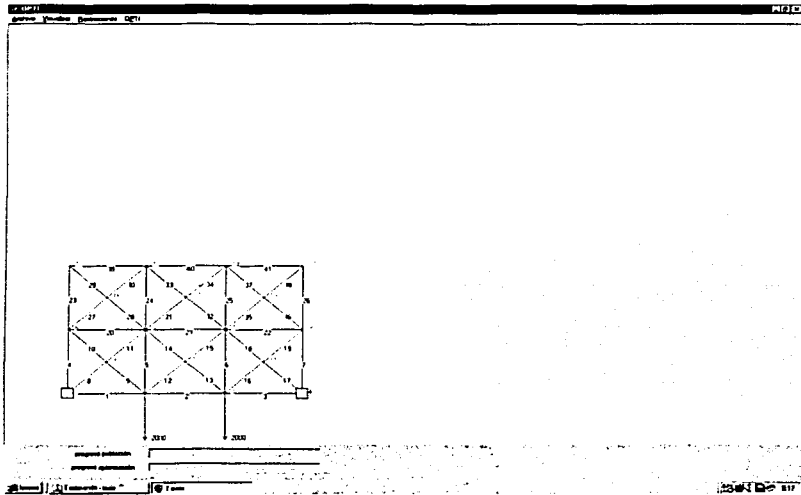


Figura 5. 2 Ventana principal del programa OPTI

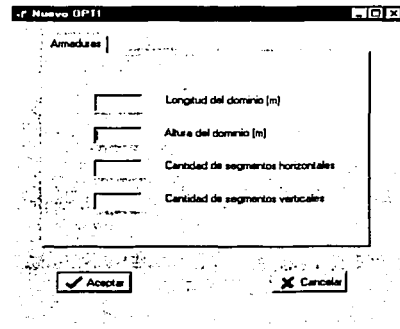


Figura 5. 3 Ventana del submenú Nuevo

5.3 VIZUALIZAR

Una vez que se han alimentado los datos geométricos del dominio de diseño y aceptar en la ventana, se puede acceder al submenú Visualizar. Éste genera una ventana como la que se muestra en la Figura 5.4.

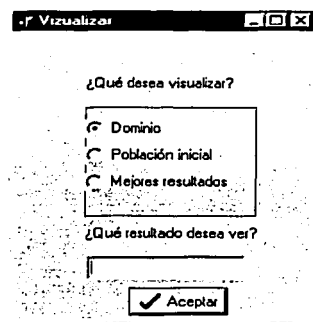


Figura 5.4 Ventana del submenú Visualizar

Hay tres opciones de visualización: Dominio, Población inicial y Mejores resultados. La primera aparece siempre que se abre la ventana. La segunda opción es visible siempre y cuando la población inicial haya sido generada, de otra forma no es visible. La tercera es visible a partir de que el proceso de optimización concluye.

Al elegir Dominio se visualiza en la ventana principal el dominio de diseño con la numeración de los elementos y los nodos, las restricciones y las fuerzas. Población inicial al ser elegida permite visualizar de manera secuencial las armaduras que componen la población inicial del proceso de optimización. Mejores resultados se activa para visualizar al mejor individuo de cada generación. Para elegir la generación se introduce el número de ésta en el cuadro de texto debajo de la pregunta ¿Qué resultado desea ver? En la ventana principal aparecerá la mejor armadura de la generación indicada.

5.4 RESTRICCIONES Y MATERIAL

El submenú Restricciones se compone de tres partes: Nodos, Cargas y Material. Nodos permite indicar los nodos que están restringidos en el dominio y el tipo de restricción. Dado que los nodos son en realidad articulaciones sin fricción, las restricciones posibles son sólo dos: el movimiento en la dirección horizontal y el movimiento en la dirección vertical. Para alimentar esta información al programa, aparece la ventana Nodos Restringidos, Figura 5.5. Gráficamente los nodos restringidos aparecen como cuadrados de tres diferentes colores dependiendo el tipo de restricción, negros si la restricción es total, rojos si la restricción es en la dirección X y azules si la restricción es en la dirección Y.

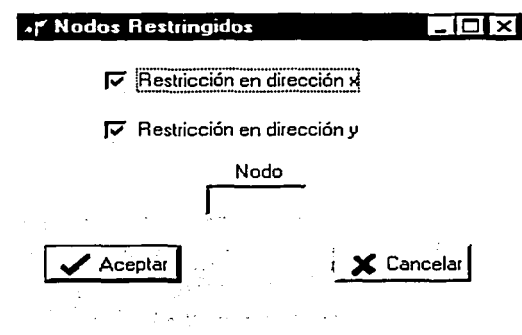


Figura 5.5 Ventana para restringir nodos

Para introducir las cargas aplicadas en el dominio de diseño al programa se emplea la opción Cargas. En la ventana Cargas se elige el nodo en el cual se aplica una fuerza, su magnitud en Newton y su orientación dada por un ángulo, Figura 5.6.

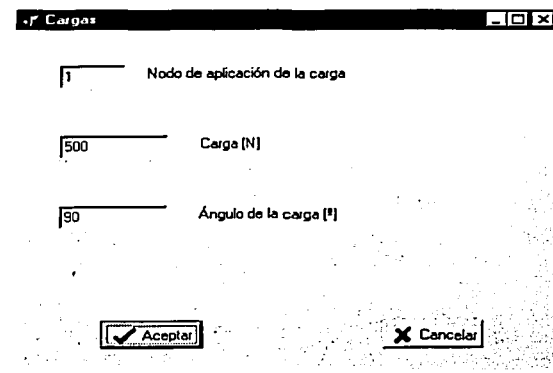


Figura 5. 6 Ventana para ingresar las cargas aplicadas sobre el dominio de diseño

La opción Material despliega la ventana mostrada en la Figura 5.7. En ésta se deben alimentar los valores de la sección transversal en m^2 , el módulo de elasticidad y el esfuerzo de fluencia, ambos en Pascales, del material que se considera para la armadura.

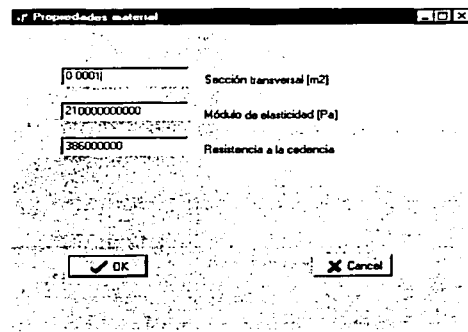


Figura 5. 7 Ventana para indicar las propiedades del material considerado en la armadura

5.5 OPTIMIZACIÓN

Una vez que se alimenta la información del dominio de diseño, se habilita el submenú Opti. Éste se compone de dos secciones: Generar población y Optimizar. La segunda sólo aparece después de ejecutar la primera. Generar población da inicio al proceso de optimización creando la población inicial. Para ello es necesario ingresar el Tamaño de la población, el Máximo número de generaciones, la probabilidad de Crossover y la probabilidad de Mutación. Además se puede elegir Crossover de doble punto habilitando la casilla de verificación correspondiente. Es posible emplear Elitismo indicando el porcentaje del mismo para cada generación, Figura 5.8.

Con la información anterior se genera la población inicial y se prepara el algoritmo para el proceso de optimización. La barra de progreso superior de la ventana principal señala el avance en la generación de la población inicial. Antes de iniciar el proceso de optimización es posible observar la población inicial seleccionando tal opción en el submenú Visualizar.

TESIS CON
FALLA DE ORIGEN

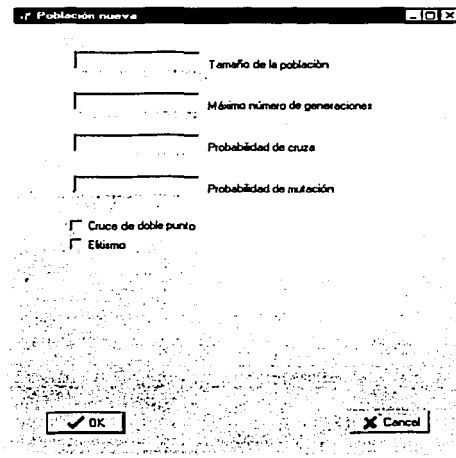


Figura 5. 8 Ventana para alimentar los parámetros para el proceso de optimización

Al seleccionar Optimizar en el menú Opti se inicia la optimización. El tiempo de proceso puede variar dependiendo las características y dimensiones del dominio de diseño. La barra de progreso superior indica el progreso de cada generación y la barra inferior muestra el avance de la optimización tomando como base el máximo número de generaciones.

Un letrero indicando la generación en la cual la mejor armadura fue encontrada aparece al terminar el proceso. A partir de este evento es posible utilizar todas las opciones del submenú Visualizar. La ejecución del programa OPTI llega a su fin con la creación de dos archivos de texto llamados Reporte.txt y Mejores.txt en los que se plasma un resumen de la ejecución y la aptitud promedio de cada generación respectivamente.

CAPÍTULO 6
APLICACIÓN DEL PROGRAMA
OPTI EN LA DETERMINACIÓN
DE ARMADURAS PLANAS
ÓPTIMAS

TESIS CON
FALLA DE ORIGEN

CAPÍTULO 6. APLICACIÓN DEL PROGRAMA OPTI EN LA DETERMINACIÓN DE ARMADURAS PLANAS ÓPTIMAS

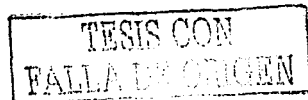
En este capítulo se presentan los resultados generados después de varias pruebas con el programa OPTI. Se demuestra la funcionalidad de la implementación de los GA al problema de optimización de armaduras planas.

6.1 CASO 1

Con el fin de determinar los valores adecuados de los parámetros de optimización en el problema de optimización de armaduras planas, se realizaron varias pruebas sobre el dominio de diseño más simple. Tal dominio se compone de ocho elementos y cinco nodos. Los nodos 1 y 2 son restringidos totalmente y una carga de 500N a 90° se ejerce sobre el nodo 5. El material considerado es acero 1018 con módulo de elasticidad 210 GPa y esfuerzo de fluencia 386 MPa. La sección transversal de los elementos es 0.0009m². Los valores de los parámetros de optimización con los que se experimenta son los siguientes:

- ▶ Tamaño de población(TP): 25, 50, 75 Individuos
- ▶ Generaciones(Gen): 2000
- ▶ Probabilidad de Crossover(Pc): 0.4, 0.6, 0.75
- ▶ Probabilidad de mutación(Pm): 0.01, 0.05, 0.1
- ▶ Elitismo(%E): 10% si aplica
- ▶ Crossover: sencillo o de doble punto

En la Figura 6.1 se muestran algunos de los individuos que componen la población inicial. La diversidad que existe en la población favorece el proceso de optimización.



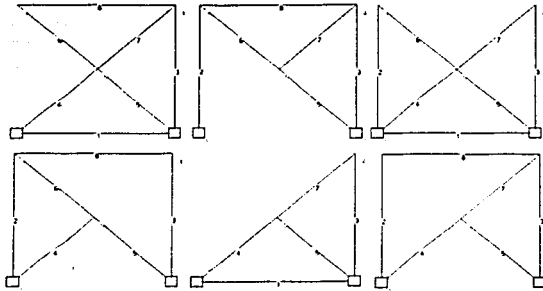


Figura 6.1 Diversidad de la población inicial en el Caso 1

En la Figura 6.2 se observa la armadura óptima para este caso. Su volumen es 20.71% el volumen del dominio de diseño. El esfuerzo promedio en el dominio de diseño es 132.2 KPa y el del óptimo es 392.8 KPa que implica un aumento en el nivel de esfuerzos promedio del 297.06%.



Figura 6.2 Armadura óptima Caso 1

En primer lugar se experimentó con el tamaño de la población dejando fijos los valores de $P_c=0.7$, $P_m=0.01$, $Gen=2000$, Crossover de punto sencillo y Elitismo nulo. Los resultados obtenidos muestran que un tamaño de población mayor permite encontrar estructuras de mejor desempeño más rápidamente. Para tamaños de población 25, 50, 75 la estructura de mayor desempeño se encontró en 756, 150 y 45 generaciones respectivamente. Sin embargo, en ningún caso se obtuvo el óptimo. La armadura obtenida es la mostrada en la Figura 6.3.

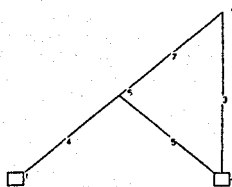


Figura 6. 3 Mejor armadura obtenida con la variación del tamaño de la población. Claramente no es la óptima solución

Para experimentar con el valor de P_c se utilizaron los valores 0.45, 0.60 y 0.75; $TP=25$, $Gen=2000$, $P_m=0.01$, Crossover sencillo y Elitismo nulo. Nuevamente no fue alcanzada la estructura óptima, sin embargo, la armadura intermedia (Figura 6.3) fue encontrada en las generaciones 561, 519 y 36 para P_c de 0.45, 0.60 y 0.75 respectivamente. Es notable el beneficio que proporciona a la búsqueda el emplear un valor alto de P_c .

La influencia del valor de P_m se determinó con valores de 0.01, 0.05 y 0.1, dejando fijos a $P_c=0.75$, Crossover sencillo y elitismo nulo, $TP=25$ y $Gen=2000$. Para esta combinación de parámetros y valor de $P_m=0.01$ no se alcanzó el óptimo. No así para los valores P_m 0.05 y 0.1, los cuales permitieron alcanzarlo en las generaciones 733 y 224 respectivamente. Los resultados indican que el valor de P_m es trascendental durante la búsqueda que desarrolla el Algoritmo.

El Crossover sencillo y el Crossover de doble punto se experimentaron con valores de $TP=25$, $Gen=2000$, $P_c=0.75$, $P_m=0.1$ y Elitismo nulo. En este caso el Crossover sencillo obtiene la estructura óptima en la generación 224 y el Crossover de doble punto tiene el mismo resultado pero en una generación anterior, la número 183, mostrando un beneficio mayor.

Por último, se revisa la influencia del elitismo con los siguientes valores: $TP=25$, $Gen=2000$, $P_c=0.75$, $P_m=0.1$, y elitismo del 10%. Valores de elitismo muy altos conllevan pérdida de diversidad en la población, por lo que debe utilizarse moderadamente lo mismo que

CAPÍTULO 6. APLICACIÓN DEL PROGRAMA OPTI

el operador de mutación. El efecto del elitismo en la búsqueda es destacable. La armadura de más alto desempeño se alcanzó en la generación 11.

En resumen, los resultados obtenidos en este caso manifiestan que el valor de TP tiene una influencia menor que la que tienen los valores de Pc y Pm. Por otra parte, el Crossover de doble punto demuestra ser superior que el Crossover sencillo; además, el uso de un porcentaje de elitismo moderado trae beneficios considerables.

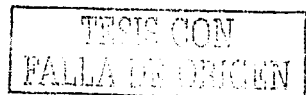
De lo anterior se desprende que una combinación de los parámetros de optimización compuesta por Pc oscilando alrededor de 0.75, Pm de 0.1, Crossover de doble punto y elitismo moderado es adecuada en la optimización de armaduras.

6.2 CASO 2

Fue considerado un dominio de 15 elementos de 1m de altura y 0.5m de base, dos divisiones verticales. Los nodos de la base fueron fijados y se aplicó una carga de 500N en el nodo que se indica en la Figura 6.5(a). El material de la estructura fue acero 1018 de sección transversal 0.0009m^2 . El conjunto de parámetros alimentados al programa es:

- ▶ TP=20
- ▶ Gen=2000
- ▶ Pc=0.7
- ▶ Pm=0.1
- ▶ Crossover de doble punto
- ▶ 2 individuos elitistas por generación

El volumen del dominio es $5.7 \times 10^{-3}\text{m}^3$ y su esfuerzo promedio 137.3KPa. En la búsqueda por la armadura de mejor desempeño se generan las configuraciones mostradas en la Figura 6.4. Existe una eliminación gradual de material evidente durante el proceso de optimización, comenzando en la base de la estructura.



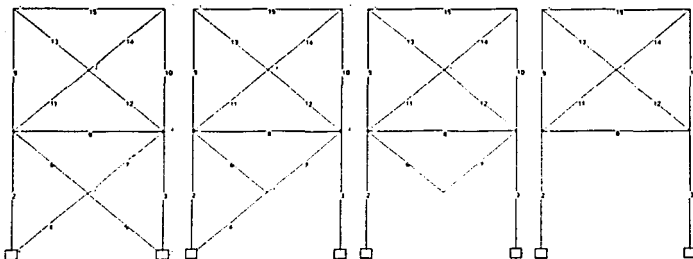


Figura 6.4 Armaduras de alto desempeño precedentes a la óptima Caso2

Los resultados obtenidos son los siguientes: el volumen del óptimo es $2.4 \times 10^{-3} \text{m}^3$, 42.8% del volumen del dominio, y el esfuerzo promedio se incrementó 136% siendo de 323.8 KPa. La Figura 6.5(b) muestra la armadura óptima correspondiente que apareció en la generación 113.

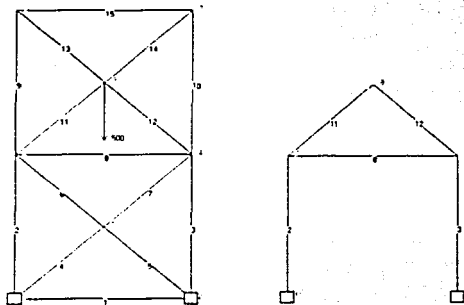


Figura 6.5 (a) Dominio de diseño Caso 2, (b) Estructura óptima

El gráfico de la Figura 6.6 corresponde a la aptitud relativa de los mejores individuos en cada generación. La aptitud relativa se obtiene dividiendo la aptitud de cada individuo entre

la del dominio de diseño. El ascenso continuo de la aptitud es resultado de la aplicación de elitismo en el proceso de optimización. Al conservar generación a generación los mejores resultados del proceso, se eleva la aptitud promedio de las generaciones en cada iteración.

Los individuos elitistas aportan a sus descendientes la información, contenida en las cadenas codificadas, que les proporciona una aptitud elevada. Esta herencia se difunde a la población con el paso de las generaciones incrementando paulatinamente el desempeño general de la población. Sin embargo, gracias a que el porcentaje de elitismo es pequeño, se permite que la diversidad de la población pueda variar, lo que abre la posibilidad de explorar una amplia gama de opciones para hallar mejores individuos que reemplacen a los elitistas.

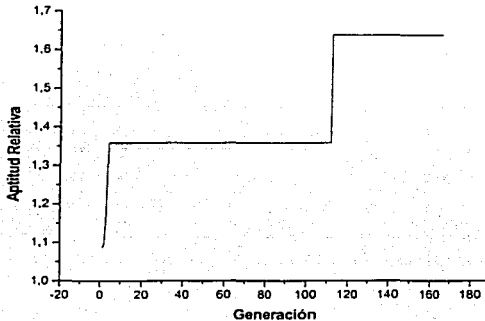


Figura 6. 6 Aptitud relativa de las mejores estructuras Caso 2

6.3 CASO 3

En este caso el dominio de diseño tiene una longitud de 0.75m y una altura de 0.25m, tres divisiones horizontales. Los nodos del extremo izquierdo fueron fijados y una carga de 1000N se aplicó al nodo superior derecho. El total de elementos para esta configuración es 22.



CAPÍTULO 6. APLICACIÓN DEL PROGRAMA OPTI

El material empleado fue acero 1018 con sección transversal 0.0009m^2 . Los parámetros de optimización fueron fijados en los siguientes valores.

- ▶ TP=20
- ▶ Gen=1000
- ▶ Pc=0.75
- ▶ Pm=0.1
- ▶ Crossover de doble punto
- ▶ Elitismo nulo

El análisis del dominio de diseño indica que éste tiene un volumen igual a $4.16 \times 10^{-3} \text{ m}^3$ y un esfuerzo promedio de 908315Pa. Se puede observar en la siguiente Figura 6.7.

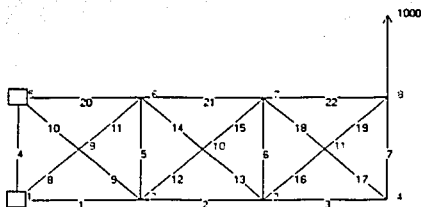


Figura 6. 7 Dominio de diseño Caso 3

La evolución hacia la armadura de mejor desempeño arrojó las configuraciones de la Figura 6.8. Conforme el proceso de optimización avanza existe una eliminación gradual de material. Lo anterior se debe a la definición de la función de aptitud, la cual se compone por la combinación entre la energía de deformación y el volumen.

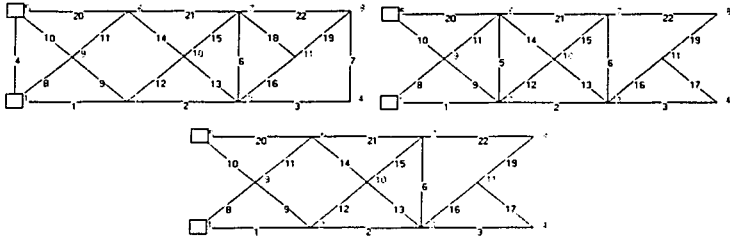


Figura 6. 8 Configuraciones anteriores al óptimo Caso 3

La armadura de mejor desempeño se determinó hasta la generación 85 y presentó las siguientes características. Volumen 74.5% el del dominio de diseño y esfuerzo promedio 1173505.3Pa, es decir, aproximadamente 30% mayor al del dominio incrementando la eficiencia de la armadura.

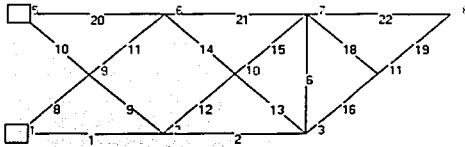


Figura 6. 9 Armadura óptima Caso 3

A pesar de que en esta ocasión se descartó el uso de elitismo, se llegó a la meta rápidamente. Esto se debe a que los diferentes parámetros de optimización participan de manera activa en el proceso, por lo que la ausencia de alguno de ellos o un valor bajo del mismo puede ser compensado.

El comportamiento de la aptitud de los mejores individuos se muestra en la Figura 6.10. La tendencia es claramente ascendente al principio, posteriormente converge a partir de la generación la generación 85.

TESIS CON
FALLA DE ORIGEN

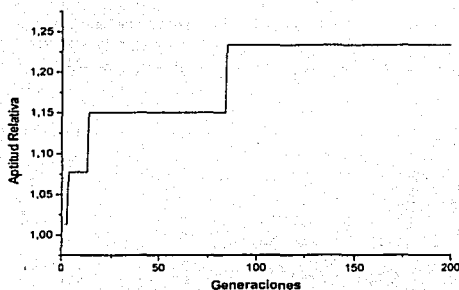


Figura 6. 10 Aptitud relativa de las mejores armaduras Caso 3

6.4 CASO 4

En un dominio de 41 elementos se realizó la experimentación de este caso. El dominio de diseño de dimensiones 0.75m de base y 0.5m de altura fue segmentado en 3 secciones horizontales y dos verticales. Dos nodos de la base se restringieron totalmente y se aplicó en el extremo derecho una fuerza de 1000N. El resultado fue la configuración mostrada en la Figura 6.11.

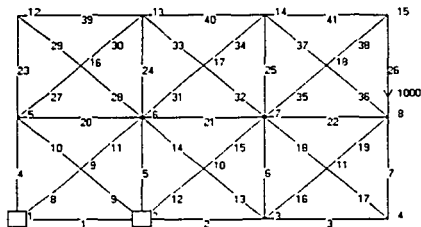


Figura 6. 11 Dominio de diseño Caso-4

TRABAJO CON
FALLA DE ORIGEN

CAPÍTULO 6. APLICACIÓN DEL PROGRAMA OPTI

Para este dominio el volumen es de $7.6 \times 10^{-3} \text{ m}^3$. El esfuerzo promedio bajo el cual está sometida esta armadura es 512.14 KPa.

Los parámetros de optimización seleccionados para resolver este caso son los siguientes:

- ▶ TP=25
- ▶ Gen=2500
- ▶ Pc=0.75
- ▶ Pm=0.1
- ▶ Crossover de doble punto
- ▶ Elitismo 10%

La solución al problema de optimizar se presentó gradualmente generando las armaduras intermedias ilustradas en la Figura 6.12.

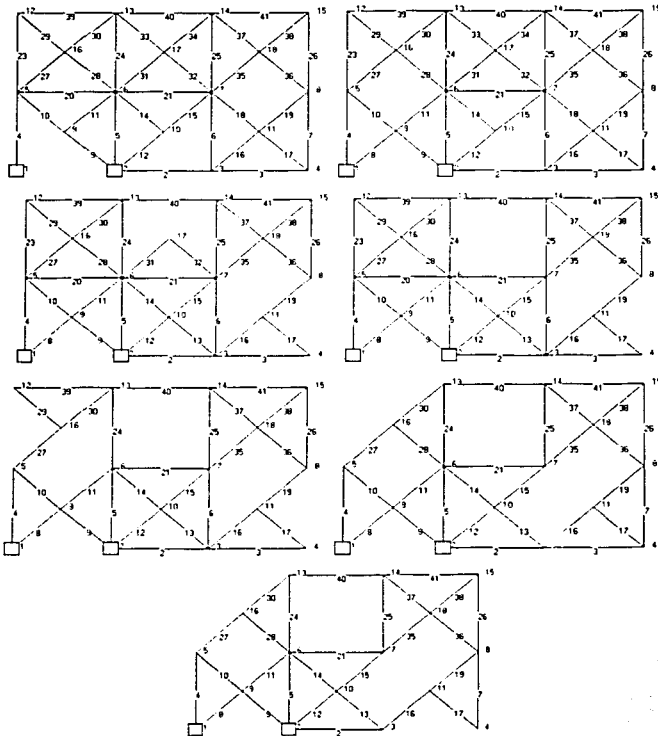


Figura 6.12 Evolución hacia la armadura óptima Caso 4

La solución óptima se halló en la generación 391 y tiene las siguientes características. El volumen 36% menor siendo de $4.9 \times 10^{-3} \text{ m}^3$, el esfuerzo promedio se incrementa 44.2% y su valor es 738.6 KPa. La configuración de esta armadura se puede observar en la Figura 6.13.

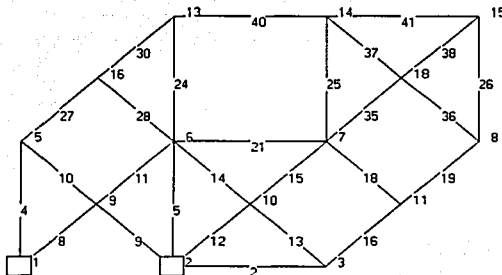


Figura 6. 13 Configuración óptima para el Caso 4

El desarrollo de la aptitud en la población, generación a generación, corrobora lo mostrado en la Figura 6.12. Gradualmente la aptitud del mejor individuo en las generaciones se incrementa hasta la generación 391. En este punto el valor de la aptitud no se incrementa y la ejecución del programa termina por repetición del mejor individuo (Figura 6.14).

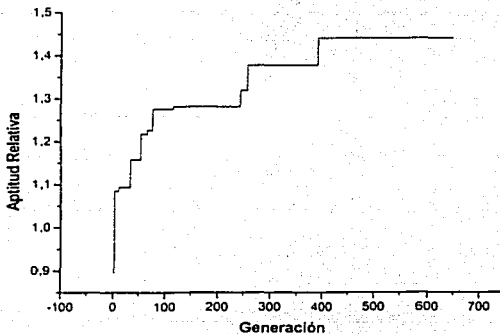
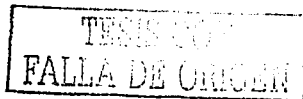


Figura 6. 14 Desarrollo de la aptitud Caso 4



6.5 CASO 5

El objetivo es encontrar la armadura óptima para el problema siguiente. El dominio mide 0.75m de largo y 0.5m de altura, 3 divisiones horizontales y 2 verticales. El material acero 1018 de sección transversal 0.0009m^2 . Los nodos extremos de la base se restringieron completamente y se aplicaron dos cargas verticales hacia abajo de magnitud 1000N en los nodos 13 y 14 según la numeración de la Figura 6.15.

El volumen inicial del dominio es $7.6 \times 10^{-3} \text{ m}^3$ y su esfuerzo promedio 286KPa. El siguiente conjunto de valores fue utilizado para los parámetros de optimización:

- ▶ TP=25
- ▶ Gen=1600
- ▶ $P_c=0.75$
- ▶ $P_m=0.1$
- ▶ Crossover de doble punto
- ▶ Elitismo 10%

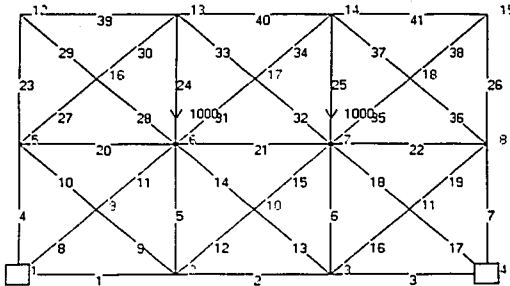
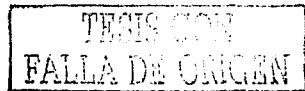


Figura 6. 15 Dominio de diseño Caso 5



Óptimos locales fueron configurados antecedendo al óptimo general. Estos son mostrados en la Figura 6.16.

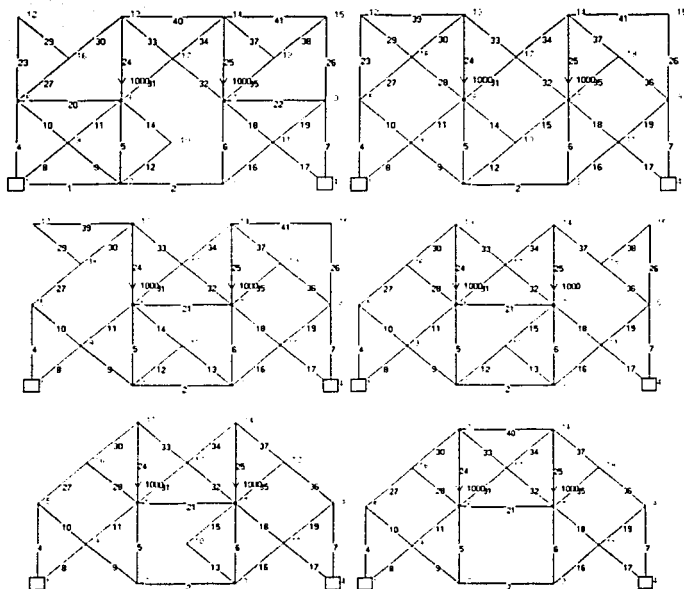


Figura 6.16 Antecesoros de la solución óptima

Claramente la tendencia durante el proceso define una estructura simétrica, lo cual era de esperarse debido a la simetría del problema en cuanto a restricciones y cargas.

La estructura óptima se muestra en la Figura 6.17 y tiene las siguientes características. Esfuerzo promedio 276Kpa y volumen de $4.4 \times 10^{-3} \text{ m}^3$. Lo anterior representa 96.5% el esfuerzo promedio del dominio y 57.9% del volumen.

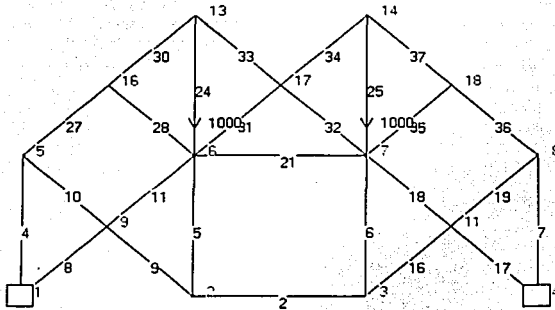


Figura 6. 17 Armadura óptima para el Caso 5

En el gráfico de la Figura 6.18 se muestra la variación de la aptitud relativa. Es evidente el crecimiento de ésta durante la optimización. Existen varios cambios en la aptitud relativa lo cual indica que el algoritmo computacional explora gran número de opciones de solución.

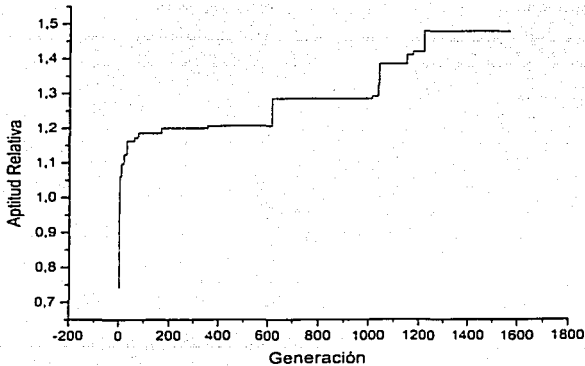


Figura 6. 18 Gráfico de aptitud relativa Caso 5

6.6 CASO 6

Nuevamente se empleo el programa OPTI en un dominio de 0.75m de largo por 0.5m de altura. En esta ocasión se aplicaron dos fuerzas simétricas oblicuas sobre dos nodos extremos de la armadura. Dos nodos superiores se fijaron completamente como se muestra en la Figura 6.19.

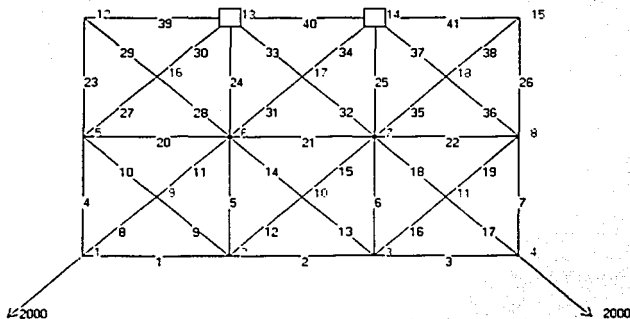


Figura 6. 19 Dominio de diseño problema Caso 6

Los parámetros alimentados al programa son los siguientes:

- ▶ TP=25
- ▶ Gen=2000
- ▶ Pc=0.75
- ▶ Pm=0.08
- ▶ Crossover de doble punto
- ▶ Elitismo 10%

TESIS CON
FALLA DE ORIGEN

Para este dominio de diseño el volumen inicial es $7.6 \times 10^{-3} \text{ m}^3$ y su esfuerzo promedio es 431 KPa.

Como en los casos anteriores al proceso evolutivo involucra la aparición de armaduras que son óptimos locales. Tales estructuras se encuentran en la Figura 6.20.

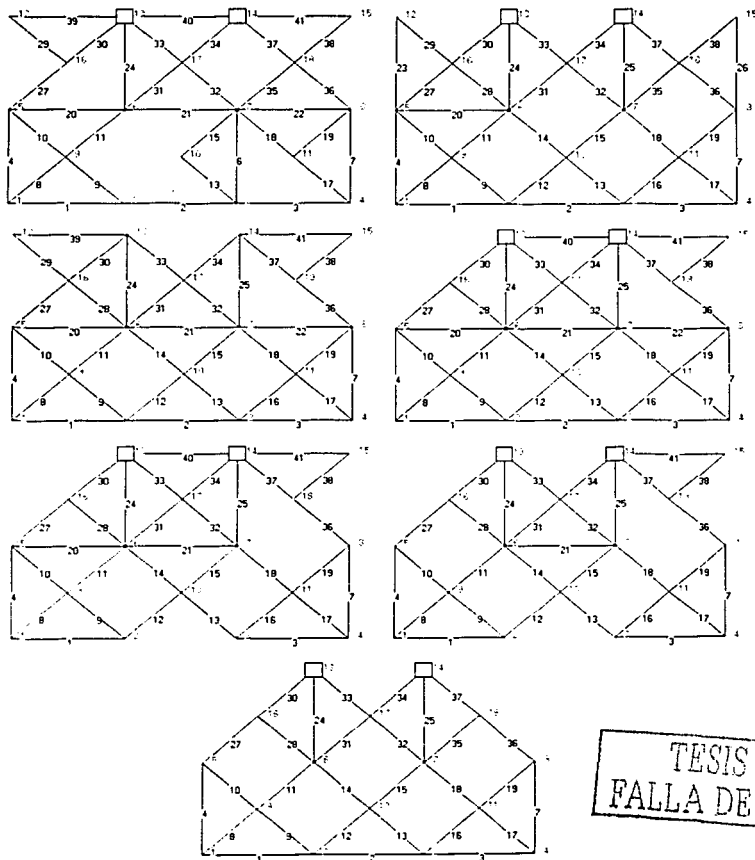


Figura 6. 20 Estructuras antecesoras de la óptima Caso 6

CAPÍTULO 6. APLICACIÓN DEL PROGRAMA OPTI

La armadura óptima se presenta en la generación 265 y está caracterizada por un volumen de $4.85 \times 10^{-3} \text{ m}^3$ y un esfuerzo promedio igual a 440 KPa. Estos valores representan 64% del volumen del dominio de diseño y 102% el esfuerzo promedio original.

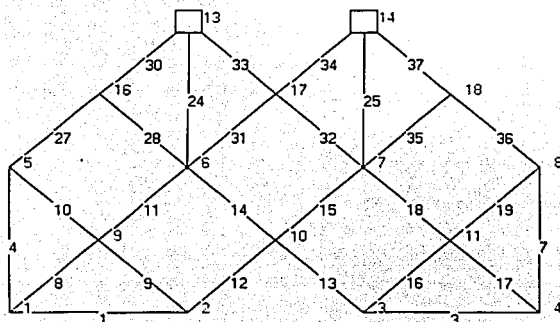


Figura 6. 21 Armadura óptima Caso 6

El gráfico de la Figura 6.22 presenta el comportamiento ascendente de la aptitud de los mejores individuos durante la ejecución del programa.

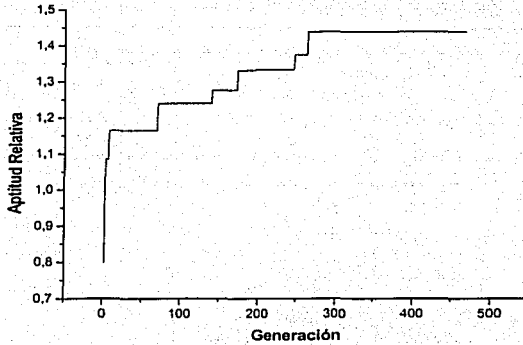


Figura 6. 22 Comportamiento de la aptitud relativa Caso 6

TESIS CON
FALLA DE ORIGEN

DISCUSIÓN DE RESULTADOS Y CONCLUSIONES

DISCUSIÓN DE RESULTADOS Y CONCLUSIONES

Los resultados obtenidos al aplicar el programa OPTI en la solución de armaduras planas óptimas demuestran la efectividad del algoritmo computacional implementado. Para cada uno de los seis casos presentados se obtiene una solución que satisface los requerimientos impuestos, es decir, cargas y restricciones. Además, el incremento consecutivo en la aptitud de las diferentes soluciones corrobora que la solución final de cada caso de estudio corresponde a una armadura de alto desempeño y que con seguridad se aproxima a una estructura óptima.

Por otra parte, la variedad de soluciones que explora el programa OPTI, que es resultado de emplear GA, le da confiabilidad a los resultados que genera.

El programa OPTI es resultado de llevar a la práctica el concepto DOPE ilustrado en el Capítulo I. En el mismo se asevera que existen grandes ventajas al aplicar DOPE en lugar de diseño convencional. OPTI tiene capacidad para manejar gran cantidad de soluciones a la vez para un problema dado. Por ejemplo, en el caso de estudio 5 se fijó un tamaño de población de 25 y 1600 generaciones, lo que significa que se analizaron un máximo de 40000 posibles soluciones (evidentemente con repeticiones) durante la ejecución. En este sentido, es claro que un diseñador no podría analizar el mismo número de posibles soluciones. El poder de exploración de los GA es una característica que le da carácter robusto a estos métodos de búsqueda.

Como se indica en el apartado 6.1, en el desarrollo de los casos de estudio se emplearon los siguientes valores de los parámetros de optimización: P_c y P_m alrededor de 0.75 y 0.1 respectivamente, Crossover de doble punto y Elitismo del 10%. Estos valores fueron determinados experimentalmente al emplear OPTI para encontrar la armadura óptima para un dominio de diseño muy sencillo.

Algunas personas involucradas con los GA (por ejemplo De Jong, 1975 y Goldberg, 1989) han reportado valores recomendados para los parámetros de optimización y algunos estudios en los que se obtienen relaciones entre ellos y se demuestra que no son lineales. Con

DISCUSIÓN DE RESULTADOS Y CONCLUSIONES

base en esto, es probable determinar combinaciones de valores de los parámetros de optimización diferentes a los arriba enunciados para cada caso de estudio que produzcan un mejor desempeño de OPTI.

La adición de filtros para detectar soluciones no factibles (mecanismos o estructuras no rígidas) delimita el dominio de búsqueda a través del cual el algoritmo computacional explora lo cual se traduce en un menor tiempo de ejecución.

A continuación se presentan las conclusiones generadas de este trabajo.

- ▶ El algoritmo computacional desarrollado con base en el concepto de Diseño Evolutivo Óptimo (DOPE) es una herramienta factible y efectiva en el diseño de armaduras planas
- ▶ DOPE es superior al diseño convencional en cuanto a la capacidad de generar y analizar gran cantidad de soluciones a la vez
- ▶ La facilidad de uso de la aplicación desarrollada permite que sea empleada no sólo por diseñadores de armaduras
- ▶ Los resultados obtenidos demuestran que OPTI genera soluciones cercanas al óptimo más que sólo soluciones para armaduras planas.

PERSPECTIVAS

OPTI es una aplicación desarrollada para encontrar armaduras planas óptimas sometidas a restricciones y cargas nodales. En este sentido, su utilización se restringe a resolver problemas en los cuales se empleen armaduras planas o en los que sea posible simplificar una armadura tridimensional en una plana.

En general, los métodos de la computación evolutiva requieren tiempos de ejecución elevados. Esta característica es heredada por OPTI. Por citar un ejemplo, el tiempo empleado en determinar la solución del caso de estudio 6 fue aproximadamente de 40 horas⁴.

Por otra parte, la capacidad de exploración de los métodos de la computación evolutiva, en particular de los GA, puede aprovecharse llevando el concepto DOPE al diseño de elementos estructurales o mecánicos más complejos que las armaduras planas.

Con base en lo anterior, se plantean las siguientes perspectivas para el trabajo actual

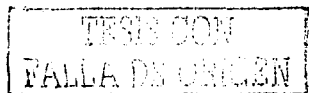
- Aplicar el concepto DOPE al diseño de elementos más complejos, por ejemplo sólidos.
- Emplear técnicas computacionales que permitan reducir drásticamente el tiempo de ejecución, por ejemplo Procesamiento Paralelo.

⁴ Utilizando un procesador Pentium IV a 1.5 MHz y 256 MB en RAM

REFERENCIAS

REFERENCIAS

1. Alexandrov, N.M. et. al. (1998) *A Trust Region Framework for Managing the Use of Approximation Models in Optimization*. NASA Langley Research Center, Hampton VA.
2. Alexandrov, N.M. et. al. (2000) *Optimization with Variable-Fidelity Models Applied to Wing Desing*. AIAA 38^o Aerospace Sciences Meeting and Exhibit, Enero 2000, Reno Nevada.
3. Aljlal, M. y Frieder, O (1997). *Effective Arabic-English Cross-Language Information Retrieval via Machine-Readable Dictionaries and Machine Translation*. Information Retrieval Laboratory, Institute of Technology Chicago, IL
4. Bendsoe, M.P. (1995) *Optimization of Structural Topology, Shape, and Material*, Springer Heidelberg
5. Chandrupatla, T., Belegundu, A (1999). *Introducción al Estudio del Elemento Finito en Ingeniería*. 2^o Edición. Prentice Hall. México
6. Davidson, R. y Harel, D. (1996) *Drawing Graphs Nicely Using Simulated Annealing*. ACM Transactions on Graphics, Vol. 15, No. 4 pp 301-331
7. Davis, M., and Dunning, T (1995). *Query Translation using Evolutionary Programming for Multilingual Information Retrieval*. The 4th Evolutionary Programming Conf.
8. Fleming, P.J. y Purshouse, R.C. (2001) *Genetic Algorithms in Control Systems Engineering*. Research Report. Department of Automatic Control and Systems Engineering, University of Sheffield. Sheffield, Gran Bretaña
9. Floreano, D., Nolfi, S., Mondada, F. (1998) *Competitive Co-Evolutionary Robotics: From Theory to Practice*. Laboratory of Microcomputing Swiss Federal Institute of Technology
10. Fogel, D.B (1993a) *Applying Evolutionary Programming to Selected Traveling Salesman Problems*. Cybernetics and Systems, Vol. 24, pp. 27-36
11. Fogel, D.B. (1993b) *Empirical Estimation of the Computation Required to Discover Approximate Solutions to the Traveling Salesman Problem Using Evolutionary Programming*. Proceedings of Second Annual Conference on Evolutionary Programming. D.B. Fogel y W. Atmar, Evolutionary Programming Society. Pp. 56-61
12. Goldberg, D.E. (1989) *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Publishing Company. EUA



13. González, O. (2002) *Análisis Estructural*. Universidad Autónoma Metropolitana. LIMUSA-NORIEGA Editores. México
14. Gross, R., et. Al (1996). *Evolving Chess Playing Programs*. Department of Computer Science, University of Dortmund. Dortmund, Alemania
15. Hadamard, J. (1949) *The Psychology of Invention in the Mathematical Field*. Princenton University Press, Nueva York
16. Holland, J.H. (1975) *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. Ann Arbor, MI: University of Michigan Press. 2ª Edición
17. Housner, G.W. y Houdson, D.E (1965). *Mecánica Aplicada Estática*. Compañía Editorial Continental. México
18. Kane, C. y Schoenauer M (1996). *Topological Optimum Design using Genetic Algorithms. Control and Cybernetics*, Vol. 25 No. 5, 1996
19. Koza, J.R. (1992) *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA: The MIT Press
20. Koza, J.R. (1997) *Genetic Programming*. Encyclopedia of Computer Science and Technology. EUA
21. Lee, K.J. y Zhang, B.T. (1994) *Learning Robot Behaviors by Evolving Genetic Programs*. IEEE Journal of Robotics and Automation
22. Louis, D. (2000) *Gran Libro DELPHI 5*. Alfaomega Marcombo. España
23. Mahlers, Y.P. (2002) *Core Reload Optimization for Equilibrium Cycles using Simulated Annealing and Successive Linear Programming*. Annals of Nuclear Energy 29 (2002). Pergamon
24. Michalewicz, Z. (1992) *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, Berlin
25. Michalewicz, Z. et. al., (1996) *Evolutionary Algorithms for Constrained Engineering Problems*. Dept. of Computer Science, Universidad de Michigan
26. O'Sullivan, B. (1998) *Interactive Constraint – Aided Conceptual Design*. Cork Constraint Science, University College Cork, Irlanda
27. Querin, O.M. (1997) *Evolutionary Structural Optimization: Stress Based Formulation and Implementation*. Ph. D. Dissertation. University of Sydney, Australia.

REFERENCIAS

28. Rozvany, G.I.M. (1989) *Structural Design via Optimality Criteria*. Kluwer Academic Publishers, Dordrecht
29. Ruhl, M., Hartenstein, H., Saupe, D. (1997) *Adaptive Partitionings for Fractal Image Compression*. IEEE International Conference on Image Processing 1997
30. Vanderplaats, G.N. (1984) *Numerical Optimization Techniques for Engineering Design: with Applications*. McGraw-Hill, New York
31. Voss, M. (1999) *Evolutionary Algorithm for Structural Optimization*. Proceedings of Genetic and Evolutionary Computation Conference. Orlando, FL.
32. Voss, M. y Foley, C.M. (1999) *Rank-Based Evolutionary Algorithm for Structural Optimization*. Computers and Structures International Journal
33. Xie, M. y Steven, G.P. (1997) *Evolutionary Structural Optimization*. Springer – Verlag, Londres
34. Zhang, B.T. y Mühlenbein, H. (1993) *Genetic Programming of Minimal Neural Nets Using Occam's Razor*. Processing of the 5^o International Conference on Genetic Algorithms (ICGA 93) Morgan Kaufmann pp 342-349

TESIS CON
FALLA DE ORIGEN