



**UNIVERSIDAD POLITÉCNICA DE CARTAGENA
DEPARTAMENTO DE ESTRUCTURAS Y CONSTRUCCIÓN**

**“DISEÑO ÓPTIMO SIMULTÁNEO
DE TOPOLOGÍA Y GEOMETRÍA
DE ESTRUCTURAS ARTICULADAS
MEDIANTE TÉCNICAS DE CRECIMIENTO”**

Pedro Jesús Martínez Castejón

2003



UNIVERSIDAD POLITÉCNICA DE CARTAGENA
DEPARTAMENTO DE ESTRUCTURAS Y CONSTRUCCIÓN

PROGRAMA DE DOCTORADO:
ANÁLISIS Y DISEÑO AVANZADO DE ESTRUCTURAS

**“DISEÑO ÓPTIMO SIMULTÁNEO
DE TOPOLOGÍA Y GEOMETRÍA
DE ESTRUCTURAS ARTICULADAS
MEDIANTE TÉCNICAS DE CRECIMIENTO”**

TESIS DOCTORAL

Presentada por: D. Pedro Jesús Martínez Castejón

Dirigida por: Dr. Pascual Martí Montrull

Cartagena, julio de 2003



UNIVERSIDAD POLITÉCNICA DE CARTAGENA
Departamento de Estructuras y Construcción

AUTORIZACIÓN DEL DIRECTOR DE LA TESIS

D. Pascual Martí Montrull, Profesor Doctor del Área de Mecánica de Medios Continuos y Teoría de Estructuras, en el Departamento de Estructuras y Construcción

AUTORIZA:

La presentación de la Tesis Doctoral titulada “Diseño óptimo simultáneo de topología y geometría de estructuras articuladas mediante técnicas de crecimiento”, realizada por D. Pedro Jesús Martínez Castejón, bajo su dirección y supervisión, en el Departamento de Estructuras y Construcción, y que presenta para la obtención del grado de Doctor por la Universidad Politécnica de Cartagena.

En Cartagena, a 7 de julio de 2003

EL DIRECTOR DE LA TESIS

Fdo.: Pascual Martí Montrull



UNIVERSIDAD POLITÉCNICA DE CARTAGENA
Departamento de Estructuras y Construcción

AUTORIZACIÓN DEL DEPARTAMENTO RESPONSABLE

D. Pascual Martí Montrull, Director del Departamento de Estructuras y Construcción

INFORMA:

Que la Tesis Doctoral titulada “Diseño óptimo simultáneo de topología y geometría de estructuras articuladas mediante técnicas de crecimiento”, ha sido realizada por D. Pedro Jesús Martínez Castejón, bajo la dirección y supervisión de D. Pascual Martí Montrull y que el Departamento ha dado su conformidad para que sea presentada ante la Comisión de Doctorado.

En Cartagena, a 9 de julio de 2003

EL DIRECTOR DEL DEPARTAMENTO

Fdo.: Pascual Martí Montrull

A Fuensanta

AGRADECIMIENTOS

Quiero agradecer al Dr. Pascual Martí Montrull sus inestimables consejos y el continuo estímulo para llevar a buen término esta tesis doctoral. Recuerdo de forma especial muchos momentos en los que sus sugerencias sutiles y razonamientos elegantes orientaban eficazmente este trabajo.

Quiero agradecer también a mi esposa, Fuensanta, su apoyo incondicional. Espero que algún día pueda recuperar algunos de los momentos que perdí con ella por la realización de esta tesis.

Y, finalmente, a mi familia, a mis compañeros del área de Mecánica de Medios Continuos y Teoría de Estructuras, especialmente a Conchita y Santiago, y a mis amigos, por su ánimo y apoyo desinteresados.

Índice

LISTA DE TABLAS	xxi
LISTA DE FIGURAS	xxiii
LISTA DE SÍMBOLOS	xxix
CAPÍTULO 1 INTRODUCCIÓN Y OBJETIVOS	
1.1 INTRODUCCIÓN	1
1.2 DISEÑO ÓPTIMO DE TOPOLOGÍA Y GEOMETRÍA DE ESTRUCTURAS ARTICULADAS	1
1.2.1 Métodos del universo estructural (<i>ground structure</i>)	1
1.2.2 Métodos de crecimiento	2
1.2.3 Métodos de homogeneización	2
1.3 OBJETIVOS DE LA TESIS	2
1.4 ORGANIZACIÓN DE LA TESIS	2
CAPÍTULO 2 FORMULACIÓN Y RESOLUCIÓN DEL PROBLEMA DE DISEÑO ÓPTIMO DE ESTRUCTURAS	
2.1 INTRODUCCIÓN	5
2.2 FORMULACIÓN DEL PROBLEMA DE DISEÑO ÓPTIMO DE ESTRUCTURAS	6
2.2.1 Variables de diseño	7
2.2.2 Función objetivo	7
2.2.3 Restricciones	8
2.3 TIPOS DE OPTIMIZACIÓN ESTRUCTURAL	9

2.3.1	Optimización de propiedades de la sección transversal.....	9
2.3.2	Optimización de geometría.....	10
2.3.3	Optimización de topología.....	10
2.4	FORMULACIÓN MATEMÁTICA DEL PROBLEMA DE OPTIMIZACIÓN.....	10
2.5	RESOLUCIÓN DEL PROBLEMA DE OPTIMIZACIÓN.....	11
2.5.1	Introducción.....	11
2.5.2	Criterios de optimalidad.....	12
2.5.2.1	<i>Características del método FSD</i>	12
2.5.3	Programación matemática.....	13
2.5.3.1	<i>Búsqueda unidimensional</i>	14
2.5.3.2	<i>Dirección de búsqueda</i>	14
2.5.3.3	<i>Problemas con restricciones</i>	15
2.5.4	Programación cuadrática sucesiva.....	17
2.5.4.1	<i>Introducción</i>	17
2.5.4.2	<i>Descripción general</i>	18
2.5.4.3	<i>Algoritmo de Schittkowski</i>	21
2.5.4.3.1	<i>Descripción</i>	21
2.5.4.3.2	<i>Algoritmo</i>	23

CAPÍTULO 3 *DISEÑO ÓPTIMO DE TOPOLOGÍA DE ESTRUCTURAS ARTICULADAS. MÉTODOS DEL UNIVERSO ESTRUCTURAL*

3.1	INTRODUCCIÓN.....	25
3.2	ANTECEDENTES.....	25
3.3	UNIVERSO ESTRUCTURAL.....	26
3.3.1	Introducción.....	26
3.3.2	Generación.....	26
3.3.2.1	<i>Introducción</i>	26
3.3.2.2	<i>Mapeado de puntos nodales</i>	26
3.3.2.3	<i>Orden de vecindad</i>	28
3.3.2.4	<i>Generación de barras</i>	28
3.3.2.5	<i>Implementación</i>	29
3.3.3	Ejemplos.....	31
3.4	MÉTODOS DE SOLUCIÓN.....	33
3.4.1	Introducción.....	33
3.4.2	Método <i>stress-ratio</i>	33

3.4.2.1	<i>Formulación</i>	33
3.4.2.2	<i>Algoritmo de resolución</i>	33
3.4.2.3	<i>Implementación</i>	33
3.4.2.3.1	Eliminación de barras	33
3.4.3	Método de la <i>compliance</i>	34
3.4.3.1	<i>Formulación</i>	34
3.4.3.2	<i>Algoritmo de resolución</i>	36
3.4.3.3	<i>Implementación</i>	36
3.4.4	Método de Pedersen.....	36
3.4.4.1	<i>Formulación</i>	36
3.4.4.2	<i>Algoritmo de resolución</i>	39
3.4.4.2.1	Variables positivas y negativas.....	40
3.4.4.2.2	Función objetivo no lineal	42
3.4.4.3	<i>Implementación</i>	42
3.4.4.3.1	Generación automática de la topología inicial	43
3.4.4.3.2	Varios estados de cargas	43
3.4.4.3.3	Degeneración del <i>simplex</i>	43
3.4.5	Método de Achtziger	44
3.4.5.1	<i>Formulación</i>	44
3.4.5.2	<i>Algoritmo de resolución</i>	46
3.4.5.3	<i>Implementación</i>	48
3.4.6	Método de búsqueda exhaustiva	48
3.4.6.1	<i>Introducción</i>	48
3.4.6.2	<i>Procedimiento</i>	49
3.4.6.2.1	Generación de topologías.....	49
3.4.6.2.2	Detección de topologías inestables	50
3.4.6.2.3	Detección de topologías equivalentes.....	51
3.4.6.2.4	Análisis y obtención de la función objetivo.....	52
3.4.6.2.5	Elección del diseño óptimo.....	52
3.4.7	Ejemplo.....	53
3.4.7.1	<i>Introducción</i>	53
3.4.7.2	<i>Voladizo con carga puntual</i>	53

CAPÍTULO 4 *DISEÑO ÓPTIMO DE TOPOLOGÍA Y GEOMETRÍA DE ESTRUCTURAS ARTICULADAS. MÉTODOS DE CRECIMIENTO*

4.1	INTRODUCCIÓN	65
4.2	ANTECEDENTES	65

4.3	MÉTODOS DE SOLUCIÓN	66
4.3.1	Método de Rule	66
4.3.1.1	<i>Formulación</i>	66
4.3.1.2	<i>Algoritmo de resolución</i>	67
4.3.2	Método de McKeown	67
4.3.2.1	<i>Formulación</i>	67
4.3.2.2	<i>Algoritmo de resolución</i>	68
4.3.3	Método de Bojczuk y Mróz	68
4.3.3.1	<i>Formulación</i>	68
4.3.3.2	<i>Algoritmo de resolución</i>	71

CAPÍTULO 5 ***DISEÑO ÓPTIMO SIMULTÁNEO DE TOPOLOGÍA Y
GEOMETRÍA DE ESTRUCTURAS ARTICULADAS.
MÉTODO PROPUESTO***

5.1	INTRODUCCIÓN	73
5.2	DISEÑO ÓPTIMO SIMULTÁNEO DE TOPOLOGÍA Y GEOMETRÍA.....	73
5.3	OBTENCIÓN DE LA TOPOLOGÍA INICIAL	75
5.3.1	Introducción.....	75
5.3.2	Definición y/o generación de los nudos iniciales.....	75
5.3.3	Obtención de la topología inicial	77
5.4	OPTIMIZACIÓN DE TOPOLOGÍA.....	79
5.4.1	Introducción.....	79
5.4.2	Optimización de topología mediante los métodos basados en el universo estructural	79
5.5	OPTIMIZACIÓN DE GEOMETRÍA.....	81
5.5.1	Introducción.....	81
5.5.2	Formulación del problema de optimización de geometría.....	81
5.5.2.1	<i>Variables de diseño</i>	81
5.5.2.2	<i>Función objetivo</i>	82
5.5.2.3	<i>Restricciones de diseño</i>	82
5.5.2.4	<i>Formulación matemática del problema de optimización de geometría</i>	82
5.5.3	Cálculo de las derivadas analíticas de la función objetivo.....	83
5.5.4	Cálculo de las derivadas segundas analíticas de la función objetivo	86
5.5.5	Resolución del problema de optimización de geometría	90
5.6	CRECIMIENTO DE LA TOPOLOGÍA	91
5.6.1	Introducción.....	91

5.6.2	Definición de los nuevos nudos	91
5.6.2.1	<i>Método general</i>	91
5.6.2.1.1	Puntos de una rejilla en el dominio de definición de la estructura	91
5.6.2.1.2	Puntos aleatorios en el dominio de definición de la estructura	91
5.6.2.1.3	Entorno de los puntos medios de las barras	92
5.6.2.1.4	Entorno de los cruces de las barras	92
5.6.2.2	<i>Método basado en las direcciones principales de tensión</i>	92
5.6.2.2.1	Mejora de la ortogonalidad de las barras	93
5.6.2.2.2	División de una barra	93
5.6.2.2.3	División de dos barras que se cruzan	95
5.6.2.3	<i>Elección de la posición del nuevo nudo</i>	96
5.6.2.3.1	División de una barra	96
5.6.2.3.2	División de dos barras que se cruzan	97
5.6.3	Elección de las nuevas barras a añadir	97
5.6.3.1	<i>División de una barra</i>	97
5.6.3.2	<i>División de dos barras que se cruzan</i>	99
5.6.3.3	<i>Criterios prácticos para limitar el número de barras a añadir al nuevo nudo</i>	100
5.6.3.3.1	Limitar el número de cruces de barras	100
5.6.3.3.2	Limitar el número máximo de barras en exceso	101
5.6.4	Elección de la barra o cruce de barras a dividir	102
5.6.5	Resumen del proceso de crecimiento de la topología	104
5.7	COMPROBACIÓN DE LA TOPOLOGÍA ÓPTIMA EN LA NUEVA GEOMETRÍA	105
5.8	EJEMPLOS	107
5.8.1	Viga con apoyos fijos	107
5.8.2	Viga con disco circular rígido	109

CAPÍTULO 6 DISEÑO ÓPTIMO SIMULTÁNEO DE TOPOLOGÍA Y GEOMETRÍA DE ESTRUCTURAS ARTICULADAS. APLICACIÓN INFORMÁTICA

6.1	INTRODUCCIÓN	113
6.2	CLASES DEFINIDAS	113
6.2.1	Introducción	114
6.2.2	Archivo: Principal.h	114
6.2.3	Archivo: Mef.h	114

6.2.4	Archivo: Analisis.h	115
6.2.5	Archivo: Optimizacion.h	117
6.2.6	Archivo: Matriz.h	120
6.2.7	Archivo: Expresion.h	120
6.2.8	Archivo: Punto.h	120
6.2.9	Archivo: Linea.h	121
6.2.10	Archivo: Area.h	122
6.2.11	Archivo: Volumen.h	123
6.2.12	Archivo: Material.h	124
6.2.13	Archivo: Propiedad.h	124
6.2.14	Archivo: Nodo.h	125
6.2.15	Archivo: Elemento.h	125
6.2.16	Archivo: Forma.h	128
6.2.17	Archivo: Gauss.h	129
6.2.18	Archivo: Numeracion.h	129
6.2.19	Archivo: Simplex.h	130
6.2.20	Archivo: OpenGL.h	132
6.3	APLICACIÓN INFORMÁTICA	132
6.3.1	Descripción general	132
6.3.2	Análisis	133
6.3.3	Optimización	140
6.4	ORDENADOR UTILIZADO	141

**CAPÍTULO 7 EJEMPLOS DE APLICACIÓN DEL MODELO DE DISEÑO
ÓPTIMO PROPUESTO**

7.1	INTRODUCCIÓN	143
7.2	EJEMPLOS DE VALIDACIÓN	143
7.2.1	Introducción	143
7.2.2	Voladizo de Michell	144
7.2.2.1	Longitud adimensional $\bar{L} = 0,5$	145
7.2.2.2	Longitud adimensional $\bar{L} = 1,82196$	146
7.2.2.3	Longitud adimensional $\bar{L} = 3,35889$	154
7.2.3	Viga de Messerschmidt-Bölkow-Blohm	160

7.2.3.1	Longitud adimensional $\bar{L} = 2,40196$	161
7.2.3.2	Longitud adimensional $\bar{L} = 5,49846$	166
7.2.4	Viga con disco circular rígido.....	171
7.3	APLICACIONES.....	177
7.3.1	Introducción.....	177
7.3.2	Estructuras de barras articuladas.....	177
7.3.2.1	Introducción	177
7.3.2.2	Voladizo de Michell con carga variable.....	177
7.3.3	Método de bielas y tirantes para el hormigón armado	193
7.3.3.1	Introducción	193
7.3.3.2	Ménsula corta.....	193
7.3.4	Optimización de topología y geometría de mecanismos.....	197
7.3.4.1	Introducción	197
7.3.4.2	Método de Frecker.....	197
7.3.4.3	Método de crecimiento propuesto.....	199
7.3.4.4	Mecanismo extractor de disquetes.....	199
CAPÍTULO 8 CONCLUSIONES Y TRABAJOS FUTUROS		
8.1	INTRODUCCIÓN.....	205
8.2	TRABAJOS REALIZADOS	205
8.3	CONCLUSIONES.....	206
8.4	TRABAJOS FUTUROS.....	207
	REFERENCIAS BIBLIOGRÁFICAS	209

Lista de tablas

Tabla 3.1	Voladizo con carga puntual. Diseños óptimos obtenidos mediante el método <i>stress-ratio</i> (sin pandeo)	56
Tabla 3.2	Voladizo con carga puntual. Diseños óptimos obtenidos mediante el método de la <i>compliance</i> (sin pandeo).....	58
Tabla 3.3	Voladizo con carga puntual. Diseños óptimos obtenidos mediante el método de Pedersen (sin pandeo)	59
Tabla 3.4	Voladizo con carga puntual. Diseños óptimos obtenidos mediante el método de Achtziger (sin pandeo)	60
Tabla 3.5	Voladizo con carga puntual. Diseños óptimos obtenidos mediante el método de búsqueda exhaustiva (sin pandeo).....	63
Tabla 7.1	Voladizo de Michell ($\bar{L} = 1,82196$). Diseños óptimos obtenidos mediante el método <i>stress-ratio</i>	150
Tabla 7.2	Voladizo de Michell ($\bar{L} = 1,82196$). Diseños óptimos obtenidos mediante el método de la <i>compliance</i>	151
Tabla 7.3	Voladizo de Michell ($\bar{L} = 1,82196$). Diseños óptimos obtenidos mediante el método de Pedersen	151
Tabla 7.4	Voladizo de Michell ($\bar{L} = 1,82196$). Diseños óptimos obtenidos mediante el método de Achtziger	151
Tabla 7.5	Voladizo de Michell ($\bar{L} = 1,82196$). Diseños óptimos obtenidos mediante el método de búsqueda exhaustiva.....	152
Tabla 7.6	Voladizo de Michell ($\bar{L} = 1,82196$). Diseños óptimos obtenidos mediante el método de crecimiento propuesto	152
Tabla 7.7	Voladizo de Michell ($\bar{L} = 3,35889$). Diseños óptimos obtenidos mediante el método de la <i>compliance</i>	158
Tabla 7.8	Voladizo de Michell ($\bar{L} = 3,35889$). Diseños óptimos obtenidos mediante el método de crecimiento propuesto (añadiendo los nudos de uno en uno y de dos en dos).....	158
Tabla 7.9	Viga MBB ($\bar{L} = 2,40196$). Diseños óptimos obtenidos mediante el método de la <i>compliance</i>	165

Tabla 7.10	Viga MBB ($\bar{L} = 2,40196$). Diseños óptimos obtenidos mediante el método de crecimiento propuesto	165
Tabla 7.11	Viga MBB ($\bar{L} = 5,49846$). Diseños óptimos obtenidos mediante el método de la <i>compliance</i>	169
Tabla 7.12	Viga MBB ($\bar{L} = 5,49846$). Diseños óptimos obtenidos mediante el método de crecimiento propuesto	169
Tabla 7.13	Viga con disco circular rígido. Diseños óptimos obtenidos mediante el método de la <i>compliance</i>	175
Tabla 7.14	Viga con disco circular rígido. Diseños óptimos obtenidos mediante el método de crecimiento propuesto	175

Lista de figuras

Figura 2.1	Diseño por prueba y error	5
Figura 2.2	Diseño optimizado	6
Figura 2.3	Método <i>Steepest Descent</i>	15
Figura 2.4	Efecto de las restricciones sobre el mínimo. Mínimos relativos	16
Figura 3.1	Proceso de mapeo y transformación (bidimensional)	27
Figura 3.2	Diferentes órdenes de vecindad de un nudo (bidimensional).....	28
Figura 3.3	Generación de barras en las líneas (bidimensional)	28
Figura 3.4	Generación de barras para órdenes de vecindad 1, 2 y 3 (bidimensional)	29
Figura 3.5	Modelado de un dominio con cuadriláteros cuadráticos (8 nodos).....	29
Figura 3.6	Universo estructural para un dominio cuadrado	31
Figura 3.7	Universo estructural para un dominio cuadrado con un agujero.....	31
Figura 3.8	Universo estructural para una ménsula	32
Figura 3.9	Universo estructural para una torre.....	32
Figura 3.10	Función área.....	38
Figura 3.11	Universo estructural de 15 barras y 12 grados de libertad	49
Figura 3.12	Universo estructural de 15 barras y 12 grados de libertad. Topología isostática (“10101011111111”)... ..	49
Figura 3.13	Universo estructural de 15 barras y 12 grados de libertad. Topologías inestables por falta de vínculos independientes (“xxxxxx0xx0xxxxx”)	50
Figura 3.14	Universo estructural de 15 barras y 12 grados de libertad. Topologías inestables por hiperestaticidad local (“xxxx1lxxlxxlxx1”).....	51
Figura 3.15	Universo estructural de 15 barras y 12 grados de libertad. Topologías equivalentes (misma función objetivo)	52
Figura 3.16	Voladizo con carga puntual en la parte inferior del extremo libre (Achtziger, 1999b).....	53
Figura 3.17	Voladizo con carga puntual en la parte inferior del extremo libre. Evolución del diseño óptimo obtenido mediante el método <i>stress-ratio</i> y un universo 15x3 (sin pandeo).....	54

Figura 3.18	Voladizo con carga puntual en la parte inferior del extremo libre. Diseños óptimos obtenidos mediante el método <i>stress-ratio</i> (sin pandeo).....	55
Figura 3.19	Voladizo con carga puntual en la parte inferior del extremo libre. Diseños óptimos obtenidos mediante el método de la <i>compliance</i> (sin pandeo).....	57
Figura 3.20	Voladizo con carga puntual en la parte inferior del extremo libre. Diseños óptimos obtenidos mediante el método de Pedersen (sin pandeo).....	59
Figura 3.21	Voladizo con carga puntual en la parte inferior del extremo libre. Diseños óptimos obtenidos mediante el método de Achtziger (sin pandeo).....	60
Figura 3.22	Voladizo con carga puntual en la parte inferior del extremo libre. Evolución del diseño óptimo obtenido mediante el método de búsqueda y un universo 5x1x1 (sin pandeo).....	61
Figura 3.23	Voladizo con carga puntual en la parte inferior del extremo libre. Diseños óptimos obtenidos mediante el método de búsqueda exhaustiva (sin pandeo)	62
Figura 3.24	Voladizo con carga puntual en la parte inferior del extremo libre (sin pandeo). (a) Evolución tiempo/nº barras; (b) Evolución función objetivo/tiempo	63
Figura 3.25	Voladizo con carga puntual en la parte inferior del extremo libre. Diseños óptimos obtenidos (con pandeo)	64
Figura 4.1	Estructuras base iniciales: (a) 2 apoyos, 1 carga; (b) 2 apoyos, 3 cargas; (c) 3 apoyos, 2 cargas.....	66
Figura 4.2	Estrategias para añadir nuevos nudos y barras: (a) barra exterior; (b) barra interior.....	67
Figura 4.3	Modos de variación topológica: (a) generación de un nuevo nudo y barra de conexión; (b) nudo de separación e introducción de una barra de conexión; (c) introducción de dos nudos y barras de conexión	70
Figura 5.1	Diagrama de flujo para el diseño óptimo simultáneo de topología y geometría	74
Figura 5.2	Ejemplos de generación de los nudos iniciales, a partir de los datos iniciales, para diferentes casos de desplazamientos impedidos y de tipos de cargas	76
Figura 5.3	Diagrama de flujo para la obtención de la topología inicial.....	77
Figura 5.4	Ejemplos de obtención de las topologías iniciales, a partir de los nudos iniciales, para diferentes casos de desplazamientos impedidos y de tipos de cargas	78
Figura 5.5	Diagrama de flujo para la optimización de topología	80
Figura 5.6	Ejemplos de obtención de las topologías iniciales óptimas para diferentes casos de desplazamientos impedidos y de tipos de cargas.....	80
Figura 5.7	Diagrama de flujo para la optimización de geometría	90
Figura 5.8	Entorno de los puntos medios de las barras	92
Figura 5.9	Entorno de los cruces de las barras	92
Figura 5.10	Mejora de la ortogonalidad al dividir una barra. En cada uno de los nudos de la barra están unidas dos barras con esfuerzos axiales de diferente signo al suyo.....	94

Figura 5.11	Mejora de la ortogonalidad al dividir una barra. En cada uno de los nudos de la barra está unida una barra con esfuerzo axial de diferente signo al suyo.....	94
Figura 5.12	Mejora de la ortogonalidad al dividir una barra. En uno de los nudos de la barra no hay ninguna barra conectada con esfuerzo axial de diferente signo al suyo y en el otro hay una.....	95
Figura 5.13	Mejora de la ortogonalidad al dividir dos barras que se cruzan.....	95
Figura 5.14	Valor medio para una aproximación polinómica cúbica.....	96
Figura 5.15	Valor máximo para una aproximación polinómica cúbica.....	97
Figura 5.16	Ejemplos donde la mejor barra a añadir no es la más perpendicular o la más corta.....	98
Figura 5.17	Universo estructural completo del nuevo nudo al dividir una barra.....	99
Figura 5.18	Universo estructural del lado opuesto del nuevo nudo al dividir una barra.....	99
Figura 5.19	Topología óptima al dividir una barra.....	99
Figura 5.20	Topologías al dividir dos barras que se cruzan.....	100
Figura 5.21	Universo estructural del nuevo nudo con un cruce.....	100
Figura 5.22	Universo estructural del nuevo nudo con un cruce y un grado de hiperestaticidad máximo de 3.....	101
Figura 5.23	Topología en la que al dividir dos barras que se cruzan, alguna de las barras en las que se dividen las originales no forma parte de la topología óptima.....	101
Figura 5.24	Ejemplos donde la mejor barra a dividir no es la más larga.....	102
Figura 5.25	Mejor barra o cruce de barras a dividir para obtener la nueva topología óptima. Cuatro primeras iteraciones del voladizo de Michell.....	103
Figura 5.26	Diagrama de flujo para el proceso de crecimiento de la topología.....	104
Figura 5.27	Diagrama de flujo para la comprobación de la topología óptima en la nueva geometría.....	105
Figura 5.28	Cambio de la topología óptima al optimizar la geometría.....	106
Figura 5.29	Evolución del diseño óptimo en las tres primeras iteraciones. Viga con apoyos fijos.....	107
Figura 5.30	Evolución del diseño óptimo en las tres primeras iteraciones. Viga con disco circular rígido.....	109
Figura 6.1	Clase base <i>Analisis</i> y sus clases derivadas.....	115
Figura 6.2	Clase base <i>Optimizacion</i> y sus clases derivadas.....	117
Figura 6.3	Clase base <i>Linea</i> y sus clases derivadas.....	121
Figura 6.4	Clase base <i>Area</i> y sus clases derivadas.....	122
Figura 6.5	Clase base <i>Volumen</i> y sus clases derivadas.....	123
Figura 6.6	Clase base <i>Elemento</i> y sus clases derivadas.....	125
Figura 6.7	Clase base <i>Numeración</i> y sus clases derivadas.....	130
Figura 6.8	Clase base <i>Simplex</i> y sus clases derivadas.....	131
Figura 6.9	Pantalla principal de TTO sin datos.....	133
Figura 6.10	Pantalla principal de TTO con un problema de barras articuladas.....	134

Figura 6.11	Pantalla principal de TTO con un problema de barras de nudos rígidos.....	134
Figura 6.12	Pantalla principal de TTO con un problema de elasticidad plana	135
Figura 6.13	Pantalla principal de TTO con un problema de campos	135
Figura 6.14	Pantalla principal de TTO con un problema de optimización de barras articuladas	136
Figura 6.15	Mallado generado para un diente de engranaje.....	136
Figura 6.16	Matriz de rigidez sin y con reenumeración de los nodos del problema de la Fig. 6.15.....	137
Figura 6.17	Resultados del análisis del problema de la Fig. 6.10	138
Figura 6.18	Resultados del análisis del problema de la Fig. 6.11	138
Figura 6.19	Resultados del análisis del problema de la Fig. 6.12	139
Figura 6.20	Resultados del análisis del problema de la Fig. 6.13	139
Figura 6.21	Resultados de la optimización del problema de la Fig. 6.14 sin optimización de geometría.....	140
Figura 6.22	Resultados de la optimización del problema de la Fig. 6.14 con optimización de geometría.....	141
Figura 7.1	(a) Voladizo de Michell. (b) Solución analítica. (c) Valores óptimos de la masa (Rozvany, 1998)	144
Figura 7.2	Voladizo de Michell ($\bar{L} = 0,5$). Diseños óptimos obtenidos mediante todos los métodos del universo estructural y optimización final de geometría	146
Figura 7.3	Voladizo de Michell ($\bar{L} = 0,5$). Diseño óptimo obtenido mediante el método de crecimiento propuesto	146
Figura 7.4	Voladizo de Michell ($\bar{L} = 1,82196$). Diseños óptimos obtenidos mediante métodos del universo estructural.....	147
Figura 7.5	Voladizo de Michell ($\bar{L} = 1,82196$). Diseños óptimos obtenidos mediante el método de crecimiento propuesto	149
Figura 7.6	Voladizo de Michell ($\bar{L} = 1,82196$). Métodos del universo estructural (sin optimización de geometría) y de crecimiento propuesto. (a) Evolución error/nº nudos; (b) Evolución error/tiempo	153
Figura 7.7	Voladizo de Michell ($\bar{L} = 1,82196$). Métodos del universo estructural (con optimización de geometría) y de crecimiento propuesto. (a) Evolución error/nº nudos; (b) Evolución error/tiempo	153
Figura 7.8	Voladizo de Michell ($\bar{L} = 3,35889$). Diseños óptimos obtenidos mediante el método de la <i>compliance</i>	155
Figura 7.9	Voladizo de Michell ($\bar{L} = 3,35889$). Diseños óptimos obtenidos mediante el método de crecimiento propuesto, añadiendo los nudos de uno en uno.....	156
Figura 7.10	Voladizo de Michell ($\bar{L} = 3,35889$). Diseños óptimos obtenidos mediante el método de crecimiento propuesto, añadiendo los nudos de dos en dos.....	157
Figura 7.11	Voladizo de Michell ($\bar{L} = 3,35889$). Método de la <i>compliance</i> (sin y con optimización de geometría) y de crecimiento propuesto	

	(añadiendo los nudos de uno en uno y de dos en dos).	
	(a) Evolución error/nº nudos; (b) Evolución error/tiempo	159
Figura 7.12	(a) Viga MBB. (b) Solución analítica. (c) Valores óptimos de la masa (Rozvany, 1998).....	160
Figura 7.13	Viga MBB ($\bar{L} = 2,40196$). Diseños óptimos obtenidos mediante el método de la <i>compliance</i>	161
Figura 7.14	Viga MBB ($\bar{L} = 2,40196$). Diseños óptimos obtenidos mediante el método de crecimiento propuesto	164
Figura 7.15	Viga MBB ($\bar{L} = 2,40196$). Método de la <i>compliance</i> (sin y con optimización de geometría) y de crecimiento propuesto. (a) Evolución error/nº nudos; (b) Evolución error/tiempo	166
Figura 7.16	Voladizo de Michell ($\bar{L} = 5,49846$). Diseños óptimos obtenidos mediante el método de la <i>compliance</i>	166
Figura 7.17	Voladizo de Michell ($\bar{L} = 5,49846$). Diseños óptimos obtenidos mediante el método de crecimiento propuesto	168
Figura 7.18	Viga MBB ($\bar{L} = 5,49846$). Método de la <i>compliance</i> (sin y con optimización de geometría) y de crecimiento propuesto. (a) Evolución error/nº nudos; (b) Evolución error/tiempo	170
Figura 7.19	(a) Viga con disco circular rígido. (b) Solución analítica (Rozvany, 1998).....	171
Figura 7.20	Viga con disco circular rígido. Superficies a mapear para crear el universo estructural.....	172
Figura 7.21	Viga con disco circular rígido. Diseños obtenidos mediante el método de la <i>compliance</i>	172
Figura 7.22	Viga con disco circular rígido. Diseños obtenidos mediante el método de crecimiento propuesto	174
Figura 7.23	Viga con disco circular rígido. Método de la <i>compliance</i> (sin y con optimización de geometría) y de crecimiento propuesto. (a) Evolución error/nº nudos; (b) Evolución error/tiempo	176
Figura 7.24	Voladizo de Michell con carga variable.....	177
Figura 7.25	Voladizo de Michell ($\bar{L} = 0,5$). Diseño óptimo: $\alpha = 0^\circ$ (sin pandeo).....	178
Figura 7.26	Voladizo de Michell ($\bar{L} = 0,5$). Diseño óptimo: $\alpha = 15^\circ$ (sin pandeo).....	178
Figura 7.27	Voladizo de Michell ($\bar{L} = 0,5$). Diseño óptimo: $\alpha = 30^\circ$ (sin pandeo).....	178
Figura 7.28	Voladizo de Michell ($\bar{L} = 0,5$). Diseño óptimo: $\alpha = 45^\circ$ (sin pandeo).....	179
Figura 7.29	Voladizo de Michell ($\bar{L} = 0,5$). Diseño óptimo: $\alpha = 60^\circ$ (sin pandeo).....	179
Figura 7.30	Voladizo de Michell ($\bar{L} = 0,5$). Diseño óptimo: $\alpha = 75^\circ$ (sin pandeo).....	179
Figura 7.31	Voladizo de Michell ($\bar{L} = 0,5$). Diseño óptimo: $\alpha = 90^\circ$ (sin pandeo).....	179
Figura 7.32	Voladizo de Michell ($\bar{L} = 0,5$). Evolución masa adimensional/ángulo (sin pandeo).....	180
Figura 7.33	Voladizo de Michell ($\bar{L} = 1,82196$). Diseño óptimo: $\alpha = 0^\circ$ (sin pandeo)	180
Figura 7.34	Voladizo de Michell ($\bar{L} = 1,82196$). Diseño óptimo: $\alpha = 15^\circ$ (sin pandeo).....	180

Figura 7.35	Voladizo de Michell ($\bar{L} = 1,82196$). Diseño óptimo: $\alpha = 30^\circ$ (sin pandeo)	181
Figura 7.36	Voladizo de Michell ($\bar{L} = 1,82196$). Diseño óptimo: $\alpha = 45^\circ$ (sin pandeo)	182
Figura 7.37	Voladizo de Michell ($\bar{L} = 1,82196$). Diseño óptimo: $\alpha = 60^\circ$ (sin pandeo)	183
Figura 7.38	Voladizo de Michell ($\bar{L} = 1,82196$). Diseño óptimo: $\alpha = 75^\circ$ (sin pandeo)	184
Figura 7.39	Voladizo de Michell ($\bar{L} = 1,82196$). Diseño óptimo: $\alpha = 90^\circ$ (sin pandeo)	185
Figura 7.40	Voladizo de Michell ($\bar{L} = 1,82196$). Evolución masa adimensional/ángulo (sin pandeo).....	186
Figura 7.41	Voladizo de Michell ($\bar{L} = 0,5$). Diseño óptimo: $\alpha = 0^\circ$ (con pandeo)	186
Figura 7.42	Voladizo de Michell ($\bar{L} = 0,5$). Diseño óptimo: $\alpha = 15^\circ$ (con pandeo).....	186
Figura 7.43	Voladizo de Michell ($\bar{L} = 0,5$). Diseño óptimo: $\alpha = 30^\circ$ (con pandeo)	187
Figura 7.44	Voladizo de Michell ($\bar{L} = 0,5$). Diseño óptimo: $\alpha = 45^\circ$ (con pandeo)	187
Figura 7.45	Voladizo de Michell ($\bar{L} = 0,5$). Diseño óptimo: $\alpha = 60^\circ$ (con pandeo)	187
Figura 7.46	Voladizo de Michell ($\bar{L} = 0,5$). Diseño óptimo: $\alpha = 75^\circ$ (con pandeo)	188
Figura 7.47	Voladizo de Michell ($\bar{L} = 0,5$). Diseño óptimo: $\alpha = 90^\circ$ (con pandeo)	188
Figura 7.48	Voladizo de Michell ($\bar{L} = 0,5$). Evolución masa adimensional/ángulo (con pandeo)	188
Figura 7.49	Voladizo de Michell ($\bar{L} = 1,82196$). Diseño óptimo: $\alpha = 0^\circ$ (con pandeo)	189
Figura 7.50	Voladizo de Michell ($\bar{L} = 1,82196$). Diseño óptimo: $\alpha = 15^\circ$ (con pandeo).....	189
Figura 7.51	Voladizo de Michell ($\bar{L} = 1,82196$). Diseño óptimo: $\alpha = 30^\circ$ (con pandeo)	189
Figura 7.52	Voladizo de Michell ($\bar{L} = 1,82196$). Diseño óptimo: $\alpha = 45^\circ$ (con pandeo)	190
Figura 7.53	Voladizo de Michell ($\bar{L} = 1,82196$). Diseño óptimo: $\alpha = 60^\circ$ (con pandeo)	190
Figura 7.54	Voladizo de Michell ($\bar{L} = 1,82196$). Diseño óptimo: $\alpha = 75^\circ$ (con pandeo)	191
Figura 7.55	Voladizo de Michell ($\bar{L} = 1,82196$). Diseño óptimo: $\alpha = 90^\circ$ (con pandeo)	191
Figura 7.56	Voladizo de Michell ($\bar{L} = 1,82196$). Evolución masa adimensional/ángulo (con pandeo)	192
Figura 7.57	Ménsula corta.....	194
Figura 7.58	Ménsula corta. Diseño óptimo obtenido por Adebar	195
Figura 7.59	Ménsula corta. Diseños óptimos obtenidos mediante el método de crecimiento propuesto	195
Figura 7.60	Mecanismo. (a) Diseño a flexibilidad; (b) Diseño a rigidez	197
Figura 7.61	Mecanismo extractor de disquetes	199
Figura 7.62	Mecanismo extractor de disquetes. Estados de cargas.....	200
Figura 7.63	Mecanismo extractor de disquetes. Diseño óptimo obtenido por Frecker	201
Figura 7.64	Mecanismo extractor de disquetes. Diseño óptimo obtenido mediante el método de crecimiento propuesto	202

Lista de símbolos

Escalares

Caracteres latinos (minúsculas)

c	cadena de barras
c	coste de una barra
d	número de dimensiones del problema
e	cota superior
f	función de transformación
f, \tilde{f}	función objetivo
f_p	función objetivo penalizada
g_k	restricción de diseño de desigualdad k
h_j	restricción de diseño de igualdad j
h	altura del dominio de existencia de una estructura
m	pendiente de una barra
m	número de barras
m	número total de restricciones
m_d	número de restricciones de desigualdad
m_i	número de restricciones de igualdad
n	número de nudos
n	número de grados de libertad de una estructura
n	número de variables
n_k	número de estados de cargas
r	peso de la penalización de f_p sobre las funciones de las restricciones G_i
r	radio
s	coeficiente de seguridad
s	número de grados de libertad con desplazamientos impedidos
s	parámetro topológico
u_j	multiplicador de Lagrange de la restricción g_j

u, v, w	desplazamientos en las direcciones de los ejes globales
v	orden de vecindad en la generación del universo estructural
x, y, z	ejes de referencia en coordenadas locales
x_i	variable de diseño i
x_i^I	límite inferior de la variable de diseño x_i
x_i^S	límite superior de la variable de diseño x_i

Caracteres latinos (mayúsculas)

A	área de la sección transversal de una barra
A_E	área correspondiente a la carga crítica de Euler
C, \tilde{C}, \hat{C}	<i>compliance</i>
C	coste de la estructura
C	conjunto de cadenas
C^0	conjunto de cadenas activas
E	módulo de Young
G_i	función de la restricción g_i para la penalización de f_p
G	universo estructural
I	momento de inercia de la sección de una barra
L	función lagrangiana
L	longitud de una barra
L	longitud del dominio de existencia de una estructura
\bar{L}	longitud adimensional del dominio de existencia de una estructura
M	masa total de una estructura
\bar{M}	masa adimensional total de una estructura
M_t	momento torsor
P	carga puntual
P_E	carga crítica de Euler
V	volumen de una estructura
X, Y, Z	ejes de referencia en coordenadas globales

Caracteres griegos (minúsculas)

α_k	longitud de movimiento en la dirección de búsqueda \mathbf{d}_k para la iteración k
β	inercia / área ² de una barra
α, β, γ	ángulos con respecto a los ejes globales
δ	tolerancia para las restricciones
ε	constante positiva de valor muy pequeño
ε	error relativo
ε	conjunto de conexiones
λ	esbeltez mecánica de una barra

μ, μ'	variables asociadas a los multiplicadores de Lagrange
ρ_j	densidad de la barra j
σ_T	límite elástico a tracción
σ_C	límite elástico a compresión
σ_P	límite de proporcionalidad
\mathcal{G}	conjunto de posiciones nodales
ξ, η, ζ	ejes de referencia en coordenadas normalizadas

Caracteres griegos (mayúsculas)

Γ	conjunto de nudos intermedios de una cadena
Ω, Ω'	dominios de existencia de una estructura

Vectores y matrices

Caracteres latinos (minúsculas)

a	vector de áreas de las barras
b_i	vector de cosenos directores de la barra i
b, \tilde{b}	vectores de coeficientes independientes
c	vector de costes
d_k	dirección de búsqueda para la iteración k
f	vector de cargas nodales de una estructura
k	matriz de rigidez por unidad de área de una barra
q	vector de esfuerzos axiales
u	vector de desplazamientos nodales de una estructura
u, u'	vectores de multiplicadores de Langrange
v	variables asociadas a los multiplicadores de Lagrange
v	vector para modificar el hessiano
x, \tilde{x}	vectores de variables de diseño

Caracteres latinos (mayúsculas)

A, \tilde{A}	matrices de coeficientes
I	matriz identidad
J	matriz jacobiana de una transformación geométrica
K	matriz de rigidez de una estructura
N	vector función de forma para la transformación geométrica
R	matriz de cosenos directores de una estructura

Caracteres griegos (minúsculas)

λ	vector de multiplicadores de Lagrange
-----------	---------------------------------------

Símbolos especiales

∇	operador nabra (gradiente)
∇^2	operador laplaciano (hessiano)
Δ	incremento

Introducción y objetivos

1.1 INTRODUCCIÓN

La optimización estructural ha progresado de forma importante en los últimos años y ahora es reconocida como una herramienta de diseño práctica. Hay tres tipos de problemas de optimización estructural: de propiedades, de geometría y de topología. Un considerable número de trabajos han sido realizados en las tres últimas décadas, la mayoría de ellos están relacionados con la optimización de propiedades y de geometría pero pocas contribuciones han sido dedicadas al diseño de topología óptima debido a su complejidad, a pesar de que es reconocido que la optimización de topología y geometría puede mejorar el diseño en gran medida.

1.2 DISEÑO ÓPTIMO DE TOPOLOGÍA Y GEOMETRÍA DE ESTRUCTURAS ARTICULADAS

Hay tres formas de abordar la optimización de topología de estructuras de barras articuladas.

1.2.1 Métodos del universo estructural (*ground structure*)

La mayoría de los autores recurren a métodos basados en el universo estructural (*ground structure*) que consiste en crear una rejilla de nudos en el dominio de existencia de la estructura y colocar barras en todas o en algunas de las posibles conexiones entre los nudos. El algoritmo de optimización deberá determinar cuales son las barras que constituyen la topología óptima.

Para que los resultados sean buenos la rejilla debe ser suficientemente tupida, lo que implica que el número de barras ($\text{nudos}(\text{nudos} - 1) / 2$) debe ser muy alto, con lo que el coste computacional también lo es.

1.2.2 Métodos de crecimiento

Los métodos llamados de crecimiento parten de una estructura muy sencilla y van modificando la topología añadiendo en cada iteración nudos y barras de forma que la función objetivo disminuya.

Este método es usado por Rule (1994), McKeown (1998), y Bojczuk y Mróz (1998 y 1999).

1.2.3 Métodos de homogeneización

La optimización de estructuras articuladas es posible usando el método de homogeneización, para ello se resuelve como si fuera un problema de elasticidad y la solución se aproxima a una estructura articulada (normalmente se parecen).

Este método es usado por Díaz y Belding (1993).

1.3 OBJETIVOS DE LA TESIS

Aunque la optimización de topología de estructuras articuladas mediante métodos basados en el universo estructural ha experimentado un notable avance en los últimos años, no se ha producido el mismo avance en los métodos de crecimiento para la optimización de topología y geometría de estructuras articuladas.

El objetivo de la tesis es desarrollar un nuevo método de crecimiento para el diseño óptimo simultáneo de topología y geometría de estructuras articuladas.

1.4 ORGANIZACIÓN DE LA TESIS

La tesis se ha estructurado en ocho capítulos, incluyendo este primer capítulo introductorio.

En el capítulo 1 se ha realizado una introducción general al problema de diseño óptimo, y en particular al problema de diseño óptimo de topología y geometría de estructuras articuladas, y se han planteado los objetivos y descrito la organización de la tesis.

El capítulo 2 plantea el problema del diseño óptimo de estructuras desde un punto de vista general y recoge algunos de los métodos actuales de resolución del problema de optimización.

El capítulo 3 estudia el diseño óptimo de topología de estructuras articuladas mediante métodos basados en el universo estructural.

El capítulo 4 trata sobre el diseño óptimo de topología y geometría de estructuras articuladas mediante métodos de crecimiento.

El capítulo 5 presenta el método de crecimiento propuesto para la optimización de estructuras articuladas.

El capítulo 6 hace una breve descripción de la aplicación informática desarrollada para la realización de esta tesis.

El capítulo 7 muestra ejemplos numéricos para demostrar la validez del método de crecimiento propuesto y compararlo con los métodos del universo estructural, y muestra diferentes aplicaciones de la optimización de topología de estructuras articuladas.

Finalmente, el capítulo 8 recoge las conclusiones de la tesis y algunas sugerencias para trabajos futuros.

Formulación y resolución del problema de diseño óptimo de estructuras

2.1 INTRODUCCIÓN

Antes de la aplicación de las modernas técnicas de optimización al diseño, el proceso de diseño óptimo requería fundamentalmente una gran experiencia por parte del diseñador, el cual debía utilizarla en casi todas las etapas del proceso de diseño. El método que se empleaba era el siguiente: el ingeniero definía un diseño inicial cuyo comportamiento era analizado numéricamente; de los resultados de dicho análisis se podían deducir (por experiencia o intuición) los cambios a realizar para mejorar dicho diseño. Se finalizaba el proceso cuando se consideraba que el diseño era lo suficientemente bueno.



Figura 2.1 Diseño por prueba y error

Tal método, denominado de prueba y error (Fig. 2.1), presentaba el inconveniente de que las modificaciones dependían totalmente de la experiencia del diseñador, lográndose soluciones buenas, pero no las mejores, y además con un alto precio en tiempo por parte del diseñador.

Para conseguir una resolución del problema más económica en tiempo empleado, pudiéndose practicar un mayor número de soluciones de forma que el resultado se acerque al óptimo, se intentó formular el problema de forma que fuese apto para resolverse automáticamente. Así, se reformó el proceso de diseño hasta el esquema mostrado en la Fig. 2.2.

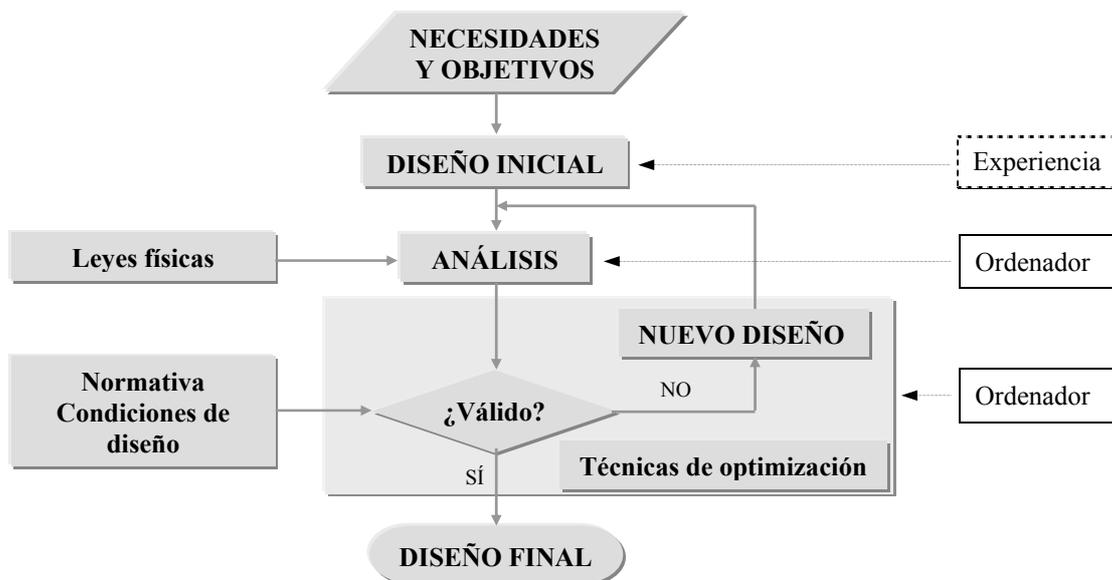


Figura 2.2 Diseño optimizado

Un paso previo a la tarea de diseño es el de definir una idealización del objeto a diseñar, a fin de obtener un modelo que incluya un número finito de los aspectos más importantes del diseño y sea más sencillo de manejar que el objeto real. Tal proceso de idealización debe realizarlo el diseñador, utilizando para ello la experiencia adquirida en diseños precedentes. Sin embargo, lejos de encontrarse desasistido en tal labor, el diseñador cuenta con una serie de pautas establecidas y generalizadas que, como se verá más adelante, no puede rechazar arbitrariamente.

2.2 FORMULACIÓN DEL PROBLEMA DE DISEÑO ÓPTIMO DE ESTRUCTURAS

El problema de diseño óptimo se entiende actualmente como aquél que se plantea determinar el valor de una serie de variables de manera que se minimice el valor de una función objetivo a la vez que se cumplen una serie de restricciones impuestas.

Con este planteamiento quedan expuestas las tres características fundamentales del problema: (1) las variables de diseño, (2) la función objetivo, y (3) las restricciones.

A continuación se van a definir y describir cada uno de estos aspectos del proceso de diseño.

2.2.1 Variables de diseño

Las magnitudes que intervienen en un diseño se pueden dividir en:

- Parámetros del problema.
- Variables del problema.

Los parámetros del problema representan magnitudes cuyo valor es fijo para todo el diseño y son consecuencia, en general, de una serie de condiciones impuestas externamente al problema.

Las variables, por contra, hacen referencia a magnitudes cuyos valores cambian conforme evoluciona el diseño.

En el caso más general, se distinguen cuatro tipos de variables, en función de la complejidad que plantea la optimización:

- Propiedades de la sección transversal de la pieza (áreas, espesores, momentos de inercia).
- Geometría de la estructura (dimensiones, contornos).
- Topología de la estructura.
- Propiedades del material que constituye la pieza.

El tipo de optimización a realizar dependerá de cuales de estas variables se consideren.

Actualmente no existen técnicas de optimización que consideren eficientemente los cuatro tipos de variables. Por ello, lo habitual es considerar como parámetros al material e incluso a la topología.

Según su campo de existencia, las variables se pueden clasificar en:

- Continuas.
- Discretas.

Los métodos de optimización disponibles actualmente trabajan, en su mayoría, sólo con variables continuas. Por lo que el uso de variables discretas está limitado a algunos métodos aún no muy establecidos.

2.2.2 Función objetivo

La función objetivo suele ser una función escalar de las variables que intervienen en el diseño, y la condición habitual que se le exige a dicha función es que para la solución óptima tome un valor mínimo.

Hasta la fecha, la función objetivo que se ha utilizado con más frecuencia es la masa de la pieza, debido a que las primeras optimizaciones estructurales se investigaron en el campo de la aeronáutica, en el cual la masa es un condicionamiento esencial.

El planteamiento tradicional es, pues, expresar la masa de la pieza en función de las variables de diseño, y tratar de obtener el mínimo valor de dicha masa.

Esta práctica se sigue realizando actualmente con una gran eficacia, puesto que la masa de la estructura está directamente relacionada con el coste del material de la misma. Cuanto menos material se utilice más barata será la pieza en general.

Sin embargo, en la evaluación del coste total entran en juego aspectos tales como costes de fabricación, mano de obra, etc. Estos aspectos pueden llegar a ser decisivos a la hora de evaluar la rentabilidad de uno u otro diseño, y es por ello que conviene tenerlos en cuenta al plantear la función objetivo.

El coste de la pieza es una de las funciones objetivo más usuales, que se define por medio de unos coeficientes de ponderación que suelen ser costes unitarios de materiales o procesos de fabricación.

Asimismo, pueden emplearse como función objetivo otras cantidades, tales como la fiabilidad, el desplazamiento de algún punto, la rigidez, la frecuencia fundamental, etc.

2.2.3 Restricciones

Las restricciones son condiciones que debe cumplir el diseño para que pueda ser considerado válido. Estas condiciones se plantean como funciones de las variables, a las que se les exige tener algún valor o mantenerse dentro de unos límites.

Una primera clasificación de las restricciones las divide en:

- Explícitas.
- Implícitas.

Las primeras actúan directamente imponiendo condiciones a una variable (o un grupo de ellas), mientras que las segundas imponen condiciones sobre magnitudes que dependen a su vez de las variables. Es importante la distinción entre ambas porque las explícitas tienen un tratamiento más sencillo en la mayoría de los métodos de diseño.

Otra clasificación agrupa a las restricciones en:

- Restricciones de igualdad.
- Restricciones de desigualdad.

Las restricciones de igualdad suelen estar asociadas a las relaciones que fijan el comportamiento de la estructura, tales como condiciones de equilibrio, compatibilidad, ley de comportamiento del material, etc. Otras relaciones de igualdad están asociadas a las relaciones entre las variables de diseño del problema (condiciones de simetría, tangencia, curvaturas, etc.).

Las restricciones de desigualdad suelen estar asociadas a limitaciones impuestas a la respuesta del elemento estructural, tales como tensiones máximas, deformaciones máximas, frecuencias de vibración, etc. Otro tipo de restricciones de desigualdad son las que delimitan el rango de posibles valores de las variables.

2.3 TIPOS DE OPTIMIZACIÓN ESTRUCTURAL

Los diferentes tipos de optimización estructural dependen de cuáles sean las variables de diseño utilizadas. De los cuatro tipos de variables comentados anteriormente, el material de la estructura suele plantearse como parámetros de la estructura, siendo fijado por el diseñador.

Quedan pues, como variables de diseño más habituales, las propiedades de la sección, la geometría y la topología de la estructura. Estos tres tipos de variables son el origen de tres tipos de optimización estructural distintos, cuyas características se pasan a exponer.

2.3.1 Optimización de propiedades de la sección transversal

En este tipo de optimización estructural, las variables de diseño están asociadas a propiedades geométricas de la sección transversal de los elementos que integran la estructura, tales como áreas de barras, espesores de placas, etc. Éste es el tipo más sencillo de optimización estructural y sus fundamentos están bastante bien establecidos.

Para la selección de las variables de diseño hay que distinguir entre sistemas discretos y sistemas continuos. En el caso de sistemas continuos, las variables de diseño suelen ser los espesores de las piezas. Ahora bien, dado que el análisis se suele realizar por elementos finitos, hay que asignar una variable a cada uno de los elementos. Esto obliga a adoptar algún tipo de relación que reduzca el número de variables de diseño.

En los sistemas discretos las variables de diseño suelen ser varias propiedades de la sección transversal (áreas, momentos de inercia, módulos de torsión, etc.) por cada uno de los elementos de la estructura. Esto produce dos problemas: un elevado número de variables, y unos resultados que no se corresponden con los perfiles normalizados.

Con el fin de resolver estos problemas, se recurre a diferentes técnicas. Una de estas técnicas consiste en relacionar los parámetros del modelo de análisis (\mathbf{X}) con un número más reducido de variables de diseño (\mathbf{D}). Una forma de hacerlo es a través de la relación lineal:

$$\mathbf{D} = \mathbf{D}_0 + \mathbf{TX}$$

siendo \mathbf{D}_0 un vector de constantes y \mathbf{T} una matriz.

En el caso de los sistemas discretos, además de utilizarse el agrupamiento anterior, suele emplearse otro tipo de agrupamiento, que consiste en asignar una variable de diseño a una de las propiedades de la sección transversal (el área generalmente), considerando al resto como variables dependientes (momentos de inercia, módulo de torsión pura, etc.), que se obtienen a partir de ajustes utilizando parámetros obtenidos a partir de los datos de los perfiles normalizados.

2.3.2 Optimización de geometría

En la optimización de geometría, además de las variables empleadas en la optimización de propiedades, se utilizan variables asociadas a la forma geometría de la estructura. Estas variables controlan la geometría de la estructura y requieren un modelo de análisis que cambie durante el proceso de optimización.

El interés creciente por la optimización de geometría se debe a dos causas: en primer lugar, la optimización de geometría proporciona diseños mucho mejores que la optimización de propiedades; en segundo lugar, la mayor disponibilidad actual de herramientas de análisis y optimización permite resolver los problemas que plantea este tipo de optimización.

2.3.3 Optimización de topología

En los últimos años ha habido un notable progreso en la optimización de topología de estructuras tanto discretas como continuas. Las técnicas de optimización de topología actuales están consiguiendo un nivel de aplicabilidad práctico en especial en problemas de la vida real. La optimización de topología ha comenzado un proceso que puede cambiar el diseño estructural clásico en el futuro.

En la optimización de topología, además de las variables empleadas en la optimización de propiedades, se utilizan variables asociadas a la topología de la estructura, esto es, al número de elementos y a su interconexión.

El interés creciente por la optimización de topología se debe a que se evita la necesidad de que el diseñador tenga que definir la topología, lo que supone que debe tener una cierta experiencia para saber cuál es la topología óptima.

2.4 FORMULACIÓN MATEMÁTICA DEL PROBLEMA DE OPTIMIZACIÓN

El objetivo del diseño óptimo de estructuras es obtener un diseño, es decir, un conjunto de valores de las variables de diseño, que haga mínima una función objetivo, por ejemplo el coste, y cumpla una serie de restricciones que dependen de las mismas variables.

En términos matemáticos, el problema se puede formular como:

Encontrar el vector de variables de diseño \mathbf{x} que

$$\begin{aligned}
 &\text{minimice :} && f(\mathbf{x}) \\
 &\text{sujeto a :} && h_j(\mathbf{x}) = 0 && j = 1, 2, \dots, m_i \\
 &&& g_k(\mathbf{x}) \geq 0 && k = 1, 2, \dots, m_d \\
 &&& x_i^l \leq x_i \leq x_i^s && i = 1, 2, \dots, n
 \end{aligned} \tag{2.1}$$

siendo:

\mathbf{x}	Vector n -dimensional de variables de diseño.
$f(\mathbf{x})$	Función objetivo, representa el criterio óptimo.
$h_j(\mathbf{x})$	Restricción de diseño de igualdad j .
$g_k(\mathbf{x})$	Restricción de diseño de desigualdad k .
m_i	Número de restricciones de igualdad.
m_d	Número de restricciones de desigualdad.
n	Número de variables.
x_i	Variable de diseño i .
x_i^I	Límite inferior de la variable de diseño i .
x_i^S	Límite superior de la variable de diseño i .

Al espacio n -dimensional definido por el vector de variables \mathbf{x} se le denomina espacio de diseño. Las restricciones definen hipersuperficies que acotan un entorno en el espacio de diseño. Un conjunto de variables de diseño define un punto en el espacio de diseño. Si un punto del espacio es tal que se cumplen todas las restricciones, ese punto es un diseño válido; si, por el contrario, viola alguna restricción, el punto del espacio de diseño corresponde a un diseño no válido.

2.5 RESOLUCIÓN DEL PROBLEMA DE OPTIMIZACIÓN

2.5.1 Introducción

En la actualidad existen tres aproximaciones para la resolución del problema de optimización formulado en 2.4:

- Criterios de optimalidad.
- Programación matemática.
- Algoritmos genéticos.

La primera aproximación es la formada por los métodos llamados “indirectos”. Esto es debido a que lo que se busca en dichos métodos es obtener un diseño que satisfaga un criterio especificado, que a su vez, implique el cumplimiento del objetivo buscado. El criterio puede ser intuitivo o deducido matemáticamente a partir de las características particulares del problema a tratar.

La segunda aproximación es más general. En lugar de basarse en aspectos “físicos” del problema, trata de llegar a una formulación matemática del mismo, intentando que dicha formulación matemática sea general y susceptible de implementación en computadores.

La coexistencia de ambas tendencias se debe a que los resultados obtenidos con los criterios de optimalidad dependen del problema y no pueden ser generalizados, por lo cual se requiere un método para cada tipo de problema y un mínimo de experiencia

para elegir el método apropiado. Por el contrario, los métodos desarrollados hasta la fecha a partir de la formulación matemática son capaces de resolver problemas generales, pero resultan menos eficientes que los basados en criterios de optimalidad para los casos en que estos últimos tienen solución.

2.5.2 Criterios de Optimalidad

Los criterios de optimalidad son formas de solucionar problemas concretos de optimización, dirigiendo la solución por medio de la aplicación de criterios que se sabe (o se cree) que son apropiados para el problema tratado. Algunos de los criterios de optimalidad tienen un claro sentido físico; tal es el caso del diseño FSD (*Fully Stressed Design*), que es aquél en el cual cada elemento de la estructura soporta una tensión límite bajo, al menos, uno de los estados de cargas especificados. El FSD es uno de los conceptos tradicionales en el diseño óptimo de estructuras, y mantiene su interés por su carácter intuitivo y por ser un procedimiento rápido y fácil de implementar en comparación con los basados en programación matemática.

2.5.2.1 Características del método FSD

El FSD es un diseño óptimo a resistencia, con variables de propiedades de los elementos. Inicialmente, el método no considera otro tipo de restricciones, si bien se puede generalizar para tratar restricciones de desplazamiento. La característica más destacada del FSD es la ausencia de función objetivo; por tanto no existe una cantidad a minimizar, y no se puede asegurar que un algoritmo para calcular FSD converja al diseño de mínima masa, ya que no se puede explicitar dicha condición.

Para una estructura sometida a un único estado de cargas el FSD ofrece la solución de mínima relación masa/resistencia (Michell, 1904). Sin embargo, esta condición no es generalizable para más estados de cargas. Esto se debe a que un diseño FSD no es único para diferentes estados de cargas. Por tanto, un FSD puede conducir a una solución óptima en ciertos casos pero también puede llevar a una solución no óptima o incluso puede no encontrarse solución.

En general, la experiencia ha demostrado que salvo en los casos de estructuras con comportamientos extraños, el FSD da un diseño a tensión que es el óptimo o muy cercano a él. La posibilidad de que la solución obtenida no sea la óptima queda compensada por la facilidad de implementación del método y la rapidez con que converge.

Una referencia más detallada del método se puede encontrar en la bibliografía sobre el tema, especialmente en los trabajos de Gellatly y Berke (1973).

2.5.3 Programación matemática

La programación matemática obtiene la solución al problema de diseño aplicando métodos numéricos de minimización (o maximización) de funciones objetivo sujetas a restricciones.

En la mayoría de las aplicaciones de diseño de estructuras la función objetivo a minimizar es la masa de la misma y las restricciones que se le imponen suelen derivarse de limitaciones de tensión, de pandeo o de desplazamientos de puntos de la estructura.

Recordando lo expuesto en el epígrafe 2.4, el problema de optimización se planteaba de la forma:

Encontrar el vector de variables de diseño \mathbf{x} que

$$\begin{aligned} \text{minimice :} & \quad f(\mathbf{x}) \\ \text{sujeto a :} & \quad h_j(\mathbf{x}) = 0 \quad j = 1, 2, \dots, m_i \\ & \quad g_k(\mathbf{x}) \geq 0 \quad k = 1, 2, \dots, m_d \\ & \quad x_i^I \leq x_i \leq x_i^S \quad i = 1, 2, \dots, n \end{aligned} \quad (2.1)$$

con la notación utilizada en el citado epígrafe.

Según sean las variables \mathbf{x} de diseño, la función objetivo $f(\mathbf{x})$ y las restricciones de diseño se tienen diferentes problemas de programación matemática.

Atendiendo al tipo de variables se tiene la programación continua, cuando las variables pueden adoptar cualquier valor; programación entera, cuando las variables sólo pueden adoptar valores enteros, y programación discreta cuando las variables sólo pueden adoptar valores, enteros o no, de una gama predefinida.

Otras clasificaciones se realizan atendiendo a la linealidad o no linealidad de las funciones. Si la función objetivo y todas las restricciones son funciones lineales, el problema es de programación lineal. Si cualquiera de las funciones del problema es no lineal, el problema es de programación no lineal.

El problema planteado en (2.1) es un problema de programación convexa cuando $f(\mathbf{x})$ es una función convexa y cada una de las restricciones es una función cóncava. Este caso tiene gran interés, ya que el espacio de diseño es un conjunto convexo, y cualquier mínimo local en este espacio de diseño, es un mínimo global.

Ninguno de los métodos de programación matemática propuestos hasta la fecha es capaz de resolver eficientemente todos los problemas del tipo (2.1). La solución habitual es la de emplear varios métodos, que se adapten mejor a diferentes tipos de problemas, y elegir en cada caso el más apropiado.

Las características del problema de optimización de estructuras son las que condicionan el método numérico más apropiado para obtener la solución. Estas características son:

- El problema es n -dimensional (pudiendo ser n muy grande).
- La función objetivo y las restricciones no son lineales, generalmente.

- Las variables no pueden tomar cualquier valor (existen límites de las variables).

La mayoría de los algoritmos de optimización resuelven el problema (2.1) de una forma iterativa según la expresión

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k \quad (2.2)$$

es decir, que el vector de variables de diseño (punto de diseño) para la iteración $k+1$ se obtiene a partir del punto de diseño anterior (\mathbf{x}_k). La actualización viene dada por el producto $\alpha_k \mathbf{d}_k$. En este producto, \mathbf{d}_k es una dirección de búsqueda en el espacio n -dimensional de las variables de diseño y α_k es la longitud del movimiento en esa dirección.

Así pues, la actualización consta de dos partes:

- Determinación de la dirección de búsqueda \mathbf{d}_k .
- Cálculo de la longitud α_k del movimiento.

Conceptualmente el proceso iterativo es el siguiente: si el diseño actual es un diseño válido, la dirección de búsqueda reduce el valor de la función objetivo sin violar las restricciones. Si el diseño actual no es válido (viola alguna restricción), la dirección de búsqueda se dirigirá hacia la zona válida, aún a costa de aumentar la función objetivo.

2.5.3.1 Búsqueda unidimensional

La obtención del valor de α_k es lo que se llama búsqueda unidimensional porque coincide con la búsqueda de la solución óptima para un problema con una sola variable. Se trata en esta búsqueda de obtener el valor de α_k que mejore el diseño lo máximo posible. Si el diseño actual está en la zona válida se busca el valor de α_k que reduzca $f(\mathbf{x})$ lo máximo posible sin violar ninguna de las restricciones. Si el diseño actual está fuera de la zona válida se busca el valor de α_k que elimine o reduzca al máximo la violación de restricciones.

Dado que la búsqueda unidimensional supone evaluar las funciones del problema, y en el caso del análisis por elementos finitos estas evaluaciones son costosas, se ha desarrollado una nueva tendencia en la búsqueda unidimensional que consiste en buscar una solución que simplemente mejore a la anterior sin exigir que sea la óptima.

2.5.3.2 Dirección de búsqueda

Encontrar una dirección de búsqueda \mathbf{d}_k consiste en elegir una dirección en el espacio de diseño de forma que, partiendo del punto de diseño actual, según esa dirección, se pueda hacer una búsqueda unidimensional que conduzca a un nuevo punto de diseño.

Los métodos que se emplean para determinar la dirección de búsqueda se diferencian entre sí en función del tipo de información que utilizan para lograr su propósito. Según ello se tienen:

a) Métodos de Orden Cero

Utilizan únicamente el valor de la función objetivo. El más sencillo de estos métodos se denomina *búsqueda aleatoria* y consiste en elegir una dirección aleatoria tal que el valor de la función sea menor que el valor en el punto de partida.

Un método más eficiente se puede derivar del anterior sin más que disponer algún criterio objetivo para determinar la dirección de búsqueda. La forma más elemental para hacer esto consiste en no cambiar de dirección mientras no nos conduzca a un punto peor que el último obtenido.

b) Métodos de Primer Orden

Utilizan información de las primeras derivadas. El más elemental es el *Steepest Descent* (Fig. 2.3), que se basa en que en cualquier punto la dirección de máximo descenso es la opuesta al gradiente de la función en dicho punto. Este método es bueno para funciones “redondas” pero es muy lento para funciones excéntricas.

Para funciones no redondas es apropiado un método del tipo del de direcciones conjugadas pero que aproveche la información de las derivadas. Tal es el método del *gradiente conjugado* de Fletcher y Reeves.

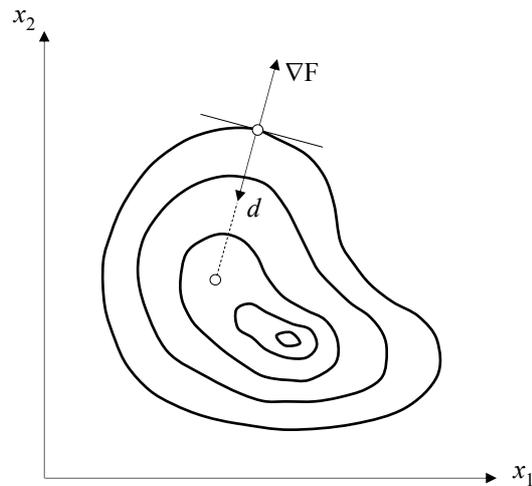


Figura 2.3 Método *Steepest Descent*

c) Métodos de Segundo Orden

Estos métodos utilizan información de las segundas derivadas. Destaca entre ellos el *método de Newton*, que consiste en hacer un desarrollo en serie de la función objetivo truncado en las segundas derivadas. Derivando dicha expresión e igualando a cero se obtiene el mínimo. Si la función objetivo es cuadrática, se obtiene el mínimo en la primera evaluación. En caso contrario se debe establecer un proceso iterativo.

2.5.3.3 Problemas con restricciones

En el caso más general el problema de optimización es un problema de minimizar una función objetivo sometido a restricciones.

El efecto de las restricciones es reducir el espacio de diseños válido pudiendo ocurrir que el diseño óptimo sin restricciones (Fig. 2.4) quede fuera de dicho espacio.

Un segundo efecto de las restricciones es que pueden hacer aparecer varios mínimos relativos (incluso en un problema caracterizado por una función objetivo unimodal) como se muestra en la Fig. 2.4.

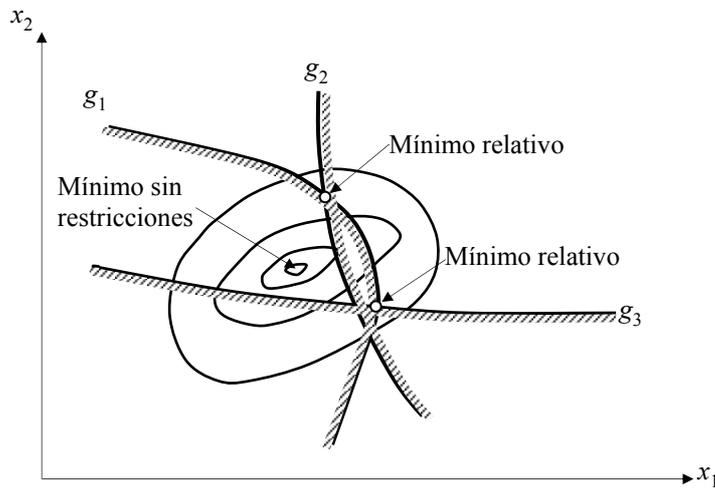


Figura 2.4 Efecto de las restricciones sobre el mínimo. Mínimos relativos

Para resolver el problema con restricciones hay dos estrategias diferentes:

- Utilizar un método de resolución sin restricciones, al que se añaden indirectamente las condiciones que implican las restricciones.
- Utilizar métodos que abordan el problema completo.

Los primeros se denominan genéricamente *métodos indirectos*. La forma más extendida de incluir las restricciones en un método que no las considera consiste en modificar el método para que tenga en cuenta las restricciones. Para ello se puede modificar la función objetivo de forma que incluya el efecto de las restricciones. Esta idea da lugar a los llamados *métodos de penalización*. Para estos métodos la expresión de la función objetivo queda convertida en

$$f_p(\mathbf{x}, r) = f(\mathbf{x}) + r \sum_{i=1}^m G_i(g_i) \quad (2.3)$$

siendo:

- $f_p(\mathbf{x}, r)$ Función objetivo penalizada.
- $f(\mathbf{x})$ Función objetivo original.
- r “Peso” de la penalización.
- $G_i(g_i)$ Función de las restricciones.
- m Número de restricciones.

La elección de las funciones G_i da lugar a diferentes métodos de penalización entre los que se pueden distinguir dos variantes:

- Métodos de penalización interior (o de barrera).
- Métodos de penalización exterior.

La diferencia entre ambos estriba en que en los primeros se llega a la solución a través de soluciones intermedias válidas mientras que en los segundos se llega desde soluciones intermedias no válidas. La ventaja de los primeros es que cualquier solución intermedia es buena (aunque no la mejor) mientras que en los segundos sólo es buena la última. A cambio, los primeros necesitan un punto de partida válido.

El segundo tipo de métodos son los denominados *métodos directos*. El más representativo de los métodos directos es el denominado *método de direcciones posibles*. La estrategia de este método es básicamente una modificación del *Steepest Descent*. Se trata de encontrar una dirección de descenso, para lo cual se utiliza la dirección del gradiente. Es decir, que la dirección \mathbf{d} debe cumplir

$$\mathbf{d}^T \nabla f \leq 0 \quad (2.4)$$

Pero, en este caso, además se debe cumplir que la dirección no lleve inmediatamente fuera de la región válida. Es decir

$$\mathbf{d}^T \nabla g_j = 0; \quad j = 1, 2, \dots, m \quad (2.5)$$

siendo m el número de restricciones activas.

Una vez determinada la dirección de búsqueda se debe aplicar alguno de los métodos de búsqueda unidimensional comentados para encontrar la longitud de avance. Debe hacerse notar que, según la longitud calculada, puede hacerse activa o violada alguna restricción, por lo que hay que añadir un control para que esto no ocurra.

2.5.4 Programación cuadrática sucesiva

2.5.4.1 Introducción

La *programación cuadrática sucesiva* (*Recursive Quadratic Programming* (RQP) en la bibliografía inglesa original) es un método directo de programación matemática con restricciones cuyas características generales han sido descritas en el apartado 2.5.3.3. En esencia, el método consiste en reducir el problema a un subproblema cuadrático para cada iteración del proceso de búsqueda del óptimo.

La idea básica de la RQP fue desarrollada por Wilson (1963). Han (1977) desarrolló según esta técnica un algoritmo que fue implementado por Powell (1978) con algunas modificaciones. Independientemente, Pshenichny (1970) publicó otro algoritmo RQP en el que incluyó una estrategia de conjunto activo de restricciones.

En la actualidad los métodos RQP tienen una gran aceptación debido a dos características de los mismos:

- Se ha probado que convergen globalmente (desde cualquier punto arbitrario de partida).

- Tienen relación lineal (o incluso superior) de convergencia.

A continuación se va a hacer una descripción general del método RQP que se ampliará y particularizará después con una descripción del algoritmo de Schittkowsky (1983) que es el empleado en el sistema DISSENY (Martí, 1985). En Belegundu (1982) se puede encontrar una descripción comparada de los algoritmos de Han (1977) y Psenichny (1970).

2.5.4.2 Descripción general

Se pretende resolver el problema general de optimización no lineal

$$\begin{aligned} \text{minimizar : } & f(\mathbf{x}) \\ \text{sujeto a : } & g_j(\mathbf{x}) \geq 0 \quad j = 1, 2, \dots, m \end{aligned} \quad (2.6)$$

Para ello se va a emplear un método iterativo que nos permita encontrar un nuevo punto de diseño \mathbf{x}_{k+1} que mejore el punto actual \mathbf{x}_k . La expresión que nos dará el nuevo punto de diseño es

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k \quad (2.7)$$

siendo \mathbf{d}_k un vector *dirección* y α_k una *longitud de paso* desde el punto de diseño actual al punto mejorado. Las formas generales de obtener α_k han sido discutidas en el epígrafe 2.5.3.1, por lo que se va a centrar el cálculo en \mathbf{d}_k .

Se puede decir que encontrar el nuevo punto de diseño es encontrar un cambio en el diseño actual tal que minimice el valor de la función objetivo y cumpla las restricciones.

Según lo anterior el problema se puede reformular como

$$\begin{aligned} \text{minimizar : } & f(\mathbf{x}_k + \mathbf{d}_k) \\ \text{sujeto a : } & g_j(\mathbf{x}_k + \mathbf{d}_k) \geq 0 \quad j = 1, 2, \dots, m \end{aligned} \quad (2.8)$$

La idea básica de todos los métodos RQP para calcular \mathbf{d}_k consiste en convertir el problema en un subproblema cuadrático que se resuelve por las técnicas conocidas para dicho tipo de problemas. Hay diferentes aproximaciones para hacer la conversión del problema pero todas ellas se basan en sustituir las funciones f y g_j por sus desarrollos en serie alrededor del punto de diseño \mathbf{x}_k .

En todas las aproximaciones se hace uso de la función lagrangiana

$$L(\mathbf{x}, \mathbf{u}) = f(\mathbf{x}) - \sum_{j=1}^m u_j g_j(\mathbf{x}) \quad (2.9)$$

en donde los u_j son los multiplicadores de Lagrange.

El que todas las aproximaciones utilicen la función lagrangiana se debe a que una vez hecha la aproximación correspondiente para convertir en cuadrático el problema se exige a ésta el cumplimiento de las condiciones de optimalidad de Kuhn-Tucker

$$\begin{aligned}
 \nabla L(\mathbf{x}, \mathbf{u}) &= 0 \\
 g_j(\mathbf{x}) &\geq 0 \quad j = 1, 2, \dots, m \\
 u_j &\geq 0 \quad j = 1, 2, \dots, m \\
 g_j(\mathbf{x})u_j &= 0 \quad j = 1, 2, \dots, m
 \end{aligned} \tag{2.10}$$

La forma más directa de convertir el problema en cuadrático es hacer una aproximación cuadrática de la función objetivo. Hacer también una aproximación cuadrática de las restricciones conduce a un problema para el que se han propuesto algunas técnicas iterativas (Belegundu, 1982). Por ello, lo habitual es hacer una aproximación lineal de las restricciones. Con estas aproximaciones el problema se puede reformular como

$$\begin{aligned}
 \text{minimizar : } & 1/2 \mathbf{d}_k^T \nabla^2 f(\mathbf{x}_k)^T \mathbf{d}_k + \nabla f(\mathbf{x}_k)^T \mathbf{d}_k \\
 \text{sujeto a : } & \nabla g_j(\mathbf{x}_k)^T \mathbf{d}_k + g_j(\mathbf{x}_k) \geq 0 \quad j = 1, 2, \dots, m
 \end{aligned} \tag{2.11}$$

Se añade una condición para asegurar que la solución esté acotada

$$1/2 \mathbf{d}_k^T \mathbf{W}_k \mathbf{d}_k \leq e^2 \tag{2.12}$$

donde \mathbf{W}_k es una matriz que debe ser definida positiva; por tanto, basta tomar la matriz identidad. No obstante, Belegundu (1982) muestra cómo incluyendo en dicha matriz información de curvatura de las funciones se aumenta la relación de convergencia.

El problema así formulado consiste en encontrar una dirección \mathbf{d}_k tal que conduzca a la región válida (ninguna restricción violada) con el menor valor posible para la función objetivo y a una distancia no superior a e . No obstante, si e es lo suficientemente pequeño o el k -ésimo punto de diseño está suficientemente lejos de la región válida será imposible encontrar una solución que cumpla las Ecs. (2.11) y (2.12) al mismo tiempo.

Las condiciones de optimalidad del punto elegido quedan más patentes planteando las condiciones de Kuhn-Tucker

$$\begin{aligned}
 \nabla f(\mathbf{x}_k) + (\nabla^2 L(\mathbf{x}_k, \mathbf{u}') + \mu' \mathbf{I}) \mathbf{d}_k - \sum_{j=1}^m u_j' \nabla g_j(\mathbf{x}_k)^T &= 0 \\
 u_j' (\nabla g_j(\mathbf{x}_k)^T \mathbf{d}_k + g_j(\mathbf{x}_k)) &= 0 \quad j = 1, 2, \dots, m \\
 \mu' (1/2 \mathbf{d}_k^T \mathbf{W}_k \mathbf{d}_k - e^2) &= 0 \\
 u_j' &\geq 0 \quad j = 1, 2, \dots, m \\
 \mu' &\geq 0
 \end{aligned} \tag{2.13}$$

En estas condiciones es importante observar que la expresión de la función lagrangiana se ha obtenido haciendo una aproximación cuadrática tanto de la función objetivo como de las restricciones.

De estas condiciones se deduce que tomando $\mu' > 0$ se asegura indirectamente que la región válida se puede alcanzar independientemente de lo lejano que esté el punto de diseño \mathbf{x}_k . Por lo tanto, imponiendo esta condición en la Ec. (2.13), se obtiene

$$\begin{aligned} \nabla f(\mathbf{x}_k) - \sum_{j=1}^m u_j' \nabla g_j(\mathbf{x}_k)^T + (\nabla^2 L(\mathbf{x}_k, \mathbf{u}') + \mu' \mathbf{I}) \mathbf{d}_k &= 0 \\ u_j' (\nabla g_j(\mathbf{x}_k)^T \mathbf{d}_k + g_j(\mathbf{x}_k)) &= 0 \quad j = 1, 2, \dots, m \\ u_j' &\geq 0 \quad j = 1, 2, \dots, m \end{aligned} \quad (2.14)$$

problema que tiene siempre solución.

Afortunadamente no es necesario resolver el problema así planteado. Haciendo los cambios

$$u_j = \frac{u_j'}{\mu} \quad \mu = \frac{1}{\mu'} \quad (2.15)$$

se puede reescribir la Ec. (2.14) como

$$\begin{aligned} \mu \nabla f(\mathbf{x}_k) - \sum_{j=1}^m u_j \nabla g_j(\mathbf{x}_k)^T + \mu (\nabla^2 L(\mathbf{x}_k, \mathbf{u}) + \mu' \mathbf{I}) \mathbf{d}_k &= 0 \\ u_j (\nabla g_j(\mathbf{x}_k)^T \mathbf{d}_k + g_j(\mathbf{x}_k)) &= 0 \quad j = 1, 2, \dots, m \\ u_j &\geq 0 \quad j = 1, 2, \dots, m \end{aligned} \quad (2.16)$$

Planteando ahora el problema cuadrático

$$\begin{aligned} \text{minimizar : } & 1/2 \mathbf{d}_k^T \mathbf{W}_k \mathbf{d}_k + \mu f(\mathbf{x}_k)^T \mathbf{d}_k \\ \text{sujeto a : } & g_j(\mathbf{x}_k)^T \mathbf{d}_k + g_j(\mathbf{x}_k) \geq 0 \quad j = 1, 2, \dots, m \end{aligned} \quad (2.17)$$

Se observa que las condiciones de Kuhn-Tucker para él son las mismas de las Ecs. (2.16), sin más que tomar

$$\mathbf{W}_k = \nabla^2 L(\mathbf{x}_k, \mathbf{u}) + \mu' \mathbf{I} \quad (2.18)$$

$$\mu = 1 / \mu' = 1 \quad (2.19)$$

De la Ec. (2.18) se puede comentar que para un μ' suficientemente grande \mathbf{W}_k es definida positiva. Por tanto \mathbf{W}_k puede verse como una aproximación definida positiva al hessiano de la función lagrangiana. Esta matriz, conteniendo información de curvaturas de las funciones, es la que hace que el método tenga una relación de convergencia superlineal. No obstante, dado el esfuerzo de cálculo que requiere conocer la matriz \mathbf{W}_k es habitual recurrir a estimaciones de la misma del tipo Quasi-Newton. Por lo tanto, la relación de convergencia suele ser menor que la máxima posible aunque sigue siendo igual o mayor que la unidad mientras siga siendo definida positiva.

El significado de la Ec. (2.19) se puede ver claramente en Belegundu (1982), donde se llega a una expresión

$$\mathbf{d}_k = \mu \mathbf{d}_1 + \mathbf{d}_2 \quad (2.20)$$

en donde \mathbf{d}_1 es la componente de máximo descenso (mayor reducción de la función objetivo) mientras que \mathbf{d}_2 es la componente de máximo acercamiento a la región factible. Por tanto, tomar $\mu = 1$ significa dar el mismo peso a la minimización de la función objetivo que al cumplimiento de las restricciones.

De todo lo anterior se deduce que se puede resolver el problema de la Ec. (2.17) para encontrar el valor \mathbf{d}_k buscado en la Ec. (2.8). No obstante, es difícil encontrar la solución debido a que

$$\nabla^2 L(\mathbf{x}_k, \mathbf{u}) = \nabla^2 f(\mathbf{x}_k)^T - \sum_{j=1}^m u_j \nabla^2 g_j(\mathbf{x}_k)^T \quad (2.21)$$

es una función que depende de u , que es el vector de los multiplicadores de Lagrange en el óptimo. Dado que el valor de los multiplicadores en el óptimo no es conocido a priori, para obtener la solución se recurre a estimaciones de los mismos para las diferentes iteraciones

$$\nabla^2 L(\mathbf{x}_k, \mathbf{u}) = \nabla^2 f(\mathbf{x}_k)^T - \sum_{j=1}^m u_{kj} \nabla^2 g_j(\mathbf{x}_k)^T \quad (2.22)$$

Estas estimaciones de los multiplicadores se pueden realizar de varias formas. Una de las más habituales es

$$\mathbf{u}_k = -\mathbf{N}^+ f(\mathbf{x}_k) \quad (2.23)$$

siendo:

$$\mathbf{N}^+ = (\mathbf{N}^T \mathbf{N})^{-1} \mathbf{N}^T \quad (2.24)$$

la pseudoinversa de la matriz \mathbf{N} formada por m columnas que contienen a los vectores $g_j(\mathbf{x}_k)^T$. Esta aproximación será tanto más real conforme el proceso iterativo se vaya acercando al óptimo.

2.5.4.3 Algoritmo de Schittkowski

2.5.4.3.1 Descripción

El algoritmo propuesto por Schittkowsky (1983) es un RQP con aproximación cuadrática de la función Lagrangiana y aproximación lineal de las restricciones. El algoritmo plantea la búsqueda iterativa del punto óptimo de diseño por medio de la Ec. (2.2). Para ello establece una estrategia de cálculo diferente para la dirección de búsqueda y para la longitud.

Para la dirección de búsqueda el algoritmo plantea la resolución del problema formulado en (2.17). Si bien establece dos diferencias importantes respecto a dicha formulación:

- La consideración de restricciones de igualdad.
- La consideración del subconjunto de restricciones activas.

Considerando las restricciones de igualdad independientemente se llega a una formulación más general. Esto permite un tratamiento numérico más eficiente de cada tipo de restricciones. La formulación del problema queda convertida en

$$\begin{aligned} \text{minimizar : } & 1/2 \mathbf{d}_k^T \mathbf{W}_k \mathbf{d}_k + \nabla f(\mathbf{x}_k)^T \mathbf{d}_k \\ \text{suje to a : } & \nabla g_j(\mathbf{x}_k)^T \mathbf{d}_k + g_j(\mathbf{x}_k) = 0 \quad j = 1, 2, \dots, m_i \\ & \nabla g_j(\mathbf{x}_k)^T \mathbf{d}_k + g_j(\mathbf{x}_k) \geq 0 \quad j = m_i + 1, \dots, m \end{aligned} \quad (2.25)$$

La segunda consideración pretende aumentar la eficiencia del algoritmo eliminando cálculos innecesarios de gradientes de las restricciones no activas. La formulación expresada en (2.25) se replantea como

$$\begin{aligned} \text{minimizar : } & 1/2 \mathbf{d}_k^T \mathbf{W}_k \mathbf{d}_k + \nabla f(\mathbf{x}_k)^T \mathbf{d}_k \\ \text{suje to a : } & \nabla g_j(\mathbf{x}_k)^T \mathbf{d}_k + g_j(\mathbf{x}_k) = 0 \quad j \in J_{ak} \\ & \nabla g_j(\mathbf{x}_k)^T \mathbf{d}_k + g_j(\mathbf{x}_k) \geq 0 \quad j \in J_{ik} \end{aligned} \quad (2.26)$$

siendo:

- J_{ak} El conjunto formado por todas las restricciones de igualdad más las restricciones activas en la iteración k .
- J_{ik} Todas las restricciones no pertenecientes a J_{ak} .

Una restricción es activa si su valor es negativo o si el correspondiente multiplicador de Lagrange es mayor que cero. Para evitar situaciones inestables provocadas por restricciones no activas pero muy cercanas a serlo ($g_j(\mathbf{x}_k) > 0$; $g_j(\mathbf{x}_k) \cong 0$) se trabaja con restricciones ε -activas. Así, se define el conjunto J_{ak} como el formado por todas las restricciones de igualdad más las restricciones de desigualdad que cumplen $g_j(\mathbf{x}_k) \leq \varepsilon$ para la iteración k .

Para evitar que el problema pueda ser inconsistente se introduce esta nueva condición por medio de una nueva variable δ . De forma que el problema expresado en (2.26) queda convertido en

$$\begin{aligned} \text{minimizar : } & 1/2 \mathbf{d}_k^T \mathbf{W}_k \mathbf{d}_k + \nabla f(\mathbf{x}_k)^T \mathbf{d}_k + 1/2 r_k (\delta_k)^2 \\ \text{suje to a : } & \nabla g_j(\mathbf{x}_k)^T \mathbf{d}_k + (1 - \delta) g_j(\mathbf{x}_k) = 0 \quad j \in J_{ak} \\ & g_j(\mathbf{x}_k)^T \mathbf{d}_k + g_j(\mathbf{x}_k) \geq 0 \quad j \in J_{ik} \end{aligned} \quad (2.27)$$

en donde a la variable δ se le exige que $0 \leq \delta \leq 1$ y r es el parámetro de penalización de dicha variable.

Una vez resuelto el problema formulado en 2.27, se tiene una dirección de búsqueda \mathbf{d}_k y una estimación (p. e. la de la Ec. (2.15)) de los multiplicadores de Lagrange. Entonces, para actualizar las variables y los multiplicadores se debe plantear

$$\begin{aligned}\mathbf{x}_{k+1} &= \mathbf{x}_k + \alpha_k \mathbf{d}_k \\ \mathbf{v}_{k+1} &= \mathbf{v}_k + \alpha_k (\mathbf{u}_k - \mathbf{v}_k)\end{aligned}\quad (2.28)$$

Para encontrar la longitud de búsqueda α_k Han (1977) y Powell (1978) minimizan una función de búsqueda dada por

$$\Phi(\mathbf{x}, \mathbf{r}) = f(\mathbf{x}) + \sum_{j=1}^{m_i} r_j |g_j(\mathbf{x})| + \sum_{j=m_i+1}^m r_j |\min(0, g_j(\mathbf{x}))| \quad (2.29)$$

en donde $\mathbf{r} = \{r_1, r_2, \dots, r_m\}^T$ son los parámetros de penalización por violación de las diferentes restricciones.

Para soslayar los problemas que da la utilización de esta función (pérdida de la relación de convergencia superlineal, etc.) se ha sustituido por una función lagrangiana diferenciable

$$\Phi_r(\mathbf{x}, \mathbf{v}) = f(\mathbf{x}) + \sum_{j \in J_a} \left(v_j g_j(\mathbf{x}) - \frac{1}{2} r_j g_j(\mathbf{x})^2 \right) - \frac{1}{2} \sum_{j \in J_i} \frac{v_j^2}{r_j} \quad (2.30)$$

A partir de la Ec. (2.28) se puede definir la función a minimizar con respecto a α para obtener la longitud α_k

$$\phi(\alpha) = \Phi_r \left[\begin{bmatrix} \mathbf{x}_k \\ \mathbf{v}_k \end{bmatrix} + \alpha \begin{bmatrix} \mathbf{d}_k \\ \mathbf{u}_k - \mathbf{v}_k \end{bmatrix} \right] \quad (2.31)$$

Debe señalarse, no obstante, que la elección de los parámetros de penalización debe cumplir ciertas condiciones Rajan (1986) para evitar comportamientos anómalos del algoritmo.

2.5.4.3.2 Algoritmo

Los principales pasos del algoritmo pueden describirse de la forma:

- 1 Inicio ($k = 0$). Se elige cualquier punto de partida ($\mathbf{x}_0, \mathbf{v}_0, \mathbf{W}_0, \mathbf{r}_0$), se evalúan los valores de las funciones y sus gradientes. Luego se determina el subconjunto activo J_{ak} .
- 2 Se resuelve el subproblema cuadrático formulado en (2.27). La solución obtenida es \mathbf{d}_k, δ_k , y los multiplicadores \mathbf{u}_k .
- 3 Se determinan los nuevos parámetros de penalización $\mathbf{r}_{k+1}, \rho_{k+1}$.
- 4 Se realiza una búsqueda unidimensional mediante el problema formulado en (2.31) para obtener α_k .
- 5 Se actualizan las variables y los multiplicadores.
- 6 Se evalúan los nuevos valores de las funciones y sus gradientes y se estima el nuevo valor del hessiano.

- 7 Se comprueba el criterio de convergencia y se finaliza; o bien comienza un nuevo ciclo ($k = k+1$) volviendo al paso 1.

En cuanto a la resolución del subproblema cuadrático puede emplearse la subrutina de Gill, Murray y Wright (1981) u otra basada en la publicada por Lawson y Hanson (1974) que es la empleada en el sistema DISSENY (Martí, 1985).

Por lo que respecta al criterio de convergencia se utiliza uno combinado que tiene en cuenta tanto el valor de la función lagrangiana

$$\|L(\mathbf{x}_k, \mathbf{u})\|^2 \leq \varepsilon \quad (2.32)$$

como la información de derivadas de las funciones

$$\mathbf{d}_k^T \mathbf{W}_k \mathbf{d}_k \leq \varepsilon \quad (2.33)$$

Por último, indicar que Rajan (1986) ofrece una descripción más detallada del algoritmo que incluye ciertos parámetros de ajuste necesarios para evitar comportamientos anómalos.

Diseño óptimo de topología de estructuras articuladas. Métodos del universo estructural

3.1 INTRODUCCIÓN

En este capítulo se describen los algoritmos de optimización de topología de estructuras articuladas basados en el método del universo estructural (*ground structure*).

El capítulo comienza enumerando los antecedentes del método del universo estructural. El resto del capítulo se divide en dos bloques.

El primer bloque de este capítulo trata sobre la definición y generación de los universos estructurales. Comienza con una definición del universo estructural y una descripción general de las formas de generación. Continúa con la implementación efectuada para la generación de universos estructurales. Por último, se muestran varios ejemplos sobre generación de universos estructurales.

En el segundo bloque de este capítulo se describen varios algoritmos de optimización de topología de estructuras articuladas basados en el universo estructural. Se han implementado los siguientes algoritmos basados en el universo estructural: el método *stress-ratio*, el de la *compliance*, el de Pedersen, el de Achtziger y el de búsqueda exhaustiva. Para cada algoritmo se muestra la teoría, se describe el algoritmo y se presenta la implementación efectuada con las aportaciones añadidas a algunos métodos.

A continuación se muestra un ejemplo de optimización de topología mediante dichos algoritmos.

3.2 ANTECEDENTES

La optimización de topología de estructuras articuladas en la forma de rejilla continua es un tema clásico en el diseño estructural. Michell (1904), fue el pionero en el estudio de las propiedades fundamentales de la rejilla continua óptima, pero este interesante

campo no ha sido desarrollado hasta mucho después. La aplicación de los métodos numéricos a los problemas de optimización de estructuras articuladas tiene una corta historia, con unas primeras contribuciones de, por ejemplo, Dorn (1964) y Fleron (1964).

La optimización de topología y geometría de estructuras articuladas puede ser convenientemente formulada con el llamado método del universo estructural. Éste es el método de optimización más usado por los diferentes autores y consiste, básicamente, en, dada una estructura articulada potencial formada por un conjunto de conexiones entre un conjunto fijo de posiciones nodales, determinar las barras que forman parte del óptimo y el valor de sus áreas.

3.3 UNIVERSO ESTRUCTURAL

3.3.1 Introducción

Dada una malla de n posiciones nodales definidas en un dominio de diseño Ω , se define el universo estructural como un subconjunto de todas las $m=n(n+1)/2$ posibles conexiones entre cada dos nudos mediante barras articuladas.

La topología óptima de la estructura articulada es generada variando las áreas de las secciones transversales de las barras, permitiendo que éstas se puedan hacer nulas. Ésta es la principal ventaja de los métodos del universo estructural, que reducen los problemas de optimización de topología y geometría a un problema de optimización de propiedades.

3.3.2 Generación

3.3.2.1 Introducción

Para la generación de las posiciones nodales definidas en el dominio de diseño, se utilizan generadores de malla mapeados similares a los de los programas de elementos finitos.

3.3.2.2 Mapeado de puntos nodales

Los métodos de mapeo consisten en el mapeo de una malla de referencia y la transformación de los nodos de ésta al dominio real mediante una función de transformación.

Sea Ω un dominio real, por ejemplo un hexaedro (cuadrilátero para dominios bidimensionales); el contorno de este dominio está formado por seis caras (cuatro lados), y en cada lado se han definido una serie de puntos o divisiones indicativas del número de elementos de la malla que se desea generar, con la única condición de que coincida el número de estos puntos en los lados opuestos. Sea Ω' un dominio cúbico (cuadrado) normalizado (coordenadas de -1 a 1), en donde se definen igual número de

puntos que los definidos en el contorno del dominio Ω . En el dominio Ω' se puede obtener una malla canónica mediante la unión de puntos opuestos. Esta malla compuesta de elementos cúbicos (cuadrados) define la conectividad de los nodos de una forma simple. Sea f una función de transformación que proyecta cualquier punto de Ω' en el dominio Ω , entonces, aplicando esta transformación a todos los nodos de la malla de referencia, se obtendrá el mapa (en términos de localización de puntos y conectividades) de los elementos de la malla en el dominio real Ω . (Fig. 3.1). Esta transformación debe establecer una correspondencia biunívoca entre las coordenadas locales (ξ , η y ζ) y las globales (x , y y z).

$$\begin{Bmatrix} x \\ y \\ z \end{Bmatrix} = f \begin{Bmatrix} \xi \\ \eta \\ \zeta \end{Bmatrix} \tag{3.1}$$

La capacidad de generar mallas en dominios con geometrías más o menos complejas, va a depender de la naturaleza de la función de transformación f .

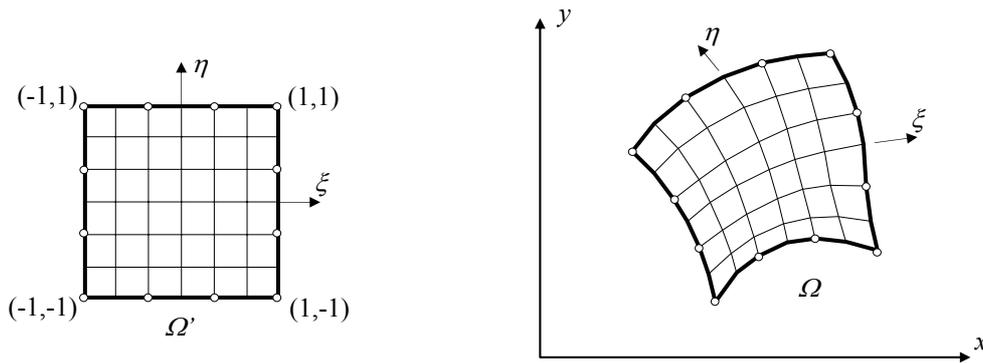


Figura 3.1 Proceso de mapeo y transformación (bidimensional)

Normalmente se suelen usar las funciones de forma del MEF como funciones de transformación, en cuyo caso

$$\begin{aligned} x &= \mathbf{N}\mathbf{x} = \{N_1 \quad N_2 \quad \dots\} \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \end{Bmatrix} \\ y &= \mathbf{N}\mathbf{y} = \{N_1 \quad N_2 \quad \dots\} \begin{Bmatrix} y_1 \\ y_2 \\ \vdots \end{Bmatrix} \\ z &= \mathbf{N}\mathbf{z} = \{N_1 \quad N_2 \quad \dots\} \begin{Bmatrix} z_1 \\ z_2 \\ \vdots \end{Bmatrix} \end{aligned} \tag{3.2}$$

3.3.2.3 Orden de vecindad

Si se utilizan todas las conexiones, una gran variedad de direcciones es posible, pero a cambio, el número de barras es muy alto. A veces, en la práctica, no son necesarias tantas posibles direcciones, por lo que es mejor definir un universo estructural donde sólo se conectan los nudos vecinos hasta un cierto orden.

Un universo estructural con orden de vecindad ν es un subconjunto del universo estructural total donde sólo se conectan los nudos vecinos hasta el orden ν . En la Fig. 3.2 se muestran los diferentes órdenes de vecindad de un nudo.

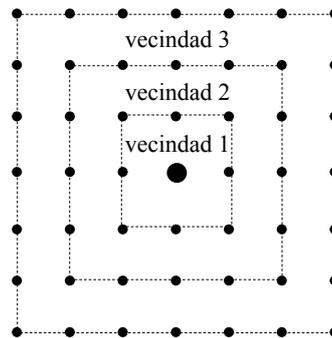


Figura 3.2 Diferentes órdenes de vecindad de un nudo (bidimensional)

3.3.2.4 Generación de barras

Una vez que se han obtenido las posiciones nodales se añaden barras en algunas o en todas las posibles conexiones entre nudos.

Primero se añaden barras en las líneas (Fig. 3.3) conectando cada nudo con su vecino; en el resto de las conexiones no se añaden barras puesto que estarían superpuestas a las anteriores.

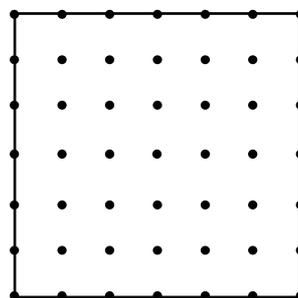


Figura 3.3 Generación de barras en las líneas (bidimensional)

Después se añaden barras en el interior de las áreas. Para evitar la duplicación de barras se utiliza el esquema de generación de la Fig. 3.4 (para un dominio bidimensional y órdenes de vecindad 1, 2 y 3). Cuando hay varios nodos alineados se eliminan las barras superpuestas

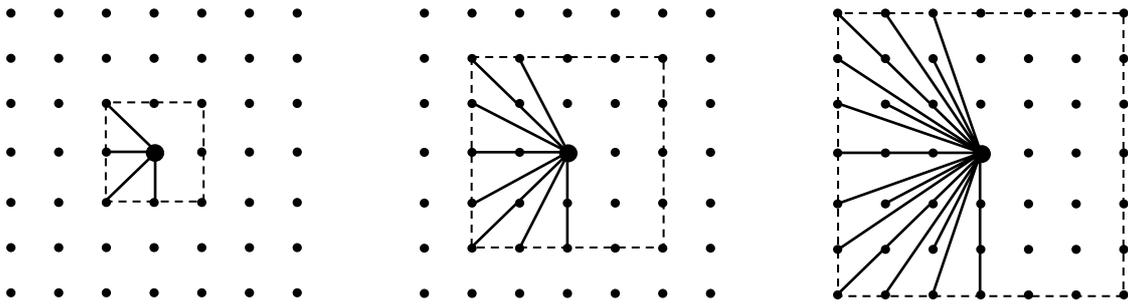


Figura 3.4 Generación de barras para órdenes de vecindad 1, 2 y 3 (bidimensional)

Y, por último, se añaden barras en el interior de los volúmenes (para dominios tridimensionales).

3.3.2.5 Implementación

La implementación efectuada consta de los siguientes pasos:

- 1 Modelado del dominio mediante hexaedros (cuadriláteros) cuadráticos serendípticos (con 1 nodo en cada vértice y 1 nodo en cada arista), que en adelante se denominarán superelementos, conectados unos con otros a lo largo de sus caras (lados). Pueden unirse varios vértices para formar tetraedros (triángulos). Para ello se definen puntos, con los puntos se definen las líneas (3 puntos por cada línea para que sea cuadrático), con las líneas se definen las áreas (3 ó 4 líneas por área) y con las áreas se definen los volúmenes (4, 5 ó 6 áreas por volumen). Los superelementos son los volúmenes (áreas). En la Fig. 3.5 se muestra un ejemplo de modelado de un dominio bidimensional mediante superelementos cuadrangulares cuadráticos (8 nodos).

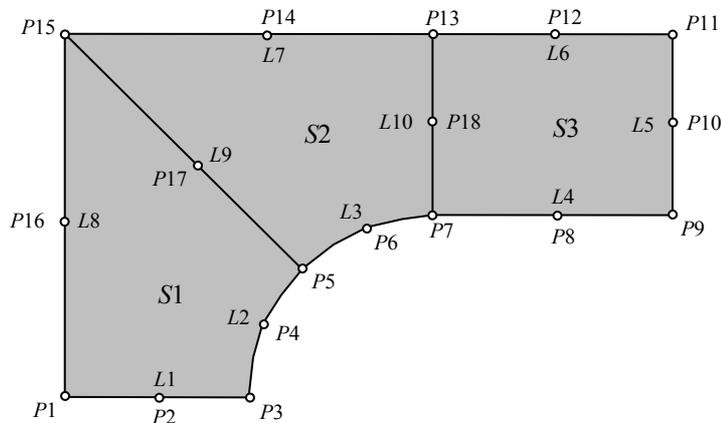


Figura 3.5 Modelado de un dominio con cuadriláteros cuadráticos (8 nodos)

Las funciones de forma, para la transformación geométrica, de un superelemento cuadrangular cuadrático serendípito (8 nodos) son las siguientes:

Nodos de vértice

$$N_i = \frac{1}{4}(1 + \xi_i \xi)(1 + \eta_i \eta)(\xi_i \xi + \eta_i \eta - 1) \quad (3.3)$$

Nodos laterales

$$\begin{aligned} \xi_i = 0 \quad N_i &= \frac{1}{2}(1 - \xi^2)(1 + \eta_i \eta) \\ \eta_i = 0 \quad N_i &= \frac{1}{2}(1 + \xi_i \xi)(1 - \eta^2) \end{aligned} \quad (3.4)$$

y las de un superelemento hexaédrico cuadrático serendípito (20 nodos) las siguientes:

Nodos de vértice

$$N_i = \frac{1}{8}(1 + \xi_i \xi)(1 + \eta_i \eta)(1 + \zeta_i \zeta)(\xi_i \xi + \eta_i \eta + \zeta_i \zeta - 2) \quad (3.5)$$

Nodos laterales

$$\begin{aligned} \xi_i = 0 \quad N_i &= \frac{1}{4}(1 - \xi^2)(1 + \eta_i \eta)(1 + \zeta_i \zeta) \\ \eta_i = 0 \quad N_i &= \frac{1}{4}(1 + \xi_i \xi)(1 - \eta^2)(1 + \zeta_i \zeta) \\ \zeta_i = 0 \quad N_i &= \frac{1}{4}(1 + \xi_i \xi)(1 + \eta_i \eta)(1 - \zeta^2) \end{aligned} \quad (3.6)$$

- 2 Mapeado de los superelementos para la obtención de las posiciones nodales.
- 3 Conexión de los nudos mediante barras articuladas según el orden de vecindad.
- 4 Eliminación de las barras superpuestas y las que unan nudos con todos sus grados de libertad restringidos.
- 5 Eliminación de las barras que violen determinadas restricciones, como puedan ser: las que tengan una longitud mayor que un valor determinado, las que crucen por un determinado volumen (área), etc.

También se pueden crear universos estructurales a partir de un número determinado de puntos, sin necesidad de definir líneas, áreas y volúmenes. Para ello, se conectan mediante barras todas las combinaciones posibles de dos puntos. En este caso, el orden de vecindad no tiene sentido, aunque el resto de restricciones se puede mantener.

3.3.3 Ejemplos

En la Fig. 3.6 se muestra el universo estructural para un dominio cuadrado y para órdenes de vecindad 1, 2, 3 y completo (sin limitación de orden de vecindad). En la Fig. 3.7 se muestra el universo del mismo dominio anterior eliminando las barras que cruzan por el dominio interior (agujero).

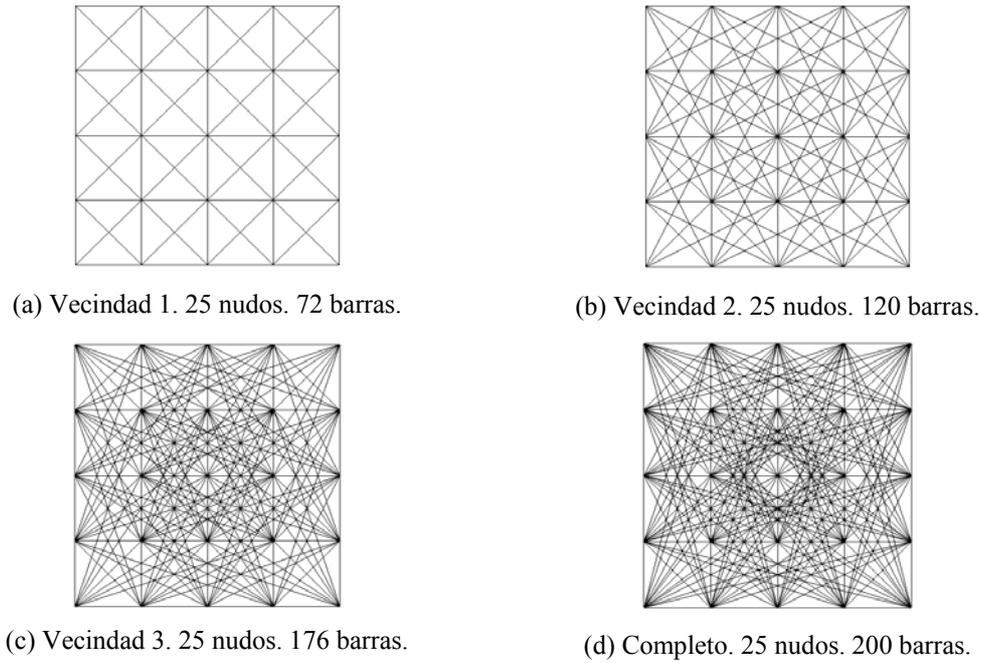


Figura 3.6 Universo estructural para un dominio cuadrado

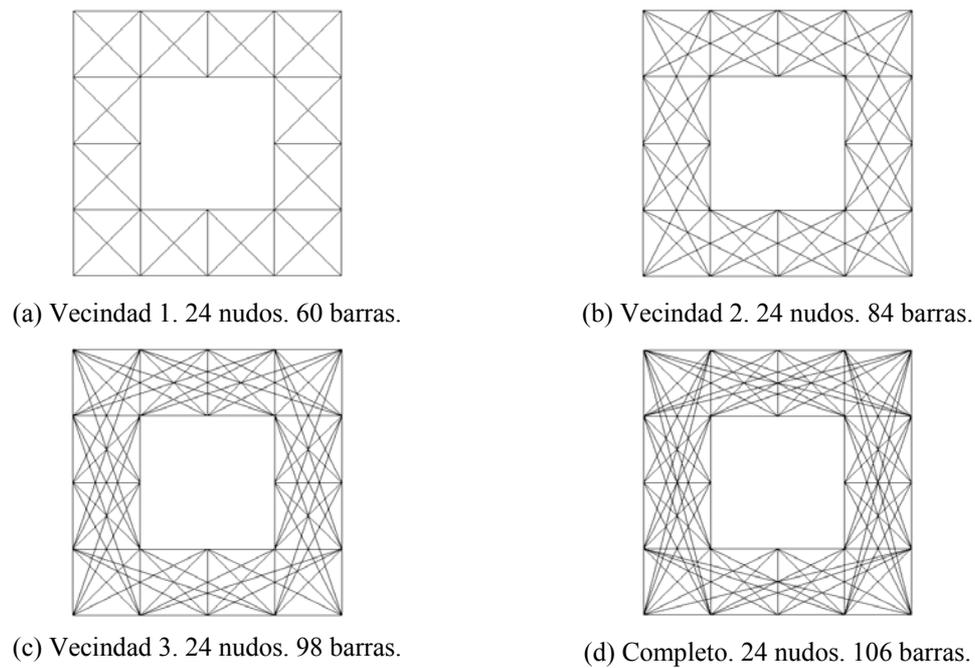


Figura 3.7 Universo estructural para un dominio cuadrado con un agujero

En las Figs. 3.8 y 3.9 se muestran los universos estructurales para una ménsula y una torre. En estos ejemplos los dominios se dividen en varias áreas para las que se crean universos independientes, de ahí que no haya barras que crucen de un dominio a otro.

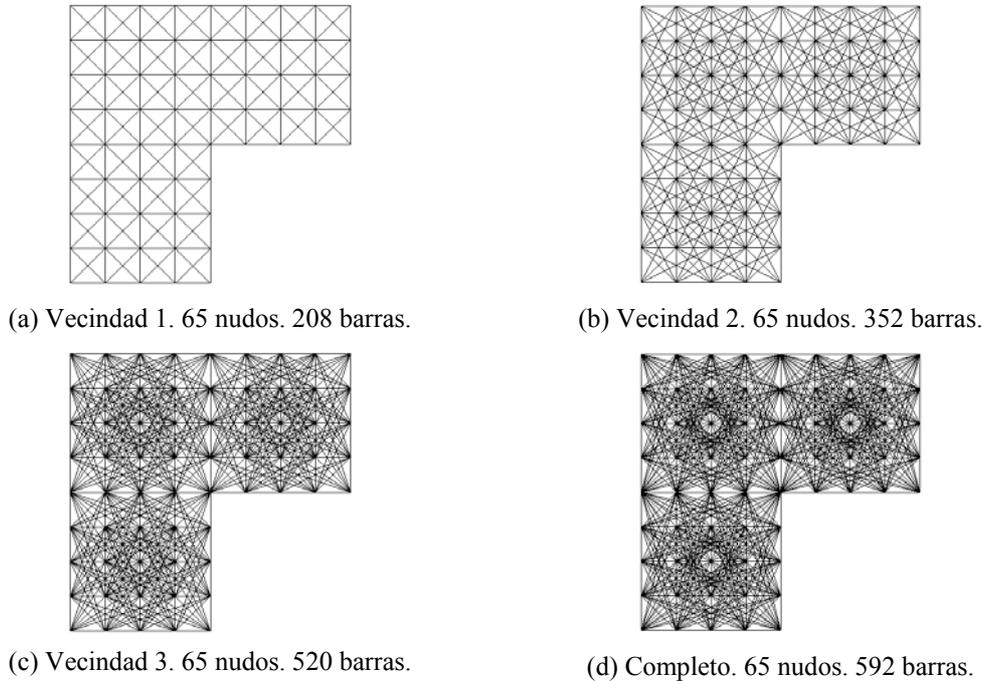


Figura 3.8 Universo estructural para una ménsula

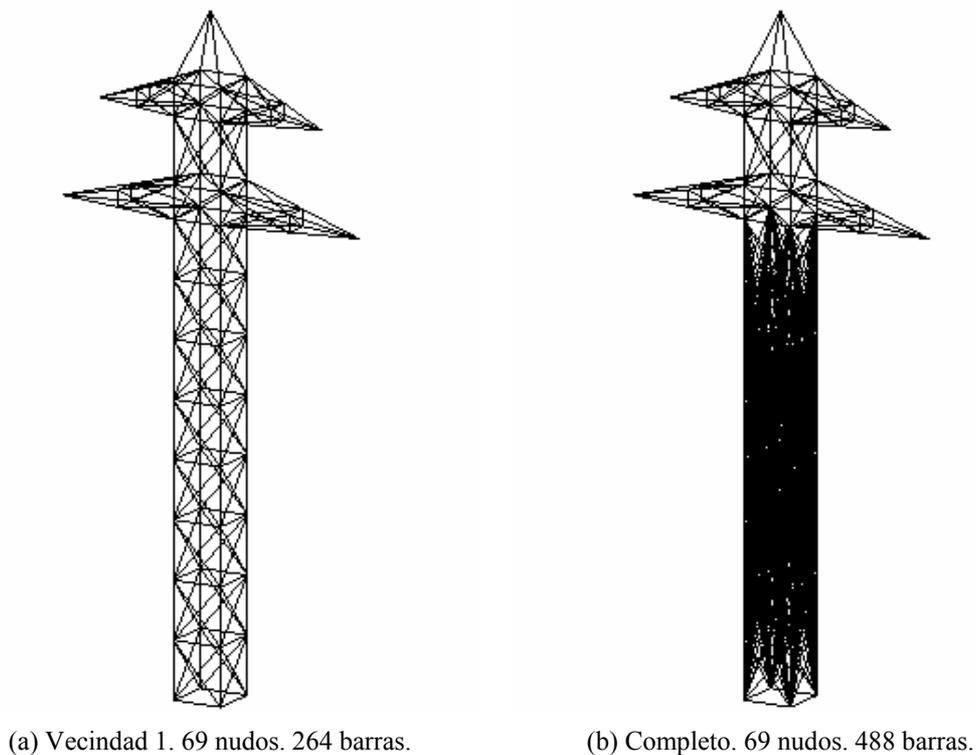


Figura 3.9 Universo estructural para una torre

3.4 MÉTODOS DE SOLUCIÓN

3.4.1 Introducción

Se han implementado el método *stress-ratio* por su sencillez, el de la *compliance* por estar muy extendido y por su rapidez (cuando sólo se usan los desplazamientos como variables), el de Pedersen por admitir restricciones de pandeo, el de Achtziger por considerar las cadenas activas (ver epígrafe 3.4.5.1), y el de búsqueda exhaustiva porque evalúa todas las posibles soluciones.

3.4.2 Método *stress-ratio*

3.4.2.1 Formulación

Este método es el más sencillo de los criterios de optimalidad *Full Stressed Design* (FSD), que considera que la estructura óptima es aquella en la cual cada barra está trabajando a la tensión máxima admisible, y se basa en el supuesto de que los esfuerzos obtenidos en una iteración no cambian al cambiar las áreas de las barras, tal como ocurre en las estructuras isostáticas.

3.4.2.2 Algoritmo de resolución

De acuerdo con los supuestos anteriores, para el caso de restricciones de tensión, la regla de recurrencia es

$$A_i^{k+1} = \frac{\sigma_i}{\sigma_{i\max}} A_i^k \quad (3.7)$$

siendo:

- A_i^k Área de la barra i en la iteración k .
- A_i^{k+1} Área de la barra i en la iteración $k+1$.
- σ_i Tensión de la barra i .
- $\sigma_{i\max}$ Tensión máxima de la barra i .

3.4.2.3 Implementación

Se parte de una estructura inicial y, en cada iteración, se hace un análisis y se modifican las áreas de las barras según la Ec. (3.7) hasta que la variación de la función objetivo de una iteración a la siguiente sea menor que un determinado valor.

3.4.2.3.1 Eliminación de barras

Para disminuir el coste computacional en cada iteración se eliminan las barras cuya área sea menor que un valor predeterminado. Al eliminar barras la estructura puede convertirse en inestable. Para evitar esta inestabilidad se establecen varias estrategias, como la eliminación de los nudos que no son apoyos en los que sólo converge una barra

(quitando también la barra), y la eliminación de los nudos en los que sólo convergen barras alineadas (uniendo las barras).

3.4.3 Método de la *compliance*

3.4.3.1 Formulación

Es conocido que el diseño de estructuras articuladas de volumen mínimo (sujeto a equilibrio de fuerzas y restricciones de tensión) coincide, previo escalado, con el diseño de estructuras de *compliance* mínima (sujeto a equilibrio estático y restricción de volumen). Parece que fue Cox (1965) el primero en detectar esta equivalencia.

La formulación de la *compliance* es la siguiente (Oberndorfer, 1996)

$$\begin{aligned} \underset{\mathbf{a}, \mathbf{u}}{\text{minimizar}} : \quad C = \mathbf{f}^T \mathbf{u} &= \sum_{j=1}^n f_j u_j \\ \text{sujeto a :} \quad \mathbf{K} \mathbf{u} &= \mathbf{f} \\ \sum_{k=1}^m A_k L_k &= V \\ A_k &\geq 0 \quad k = 1, \dots, m \end{aligned} \tag{3.8}$$

siendo:

C	<i>Compliance</i> .
m	Número de barras de la estructura.
n	Número de grados de libertad de la estructura.
L_k	Longitud de la barra k .
$\mathbf{a} = (A_1, \dots, A_n)^T$	Vector de áreas de las barras.
$\mathbf{u} = (u_1, \dots, u_n)^T$	Vector de desplazamientos nodales.
$\mathbf{f} = (f_1, \dots, f_n)^T$	Vector de cargas nodales.
\mathbf{K}	Matriz de rigidez de la estructura.
V	Volumen total de la estructura.

La *compliance* puede ser interpretada como la energía de deformación de la estructura. Hemp (1973) demostró que siempre existe una solución totalmente tensionada para el problema anterior, de ahí que pueda aplicarse un método como el *stress-ratio*. Esto requiere la regularidad de la matriz de rigidez en cada paso para calcular los desplazamientos para unas áreas determinadas. Este requerimiento de regularidad contradice la filosofía de la aproximación de topología donde, generalmente, la mayoría de las áreas serán nulas. Esto y el gran número de barras potenciales conducen a la siguiente aproximación que usa los desplazamientos nodales como únicas variables.

Para una estructura (Taylor, 1977) la *compliance* puede ser rescrita en función de la energía potencial mínima

$$-\frac{1}{2}\mathbf{f}^T \mathbf{u} = \min_{\tilde{\mathbf{u}} \in \mathcal{R}^n} \left\{ \frac{1}{2} \tilde{\mathbf{u}}^T \mathbf{K} \tilde{\mathbf{u}} - \mathbf{f}^T \tilde{\mathbf{u}} \right\} \quad (3.9)$$

donde el equilibrio elástico juega el papel de condición de optimalidad necesaria y suficiente. Esta identidad lleva a la reformulación equivalente del problema (3.8)

$$\begin{aligned} \text{minimizar : } \tilde{\mathbf{C}} = \max_{\mathbf{a}} & \left\{ \mathbf{f}^T \mathbf{u} - \frac{1}{2} \mathbf{u}^T \mathbf{K} \mathbf{u} \right\} \\ \text{sujeto a : } & \sum_{k=1}^m A_k L_k = V \\ & A_k \geq 0 \quad k = 1, \dots, m \end{aligned} \quad (3.10)$$

Comparada con la formulación original, la incómoda restricción de equilibrio ha desaparecido formalmente, pero está oculta ahora en la función objetivo, lo cual es más dificultoso de manejar (puesto que representa un problema de optimización por sí mismo). Afortunadamente, las condiciones matemáticas permiten el intercambio de máx (en \mathbf{u}) y mín (en \mathbf{a}). Entonces, para un \mathbf{u} fijo, el término resultante representa un problema de programación lineal en la variable \mathbf{a} , donde se usa la dependencia lineal de la matriz de rigidez \mathbf{K} con \mathbf{a} . Este problema lineal puede fácilmente ser resuelto analíticamente, y así la variable \mathbf{a} es eliminada del problema. La formulación resultante es

$$\text{maximizar}_{\mathbf{u}} : \hat{\mathbf{C}} = \min_{1 \leq i \leq m} \left\{ \mathbf{p}^T \mathbf{u} - \frac{V}{2L_i} \mathbf{u}^T \mathbf{k}_i \mathbf{u} \right\} \quad (3.11)$$

siendo:

$$\mathbf{k}_i = \frac{E_i}{L_i} \mathbf{b}_i \mathbf{b}_i^T \quad \text{Matriz de rigidez por unidad de área de la barra } i.$$

$$E_i \quad \text{Módulo de Young de la barra } i.$$

$$\mathbf{b}_i \quad \text{Vector de cosenos directores de la barra } i.$$

El problema de la Ec. (3.11) se puede describir como una versión escalada en forma de un problema de programación lineal simple

$$\begin{aligned} \text{maximizar}_{\mathbf{u}} : & \quad \mathbf{f}^T \mathbf{u} \\ \text{sujeto a : } & \quad -1 \leq \frac{\sqrt{E_i}}{L_i} \mathbf{b}_i^T \mathbf{u} \leq 1 \quad i = 1, \dots, m \end{aligned} \quad (3.12)$$

Si el módulo de Young es el mismo para todas las barras, el problema (3.11) se reduce a

$$\begin{aligned} \text{maximizar}_{\mathbf{u}} : & \quad \mathbf{f}^T \mathbf{u} \\ \text{sujeto a : } & \quad -1 \leq \frac{\mathbf{b}_i^T}{L_i} \mathbf{u} \leq 1 \quad i = 1, \dots, m \end{aligned} \quad (3.13)$$

Al eliminar las áreas como variables el coste computacional es mucho menor ya que en los problemas de optimización de topología el número de nudos es mucho menor al número de barras. Desafortunadamente, esta simplificación sólo puede ser usada si no hay restricciones de desplazamientos, ni de pandeo, ni de esbeltez ni de área mínima. Sin embargo es posible generalizar el concepto para varios estados de cargas.

3.4.3.2 Algoritmo de resolución

Al ser un problema lineal se puede resolver por cualquier algoritmo de programación lineal (por ejemplo el método *simplex*). De una solución \mathbf{u}^* de (3.13) y sus correspondientes vectores de multiplicadores de Lagrange puede, fácilmente, obtenerse una solución de (3.8) mediante un escalado (Beckers, 1997)

$$\begin{aligned} \mathbf{u} &= \mu^* \mathbf{u}^* \\ \mathbf{a} &= \frac{1}{E\mu^*} (\boldsymbol{\lambda}^+ + \boldsymbol{\lambda}^-) \\ \mu^* &= \frac{1}{VE} \sum_{i=1}^m (\lambda_i^+ + \lambda_i^-) \end{aligned} \quad (3.14)$$

siendo:

$$\begin{aligned} \boldsymbol{\lambda}^+ &= (\lambda_1^+, \dots, \lambda_m^+)^T && \text{Vector de multiplicadores de Lagrange (restr. +).} \\ \boldsymbol{\lambda}^- &= (\lambda_1^-, \dots, \lambda_m^-)^T && \text{Vector de multiplicadores de Lagrange (restr. -).} \end{aligned}$$

3.4.3.3 Implementación

Para resolver el problema (3.13) se ha utilizado la subrutina DDLPRS de la librería IMSL que resuelve problemas de programación lineal mediante el algoritmo *simplex* revisado.

3.4.4 Método de Pedersen

3.4.4.1 Formulación

El método de Pedersen (1993) es un algoritmo para la optimización de topología de estructuras espaciales de barras articuladas sometidas a un estado de cargas y que considera, exclusivamente, restricciones de tensión y de pandeo de los elementos de la estructura.

Pedersen demuestra, para los supuestos anteriores, que se puede encontrar una topología isostática que minimiza la masa de la estructura. Dado que la solución es isostática, el problema consiste en, partiendo de una estructura hiperestática, buscar la estructura isostática de menor masa, entre todas las estructuras isostáticas posibles.

La masa total de la estructura tendrá la forma

$$M = \sum_{j=1}^m \rho_i L_i A_i(q_i) \quad (3.15)$$

siendo:

- M Masa total de la estructura.
- ρ_i Densidad de la barra i .
- L_i Longitud de la barra i .
- A_i Área de la barra i .
- q_i Esfuerzo axial de la barra i .
- m Número de barras.

Se define una función área, como el área mínima necesaria para que no se violen las restricciones de tensión y pandeo del elemento, esto es

$$A_j = \begin{cases} 0 & \text{para } q_i = 0 \\ \frac{q_i}{\sigma_T} & \text{para } q_i > 0 & \text{tracción (límite elástico)} \\ \frac{L}{\pi \sqrt{\beta E / s}} \sqrt{-q_i} & \text{para } -P_E \leq q_i < 0 & \text{compresión (pandeo)} \\ A_E - \frac{q_i + P_E}{\sigma_C} & \text{para } q_i < -P_E & \text{compresión (límite elástico)} \end{cases} \quad (3.16)$$

siendo:

- σ_T Límite elástico a tracción.
- σ_C Límite elástico a compresión.
- σ_p Límite de proporcionalidad.
- E Módulo de Young.
- s Coeficiente de seguridad.
- β I/A^2 .
- I Momento de inercia de la sección de la barra.
- P_E $\frac{\sigma_p l^2}{\pi^2 \beta E s}$ Carga crítica de Euler.
- A_E $\frac{\sigma_p l^2}{\pi^2 \alpha^2 E}$ Área correspondiente a la carga crítica de Euler.

La Fig. 3.10 representa la función área. Cuando $s = 2 \frac{\sigma_p}{\sigma_C}$, hay continuidad en el punto correspondiente a la carga crítica de Euler.

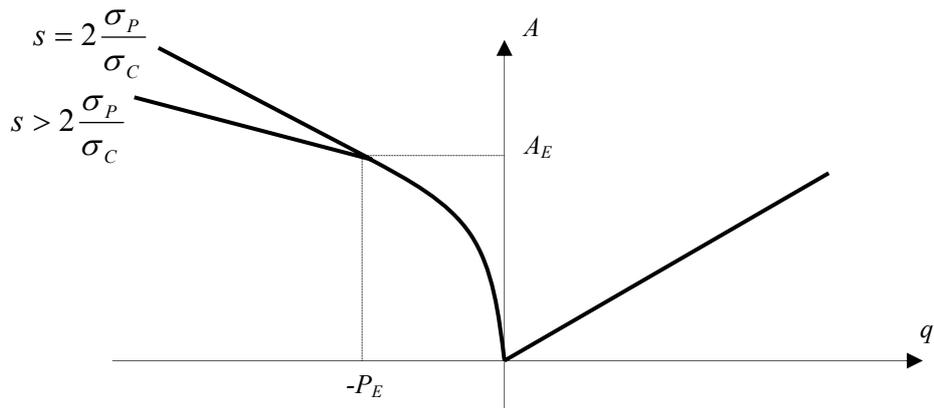


Figura 3.10 Función área

Como puede apreciarse, la masa total de la estructura es función, exclusivamente, de los esfuerzos axiales de las barras.

En la solución del problema estructural se deben satisfacer las ecuaciones de equilibrio en los nudos

$$\mathbf{Rq} = \mathbf{f} \tag{3.17}$$

siendo:

- R** Matriz de cosenos directores ($n \times m$).
- q** Vector de esfuerzos axiales (m).
- f** Vector de fuerzas externas (n).
- n Número de grados de libertad.

Al ser la solución isostática, no es necesario tener en cuenta las ecuaciones de compatibilidad. Por lo tanto, el problema consiste en minimizar (3.15) considerando las restricciones de igualdad (3.17).

La única no linealidad del problema de optimización es la debida a las barras sometidas a pandeo de elementos. Para barras sometidas a tracción o a compresión en el límite elástico, la dependencia de la masa es lineal con respecto al esfuerzo axial (\mathbf{q}_L). La masa de las barras sometidas a compresión en zona elástica, para pandeo de Euler, depende de la raíz cuadrada del esfuerzo axial ($\mathbf{q}_{\sqrt{}}$).

Así pues, el problema de optimización, en el caso más general en que se considere el pandeo de Euler, se puede poner de la forma

$$\begin{aligned} \text{minimizar : } & M = \mathbf{c}_L^T \mathbf{q}_L + \mathbf{c}_{\sqrt{}}^T \sqrt{\mathbf{q}_{\sqrt{}}} \\ \text{sujeto a : } & \mathbf{Rq} = \mathbf{f} \end{aligned} \tag{3.18}$$

donde \mathbf{c}_L y $\mathbf{c}_{\sqrt{}}$ son los vectores de costes lineales y no lineales respectivamente, Ecs. (3.15) y (3.16).

A pesar de que las restricciones son lineales, al ser la función objetivo no lineal, el problema no se puede resolver mediante métodos de programación lineal, tales como el *simplex* clásico. Este inconveniente lo subsana Pedersen (1994), mediante el denominado método *simplex* modificado.

La formulación puede ser extendida para considerar el coste de las reacciones. En realidad las reacciones se tratan mediante barras, individualmente conectadas en los apoyos (una por cada grado de libertad restringido).

3.4.4.2 Algoritmo de resolución

El método *simplex* modificado fue desarrollado por Pedersen (1994) para resolver problemas de optimización de topología de barras articuladas, aunque es totalmente general.

Las diferencias con respecto al método *simplex* clásico son:

- La reformulación para variables no negativas no es necesaria, con lo que se reduce a la mitad el número de variables y por lo tanto se reduce sustancialmente el coste computacional (decisivo cuando se parte del universo estructural).
- La función objetivo no está restringida a funciones lineales, sino que está extendida a funciones no lineales cóncavas.

Dado el problema de optimización

$$\begin{aligned} \text{minimizar : } & f(\mathbf{x}) = \sum_{i=1}^n f_i(x_i) \\ \text{sujeto a : } & \mathbf{Ax} = \mathbf{b} \end{aligned} \quad (3.19)$$

siendo:

- f Función objetivo.
- \mathbf{x} Vector de variables de diseño (n).
- \mathbf{A} Matriz de coeficientes ($m \times n$).
- \mathbf{b} Vector de coeficientes independientes (m).
- n Número de variables de diseño.
- m Número de restricciones lineales de igualdad.
- f_i Funciones que se deben satisfacer.

$$f_i(0) = 0, \quad x_i \frac{df_i}{dx_i} \geq 0, \quad x_i \frac{d^2 f_i}{dx_i^2} \leq 0 \quad \text{para } i = 1, 2, \dots, n \quad (3.20)$$

Con las condiciones de la Ec. (3.20), el problema de la Ec. (3.19) satisface la condición para que la solución óptima sea una solución básica *simplex*.

3.4.4.2.1 Variables positivas y negativas

Cuando las variables son no necesariamente no-negativas, un truco común es tratarlas con el doble del número de variables,

$$\mathbf{x} = \mathbf{x}^+ - \mathbf{x}^-; \quad \mathbf{x}^+ \geq \mathbf{0}; \quad \mathbf{x}^- \geq \mathbf{0} \quad (3.21)$$

y reformulando el problema (3.18) a

$$\text{minimizar : } f(\mathbf{x}) = \sum_{i=1}^n [f_i^+(x_i^+) + f_i^-(x_i^-)] \quad (3.22)$$

$$\text{sujeto a : } \mathbf{A}\mathbf{x}^+ - \mathbf{A}\mathbf{x}^- = \mathbf{b}$$

con lo que la matriz de coeficientes $[\mathbf{A}, -\mathbf{A}]$ es ahora el doble de grande, lo cual puede ser crítico para resolver grandes problemas.

El procedimiento *simplex* modificado, para almacenar \mathbf{A} sólo una vez, es el que se expone a continuación. En primer lugar se separan las variables en variables básicas $\tilde{\mathbf{X}}^+, \tilde{\mathbf{X}}^-$ (dos vectores de $m \times 1$) y variables no básicas $\mathbf{X}_0^+, \mathbf{X}_0^-$ (dos vectores de $(n-m) \times 1$).

La ecuación de las restricciones queda en la forma

$$\tilde{\mathbf{A}}\tilde{\mathbf{x}}^+ - \tilde{\mathbf{A}}\tilde{\mathbf{x}}^- + \mathbf{A}_0\mathbf{x}_0^+ - \mathbf{A}_0\mathbf{x}_0^- = \mathbf{b} \quad (3.23)$$

Una solución básica de la Ec. (3.23) se obtiene haciendo $\mathbf{x}_0^+ = \mathbf{x}_0^- = \mathbf{0}$

$$\tilde{\mathbf{x}}^+ - \tilde{\mathbf{x}}^- = \tilde{\mathbf{b}} = \tilde{\mathbf{A}}^{-1}\mathbf{b} \quad (3.24)$$

Se asume que $\tilde{\mathbf{A}}^{-1}$ existe, es decir, que es no-singular ($m \times m$).

En términos de componentes individuales, la interpretación de la solución de la Ec. (3.24) es

$$\begin{aligned} \tilde{b}_j \geq 0 &\Rightarrow \tilde{x}_j^+ = \tilde{b}_j \quad y \quad \tilde{x}_j^- = 0 \\ \tilde{b}_j < 0 &\Rightarrow \tilde{x}_j^+ = 0 \quad y \quad \tilde{x}_j^- = -\tilde{b}_j \end{aligned} \quad (3.25)$$

Con las combinaciones lineales $\mathbf{Z} = \tilde{\mathbf{A}}^{-1}\mathbf{A}_0$, como en el *simplex* clásico, las restricciones de la Ec. (3.23) son escritas como

$$\tilde{\mathbf{x}}^+ - \tilde{\mathbf{x}}^- = \tilde{\mathbf{b}} - \mathbf{Z}\mathbf{x}_0^+ - \mathbf{Z}\mathbf{x}_0^- \quad (3.26)$$

y con sólo una variable no básica x_i diferente de cero, la ecuación (3.26) sería entonces

$$\tilde{\mathbf{x}}^+ - \tilde{\mathbf{x}}^- = \tilde{\mathbf{b}} - \mathbf{Z}x_i^+ - \mathbf{Z}x_i^- \quad (3.27)$$

El límite del valor numérico de x_i^+ , o de x_i^- , es obtenido por la condición de que las variables básicas no cambian de signo (en el *simplex* clásico no se convierten en negativas). Dos casos, cada uno con dos subcasos, serán considerados:

- 1 Asumiendo que x_i^+ pase a ser variable básica, es decir $x_i^- = 0$, la fila j de la Ec. (3.27) es

$$\tilde{x}_j^+ - \tilde{x}_j^- = \tilde{b}_j - (z_j)_i x_i^+ \quad (3.28)$$

De acuerdo con la Ec. (3.25) se tendrán dos subcasos:

$$(a) \quad \tilde{b}_j \geq 0 \Rightarrow \tilde{x}_j^+ = \tilde{b}_j - (z_j)_i x_i^+$$

y \tilde{x}_j^+ sólo mantiene su signo para

$$\tilde{b}_j - (z_j)_i x_i^+ \geq 0 \quad (3.29)$$

$$(b) \quad \tilde{b}_j < 0 \Rightarrow -\tilde{x}_j^- = \tilde{b}_j - (z_j)_i x_i^+$$

y \tilde{x}_j^- sólo mantiene su signo para

$$\tilde{b}_j - (z_j)_i x_i^+ \leq 0 \quad (3.30)$$

2. Asumiendo que x_i^- pase a ser variable básica, es decir $x_i^+ = 0$, la fila j de la Ec. (3.27) es

$$\tilde{x}_j^+ - \tilde{x}_j^- = \tilde{b}_j + (z_j)_i x_i^- \quad (3.31)$$

Con dos subcasos:

$$(a) \quad \tilde{b}_j \geq 0 \Rightarrow \tilde{x}_j^+ = \tilde{b}_j + (z_j)_i x_i^-$$

y \tilde{x}_j^+ sólo mantiene su signo para

$$\tilde{b}_j + (z_j)_i x_i^- \geq 0 \quad (3.32)$$

$$(b) \quad \tilde{b}_j < 0 \Rightarrow -\tilde{x}_j^- = \tilde{b}_j + (z_j)_i x_i^-$$

y \tilde{x}_j^- sólo mantiene su signo para

$$\tilde{b}_j + (z_j)_i x_i^- \leq 0 \quad (3.33)$$

La condición de la Ec. (3.29) sobre x_i^+ es sólo activa cuando $(z_j)_i > 0$ y en la Ec. (3.30) cuando $(z_j)_i < 0$, luego para el caso (I) se cumple $\tilde{b}_j (z_j)_i > 0$. Análogamente para el caso (II) se cumple $\tilde{b}_j (z_j)_i < 0$. Con estas condiciones se pueden determinar los valores de x_i^+ o de x_i^- y los correspondientes nuevos valores de las variables básicas viejas.

Esta información es suficiente para chequear el criterio de optimalidad y si no es óptimo decidir entonces la iteración *simplex*.

3.4.4.2 Función objetivo no lineal

El procedimiento de Pedersen trabaja iterativamente hacia la mejor de las soluciones vecinas. Una solución vecina es aquella donde sólo una variable básica es intercambiada con una variable no básica.

El criterio de optimalidad es que la función objetivo es menor que la función objetivo para todas las soluciones vecinas. Con una función objetivo no-lineal se tiene que calcular la función objetivo directamente, y no sólo los gradientes para la solución básica.

La función objetivo para una solución básica es

$$\tilde{f} = f(\tilde{\mathbf{b}}) = \sum_{\tilde{b}_j > 0} f_j^+(\tilde{b}_j) + \sum_{\tilde{b}_j < 0} f_j^-(-\tilde{b}_j) \quad (3.34)$$

y la función objetivo para las variables básicas cambiadas cuando x_i^+ es introducido es

$$f(\tilde{\mathbf{b}} - \mathbf{Z}_i x_i^+) = \sum_{\tilde{b}_j > 0} f_m^+(\tilde{b}_j - (z_j)_i x_i^+) + \sum_{\tilde{b}_j < 0} f_j^-(-\tilde{b}_j + (z_j)_i x_i^+) \quad (3.35)$$

El valor de chequeo total para la introducción de x_i^+ es

$$\Delta f = f_i^+(x_i^+) + f(\tilde{\mathbf{b}} - \mathbf{Z}_i x_i^+) - \tilde{f} \quad (3.36)$$

Se localiza el mínimo Δf para todas las x_i^+ y x_i^- . Si el mínimo es $\Delta f < 0$ se acepta el intercambio; si el mínimo es $\Delta f > 0$ se tendrá un óptimo local; por último, si el mínimo es $\Delta f = 0$ se produce la degeneración y hay que tomar decisiones de tipo heurístico. Pedersen (1994) no comenta cómo evitar la degeneración.

3.4.4.3 Implementación

Se parte de un universo estructural, o de una estructura hiperestática en general, de donde hay que seleccionar un subconjunto de barras que formen una topología isostática, que será la topología inicial con la que comienza el procedimiento de optimización.

En cada optimización se modifica la topología, intercambiando una de las barras básicas (las que forman parte de la topología) por una de las barras no básicas (las que no forman parte de la topología), de forma que el incremento de masa de la estructura sea mínimo (y negativo).

La estrategia para localizar las barras que producen un incremento de masa mínimo al intercambiarlas es probar todas las posibilidades e ir guardando aquellos intercambios cuyo incremento de masa es menor que el incremento mínimo de los probados. Sólo al final se intercambian las barras.

El criterio de optimalidad será que el incremento de masa mínimo sea positivo, es decir, cuando todos los intercambios de barras activas por barras no activas producen un

incremento positivo de la masa, se detiene el proceso, ya que se ha alcanzado un mínimo local.

3.4.4.3.1 Generación automática de la topología inicial

Para la resolución del *simplex* modificado en el método de Pedersen hay que partir de una estructura isostática inicial. Pedersen (1993) propone que sea el usuario el que la facilite.

Smith (1997) propone un método para la generación automática de topologías iniciales. Este método está basado en los generadores de malla del MEF y consiste en generar una malla de cuadriláteros (hexaedros en estructuras espaciales) en cuyas aristas se situarán las barras. Para convertir el mecanismo resultante en estructura isostática habrá que añadir una barra en una de las diagonales de cada cuadrilátero (cuatro en las de los hexaedros).

En la presente tesis la topología inicial se genera automáticamente. Se comienza situando los nudos fijos y el resto se construye por triangulación. Para establecer el orden de triangulación se elige el nudo que esté conectado mediante barras (2 en estructuras planas y 3 en las espaciales) a los nudos a los que menos barras lleguen. Una vez que se hayan incluido todos los nudos se habrá formado una topología isostática. En cada triangulación se debe comprobar que no hay inestabilidad (las barras de estructuras planas no deben estar alineadas y las de estructuras espaciales no deben ser coplanarias).

3.4.4.3.2 Varios estados de cargas

Para tratar varios estados de cargas, mediante el método de Pedersen, Smith (1997) escoge como área de cada barra la mayor de las correspondientes a cada estado de cargas individual.

3.4.4.3.3 Degeneración del *simplex*

El método *simplex* consiste, básicamente, en intercambiar variables básicas (barras que forman parte de la topología) con variables no básicas (barras que no forman parte de la topología) de forma que el decremento de la función objetivo sea lo mayor posible. Cuando este decremento máximo es negativo termina la optimización (todos los intercambios hacen aumentar la función objetivo). Si el decremento máximo es positivo, la optimización debe continuar (algún intercambio hace disminuir la función objetivo). Pero cuando el decremento máximo es nulo (en optimización de topología se da con mucha frecuencia) si se intercambian las barras puede que se produzca un bucle infinito al volver en el siguiente intercambio a la topología anterior y si no se intercambian puede que no se llegue al óptimo. La situación se complica si hay más de un intercambio que provoque decremento nulo (también habitual en optimización de topología). Esta situación es la degeneración en el *simplex* modificado. Pedersen (1993) no propone cómo evitar dicha degeneración.

En el presente trabajo, para evitar la degeneración, se aplican cargas aleatorias muy pequeñas en todos los grados de libertad de la estructura. De esta forma, la función objetivo siempre va a cambiar al sustituir una variable básica por otra no básica.

3.4.5 Método de Achtziger

3.4.5.1 Formulación

El método de Achtziger (1999) es un algoritmo para la optimización de topología de estructuras articuladas sometidas a un estado de cargas y que considera restricciones de tensión, de pandeo y de esbeltez de los elementos de la estructura.

Sea un universo estructural G en \mathfrak{R}^d (con $d = 2$ para estructuras planas y $d = 3$ para estructuras espaciales) consistente en N puntos nodales y m barras potenciales, representado por las posiciones nodales $\mathcal{G} = \{v_1, \dots, v_N\} \subset \mathfrak{R}^d$, el conjunto $\mathcal{E} = \{e_1, \dots, e_m\}$ definiendo las conexiones, y las coordenadas nodales s donde los desplazamientos son fijos (apoyos). La barra potencial i es interpretada como el conjunto de sus dos nodos, $e_i = \{v_{k_{i,1}}, v_{k_{i,2}}\}$ con $v_{k_{i,1}}, v_{k_{i,2}} \in \mathcal{G}$, $v_{k_{i,1}} \neq v_{k_{i,2}}$. Su longitud es el valor $L_i = \|v_{k_{i,1}} - v_{k_{i,2}}\|_2$. El número de grados de libertad es $n = Nd - s$.

El problema de optimización de topología de estructuras articuladas considerando restricciones de tensión y de pandeo de elementos, y como variables las áreas y los esfuerzos axiales de las barras es el siguiente

$$\begin{aligned}
 & \underset{\mathbf{a}, \mathbf{q}}{\text{minimizar}} : && \mathbf{c}^T \mathbf{a} \\
 & \text{sujeto a} : && \mathbf{R} \mathbf{q} = \mathbf{f} \\
 & && -\sigma_i^C A_i \leq q_i \leq \sigma_i^T A_i \quad \forall i = 1 : m \\
 & && q_i \geq -\frac{s_i}{L_i^2} A_i^2 \quad \forall i = 1 : m \\
 & && A_i \geq 0 \quad \forall i = 1 : m
 \end{aligned} \tag{3.37}$$

siendo:

- \mathbf{c} Vector de costes por unidad de área de las barras ($\mathbf{c} \in \mathfrak{R}^m$).
- \mathbf{a} Vector de áreas de las barras ($\mathbf{a} \in \mathfrak{R}^m$).
- \mathbf{q} Vector de esfuerzos axiales de las barras ($\mathbf{q} \in \mathfrak{R}^m$).
- \mathbf{R} Matriz de cosenos directores ($\mathbf{R} \in \mathfrak{R}^{n \times m}$).
- \mathbf{f} Vector de fuerzas externas ($\mathbf{f} \in \mathfrak{R}^n$).
- σ_i^C Límite elástico a compresión.
- σ_i^T Límite elástico a tracción.
- s_i $\pi^2 E_i \beta_i$.
- E_i Módulo de Young de la barra i .
- β_i I_i / A_i^2 .
- A_i Área de la barra i .

- I_i Momento de inercia de la barra i .
- L_i Longitud de la barra i .
- m Número de barras.

El problema (3.37) no es suficiente para propósitos realistas en un contexto de topología. Considérese una secuencia, aquí llamada “cadena”, de barras comprimidas alineadas. Si todas las áreas de las barras no pertenecientes a la cadena, pero que están conectadas a los nudos intermedios de la cadena, son nulas, entonces, la longitud de pandeo de las barras de la cadena debe ser reemplazada por la suma de las longitudes de las barras de la cadena.

Para un universo estructural G , una secuencia $c = (e_{i_1}, \dots, e_{i_K})$ de $K \geq 2$ barras potenciales en ε es llamada “una cadena en G ”, si las siguientes condiciones son satisfechas: todas las barras e_{i_1}, \dots, e_{i_K} en c están alineadas y las barras secuenciales tienen un nudo en común ($e_{i_k} \cap e_{i_{k+1}} \neq \emptyset \quad \forall k = 1, \dots, K - 1$).

El conjunto de todas las cadenas en G es llamado C . Para cada $c = (e_{i_1}, \dots, e_{i_K}) \in C$, se usa la notación $\varepsilon(c) = \{e_{i_1}, \dots, e_{i_K}\} \subset \varepsilon$, $i(c) = i_1$, y $L(c) = \sum_{i: e_i \in \varepsilon(c)} L_i = \sum_{k=1}^K L_{i_k}$.

El conjunto de nudos intermedios de c es dado por el conjunto $\Gamma(c) = \bigcup_{k=1}^{K-1} (e_{i_k} \cap e_{i_{k+1}})$.

En lo que sigue se asume que ε no contiene barras superpuestas.

La longitud de pandeo $L(c)$ tiene que reemplazar a L_i en las restricciones de pandeo del problema (3.31) si todas las barras en la cadena c están sometidas a compresión, y si todas las barras conectadas a los nudos de $\Gamma(c)$ tienen área nula. En este caso se habla de “cadenas activas”. Esta propiedad depende de \mathbf{a} , es decir, de que la suma de las áreas de las barras de todas las barras desde fuera de la cadena c a los nudos intermedios $\mathbf{v} \in \Gamma(c)$ sea cero o un valor positivo. Esta suma es igual a $\mathbf{r}_c^T \mathbf{a}$ donde el

vector $\mathbf{r}_c \in \mathfrak{R}^m$ es definido como $[\mathbf{r}_c]_i = \begin{cases} 1 & \text{si } e_i \notin \varepsilon(c) \text{ y } e_i \cap \Gamma(c) \neq \emptyset \\ 0 & \text{en caso contrario} \end{cases}$.

Finalmente, se introduce la notación $C_i^0(\mathbf{a}) = \{c \in C \mid \mathbf{r}_c^T \mathbf{a} = 0, \quad e_i \in \varepsilon(c), A_i > 0\}$, $\forall i = 1, \dots, m$ y $C^0(\mathbf{a}) = \bigcup_i C_i^0(\mathbf{a})$. El conjunto $C_i^0(\mathbf{a})$ contiene todas las cadenas activas.

En contraste con la formulación del problema (3.37), el pandeo se denomina pandeo simple cuando no se tienen en cuenta las cadenas activas, y pandeo topológico cuando sí se tienen en cuenta.

Estas cadenas activas provocan una discontinuidad en la función objetivo si no hay restricciones de área mínima. Para evitar esta discontinuidad se propone añadir restricciones de esbeltez al problema. Si el área de una barra A_i es positiva (no nula) entonces debe ser mayor o igual que un área mínima predefinida $\varepsilon_i > 0$. Esta restricción

no permite que el área sea nula, con lo que se elimina el aspecto topológico del problema. Para solventar este problema las restricciones de esbeltez pueden rescribirse de la forma $A_i^2 \geq \varepsilon_i A_i$, $\forall i = 1, \dots, m$.

Teniendo en cuenta el pandeo topológico comentado, el problema (3.37) se convierte en el siguiente

$$\begin{aligned}
 & \underset{\mathbf{a}, \mathbf{q}}{\text{minimizar :}} && \mathbf{c}^T \mathbf{a} \\
 & \text{sujeto a :} && \mathbf{R} \mathbf{q} = \mathbf{f} \\
 & && -\sigma_i^C A_i \leq q_i \leq \sigma_i^T A_i \quad \forall i = 1 : m \\
 & && q_i \geq -\frac{s_i}{L_i^2} A_i^2 \quad \forall i = 1 : m \\
 & && q_i(c) \geq -\frac{s_i(c)}{L(c)^2} A_i^2(c) \quad \forall c \in C^0(\mathbf{a}) \\
 & && A_i(c) \leq A_i \quad \forall c \in C^0(\mathbf{a}) \quad \forall i \neq i(c) : e_i \in \mathcal{E}(c) \\
 & && A_i^2 \geq \varepsilon_i A_i \quad \forall i = 1 : m \\
 & && A_i \geq 0 \quad \forall i = 1 : m
 \end{aligned} \tag{3.38}$$

3.4.5.2 Algoritmo de resolución

El problema (3.38) no está definido de forma apropiada para un tratamiento numérico directo mediante algoritmos de programación no lineal estándar, puesto que $C^0(\mathbf{a})$ depende de \mathbf{a} , es decir, el conjunto de restricciones depende de las variables.

Para obtener una formulación equivalente en forma estándar, Achtziger propone una aproximación del problema en el cual la actividad de las cadenas es controlada por un parámetro. Para cada $c \in C$ se elige un parámetro de aproximación $P_c \in \mathfrak{R}$, $P_c \gg 0$, con el que las restricciones de pandeo de las cadenas se convierten en

$$q_i(c) \geq -\frac{s_i(c)}{l(c)^2} a_i^2(c) - P_c \mathbf{r}_c^T \mathbf{a}, \quad \forall c \in C$$

Análogamente

$$a_i(c) \leq a_i + P_c \mathbf{r}_c^T \mathbf{a} \begin{cases} \forall c \in C \text{ y} \\ \forall i \neq i(c) \\ \text{con } e_i \in \mathcal{E}(c) \end{cases}$$

Eligiendo el mismo parámetro $P_c = P \gg 0$, $\forall c \in C$, el problema (3.38) se reduce a

$$\begin{aligned}
 &\text{minimizar :} && \mathbf{c}^T \mathbf{a} \\
 &\text{a, q} \\
 &\text{sujeto a :} && \mathbf{R} \mathbf{q} = \mathbf{f} \\
 &&& -\sigma_i^C A_i \leq q_i \leq \sigma_i^T A_i \quad \forall i = 1 : m \\
 &&& q_i \geq -\frac{s_i}{L_i^2} A_i^2 \quad \forall i = 1 : m \\
 &&& q_i(c) \geq -\frac{s_i(c)}{L(c)^2} A_i^2(c) - \mathbf{Pr}_c^T \mathbf{a} \quad \forall c \in C \\
 &&& A_i(c) \leq A_i + \mathbf{Pr}_c^T \mathbf{a} \quad \forall c \in C \quad \forall i \neq i(c) : e_i \in \varepsilon(c) \\
 &&& A_i^2 \geq \varepsilon_i A_i \quad \forall i = 1 : m \\
 &&& A_i \geq 0 \quad \forall i = 1 : m
 \end{aligned} \tag{3.39}$$

Achtziger demuestra que, si P es elegido lo suficientemente grande, el problema (3.39) es equivalente al problema (3.38).

La principal dificultad para la resolución del problema (3.39) es la concavidad de las restricciones de pandeo y las condiciones de esbeltez. Este problema es casi un problema de programación lineal, excepto por el término A_i^2 . Achtziger propone una técnica, similar a la linealización, en la cual se reemplazan los términos A_i^2 por m nuevas variables \tilde{A}_i con lo que se obtiene la siguiente formulación

$$\begin{aligned}
 &\text{minimizar :} && \mathbf{c}^T \mathbf{a} \\
 &\text{a, \tilde{a}, q} \\
 &\text{sujeto a :} && \mathbf{R} \mathbf{q} = \mathbf{f} \\
 &&& -\sigma_i^C A_i \leq q_i \leq \sigma_i^T A_i \quad \forall i = 1 : m \\
 &&& q_i \geq -\frac{s_i}{L_i^2} \tilde{A}_i \quad \forall i = 1 : m \\
 &&& q_i(c) \geq -\frac{s_i(c)}{L(c)^2} \tilde{A}_i(c) - \mathbf{Pr}_c^T \mathbf{a} \quad \forall c \in C \\
 &&& A_i(c) \leq A_i + \mathbf{Pr}_c^T \mathbf{a} \quad \forall c \in C \quad \forall i \neq i(c) : e_i \in \varepsilon(c) \\
 &&& \tilde{A}_i \geq \varepsilon_i A_i \quad \forall i = 1 : m \\
 &&& A_i \geq 0 \quad \forall i = 1 : m \\
 &&& \sqrt{\tilde{A}_i} - A_i \leq 0 \quad \forall i = 1 : m
 \end{aligned} \tag{3.40}$$

Las únicas restricciones no lineales son las últimas (por la raíz cuadrada). Este problema se resuelve mediante programación lineal secuencial linealizando la raíz cuadrada mediante la expansión de Taylor en algún punto fijo $A_i' > 0$

$\sqrt{\tilde{A}_i} = \sqrt{A_i'} + \frac{1}{2\sqrt{A_i'}} (\tilde{A}_i - A_i') + \Delta$. Despreciando Δ se obtiene la restricción

$\tilde{A}_i - 2\sqrt{A_i'} A_i \leq -A_i'$, y sustituyéndola en el problema (3.40), se obtiene

$$\begin{aligned}
& \underset{\mathbf{a}, \tilde{\mathbf{a}}, \mathbf{q}}{\text{minimizar}} : && \mathbf{c}^T \mathbf{a} \\
& \text{sujeto a :} && \mathbf{R} \mathbf{q} = \mathbf{f} \\
& && -\sigma_i^C A_i \leq q_i \leq \sigma_i^T A_i \quad \forall i = 1 : m \\
& && q_i \geq -\frac{s_i}{L_i^2} \tilde{A}_i \quad \forall i = 1 : m \\
& && q_i(c) \geq -\frac{s_i(c)}{L(c)^2} \tilde{A}_i(c) - \text{Pr}_c^T \mathbf{a} \quad \forall c \in C \quad (3.41) \\
& && A_i(c) \leq A_i + \text{Pr}_c^T \mathbf{a} \quad \forall c \in C \quad \forall i \neq i(c) : e_i \in \varepsilon(c) \\
& && \tilde{A}_i \geq \varepsilon_i A_i \quad \forall i = 1 : m \\
& && A_i \geq 0 \quad \forall i = 1 : m \\
& && \tilde{A}_i - 2\sqrt{A_i} A_i \leq -A_i \quad \forall i = 1 : m
\end{aligned}$$

El punto de linealización \mathbf{a}' es un valor fijo de la variable $\tilde{\mathbf{a}}$. Así, si $(\mathbf{a}^*, \tilde{\mathbf{a}}^*, \mathbf{q}^*)$ es una solución de (3.41) entonces es bastante natural linealizar (3.40) en el punto $\tilde{\mathbf{a}}^*$ en la siguiente iteración. El proceso general del algoritmo de programación lineal secuencial para resolver el problema (3.40) es el siguiente:

- 1 Obtener una solución $(\mathbf{a}^{k*}, \tilde{\mathbf{a}}^{k*}, \mathbf{q}^{k*})$ al problema (3.35) para $\mathbf{a}' = \tilde{\mathbf{a}}^{k*}$.
- 2 Actualizar el punto de linealización $\tilde{\mathbf{a}}^k$ de la forma $\tilde{\mathbf{a}}^{k+1} = \tilde{\mathbf{a}}^{k*}$, $k = k + 1$.
- 3 Si el decremento relativo de la función objetivo en la última iteración es menor que un valor predeterminado detener el proceso, en caso contrario volver al paso 1.

3.4.5.3 Implementación

Para computar una solución al problema (3.41) en cada iteración, se ha utilizado la subrutina DDLPRS de la librería IMSL que resuelve problemas de programación lineal mediante el algoritmo *simplex* revisado.

3.4.6 Método de búsqueda exhaustiva

3.4.6.1 Introducción

El método de búsqueda exhaustiva es un procedimiento para la optimización de topología de estructuras espaciales de barras articuladas sometidas a un estado de cargas y que considera restricciones de tensión, de pandeo, de esbeltez y de área mínima. También tiene en cuenta las cadenas activas.

Este método consiste en considerar todas las soluciones posibles. Dado que la solución es isostática, por considerar un solo estado de cargas, el problema consiste en considerar todas las topologías isostáticas que se pueden formar con las barras del universo estructural, y elegir aquella topología con la que se obtenga una menor función objetivo.

3.4.6.2 Procedimiento

3.4.6.2.1 Generación de topologías

Si m es el número de barras y n es el número de grados de libertad de la estructura, el número total de topologías con n barras posibles, que incluyan todos los nudos, es igual a las combinaciones de m elementos tomados de n en n , esto es, $C_{m,n} = \frac{m!}{n!(m-n)!}$.

Para la generación de las topologías se utiliza un *bit* por cada barra, que vale 1 cuando forma parte de la topología (barra activa) y 0 cuando no forma parte de ella (barra no activa). En la Fig. 3.11 se muestra un universo estructural de 15 barras y 12 grados de libertad, y en la Fig. 3.12 la topología de 12 barras activas correspondiente a la secuencia de bits siguiente

barra	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
bit	1	0	1	0	1	0	1	1	1	1	1	1	1	1	1

En las Figs. 3.11 a 3.16 se representan con trazo grueso continuo las barras activas, con trazo fino discontinuo las barras no activas, y con trazo fino continuo las barras que pueden ser activas o no, de forma que el número total de barras activas sea n .

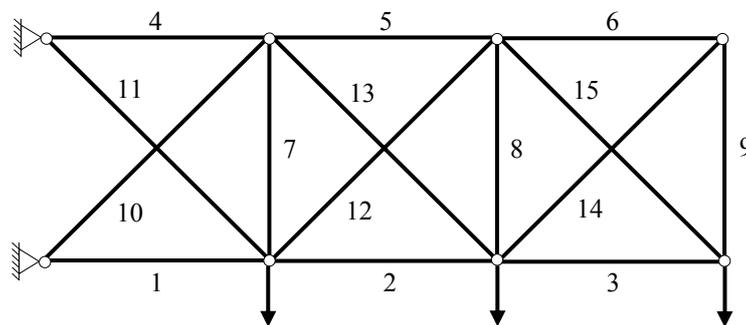


Figura 3.11 Universo estructural de 15 barras y 12 grados de libertad

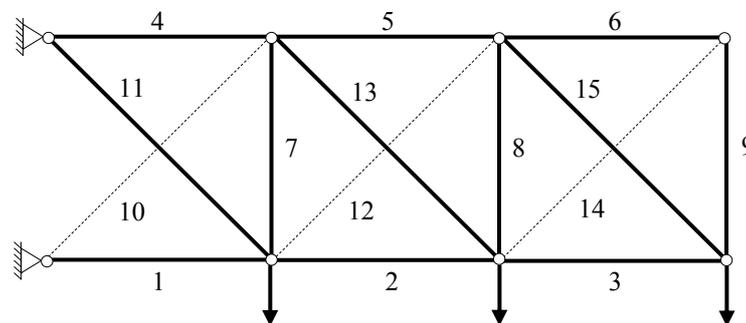


Figura 3.12 Universo estructural de 15 barras y 12 grados de libertad.
Topología isostática (“101010111111111”)

Para generar todas las topologías posibles con n barras se parte de la topología con las n primeras barras activas, y se van intercambiando barras activas por barras no

activas de forma que se cubran todas las combinaciones posibles. Para el ejemplo anterior, las primeras y las últimas topologías son

```

0001111111111111
0011111111111110
0011111111111101
...
111111111110010
111111111110100
111111111111000
    
```

De este conjunto de topologías hay que eliminar las topologías correspondientes a estructuras estáticamente inestables y las topologías que no incluyen todos los nudos. La inestabilidad estática de una topología se puede determinar comprobando la singularidad de la matriz de rigidez correspondiente; sin embargo, dado que el número de topologías inestables suele ser elevado, el coste computacional es muy alto. Por ello, se recurre a otros procedimientos que no requieren analizar la singularidad de la matriz de rigidez.

3.4.6.2.2 Detección de topologías inestables

Se puede detectar la inestabilidad de algunas topologías sin necesidad de analizar la singularidad de la matriz de rigidez.

Las topologías que incluyen algún nudo que no tiene el número de vínculos independientes suficientes (2 en el plano y 3 en el espacio) son inestables. Se determina, para cada nudo, las combinaciones de barras no activas que hacen que el número de vínculos independientes en ese nudo sea insuficiente. Todas la topologías que incluyen esa combinación de barras no activas son inestables. En la Fig. 3.13 se muestra un ejemplo con inestabilidad por falta de vínculos independientes. Todas las topologías que no incluyen las barras 6 y 9 son inestables. Estas topologías se representan con la secuencia (“xxxxxx0xx0xxxxx”), en la que las “x” se refieren a cualquier valor (0 ó 1) siempre que el número total de barras activas sea 12.

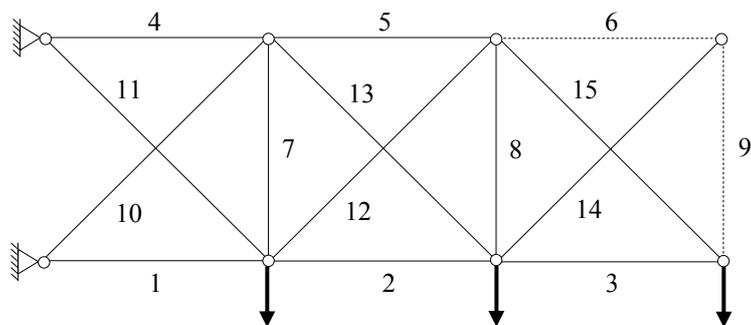


Figura 3.13 Universo estructural de 15 barras y 12 grados de libertad. Topologías inestables por falta de vínculos independientes (“xxxxxx0xx0xxxxx”)

Cuando hay una hiperestaticidad local, al ser el número de barras igual al número de grados de libertad, debe haber una inestabilidad local (si en una parte sobran barras, en otra deben faltar). Se determinan los bloques de 4 nudos (para estructuras planas) entre los que haya al menos 6 barras (5 si dos de los nudos son fijos); estos bloques son hiperestáticos. Todas las topologías que tengan activas al menos 6 barras de cada bloque tienen una hiperestaticidad local. En la Fig. 3.14 se muestra un ejemplo de inestabilidad por hiperestaticidad local; todas las topologías que incluyan las barras 1, 4, 7, 10 y 11 (“xxxx11xx1xx1”) son inestables.

Se pueden utilizar algoritmos de detección de inestabilidades más sofisticados, pero puede ocurrir que el coste de la detección sea superior al de la comprobación de la singularidad de la matriz de rigidez.

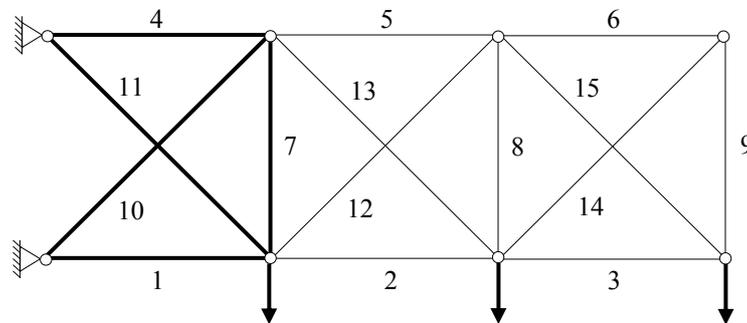


Figura 3.14 Universo estructural de 15 barras y 12 grados de libertad. Topologías inestables por hiperestaticidad local (“xxxx11xx1xx1”)

Una vez detectadas las topologías inestables más sencillas, se comprueba la singularidad de las restantes, factorizando la matriz de rigidez, para eliminar las topologías inestables no detectadas. Las topologías restantes forman el conjunto de topologías isostáticas a considerar.

3.4.6.2.3 Detección de topologías equivalentes

Si a un nudo de una estructura plana (espacial), en el que no hay apoyos ni cargas, llegan sólo dos (tres) barras activas, el esfuerzo axial de estas barras es nulo, por lo que se considera que este nudo no va a formar parte de la topología óptima (el nudo está aislado, ver epígrafe 3.4.6.2.4). Se asigna un valor nulo a las áreas de las barras, aunque haya restricciones de área mínima, por lo tanto, si se intercambia una de estas barras activas por cualquiera de las barras inactivas que lleguen al mismo nudo, la función objetivo será la misma, es decir, las topologías son equivalentes. Por lo tanto, sólo es necesario estudiar una de ellas.

Aunque las topologías equivalentes son diferentes, una vez que se elija el diseño óptimo se eliminan los nudos aislados, con lo que las topologías equivalentes se convierten en idénticas. En la Fig. 3.15 se muestra un ejemplo de topologías equivalentes.

3.4.6.2.4 Análisis y obtención de la función objetivo

Para obtener la función objetivo de las topologías isostáticas se analiza cada una de ellas, para un valor cualquiera de las áreas (los esfuerzos son independientes de las áreas). Una vez obtenidos los esfuerzos axiales se determina el área de cada barra, que será el menor valor que cumpla todas las restricciones (de tensión, pandeo, esbeltez o área mínima). Si el esfuerzo axial de todas las barras que llegan a un nudo es nulo (nudo aislado), aunque haya restricciones de área mínima, se asigna un valor nulo a las áreas (ver epígrafe 3.4.6.2.3). Si hay alguna cadena activa se cambia la longitud de pandeo. A partir de los valores de las áreas se obtiene la función objetivo de la topología.

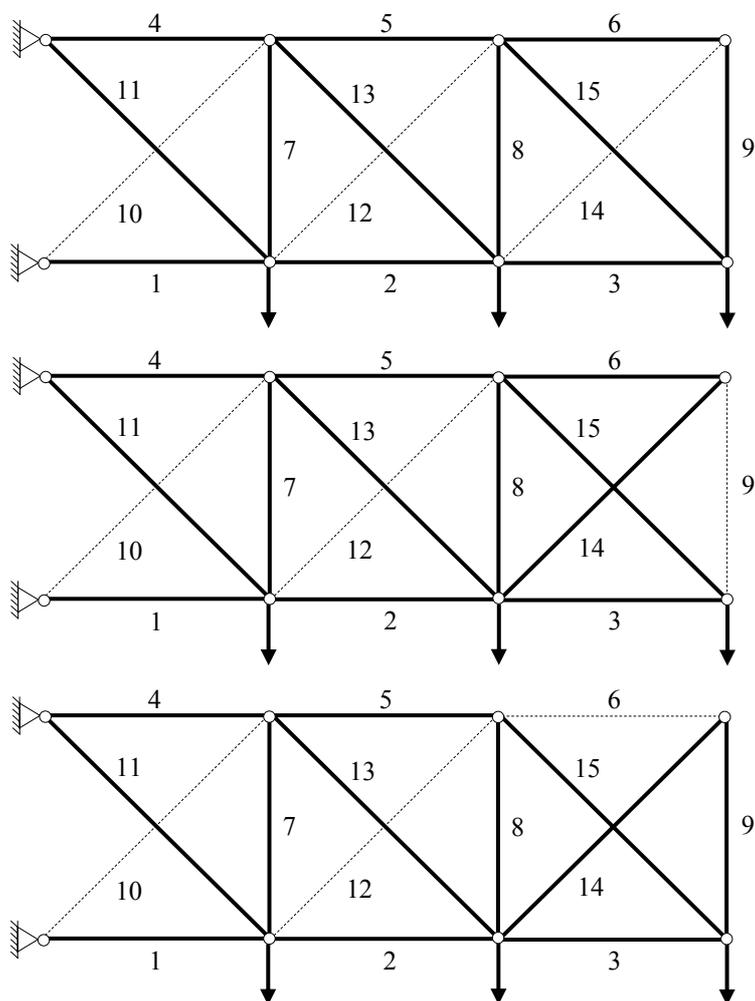


Figura 3.15 Universo estructural de 15 barras y 12 grados de libertad.
Topologías equivalentes (misma función objetivo)

3.4.6.2.5 Elección del diseño óptimo

Se elige como diseño óptimo aquel cuya topología isostática produzca una menor función objetivo. Al analizar cada una de las topologías isostáticas estables se compara con la mejor obtenida hasta ese momento y si la función objetivo es menor se almacena dicha topología.

3.4.7 Ejemplo

3.4.7.1 Introducción

Se va a resolver el ejemplo de un voladizo con carga puntual en la parte inferior del extremo libre (Achtziger, 1999b) mediante todos los métodos basados en el universo estructural implementados.

3.4.7.2 Voladizo con carga puntual

Achtziger propone el ejemplo de un voladizo con carga puntual en la parte inferior del extremo libre (Fig. 3.16), considerando restricciones de tensión, de pandeo de elementos y de área mínima (en los casos en los que se considere restricciones de pandeo).

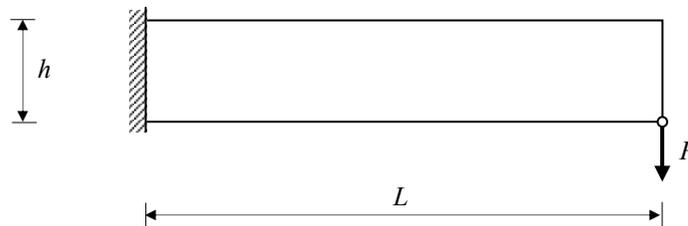


Figura 3.16 Voladizo con carga puntual en la parte inferior del extremo libre (Achtziger, 1999b)

Los datos del problema (normalizados) son: longitud $L = 10$, altura $h = 2$, módulo de Young $E = \frac{12\sqrt{2}}{\pi^2}$, densidad $\rho = 1$, límite elástico $\sigma_e = 1$ y carga puntual $P = 1$. Se consideran barras con sección transversal cuadrada y con área mínima 0,5. Las barras se dibujan a escala (en área) 1:100.

Se ha resuelto el problema, sin considerar restricciones de pandeo de elementos, con todos los métodos implementados, y con pandeo de elementos, con los métodos de Pedersen, de Achtziger y de búsqueda exhaustiva.

Cuando se consideran restricciones de pandeo de elementos, al ser el problema de optimización no convexo, los métodos de Pedersen y de Achtziger no aseguran que se obtenga el óptimo global para el universo estructural dado. Para universos estructurales muy pequeños, es aplicable el método de búsqueda exhaustiva, que es el único método con el que se obtiene siempre el óptimo global antes de optimizar geometría (después de optimizar geometría tampoco se puede asegurar que sea el óptimo global) para el universo estructural dado.

Se han elegido los universos estructurales correspondientes a 1, 2, 3, 4, 5 y 6 divisiones verticales. Para que las celdas del universo estructural sean cuadradas el número de divisiones en dirección horizontal debe ser cinco veces el número de divisiones vertical. El orden de vecindad no se ha limitado, excepto para el método de búsqueda exhaustiva, donde se adopta orden de vecindad 1 para que el coste computacional no sea excesivo, y para cuando se considera pandeo de elementos (la no

convexidad del problema dificulta la obtención del mínimo global tanto más cuanto mayor sea el tamaño del universo).

De los universos universales elegidos, para cada método, se muestran aquellos cuyo coste computacional no es muy alto.

En los universos estructurales la primera cifra se refiere al número de divisiones en dirección horizontal, la segunda al número de divisiones en dirección vertical y la tercera el grado de vecindad, que por defecto no se limita.

En la Fig. 3.17 se muestra la evolución del diseño óptimo obtenido mediante el método *stress-ratio* y un universo estructural de 15x3 divisiones, sin considerar restricciones de pandeo de elementos. En la Fig. 3.18 y en la tabla 3.1 se muestran los diseños óptimos obtenidos, para diferentes universos estructurales, mediante el método *stress-ratio*, sin pandeo de elementos.

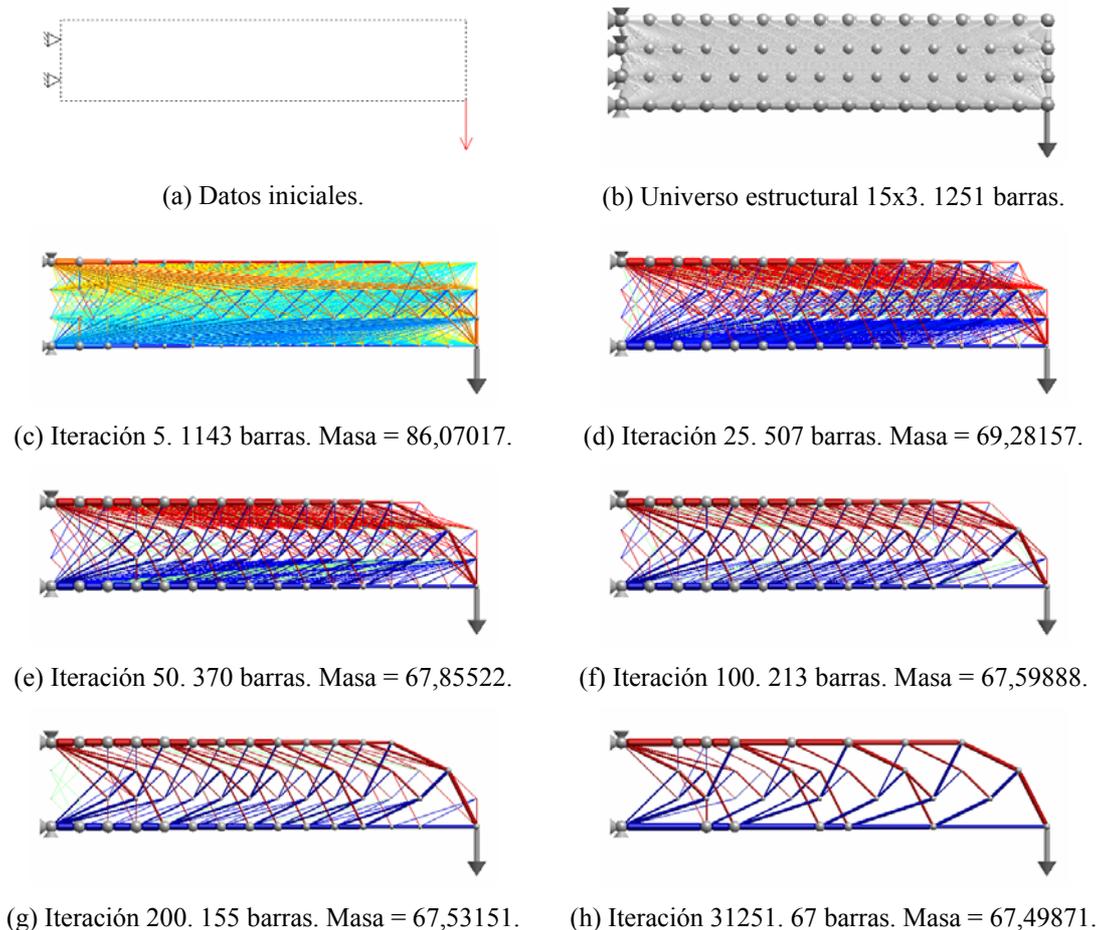


Figura 3.17 Voladizo con carga puntual en la parte inferior del extremo libre. Evolución del diseño óptimo obtenido mediante el método *stress-ratio* y un universo 15x3 (sin pandeo)

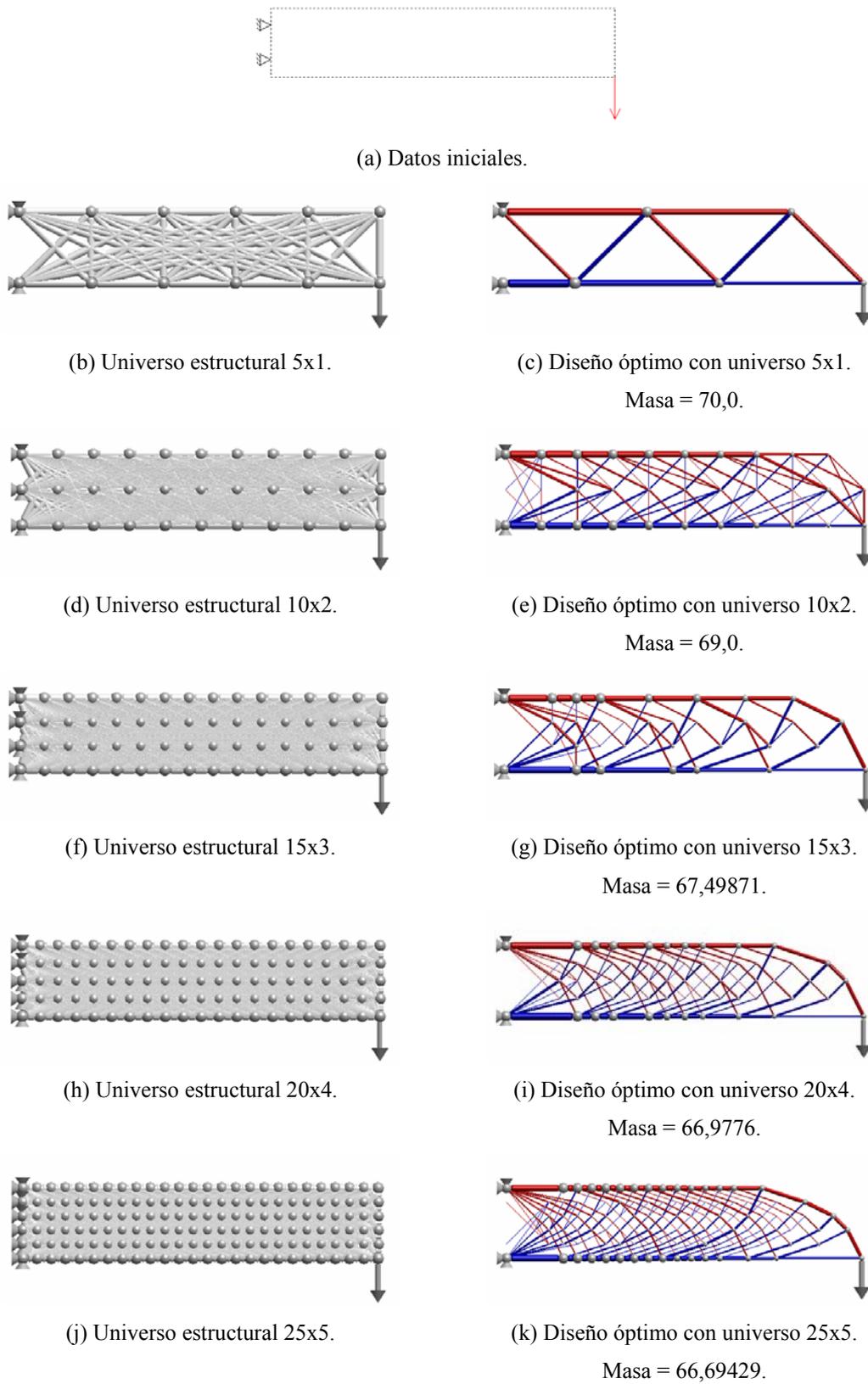


Figura 3.18 Voladizo con carga puntual en la parte inferior del extremo libre.
Diseños óptimos obtenidos mediante el método *stress-ratio* (sin pandeo)

Hay que elegir el valor del área a partir de cual se eliminan las barras. Este área de eliminación debe ser lo suficientemente pequeña para que, durante el proceso de optimización, no se eliminen barras que formen parte del óptimo, y lo suficientemente grande para no penalizar el coste computacional. Los valores elegidos se muestran en la tabla 3.1.

Tabla 3.1 Voladizo con carga puntual.
Diseños óptimos obtenidos mediante el método *stress-ratio* (sin pandeo)

div	área	nudos		barras		iter	masa (adim)	tiempo (s)
	elim	ini	fin	ini	fin			
5x1	1,00E-03	12	7	45	10	88	70,00000	0,08
10x2	1,00E-04	33	32	330	89	100	69,00000	0,44
15x3	1,00E-05	64	34	1251	67	31251	67,49871	327
20x4	1,00E-06	105	63	3328	124	2063	66,97760	56,4
25x5	1,00E-06	156	109	7443	215	8699	66,69429	747

En el método *stress-ratio* el coste computacional de cada iteración es muy bajo (sólo hay que analizar y cambiar las áreas). Sin embargo, cuando se aplica a universos estructurales, el número de iteraciones suele ser alto, sobre todo cuando alguna barra que no forma parte del óptimo está casi alineada con otras que sí forman parte del óptimo (ver tabla 3.1, universo estructural de 15x3 divisiones).

En los métodos del universo estructural es frecuente que haya problemas con más de un mínimo absoluto. El universo estructural de la Fig. 3.18d es un ejemplo de esto. En las Figs. 3.19e, 3.20e y 3.21e se muestran tres topologías diferentes con el mismo valor de la función objetivo. En estos casos, el método *stress-ratio* suele dar una solución mezcla de varias topologías óptimas (hiperestática) pero con el mismo valor de la función objetivo.

Rozvany (2001c) examina la validez del método *stress-ratio* aplicado a estructuras articuladas y encuentra que a veces se obtienen soluciones significativamente no óptimas.

En la Fig. 3.19 y en la tabla 3.2 se muestran los diseños óptimos obtenidos, para diferentes universos estructurales, mediante el método de la *compliance*, sin considerar restricciones de pandeo de elementos. No se muestra la evolución del proceso de optimización porque hasta que no se termina de resolver el *simplex* no hay solución intermedia.

El método de la *compliance*, en su formulación con los desplazamientos como únicas variables, tiene la gran ventaja de, al ser el número de variables bajo, tener un coste computacional bajo (comparado con los demás métodos). Como contrapartida, sólo admite restricciones de tensión.

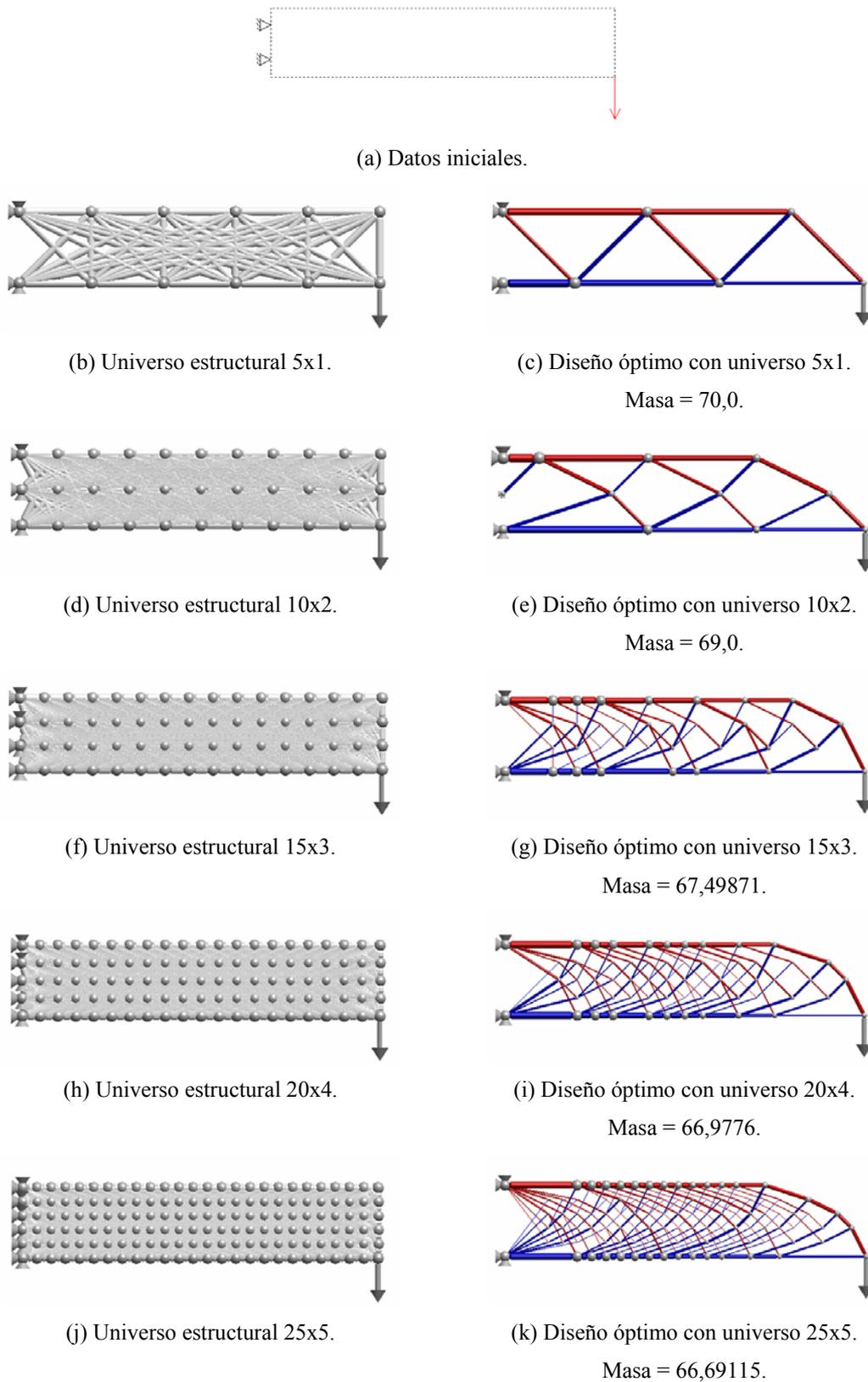


Figura 3.19 Voladizo con carga puntual en la parte inferior del extremo libre. Diseños óptimos obtenidos mediante el método de la *compliance* (sin pandeo)

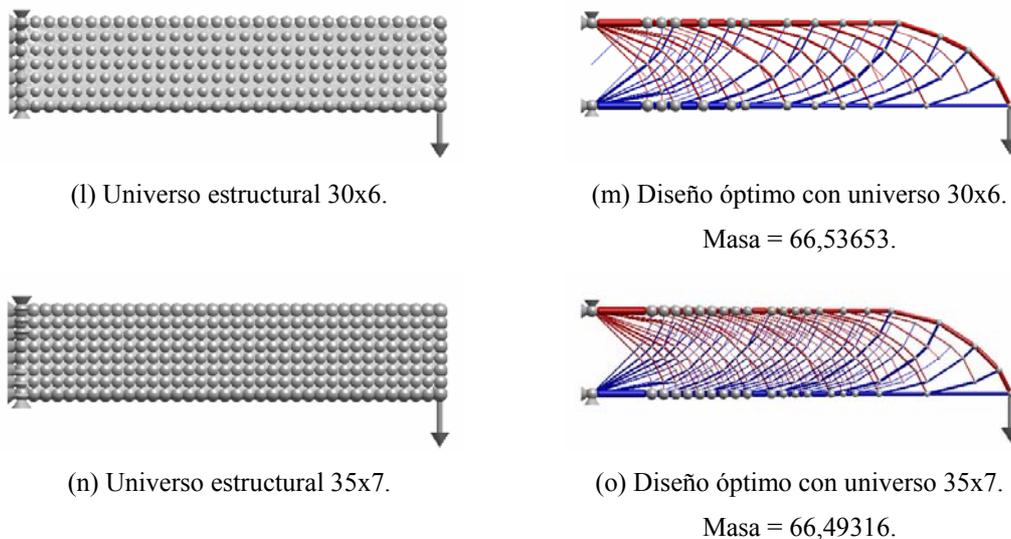


Figura 3.19 Voladizo con carga puntual en la parte inferior del extremo libre. Diseños óptimos obtenidos mediante el método de la *compliance* (sin pandeo). (Continuación)

Tabla 3.2 Voladizo con carga puntual.

Diseños óptimos obtenidos mediante el método de la *compliance* (sin pandeo)

div	nudos		barras		masa (adim)	tiempo (s)
	ini	fin	ini	fin		
5x1	12	7	45	10	70,00000	0,013
10x2	33	12	330	18	69,00000	0,13
15x3	64	35	1251	66	67,49871	1,66
20x4	105	61	3328	118	66,97760	17,5
25x5	156	103	7443	202	66,69115	155
30x6	217	99	14226	192	66,53653	418
35x7	288	168	25227	332	66,49316	850

En la Fig. 3.20 y en la tabla 3.3 se muestran los diseños óptimos obtenidos, para diferentes universos estructurales, mediante el método de Pedersen, sin considerar restricciones de pandeo de elementos. No se muestra la evolución del proceso de optimización porque hasta que no se termina de resolver el *simplex* modificado no hay solución intermedia.

El método de Pedersen es más lento que el de la *compliance*, pero admite restricciones de pandeo de elementos. No tiene en cuenta las cadenas activas (ver epígrafe 3.4.5.1), por lo que en el caso de considerar restricciones de pandeo de elementos puede que se obtengan soluciones no válidas (barras comprimidas alineadas unidas por un nudo al que sólo llegan barras de área nula), como ocurre en el ejemplo de la Fig. 3.25c.

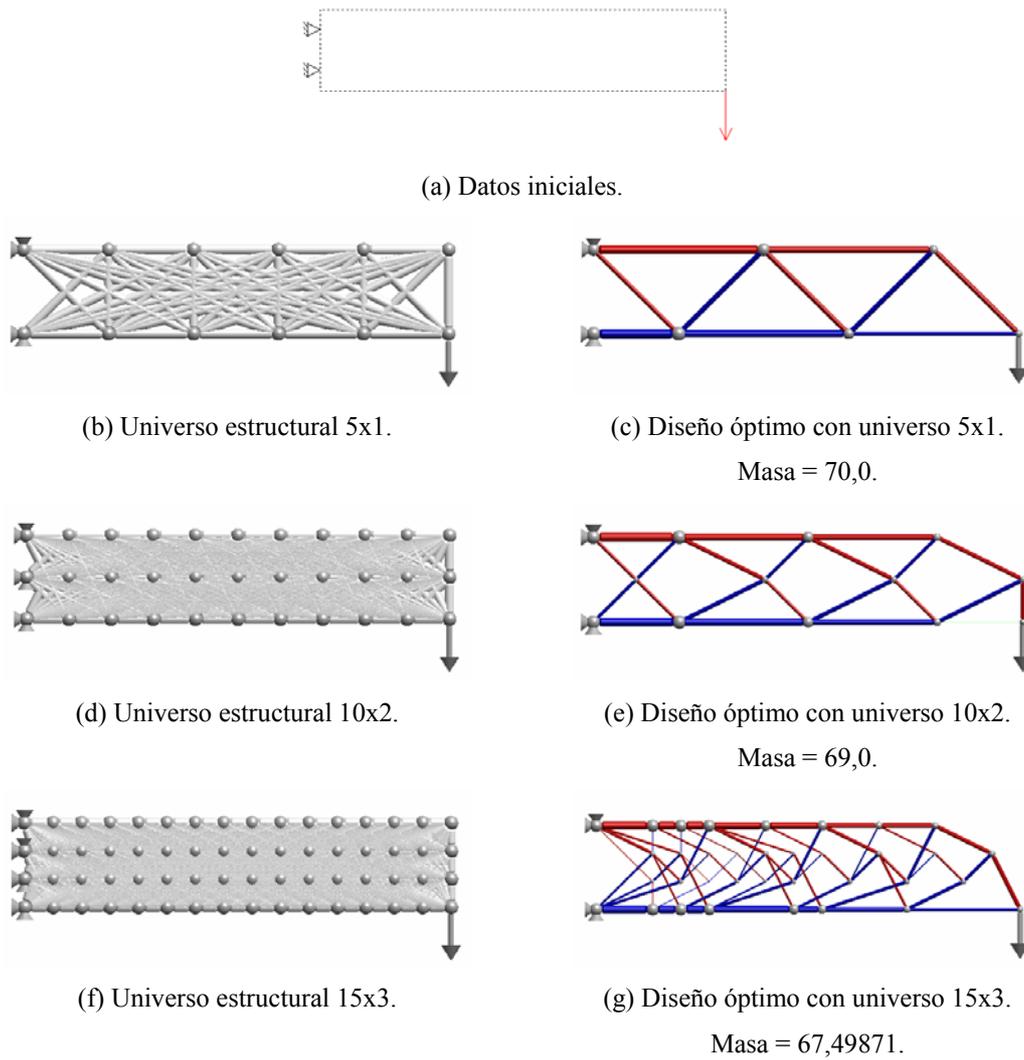


Figura 3.20 Voladizo con carga puntual en la parte inferior del extremo libre.
Diseños óptimos obtenidos mediante el método de Pedersen (sin pandeo)

Tabla 3.3 Voladizo con carga puntual.
Diseños óptimos obtenidos mediante el método de Pedersen (sin pandeo)

div	nudos		barras		masa (adim)	tiempo (s)
	ini	fin	ini	fin		
5x1	12	7	45	10	70,00000	0,02
10x2	33	13	330	22	69,00000	1,63
15x3	64	35	1251	66	67,49871	122

En la Fig. 3.21 y en la tabla 3.4 se muestran los diseños óptimos obtenidos, para diferentes universos estructurales, mediante el método de Achtziger, sin considerar restricciones de pandeo de elementos. No se muestra la evolución del proceso de optimización porque hasta que no termina el proceso de optimización no hay solución intermedia.

El método de Achtziger es más lento que el de Pedersen, pero, además de admitir restricciones de pandeo de elementos, tiene en cuenta las cadenas activas. Esto hace que sea el método más aconsejable, de los implementados, para el caso de considerar restricciones de pandeo de elementos para universos estructurales grandes.

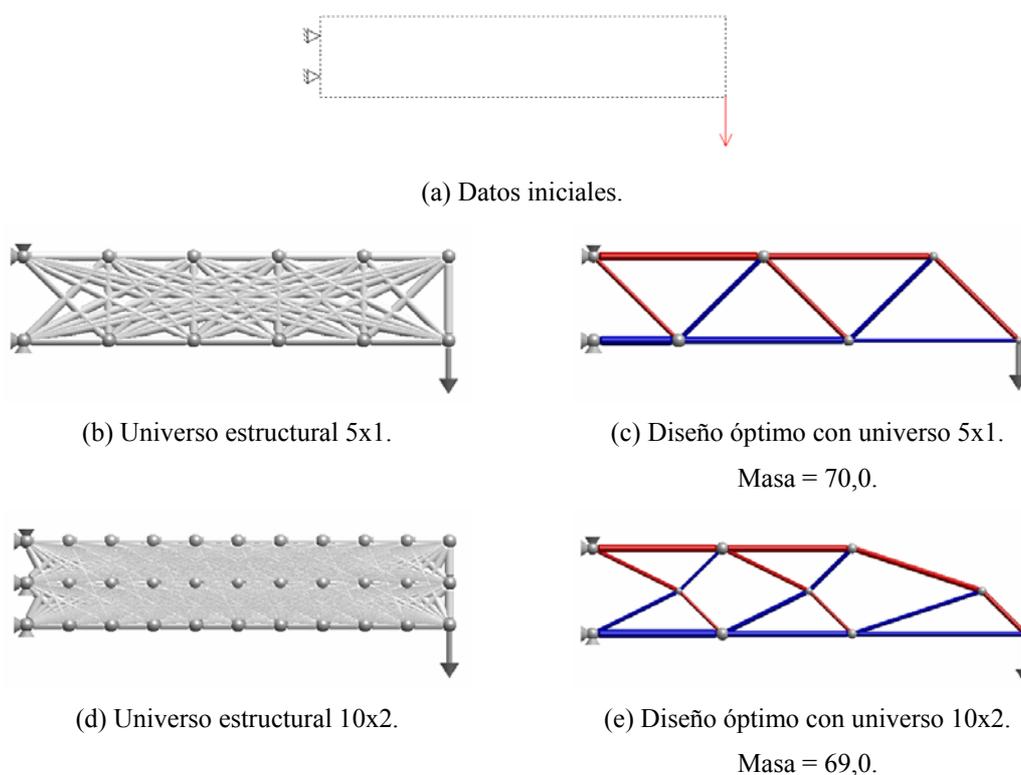


Figura 3.21 Voladizo con carga puntual en la parte inferior del extremo libre. Diseños óptimos obtenidos mediante el método de Achtziger (sin pandeo)

Tabla 3.4 Voladizo con carga puntual.

Diseños óptimos obtenidos mediante el método de Achtziger (sin pandeo)

div	nudos		barras		masa (adim)	tiempo (s)
	ini	fin	ini	fin		
5x1	12	7	45	10	70,00000	0,02
10x2	33	10	330	16	69,00000	1,63

En la Fig. 3.22 se muestra la evolución del diseño óptimo obtenido mediante el método de búsqueda exhaustiva y un universo estructural de 5x1x1 divisiones, sin considerar restricciones de pandeo de elementos. A pesar de que el universo estructural es muy pequeño, el número de combinaciones que hay que considerar es muy alto ($C_{25,20} = 53131$).

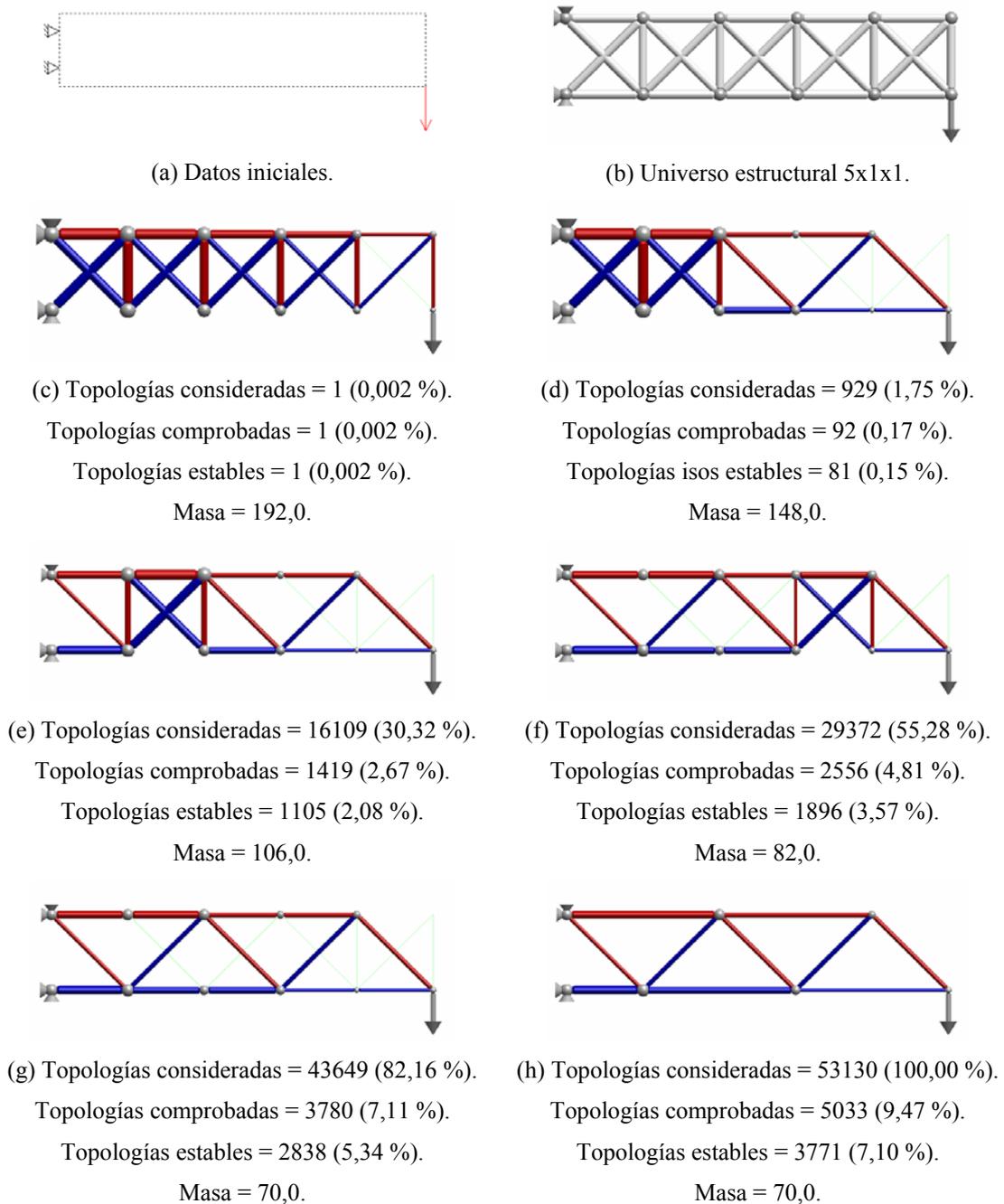


Figura 3.22 Voladizo con carga puntual en la parte inferior del extremo libre. Evolución del diseño óptimo obtenido mediante el método de búsqueda exhaustiva y un universo estructural 5x1x1 (sin pandeo)

En el diseño óptimo obtenido (Fig. 3.22h) se han eliminado las barras que no trabajan y los nudos en los que, todas las barras que concurren en él, están alineadas, uniendo dichas barras.

En la Fig. 3.23 y en la tabla 3.5 se muestran los diseños óptimos obtenidos, para diferentes universos estructurales, mediante el método de búsqueda exhaustiva, sin pandeo de elementos. Como el universo de $10 \times 2 \times 1$ divisiones tiene un coste excesivo para este método ($C_{90,60} = 6,7 \cdot 10^{23}$), se han elegido los universos estructurales $3 \times 1 \times 1$, $4 \times 1 \times 1$ y $5 \times 1 \times 1$.

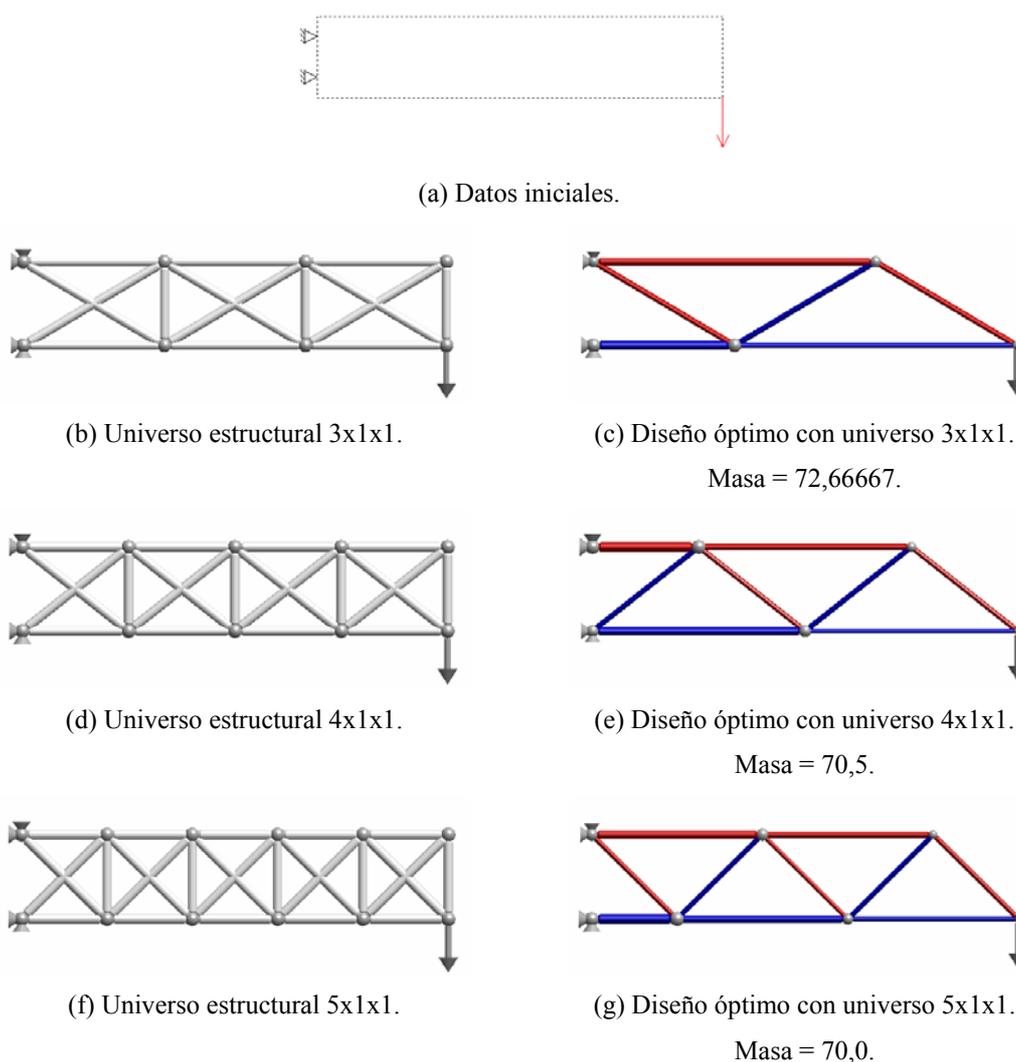


Figura 3.23 Voladizo con carga puntual en la parte inferior del extremo libre. Diseños óptimos obtenidos mediante el método de búsqueda exhaustiva (sin pandeo)

Tabla 3.5 Voladizo con carga puntual.

Diseños óptimos obtenidos mediante el método de búsqueda exhaustiva (sin pandeo)

div	nudos		barras		masa (adim)	tiempo (s)
	ini	fin	ini	fin		
3x1x1	8	5	15	6	72,66667	0,02
4x1x1	10	6	20	8	70,50000	0,07
5x1x1	12	7	25	10	70,00000	0,78

En la Fig. 3.24 se muestran los gráficos comparativos para los diferentes diseños óptimos obtenidos mediante todos los métodos del universo estructural implementados, sin considerar restricciones de pandeo de elementos. Puede apreciarse la superioridad del método de la *compliance* sobre los demás.

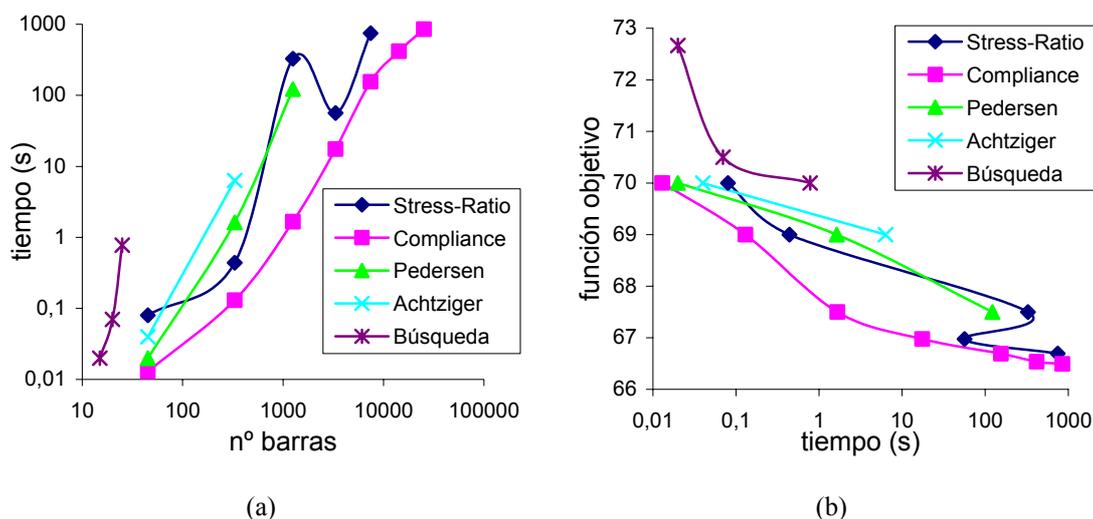


Figura 3.24 Voladizo con carga puntual en la parte inferior del extremo libre (sin pandeo).

(a) Evolución tiempo/nº barras; (b) Evolución función objetivo/tiempo

En la Fig. 3.25 se muestran los diseños óptimos obtenidos mediante los métodos de Pedersen, Achtziger y búsqueda exhaustiva, con restricciones de pandeo de elementos, sin restricciones de área mínima (primera columna) y con restricciones de área mínima (segunda columna) para el universo estructural de 5x1x1 divisiones.

Para el método de Pedersen sin restricciones de área mínima se obtiene una solución no válida, puesto que hay dos cadenas activas. Habría que eliminar las barras de área nula y duplicar las longitudes de pandeo, con lo que la función objetivo aumentaría. En cambio, si se consideran restricciones de área mínima se obtiene la solución correcta (la obtenida mediante el método de búsqueda exhaustiva).

Con el método de Achtziger ocurre lo contrario, es decir, se obtiene la solución correcta sin restricciones de área mínima, pero con restricciones de área mínima la solución es un mínimo local (debido a la no convexidad del problema).

El método de búsqueda exhaustiva obtiene los mínimos globales puesto que prueba todas las posibles topologías. En las Figs. 3.25g y 3.25h se observa que al añadir restricciones de área mínima se obtiene una solución con una función objetivo menor.

Esta contradicción es debida a que, sin restricciones de área mínima, las barras verticales no trabajan, por lo que el área asignada por el método es nula, lo que hace que aparezcan cadenas activas (el método de búsqueda exhaustiva también las considera) que hacen que la función objetivo aumente (al aumentar las longitudes de pandeo). Se obtendría una solución mejor si se añadiera una restricción que dimensionara las barras verticales de forma que fueran capaces de arriostar el nudo intermedio.

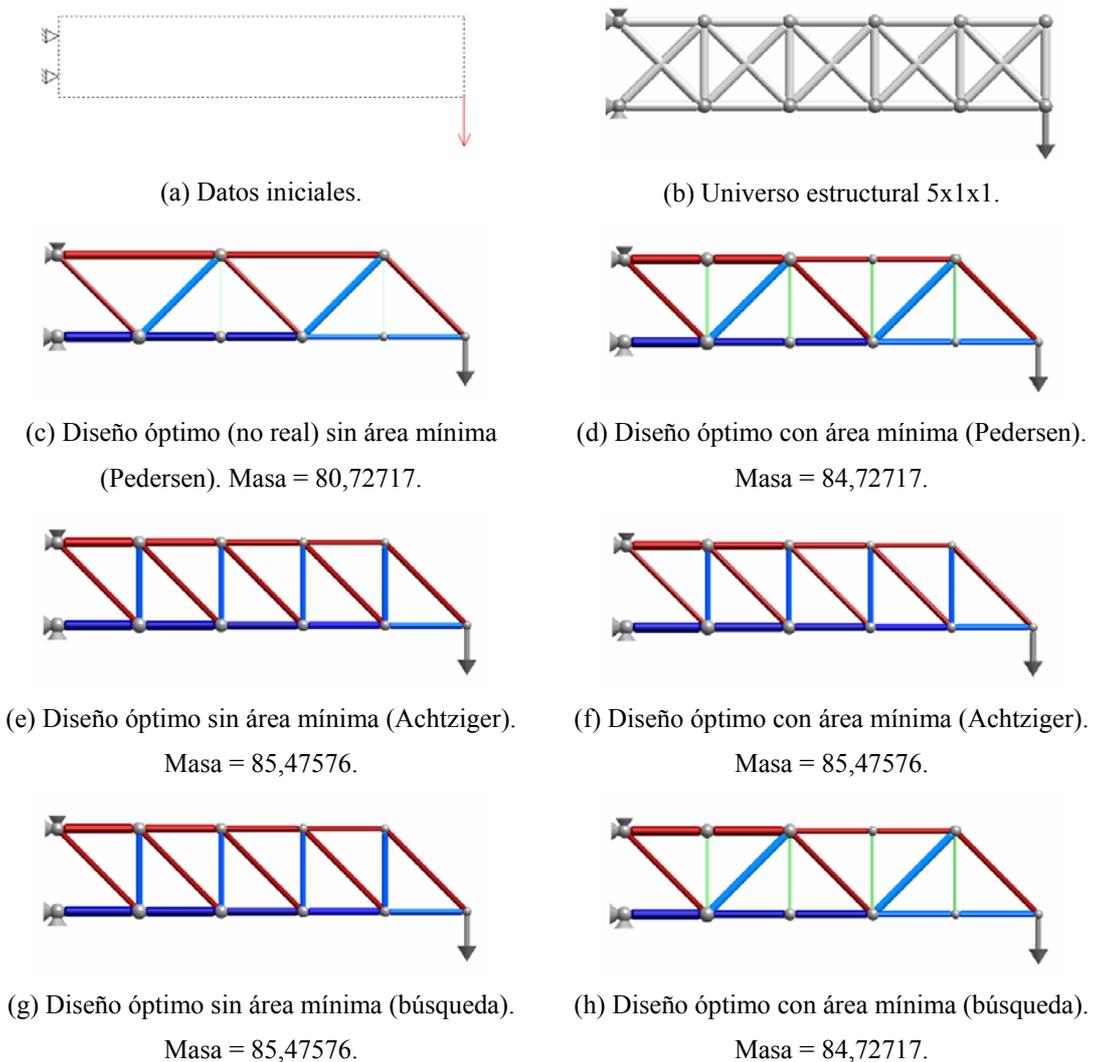


Figura 3.25 Voladizo con carga puntual en la parte inferior del extremo libre. Diseños óptimos obtenidos (con pandeo)

Diseño óptimo de topología y geometría de estructuras articuladas. Métodos de crecimiento

4.1 INTRODUCCIÓN

En este capítulo se describen los algoritmos de diseño óptimo simultáneo de topología y geometría de estructuras articuladas basados en técnicas de crecimiento.

El capítulo comienza enumerando los antecedentes del método de crecimiento. Seguidamente, se describen varios algoritmos de optimización simultánea de topología y geometría de estructuras articuladas mediante técnicas de crecimiento. Para cada algoritmo se muestra la teoría y se describe el algoritmo. A continuación, se muestran algunos ejemplos de optimización simultánea de topología y geometría mediante dichos algoritmos.

4.2 ANTECEDENTES

Los métodos del universo estructural presentan algunos inconvenientes, entre los que destacan los siguientes:

- Mediante la aproximación del universo estructural, con rejillas muy tupidas, el valor de la función objetivo será próximo al óptimo pero, en general, con topologías complejas y poco prácticas. El coste computacional aumenta de manera potencial con el tamaño de la rejilla; en cambio, la función objetivo disminuye de forma asintótica con dicho tamaño, es decir, la mejora que se obtiene al aumentar el tamaño es cada vez menor.
- El efecto del coste de los nudos no puede ser tenido en cuenta de forma sencilla con esta formulación.

Estos inconvenientes han hecho que aparezcan en escena los métodos de optimización de topología y geometría de estructuras articuladas mediante técnicas de crecimiento. Estos métodos, a partir de una topología inicial sencilla y mediante un

proceso iterativo, van añadiendo nudos y barras y optimizando la topología y la geometría en cada iteración.

Los métodos de crecimiento son recientes y cabe destacar los métodos de Rule (1994), McKeown (1998), y Bojczuk y Mróz (1998 y 1999). A continuación se da una breve descripción de dichos métodos.

4.3 MÉTODOS DE SOLUCIÓN

4.3.1 Método de Rule

4.3.1.1 Formulación

El método de Rule (1994) parte de las cargas, los apoyos y una estructura base inicial a la cual pueden añadirse barras de una manera organizada para producir una estructura práctica. La idea general consiste en generar una estructura compleja en etapas a partir de una estructura base sencilla.

El diseñador puede seleccionar el número de etapas a ser usadas en el proceso de síntesis. La estructura se hace más complicada cuantas más etapas sean usadas, puesto que un número fijo de barras son añadidas durante cada etapa. Una estructura complicada es generalmente más ligera que una sencilla, pero puede no ser más barata debido al coste de los nudos.

En la Fig. 4.1 se muestran las estructuras base iniciales para diferentes números de cargas y apoyos.

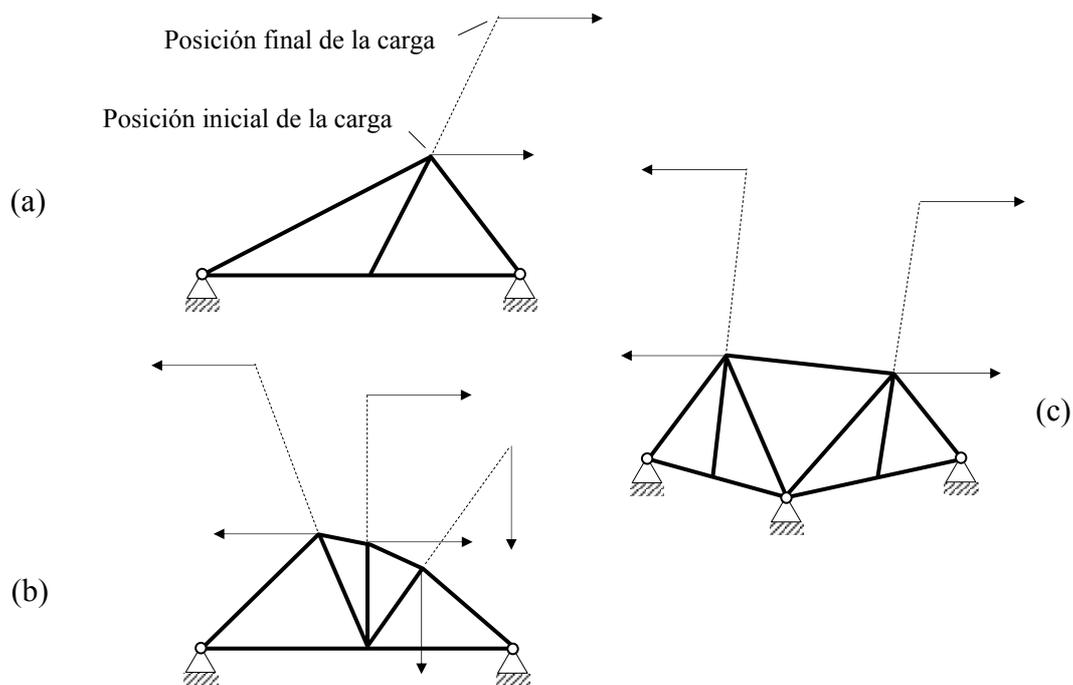


Figura 4.1 Estructuras base iniciales:

(a) 2 apoyos, 1 carga; (b) 2 apoyos, 3 cargas; (c) 3 apoyos, 2 cargas

En la estructura base los nudos cargados son situados a lo largo de líneas que unen las posiciones finales de las cargas con el centroide de las posiciones de los apoyos (líneas discontinuas).

4.3.1.2. Algoritmo de resolución

Una vez fijada la estructura base inicial se optimiza la posición de los nudos libres (todos excepto los cargados y los apoyos). Cada etapa es producida por una expansión hacia la estructura final de la etapa previa, añadiendo nuevos nudos y barras, y ajustando las posiciones de los nudos libres para minimizar la masa estructural usando técnicas de optimización convencionales (el método del gradiente conjugado, p. ej.).

Al final del proceso de expansión de cada etapa se añaden nuevos nudos y barras según la estrategia mostrada en la Fig. 4.2.

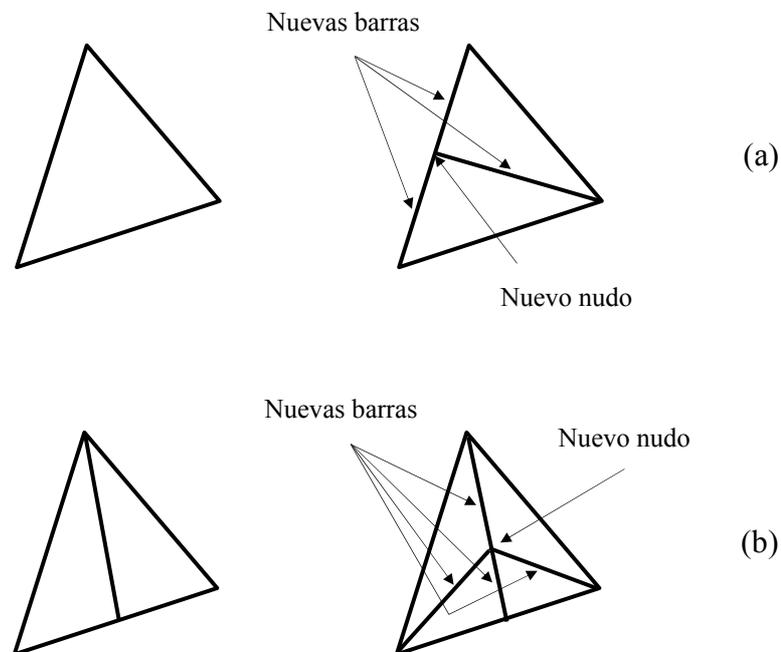


Figura 4.2 Estrategias para añadir nuevos nudos y barras:
(a) barra exterior; (b) barra interior

Para reducir la esbeltez se eligen los puntos medios de las barras más largas para la inserción de nudos según la estrategia anterior.

4.3.2 Método de McKeown

4.3.2.1 Formulación

El método de McKeown (1998) comienza con la estructura inicial más sencilla posible capaz de transmitir las cargas a los apoyos e introduce nudos uno a uno o en grupos simétricos.

En cada etapa el problema se reduce a minimizar el volumen de una estructura con un número fijo de nudos. Las variables son las posiciones y los desplazamientos de los nudos y las áreas de las barras. La formulación es la siguiente:

$$\begin{aligned}
 \underset{\mathbf{x}, \mathbf{a}, \mathbf{u}}{\text{minimizar}} : & \quad \sum_{i=1}^n L_i A_i \\
 \text{sujeto a :} & \quad A_i \geq 0 \quad i = 1, \dots, n \\
 & \quad A_i \sigma_l \leq A_i \sigma_{ij} \leq A_i \sigma_u \quad j = 1, \dots, n_k \\
 & \quad d_k(\mathbf{u}) \leq 0 \quad k = 1, \dots, m_1 \\
 & \quad d_k(\mathbf{u}) = 0 \quad k = m_1 + 1, \dots, m_2 \\
 & \quad g_k(\mathbf{x}) \leq 0 \quad k = m_2 + 1, \dots, m_3 \\
 & \quad g_k(\mathbf{x}) = 0 \quad k = m_3 + 1, \dots, m
 \end{aligned} \tag{4.1}$$

siendo:

n	Número de variables (barras).
n_k	Número de estados de cargas.
m	Número de restricciones.
m_1, m_2, m_3	Número de restricciones de desigualdad o de igualdad, de desplazamiento o de geometría.
\mathbf{x}	Coordenadas de los nudos.
\mathbf{a}	Áreas de las secciones transversales de las barras.
\mathbf{u}_i	Desplazamientos de los nudos bajo el estado de cargas P_i .
σ_{ij}	Tensión de la barra i para el estado de cargas j .
σ_l	Límite inferior de la tensión.
σ_u	Límite superior de la tensión.
$d_k(\mathbf{u})$	Restricciones de desplazamiento.
$g_k(\mathbf{x})$	Restricciones de geometría.

4.3.2.2 Algoritmo de resolución

Después de cada etapa se añaden un nudo y varias barras, pero debe ser el diseñador el que decida la posición de estos (el autor no propone ningún algoritmo automático que determine dicha posición).

4.3.3 Método de Bojczuk y Mróz

4.3.3.1 Formulación

Bojczuk y Mróz (1998 y 1999) proponen un algoritmo heurístico para el diseño óptimo de estructuras articuladas teniendo en cuenta restricciones de tensión y pandeo. Las variables de diseño están constituidas por las áreas de las secciones transversales de las barras, la configuración de los nudos y el número de nudos y de barras.

Se parte de una topología inicial y se determina el diseño óptimo de esta estructura inicial. A continuación el diseño evoluciona admitiendo una variación de topología por generación de nuevos nudos y conexiones. Los puntos de bifurcación de la topología son especificados considerando el coste y la derivada del coste con respecto al parámetro de variación topológica virtual (o crecimiento). El coste y/o la derivada del coste sufren una discontinuidad en el punto de variación de topología, y si es obtenido un valor inferior del coste o de su derivada, se asume para el posterior estudio de evolución del diseño.

Tres tipos de variación de topología son introducidos:

- 1 Un nuevo nudo es generado en el centro de una barra existente y conectado al nudo más cercano (Fig. 4.3a). El área de la sección transversal de la nueva barra es inicialmente A_{min} , la fuerza transmitida es cero, pero el coste sufre una discontinuidad. El parámetro topológico s_j corresponde a la traslación del nuevo nudo desde su posición inicial.
- 2 El nodo existente se separa y una nueva barra conectando los dos nudos es introducida (Fig. 4.3b). La barra conectada al nudo es separada en dos barras conectada a los nuevos nudos. La separación de los nudos s_j constituye el nuevo parámetro topológico.
- 3 Dos nudos son generados en el centro de una barra comprimida y separados por una barra de conexión (Fig. 4.3c). La barra existente es reemplazada por cuatro barras conectando los dos nuevos nudos a los extremos de dicha barra.

La formulación es la siguiente:

$$\begin{aligned} \underset{s_j, A_i}{\text{minimizar}} : C &= \sum_{i=1}^n c_i A_i L_i \\ \text{sujeto a : } & \begin{cases} |\sigma_i| \leq \sigma_{ai} & i = 1, \dots, n \\ A_{min} \leq A_i & i = 1, \dots, n \end{cases} \end{aligned} \quad (4.2)$$

siendo:

- C Coste de la estructura.
- n Número de variables (barras).
- c_i Coste por unidad de volumen de la barra i .
- A_i Área de la barra i .
- L_i Longitud de la barra i .
- σ_i Tensión de la barra i .
- σ_{ai} Tensión de fluencia de la barra i .
- A_{min} Área mínima.
- s_j Parámetro topológico.

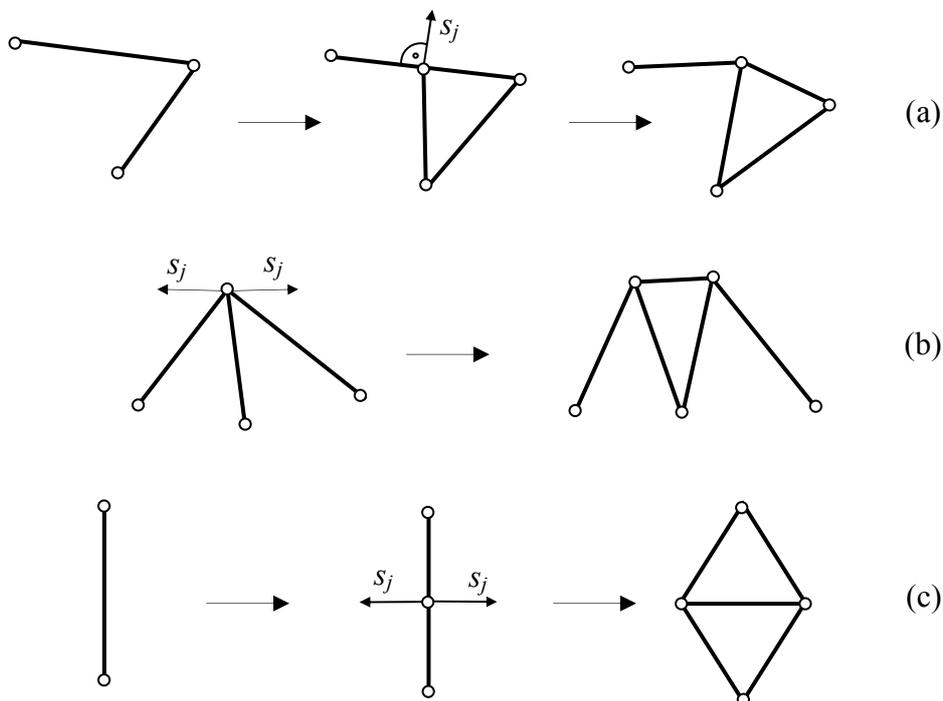


Figura 4.3 Modos de variación topológica:

- (a) generación de un nuevo nudo y barra de conexión;
- (b) nudo de separación e introducción de una barra de conexión;
- (c) introducción de dos nudos y barras de conexión

Utilizando los multiplicadores de Lagrange la función objetivo aumentada será

$$C^*(s_j, A_i, \xi_i, \mu_i) = \sum_{i=1}^n c_i A_i L_i + \sum_{i=1}^n \xi_i (|\sigma_i| - \sigma_{ai}) + \sum_{i=1}^n \mu_i (A_{\min} - A_i) \quad (4.3)$$

$$\xi_i \geq 0; \quad \mu_i \geq 0$$

siendo:

- ξ_i el multiplicador de Lagrange de la restricción de tensión de la barra i ;
- μ_i el multiplicador de Lagrange de la restricción de borde de la barra i ;

Las condiciones de optimalidad toman la forma

$$\frac{\partial C^*}{\partial s_j} = \sum_{i=1}^n c_i A_i \frac{\partial L_i}{\partial s_j} + \sum_{i=1}^n \xi_i \left(\frac{\partial |\sigma_i|}{\partial s_j} - \frac{\partial \sigma_{ai}}{\partial s_j} \right) = 0$$

$$\frac{\partial C^*}{\partial A_i} = c_i L_i + \sum_{k=1}^n \xi_k \left(\frac{\partial |\sigma_k|}{\partial A_i} - \frac{\partial \sigma_{ak}}{\partial A_i} \right) - \mu_i = 0 \quad (4.4)$$

$$\xi_i (|\sigma_i| - \sigma_{ai}) = 0; \quad \mu_i (A_{\min} - A_i) = 0$$

4.3.3.2 Algoritmo de resolución

El procedimiento de optimización consiste en la minimización de la Ec. (4.3) con respecto al parámetro topológico s_j . Se pueden distinguir tres pasos en el procedimiento:

- 1 Selección del diseño inicial. Se elige una estructura isostática que satisfaga las restricciones de tensión y borde (4.2) del problema.
- 2 Optimización del diseño inicial. El diseño inicial es modificado determinando las nuevas posiciones s_j de los nudos sin cambiar la topología.
- 3 Variación de la topología. Se modifica la topología según los modos mostrados en la Fig. 4.3, se calcula la variación de la función objetivo con respecto al diseño anterior y si es negativo la nueva topología es aceptada y se vuelve a repetir este paso mientras el decremento de la función objetivo sea negativo.

Diseño óptimo simultáneo de topología y geometría de estructuras articuladas. Método propuesto

5.1 INTRODUCCIÓN

En este capítulo se propone un método, basado en técnicas de crecimiento, para el diseño óptimo simultáneo de topología y geometría de estructuras articuladas sometidas a un estado de cargas y con restricciones de diseño de tensión, pandeo de elementos, esbeltez y área mínima.

En primer lugar se describe el método propuesto. Seguidamente se trata la obtención de la topología inicial. A continuación se formula la optimización simultánea de topología y de geometría en cada iteración. Posteriormente se discuten los criterios utilizados para la modificación de la topología en cada iteración. Finalmente se muestran dos ejemplos donde se ve la evolución de los diseños óptimos en las primeras iteraciones.

5.2 DISEÑO ÓPTIMO SIMULTÁNEO DE TOPOLOGÍA Y GEOMETRÍA

El método propuesto es un algoritmo de optimización, basado en técnicas de crecimiento, para el diseño óptimo simultáneo de topología y geometría de estructuras articuladas sometidas a un estado de cargas y con restricciones de diseño de tensión, pandeo de elementos, esbeltez y área mínima.

A causa de la complejidad de la optimización simultánea global de topología y geometría, se han considerado dos etapas en cada iteración:

- 1 Optimización de topología.
- 2 Optimización de geometría.

El proceso de diseño óptimo simultáneo de topología y geometría (Fig. 5.1) se divide en cinco partes:

- 1 Obtención de la topología inicial.
- 2 Optimización de topología.
- 3 Optimización de geometría.
- 4 Comprobación de la topología óptima en la nueva geometría.
- 5 Crecimiento de la topología.

Una vez definidos los datos, se obtiene una topología inicial a partir de la posición de los puntos con desplazamientos impedidos y de los puntos cargados.

A continuación se sigue un proceso iterativo en el cual, y en primer lugar, se optimiza la topología obtenida.

Seguidamente se optimiza la geometría de la topología óptima obtenida.

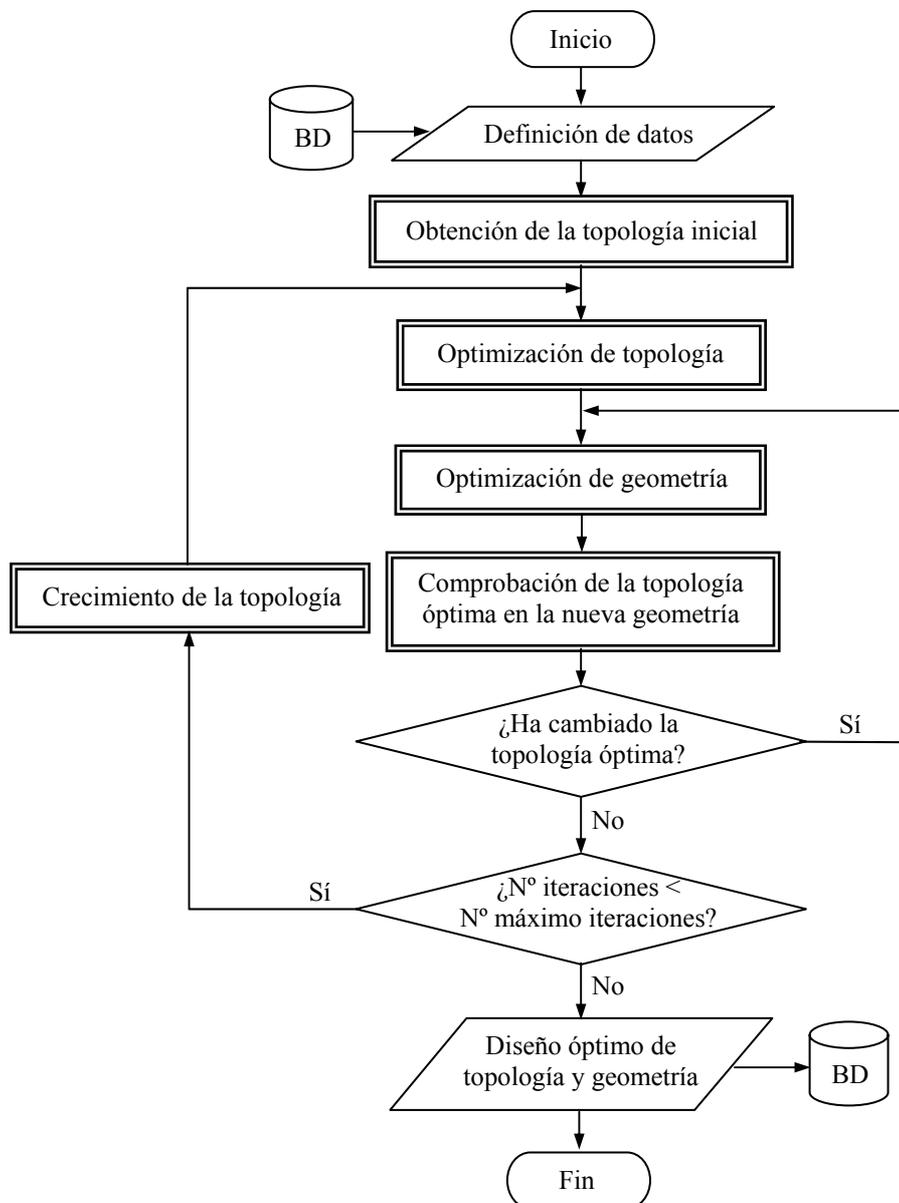


Figura 5.1 Diagrama de flujo para el diseño óptimo simultáneo de topología y geometría

Una vez se ha optimizado la geometría, se comprueba si la topología óptima ha cambiado, en cuyo caso se vuelve a optimizar la geometría.

Finalmente, se modifica la topología añadiendo un nudo (dos nudos si se impone la condición de simetría) y varias barras, y se repite el proceso iterativo hasta que se alcanza el número máximo de iteraciones previamente definido, o hasta que no se encuentre una topología que mejore la anterior.

La optimización de propiedades de la sección transversal de las barras se lleva a cabo tanto en la etapa de optimización de topología como en la de optimización de geometría.

5.3 OBTENCIÓN DE LA TOPOLOGÍA INICIAL

5.3.1 Introducción

En los métodos de optimización de topología de estructuras articuladas basados en el universo estructural, la estructura inicial es el propio universo estructural. En los métodos de crecimiento, normalmente, se define una topología inicial sencilla (Rule, 1994; McKeown, 1998, y Bojczuk y Mróz, 1998 y 1999), lo más parecida posible a la óptima, para mejorar y asegurar la convergencia hacia la solución.

Para generar el universo estructural es conveniente modelar la estructura articulada y definir una metodología de generación.

Los nudos iniciales pueden definirse directamente, o bien, generarse a partir de los datos iniciales.

5.3.2 Definición y/o generación de los nudos iniciales

Para la obtención de la topología inicial es necesario definir los nudos iniciales, que son los nudos en los que hay definidos apoyos o cargas, y aquellos nudos que se quiere que aparezcan en el diseño óptimo final.

Los nudos iniciales pueden definirse directamente en un determinado punto, sin embargo, para tener una mayor flexibilidad, es conveniente generarlos a partir de líneas, áreas o volúmenes, discretizando éstas en una rejilla de nudos.

Para discretizar líneas, áreas o volúmenes, en una rejilla de nudos, es necesario definir el número de divisiones en cada dirección (una para las líneas, dos para las áreas y tres para los volúmenes).

En la Fig. 5.2 se muestran ejemplos de generación de los nudos iniciales a partir de los datos iniciales para diferentes casos de desplazamientos impedidos y de tipos de cargas.

En las Figs. 5.2a a 5.2d se muestran dos ejemplos de generación de nudos para casos de desplazamientos impedidos y cargas en puntos, y en las Figs. 5.2e a 5.2h dos

ejemplos para casos de desplazamientos impedidos y cargas uniformemente repartidas en líneas.



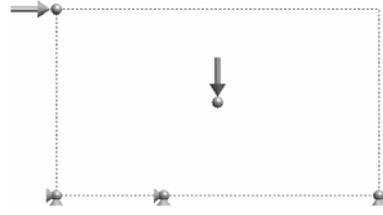
(a) Voladizo de Michell.
Datos iniciales.



(b) Voladizo de Michell.
Nudos iniciales.



(c) Disquera.
Datos iniciales.



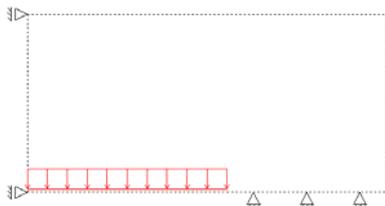
(d) Disquera.
Nudos iniciales.



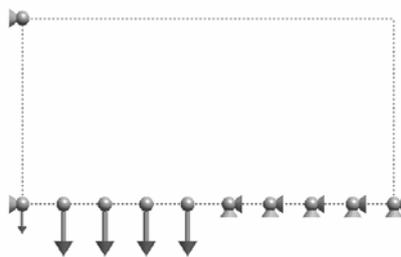
(e) Disco circular rígido.
Datos iniciales.



(f) Disco circular rígido.
Nudos iniciales.



(g) Puente.
Datos iniciales.



(h) Puente.
Nudos iniciales.

Figura 5.2 Ejemplos de generación de los nudos iniciales, a partir de los datos iniciales, para diferentes casos de desplazamientos impedidos y de tipos de cargas

Se genera un nudo en cada punto con desplazamientos impedidos o con carga puntual (Figs. 5.2a a 5.2d). Las líneas con desplazamientos impedidos o con cargas repartidas se discretizan en un número de nudos previamente definido (Figs. 5.2e a 5.2h).

En el ejemplo de las Figs. 5.2e y 5.2f se discretiza una línea con desplazamientos impedidos en 12 nudos. En las Figs. 5.2g y 5.2h se discretizan una línea con desplazamientos impedidos en 5 nudos y una línea con carga repartida uniformemente en 6 nudos. En la Fig. 5.2h, el nudo del extremo izquierdo de la línea cargada absorbe la mitad de carga que los centrales por estar en un extremo (sólo tiene carga a un lado) y en el nudo del extremo derecho no se dibuja ninguna carga porque la absorbe el apoyo (la carga está aplicada en un grado de libertad con desplazamiento impedido).

5.3.3 Obtención de la topología inicial

Si no se ha definido ninguna topología inicial, a partir de los nudos definidos y/o generados se crea el universo estructural inicial colocando barras entre cada par de nudos. Este universo estructural se adopta como topología inicial.

En la Fig. 5.3 se muestra el diagrama de flujo del proceso para la obtención de la topología inicial. Éste consiste en, una vez definidos los datos iniciales, generar los nudos iniciales como se ha explicado en el epígrafe 5.3.2 y generar las barras colocando una entre cada par de nudos (excepto cuando los dos nudos tienen todos sus grados de libertad con los desplazamientos impedidos).

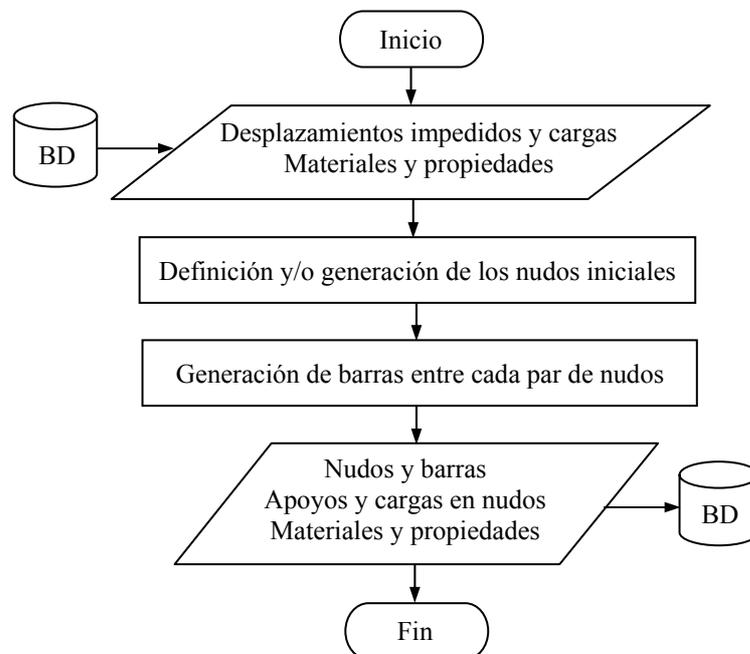


Figura 5.3 Diagrama de flujo para la obtención de la topología inicial

En la Fig. 5.4 se muestran cuatro ejemplos de obtención de las topologías iniciales a partir de los nudos iniciales para diferentes casos de desplazamientos impedidos y de tipos de cargas.

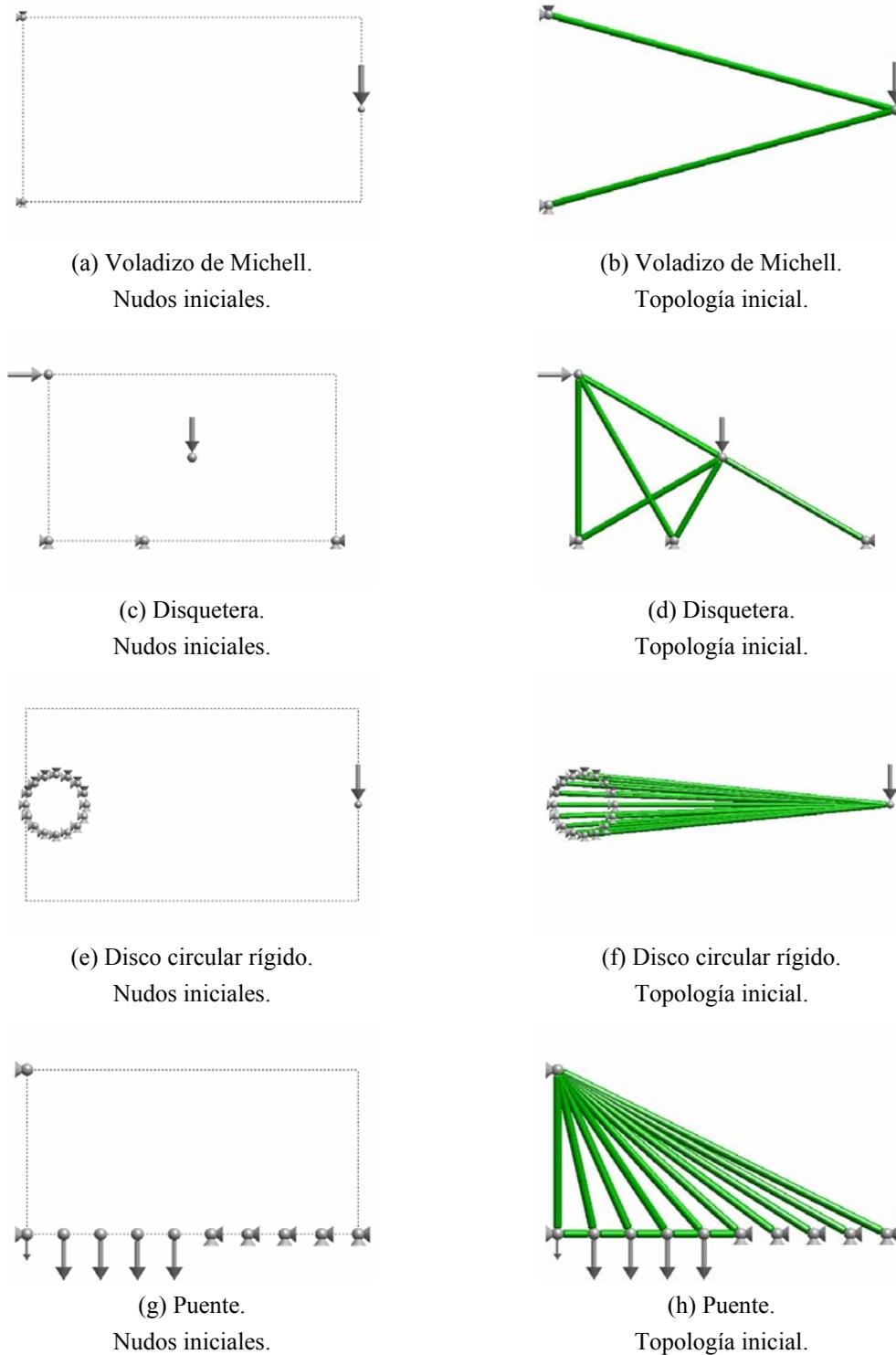


Figura 5.4 Ejemplos de obtención de las topologías iniciales, a partir de los nudos iniciales, para diferentes casos de desplazamientos impedidos y de tipos de cargas

En estos ejemplos no se ha definido una topología inicial, si no que se ha generado a partir de los diferentes casos de desplazamientos impedidos y de los tipos de cargas.

También es posible definir una topología inicial, directamente, definiendo los nudos y las barras, o indirectamente, definiendo una malla de puntos a partir de la cual se genera el universo estructural.

Para la posterior optimización de geometría hay que definir los grados de libertad de los nudos generados que tienen permitido el movimiento (variables en la optimización de geometría), y las restricciones de borde para el movimiento de dichos nudos.

5.4 OPTIMIZACIÓN DE TOPOLOGÍA

5.4.1 Introducción

La estructura inicial y las estructuras que surgen en cada iteración al modificar la topología son, normalmente, estructuras hiperestáticas. Pedersen (Pedersen, 1993) demuestra que la estructura óptima para un estado de cargas y restricciones de tensión, pandeo de elementos, esbeltez y área mínima, es isostática.

Optimizar la topología consiste, por lo tanto, en elegir la estructura isostática de menor función objetivo, de entre todas las combinaciones posibles que se pueden obtener a partir de la estructura hiperestática que se quiere optimizar.

Puesto que en las estructuras isostáticas los esfuerzos axiales son independientes de las áreas de las barras, para obtener estas áreas se determinan los esfuerzos axiales (resolviendo el sistema de ecuaciones de equilibrio) y, posteriormente, se elige como área de cada barra el menor valor que cumpla todas las restricciones de diseño.

5.4.2 Optimización de topología mediante métodos basados en el universo estructural

En el método propuesto, una vez generada la topología inicial, o aumentada en uno o dos nudos la topología óptima de la iteración anterior, se optimiza la topología mediante uno de los métodos implementados. En la Fig. 5.5 se muestra el diagrama de flujo del proceso para la optimización de topología.

En la Fig. 5.6 se muestran las topologías óptimas correspondientes a las topologías iniciales de los ejemplos de la Fig. 5.4. Las líneas de trazo continuo representan las barras que forman parte de la topología óptima, con su tamaño correspondiente, y las discontinuas las que no forman parte de ella.

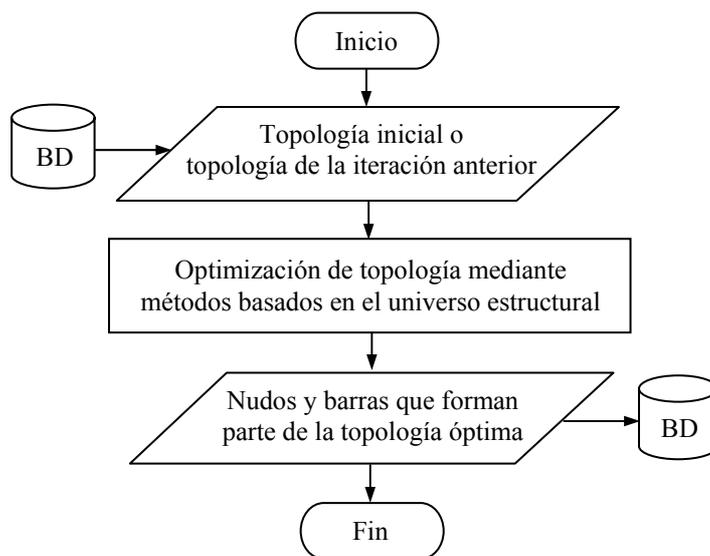


Figura 5.5 Diagrama de flujo para la optimización de topología

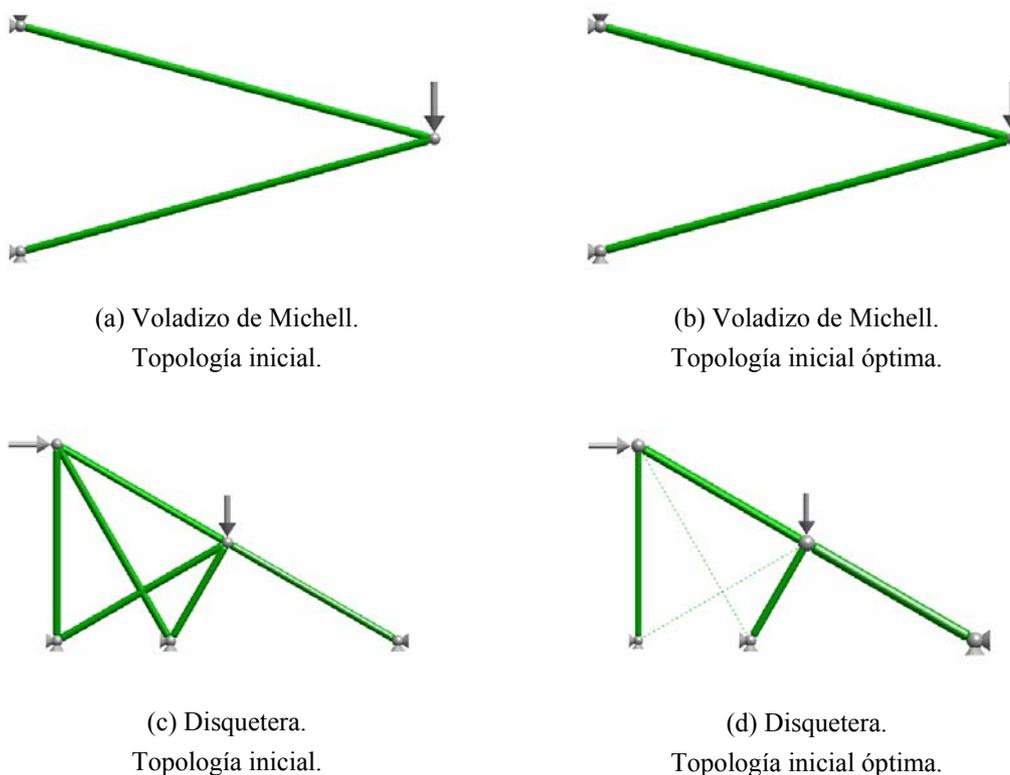
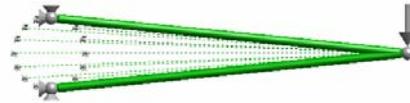


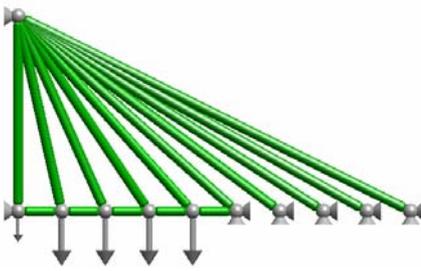
Figura 5.6 Ejemplos de obtención de las topologías iniciales óptimas para diferentes casos de desplazamientos impedidos y de tipos de cargas



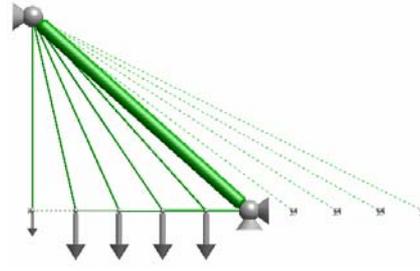
(e) Disco circular rígido.
Topología inicial.



(f) Disco circular rígido.
Topología inicial óptima.



(g) Puente.
Topología inicial.



(h) Puente.
Topología inicial óptima.

Figura 5.6 Ejemplos de obtención de las topologías iniciales óptimas para diferentes casos de desplazamientos impedidos y de tipos de cargas. (Continuación)

5.5 OPTIMIZACIÓN DE GEOMETRÍA

5.5.1 Introducción

Una vez optimizada la topología inicial, o la topología obtenida en la iteración anterior, hay que optimizar la posición de todos los nudos móviles, es decir, aquellos cuyas coordenadas son variables.

El problema de optimización de geometría de estructuras articuladas isostáticas, para el método propuesto, consiste en minimizar una función objetivo (masa, volumen o coste) no lineal con respecto a las variables de diseño (coordenadas de los nudos) y sujeto a restricciones de borde simples para las variables de diseño.

5.5.2 Formulación del problema de optimización de geometría

5.5.2.1 Variables de diseño

Las variables de diseño son las coordenadas variables de los nudos que pueden moverse.

En los nudos de la topología inicial se permite el movimiento de los grados de libertad definidos como móviles. En los nudos añadidos en cada iteración se permite el movimiento de todos sus grados de libertad.

5.5.2.2 Función objetivo

La función objetivo puede ser la masa, el volumen o el coste total de la estructura. En lo que sigue, se considera que la función objetivo es la masa.

Como la optimización de geometría se aplica a estructuras isostáticas, en las cuales los esfuerzos axiales son independientes de las áreas, para obtener la función objetivo se resuelve únicamente el sistema de ecuaciones de equilibrio, ya que las ecuaciones de compatibilidad no son necesarias por ser la estructura isostática. Una vez obtenidos los esfuerzos axiales, el área de cada barra será el menor valor que cumpla todas las restricciones (de tensión, de pandeo de elementos, de esbeltez o de área mínima). A partir de los valores de las áreas se obtiene la función objetivo de la estructura.

5.5.2.3 Restricciones de diseño

En el algoritmo de optimización de geometría sólo son necesarias restricciones de borde (si las hay). En el cálculo de la función objetivo se imponen, de forma implícita, las restricciones de tensión, pandeo de elementos, esbeltez y área mínima, luego no es necesario añadir ninguna restricción al algoritmo de optimización de geometría.

5.5.2.4 Formulación matemática del problema de optimización de geometría

En términos matemáticos, el problema de optimización de geometría de estructuras articuladas isostáticas para masa mínima se puede formular como:

$$\begin{aligned} \text{minimizar : } & M(\mathbf{x}) = \sum_{e=1}^m \rho_e L_e A_e \\ \text{sujeto a : } & x_i^I \leq x_i \leq x_i^S \quad i = 1, 2, \dots, n \end{aligned} \quad (5.1)$$

siendo:

m	Número de barras.
n	Número de variables de diseño.
\mathbf{x}	Vector de variables de diseño (coordenadas variables de los nudos).
$M(\mathbf{x})$	Masa de la estructura.
ρ_e	Densidad de la barra e .
L_e	Longitud de la barra e .
A_e	Área de la sección transversal de la barra e .
x_i	Variable de diseño i .
x_i^I	Límite inferior de la variable de diseño i .
x_i^S	Límite superior de la variable de diseño i .

que es un problema de programación matemática no lineal.

Para cada geometría analizada, se hace una optimización de propiedades, es decir, se determinan las áreas según se indica en el epígrafe 5.5.2.2.

5.5.3 Cálculo de las derivadas analíticas de la función objetivo

El algoritmo de optimización necesita las derivadas de la función objetivo respecto de las variables. Puesto que el coste computacional de las derivadas numéricas es muy superior al de las derivadas analíticas, se va a proceder al cálculo de las derivadas analíticas de la función objetivo.

Dado que la estructura cuya geometría se quiere optimizar es isostática, es posible obtener las expresiones analíticas explícitas de las derivadas de la función objetivo respecto de las variables de diseño.

Derivando la expresión de la masa total de la estructura, Ec. (5.1), respecto de la variable x_i , se obtiene

$$\frac{\partial M}{\partial x_i} = \sum_{e=1}^m \rho_e \left(\frac{\partial L_e}{\partial x_i} A_e + L_e \frac{\partial A_e}{\partial x_i} \right) \quad (5.2)$$

donde

$$L_e = \sqrt{(x_e^{fin} - x_e^{ini})^2 + (y_e^{fin} - y_e^{ini})^2 + (z_e^{fin} - z_e^{ini})^2}$$

$$\cos \alpha = \frac{x_e^{fin} - x_e^{ini}}{L_e} \quad \cos \beta = \frac{y_e^{fin} - y_e^{ini}}{L_e} \quad \cos \gamma = \frac{z_e^{fin} - z_e^{ini}}{L_e}$$

$$\frac{\partial L_e}{\partial x_i} = \begin{cases} 0 & \text{si } x_i \notin \text{barra} \\ -\cos \alpha & \text{si } x_i \equiv x_e^{ini} \\ \cos \alpha & \text{si } x_i \equiv x_e^{fin} \\ -\cos \beta & \text{si } x_i \equiv y_e^{ini} \\ \cos \beta & \text{si } x_i \equiv y_e^{fin} \\ -\cos \gamma & \text{si } x_i \equiv z_e^{ini} \\ \cos \gamma & \text{si } x_i \equiv z_e^{fin} \end{cases} \quad (5.3)$$

siendo:

- x_e^{ini} Coordenada x del nudo inicial de la barra e .
- x_e^{fin} Coordenada x del nudo final de la barra e .
- y_e^{ini} Coordenada y del nudo inicial de la barra e .
- y_e^{fin} Coordenada y del nudo final de la barra e .
- z_e^{ini} Coordenada z del nudo inicial de la barra e .
- z_e^{fin} Coordenada z del nudo final de la barra e .

Aplicando la regla de derivación en cadenas la derivada del área respecto de la variable x_i puede ponerse en la forma

$$\frac{\partial A_e}{\partial x_i} = \frac{\partial A_e}{\partial q_e} \frac{\partial q_e}{\partial x_i} \quad (5.4)$$

Derivando la función área, Ec. (3.2), respecto del esfuerzo axial de la barra se obtiene

$$\frac{\partial A_e}{\partial q_e} = \begin{cases} 0 & \text{para } q_e = 0 \\ \frac{1}{\sigma_T} & \text{para } q_e > 0 \quad \text{tracción (límite elástico)} \\ -\frac{L_e}{2\pi\sqrt{\beta E/s}} \frac{1}{\sqrt{-q_e}} & \text{para } -P_E \leq q_e < 0 \quad \text{compresión (pandeo)} \\ -\frac{1}{\sigma_C} & \text{para } q_e < -P_E \quad \text{compresión (límite elástico)} \end{cases} \quad (5.5)$$

siendo:

- q_e Esfuerzo axial de la barra e .
- σ_T Límite elástico a tracción.
- σ_C Límite elástico a compresión.
- E Módulo de Young.
- I_e Momento de inercia de la sección de la barra e .
- s Coeficiente de seguridad.
- β L_e/A_e^2 .
- P_E Carga crítica de Euler.
- A_E Área correspondiente a la carga crítica de Euler.

Derivando el sistema de ecuaciones de equilibrio, Ec. (3.11), respecto de la variable x_i , se obtiene

$$\frac{\partial \mathbf{q}}{\partial x_i} = -\mathbf{R}^{-1} \frac{\partial \mathbf{R}}{\partial x_i} \mathbf{q} \quad (5.6)$$

Las derivadas de los cosenos directores (\mathbf{R} es la matriz de cosenos directores) respecto de la variable x_i son

$$\begin{aligned}
 \frac{\partial \cos \alpha}{\partial x_i} &= \begin{cases} 0 & \text{si } x_i \notin \text{barra} \\ -\frac{1 - \cos^2 \alpha}{l_e} & \text{si } x_i \equiv x_e^{ini} \\ \frac{1 - \cos^2 \alpha}{l_e} & \text{si } x_i \equiv x_e^{fin} \\ \frac{\cos \alpha \cos \beta}{l_e} & \text{si } x_i \equiv y_e^{ini} \\ -\frac{\cos \alpha \cos \beta}{l_e} & \text{si } x_i \equiv y_e^{fin} \\ \frac{\cos \alpha \cos \gamma}{l_e} & \text{si } x_i \equiv z_e^{ini} \\ -\frac{\cos \alpha \cos \gamma}{l_e} & \text{si } x_i \equiv z_e^{fin} \end{cases} \\
 \frac{\partial \cos \beta}{\partial x_i} &= \begin{cases} 0 & \text{si } x_i \notin \text{barra} \\ \frac{\cos \beta \cos \alpha}{l_e} & \text{si } x_i \equiv x_e^{ini} \\ -\frac{\cos \beta \cos \alpha}{l_e} & \text{si } x_i \equiv x_e^{fin} \\ -\frac{1 - \cos^2 \beta}{l_e} & \text{si } x_i \equiv y_e^{ini} \\ \frac{1 - \cos^2 \beta}{l_e} & \text{si } x_i \equiv y_e^{fin} \\ \frac{\cos \beta \cos \gamma}{l_e} & \text{si } x_i \equiv z_e^{ini} \\ -\frac{\cos \beta \cos \gamma}{l_e} & \text{si } x_i \equiv z_e^{fin} \end{cases} \\
 \frac{\partial \cos \gamma}{\partial x_i} &= \begin{cases} 0 & \text{si } x_i \notin \text{barra} \\ \frac{\cos \gamma \cos \alpha}{l_e} & \text{si } x_i \equiv x_e^{ini} \\ -\frac{\cos \gamma \cos \alpha}{l_e} & \text{si } x_i \equiv x_e^{fin} \\ \frac{\cos \gamma \cos \beta}{l_e} & \text{si } x_i \equiv y_e^{ini} \\ -\frac{\cos \gamma \cos \beta}{l_e} & \text{si } x_i \equiv y_e^{fin} \\ -\frac{1 - \cos^2 \gamma}{l_e} & \text{si } x_i \equiv z_e^{ini} \\ \frac{1 - \cos^2 \gamma}{l_e} & \text{si } x_i \equiv z_e^{fin} \end{cases}
 \end{aligned} \tag{5.7}$$

5.5.4 Cálculo de las derivadas segundas analíticas de la función objetivo

Algunos algoritmos de optimización hacen uso de las derivadas segundas de la función objetivo respecto de las variables.

Derivando la Ec. (5.2) respecto de la variable x_j se obtiene

$$\frac{\partial^2 M}{\partial x_i \partial x_j} = \sum_{e=1}^m \rho_e \left(\frac{\partial^2 L_e}{\partial x_i \partial x_j} A_e + \frac{\partial L_e}{\partial x_i} \frac{\partial A_e}{\partial x_j} + \frac{\partial L_e}{\partial x_j} \frac{\partial A_e}{\partial x_i} + L_e \frac{\partial^2 A_e}{\partial x_i \partial x_j} \right) \quad (5.8)$$

donde

$$\frac{\partial^2 L_e}{\partial x_i \partial x_j} = \begin{cases} 0 & \text{si } x_i \notin \text{barra} \\ -\frac{\partial \cos \alpha}{\partial x_j} & \text{si } x_i \equiv x_e^{ini} \\ \frac{\partial \cos \alpha}{\partial x_j} & \text{si } x_i \equiv x_e^{fin} \\ -\frac{\partial \cos \beta}{\partial x_j} & \text{si } x_i \equiv y_e^{ini} \\ \frac{\partial \cos \beta}{\partial x_j} & \text{si } x_i \equiv y_e^{fin} \\ -\frac{\partial \cos \gamma}{\partial x_j} & \text{si } x_i \equiv z_e^{ini} \\ \frac{\partial \cos \gamma}{\partial x_j} & \text{si } x_i \equiv z_e^{fin} \end{cases} \quad (5.9)$$

Aplicando la regla de derivación en cadenas a la Ec. (5.4) se obtiene

$$\frac{\partial^2 A_e}{\partial x_i \partial x_j} = \frac{\partial^2 A_e}{\partial q_e^2} \frac{\partial q_e}{\partial x_i} \frac{\partial q_e}{\partial x_j} + \frac{\partial A_e}{\partial q_e} \frac{\partial^2 q_e}{\partial x_i \partial x_j} \quad (5.10)$$

Derivando la Ec. (5.5) respecto del esfuerzo axial de la barra se obtiene

$$\frac{\partial^2 A_e}{\partial q_e^2} = \begin{cases} 0 & \text{para } q_e = 0 \\ 0 & \text{para } q_e > 0 \quad \text{tracción (límite elástico)} \\ -\frac{L}{4\pi\sqrt{\beta E / s}} \frac{1}{\sqrt{-q_e^3}} & \text{para } -P_E \leq q_e < 0 \quad \text{compresión (pandeo)} \\ 0 & \text{para } q_e < -P_E \quad \text{compresión (límite elástico)} \end{cases} \quad (5.11)$$

Derivando la Ec. (5.6) respecto de la variable x_j , se obtiene

$$\frac{\partial^2 \mathbf{q}}{\partial x_i \partial x_j} = -\mathbf{R}^{-1} \left(\frac{\partial^2 \mathbf{R}}{\partial x_i \partial x_j} \mathbf{q} + \frac{\partial \mathbf{R}}{\partial x_i} \frac{\partial \mathbf{q}}{\partial x_j} + \frac{\partial \mathbf{R}}{\partial x_j} \frac{\partial \mathbf{q}}{\partial x_i} \right) \quad (5.12)$$

Las derivadas segundas de los cosenos directores respecto de las variables x_i y x_j son

$$\frac{\partial^2 \cos \alpha}{\partial x_i^2} = \begin{cases} 0 & \text{si } x_i \notin \text{barra} \\ -\frac{3 \cos \alpha (1 - \cos^2 \alpha)}{L_e^2} & \text{si } x_i \equiv x_e^{ini} \\ -\frac{3 \cos \alpha (1 - \cos^2 \alpha)}{L_e^2} & \text{si } x_i \equiv x_e^{fin} \\ -\frac{\cos \alpha (1 - 3 \cos^2 \beta)}{L_e^2} & \text{si } x_i \equiv y_e^{ini} \\ -\frac{\cos \alpha (1 - 3 \cos^2 \beta)}{L_e^2} & \text{si } x_i \equiv y_e^{fin} \\ -\frac{\cos \alpha (1 - 3 \cos^2 \gamma)}{L_e^2} & \text{si } x_i \equiv z_e^{ini} \\ -\frac{\cos \alpha (1 - 3 \cos^2 \gamma)}{L_e^2} & \text{si } x_i \equiv z_e^{fin} \end{cases}$$

$$\frac{\partial^2 \cos \alpha}{\partial x_i \partial x_j} = \begin{cases} 0 & \text{si } x_i \notin \text{barra} \\ \frac{2 \cos \alpha \frac{\partial \cos \alpha}{\partial x_j} + \frac{1 - \cos^2 \alpha}{L_e} \frac{\partial L_e}{\partial x_j}}{L_e} & \text{si } x_i \equiv x_e^{ini} \\ -\frac{2 \cos \alpha \frac{\partial \cos \alpha}{\partial x_j} + \frac{1 - \cos^2 \alpha}{L_e} \frac{\partial L_e}{\partial x_j}}{L_e} & \text{si } x_i \equiv x_e^{fin} \\ \frac{\cos \beta \frac{\partial \cos \alpha}{\partial x_j} + \cos \alpha \frac{\partial \cos \beta}{\partial x_j} - \frac{\cos \alpha \cos \beta}{L_e} \frac{\partial L_e}{\partial x_j}}{L_e} & \text{si } x_i \equiv y_e^{ini} \\ -\frac{\cos \beta \frac{\partial \cos \alpha}{\partial x_j} + \cos \alpha \frac{\partial \cos \beta}{\partial x_j} - \frac{\cos \alpha \cos \beta}{L_e} \frac{\partial L_e}{\partial x_j}}{L_e} & \text{si } x_i \equiv y_e^{fin} \\ \frac{\cos \gamma \frac{\partial \cos \alpha}{\partial x_j} + \cos \alpha \frac{\partial \cos \gamma}{\partial x_j} - \frac{\cos \alpha \cos \gamma}{L_e} \frac{\partial L_e}{\partial x_j}}{L_e} & \text{si } x_i \equiv z_e^{ini} \\ -\frac{\cos \gamma \frac{\partial \cos \alpha}{\partial x_j} + \cos \alpha \frac{\partial \cos \gamma}{\partial x_j} - \frac{\cos \alpha \cos \gamma}{L_e} \frac{\partial L_e}{\partial x_j}}{L_e} & \text{si } x_i \equiv z_e^{fin} \end{cases} \quad (5.13)$$

$$\frac{\partial^2 \cos \beta}{\partial x_i^2} = \begin{cases} 0 & \text{si } x_i \notin \text{barra} \\ -\frac{\cos \beta(1-3\cos^2 \alpha)}{L_e^2} & \text{si } x_i \equiv x_e^{ini} \\ \frac{\cos \beta(1-3\cos^2 \alpha)}{L_e^2} & \text{si } x_i \equiv x_e^{fin} \\ -\frac{3\cos \beta(1-\cos^2 \beta)}{L_e^2} & \text{si } x_i \equiv y_e^{ini} \\ \frac{3\cos \beta(1-\cos^2 \beta)}{L_e^2} & \text{si } x_i \equiv y_e^{fin} \\ -\frac{\cos \beta(1-3\cos^2 \gamma)}{L_e^2} & \text{si } x_i \equiv z_e^{ini} \\ \frac{\cos \beta(1-3\cos^2 \gamma)}{L_e^2} & \text{si } x_i \equiv z_e^{fin} \end{cases}$$

$$\frac{\partial^2 \cos \beta}{\partial x_i \partial x_j} = \begin{cases} 0 & \text{si } x_i \notin \text{barra} \\ \frac{\cos \alpha \frac{\partial \cos \beta}{\partial x_j} + \cos \beta \frac{\partial \cos \alpha}{\partial x_j} - \frac{\cos \alpha \cos \beta}{L_e} \frac{\partial L_e}{\partial x_j}}{L_e} & \text{si } x_i \equiv x_e^{ini} \\ \frac{\cos \alpha \frac{\partial \cos \beta}{\partial x_j} + \cos \beta \frac{\partial \cos \alpha}{\partial x_j} - \frac{\cos \alpha \cos \beta}{L_e} \frac{\partial L_e}{\partial x_j}}{L_e} & \text{si } x_i \equiv x_e^{fin} \\ \frac{2\cos \beta \frac{\partial \cos \beta}{\partial x_j} + \frac{1-\cos^2 \beta}{L_e} \frac{\partial L_e}{\partial x_j}}{L_e} & \text{si } x_i \equiv y_e^{ini} \\ -\frac{2\cos \beta \frac{\partial \cos \beta}{\partial x_j} + \frac{1-\cos^2 \beta}{L_e} \frac{\partial L_e}{\partial x_j}}{L_e} & \text{si } x_i \equiv y_e^{fin} \\ \frac{\cos \gamma \frac{\partial \cos \beta}{\partial x_j} + \cos \beta \frac{\partial \cos \gamma}{\partial x_j} - \frac{\cos \beta \cos \gamma}{L_e} \frac{\partial L_e}{\partial x_j}}{L_e} & \text{si } x_i \equiv z_e^{ini} \\ -\frac{\cos \gamma \frac{\partial \cos \beta}{\partial x_j} + \cos \beta \frac{\partial \cos \gamma}{\partial x_j} - \frac{\cos \beta \cos \gamma}{L_e} \frac{\partial L_e}{\partial x_j}}{L_e} & \text{si } x_i \equiv z_e^{fin} \end{cases} \quad (5.14)$$

$$\frac{\partial^2 \cos \gamma}{\partial x_i^2} = \begin{cases} 0 & \text{si } x_i \notin \text{barra} \\ -\frac{\cos \gamma (1 - 3 \cos^2 \alpha)}{L_e^2} & \text{si } x_i \equiv x_e^{ini} \\ -\frac{\cos \gamma (1 - 3 \cos^2 \alpha)}{L_e^2} & \text{si } x_i \equiv x_e^{fin} \\ -\frac{\cos \gamma (1 - 3 \cos^2 \beta)}{L_e^2} & \text{si } x_i \equiv y_e^{ini} \\ -\frac{\cos \gamma (1 - 3 \cos^2 \beta)}{L_e^2} & \text{si } x_i \equiv y_e^{fin} \\ -\frac{3 \cos \gamma (1 - \cos^2 \gamma)}{L_e^2} & \text{si } x_i \equiv z_e^{ini} \\ -\frac{3 \cos \gamma (1 - \cos^2 \gamma)}{L_e^2} & \text{si } x_i \equiv z_e^{fin} \end{cases}$$

$$\frac{\partial^2 \cos \gamma}{\partial x_i \partial x_j} = \begin{cases} 0 & \text{si } x_i \notin \text{barra} \\ \frac{\cos \alpha \frac{\partial \cos \gamma}{\partial x_j} + \cos \gamma \frac{\partial \cos \alpha}{\partial x_j} - \frac{\cos \alpha \cos \gamma}{L_e} \frac{\partial L_e}{\partial x_j}}{L_e} & \text{si } x_i \equiv x_e^{ini} \\ \frac{\cos \alpha \frac{\partial \cos \gamma}{\partial x_j} + \cos \gamma \frac{\partial \cos \alpha}{\partial x_j} - \frac{\cos \alpha \cos \gamma}{L_e} \frac{\partial L_e}{\partial x_j}}{L_e} & \text{si } x_i \equiv x_e^{fin} \\ \frac{\cos \beta \frac{\partial \cos \gamma}{\partial x_j} + \cos \gamma \frac{\partial \cos \beta}{\partial x_j} - \frac{\cos \beta \cos \gamma}{L_e} \frac{\partial L_e}{\partial x_j}}{L_e} & \text{si } x_i \equiv y_e^{ini} \\ \frac{\cos \beta \frac{\partial \cos \gamma}{\partial x_j} + \cos \gamma \frac{\partial \cos \beta}{\partial x_j} - \frac{\cos \beta \cos \gamma}{L_e} \frac{\partial L_e}{\partial x_j}}{L_e} & \text{si } x_i \equiv y_e^{fin} \\ \frac{2 \cos \gamma \frac{\partial \cos \gamma}{\partial x_j} + \frac{1 - \cos^2 \gamma}{L_e} \frac{\partial L_e}{\partial x_j}}{L_e} & \text{si } x_i \equiv z_e^{ini} \\ \frac{2 \cos \gamma \frac{\partial \cos \gamma}{\partial x_j} + \frac{1 - \cos^2 \gamma}{L_e} \frac{\partial L_e}{\partial x_j}}{L_e} & \text{si } x_i \equiv z_e^{fin} \end{cases} \quad (5.15)$$

5.5.5 Resolución del problema de optimización de geometría

El diagrama de flujo del algoritmo de optimización de geometría desarrollado e implementado es el de la Fig. 5.7.

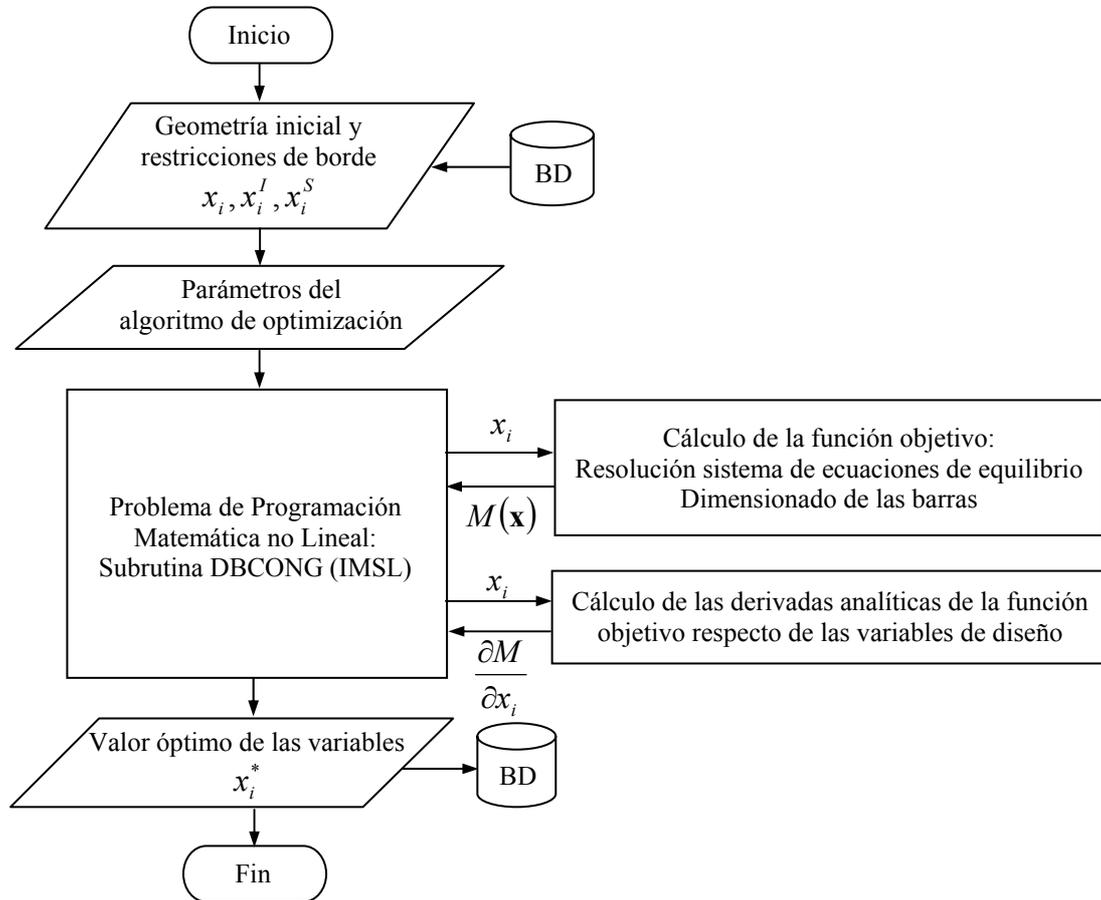


Figura 5.7 Diagrama de flujo para la optimización de geometría

Para la resolución del problema de Programación Matemática no Lineal, al que se llega con la formulación propuesta, se utiliza la subrutina DBCONG (Schittkowski, 1986) de la librería matemática IMSL[®]. Esta subrutina usa un método quasi-Newton y una estrategia de conjunto activo de restricciones para resolver problemas de optimización sujetos a restricciones de borde simples para las variables. Las derivadas de la función objetivo respecto de las variables pueden ser suministradas por el usuario (subrutina DBCONG) o calculadas por diferencias finitas (subrutina DBCONF). En esta tesis se utilizan las derivadas analíticas del epígrafe 5.5.3. Los detalles de este algoritmo se han expuesto en el epígrafe 2.5.4.3.

5.6 CRECIMIENTO DE LA TOPOLOGÍA

5.6.1 Introducción

En el método de crecimiento propuesto se sigue un proceso iterativo en el que, en cada iteración, se modifica la topología añadiendo un nudo (dos si se imponen condiciones de simetría) y varias barras que unen dicho(s) nudo(s) con el resto de nudos de la estructura.

En primer lugar, se definen las estrategias utilizadas para la definición de los nuevos nudos. A continuación se determinan las nuevas barras a añadir. Por último se muestran los criterios adoptados para la elección de la barra o el cruce de barras a dividir.

5.6.2 Definición de los nuevos nudos

Para elegir la posición del punto donde añadir un nuevo nudo se prueban diferentes posiciones y se elige aquella que, después de optimizar la topología, produce una menor función objetivo.

A continuación se enumeran las diferentes estrategias utilizadas para la definición de las posiciones de los puntos donde probar a añadir el nuevo nudo.

5.6.2.1 Método general

Por método general se entiende aquel que no sigue ningún criterio de optimalidad, que deba cumplir la estructura óptima, para limitar el número de puntos a probar.

Los criterios que se han probado para definir las posiciones de los puntos son los de los cuatro epígrafes siguientes.

5.6.2.1.1 Puntos de una rejilla en el dominio de definición de la estructura

Se calculan y comparan los valores de la función objetivo en todos los puntos de una rejilla que cubre todo el dominio de definición de la estructura. El dominio se divide de forma uniforme en cada una de las dimensiones; los puntos que delimitan estas divisiones forman la rejilla.

La función objetivo es muy dependiente de la posición del punto, de ahí que para obtener buenos resultados es necesario utilizar rejillas muy tupidas, lo que da lugar a un coste computacional muy alto.

5.6.2.1.2 Puntos aleatorios en el dominio de definición de la estructura

Se calculan y comparan los valores de la función objetivo en un número determinado de puntos aleatorios dentro del dominio de definición de la estructura. Este criterio ha sido utilizado por McKeown (1998).

Por el mismo razonamiento que en el criterio anterior es necesario un gran número de puntos aleatorios para obtener buenos resultados, lo que da lugar a un coste computacional muy alto.

5.6.2.1.3 Entorno de los puntos medios de las barras

Se calculan y comparan los valores de la función objetivo en un número determinado de puntos del entorno de los puntos medios de las barras. Este criterio ha sido utilizado por Rule (1994) y Bojczuk y Mróz (1998 y 1999).

Si se elige el punto medio de las barras, y sólo hay restricciones de tensión, no variará la función objetivo, de ahí que se desplace una cierta cantidad (por defecto se toma un 1% de la longitud de la barra) en la dirección perpendicular a la barra (Fig. 5.8).

Se han conseguido buenos resultados con este criterio.

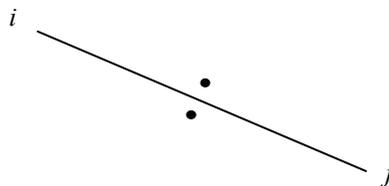


Figura 5.8 Entorno de los puntos medios de las barras

5.6.2.1.4 Entorno de los cruces de las barras

Se calculan y comparan los valores de la función objetivo en un número determinado de puntos del entorno de los puntos de cruce de las barras (Fig. 5.9). En cierto modo, este criterio es una variante del anterior.

Si se elige el punto de cruce de las barras, y sólo hay restricciones de tensión, no variará la función objetivo, de ahí que se desplace una cierta cantidad (por defecto se toma un 1% de la longitud media de las barras) en la dirección de las bisectrices de las barras (Fig. 5.9).

Se han conseguido buenos resultados con este criterio.

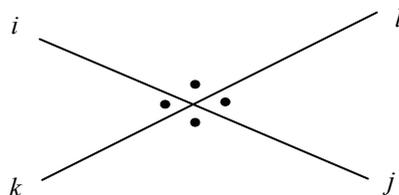


Figura 5.9 Entorno de los cruces de las barras

5.6.2.2 Método basado en las direcciones principales de tensión

La optimización de topología de estructuras articuladas (planas o tridimensionales), sometidas a restricciones de tensión, puede asimilarse a un problema de elasticidad.

Para que las tensiones normales sean máximas deben ser tensiones principales; por lo tanto, en la estructura óptima ideal las barras traccionadas (tensión principal máxima) que llegan a un nudo deben ser perpendiculares a las barras comprimidas (tensión principal mínima) que llegan a ese mismo nudo.

Este criterio de ortogonalidad de las barras en los nudos es el que se ha tenido en cuenta a la hora de elegir los puntos donde añadir nuevos nudos para limitar la búsqueda.

5.6.2.2.1 Mejora de la ortogonalidad de las barras

Si se modifica la topología de la estructura articulada, de manera que los ángulos que forman las barras que convergen en un nudo y que tienen esfuerzos axiales opuestos se acerquen más al ángulo recto, la función objetivo normalmente disminuye (sobre todo si sólo hay restricciones de tensión).

Para mejorar la ortogonalidad de las barras se han seguido dos criterios:

- 1 división de una barra, y
- 2 división de dos barras que se cruzan.

5.6.2.2.2 División de una barra

Una forma de mejorar la ortogonalidad de las barras en los nudos consiste en dividir una barra en dos añadiendo un nudo intermedio, añadir una nueva barra (en estructuras planas) para mantener la isostaticidad, y mover el nudo intermedio de forma que mejoren los ángulos en los dos nudos de la barra que se divide.

En el óptimo continuo, las barras que convergen en un nudo y tienen esfuerzo axial de diferente signo deben ser ortogonales y las del mismo signo alineadas. Como en la práctica esto no ocurre, se puede comparar el ángulo de la barra que se quiere dividir con el ángulo de la bisectriz de las dos barras que convergen en los nudos de los extremos de esta barra y trabajan en sentido opuesto al de la barra a dividir. Moviendo el nudo intermedio hacia la zona del mayor de los ángulos de las dos bisectrices, mejorará la ortogonalidad de las barras en los dos nudos.

Cuando en un nudo hay más de dos barras con esfuerzo axial opuesto al de la barra que se quiere dividir, se consideran las dos que forman el menor ángulo a cada lado con la barra. Si hay menos de dos, se supone que la que falta es perpendicular a la barra a dividir.

En la Fig. 5.10 se muestra un ejemplo de cómo mejorar la ortogonalidad de las barras, al dividir una barra, cuando en cada uno de sus nudos están unidas dos barras con esfuerzos axiales de diferente signo al de la barra a dividir. Se trazan las bisectrices de las barras de los extremos y la que forma un ángulo mayor con la barra a dividir es la que define la zona hacia donde hay que desplazar el nudo añadido.

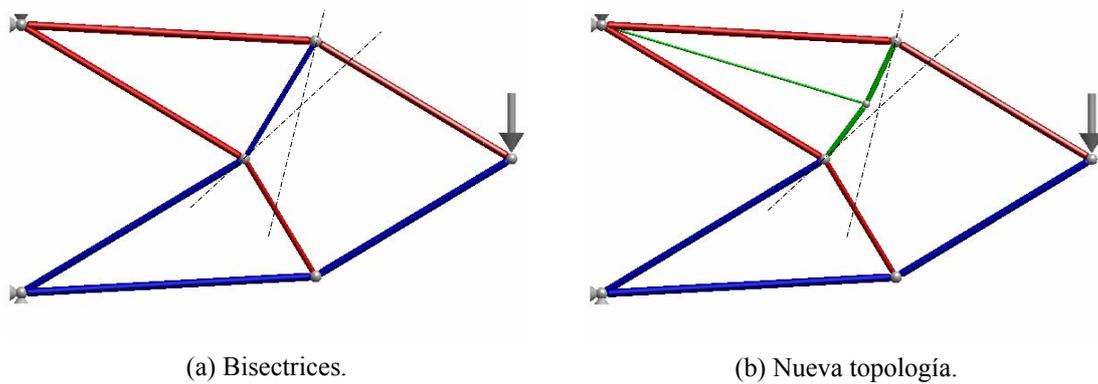


Figura 5.10 Mejora de la ortogonalidad al dividir una barra. En cada uno de los nudos de la barra están unidas dos barras con esfuerzos axiales de diferente signo al suyo

En la Fig. 5.11 se muestra un ejemplo de cómo mejorar la ortogonalidad de las barras, al dividir una barra, cuando en cada uno de sus nudos está unida una barra con esfuerzos axiales de diferente signo al de la barra a dividir. La barra que falta se supone que es perpendicular a la barra a dividir (línea de puntos azul). Una vez que se tienen las dos barras en cada extremo se trazan las bisectrices y se sigue el mismo criterio que en el ejemplo anterior.

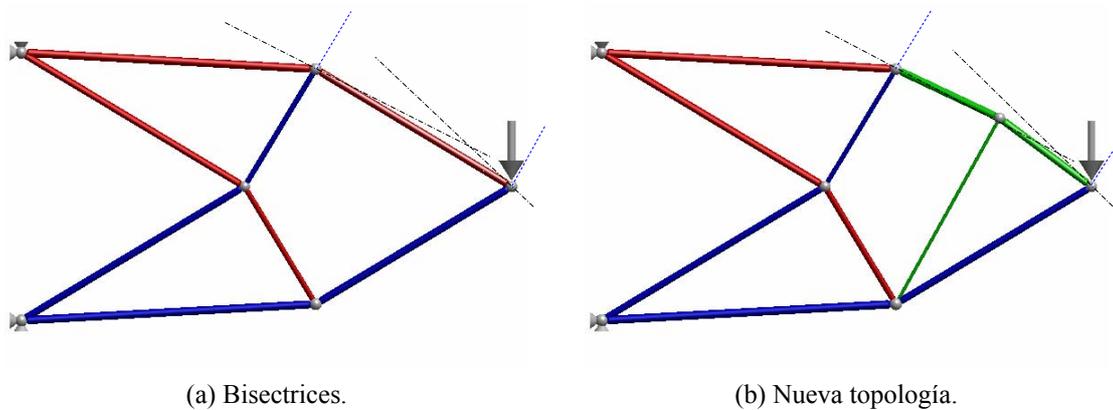


Figura 5.11 Mejora de la ortogonalidad al dividir una barra. En cada uno de los nudos de la barra está unida una barra con esfuerzo axial de diferente signo al suyo

En la Fig. 5.12 se muestra un ejemplo de cómo mejorar la ortogonalidad de las barras, al dividir una barra, cuando en uno de sus nudos no hay ninguna barra conectada con esfuerzos axiales de diferente signo al suyo, y en el otro nudo, sí que hay una. Las barras que faltan se supone que son perpendiculares a la barra a dividir. En el extremo inicial la bisectriz es paralela a la barra a dividir. Una vez que se tienen las dos barras en cada extremo se trazan las bisectrices y se sigue el mismo criterio que en los ejemplos anteriores.

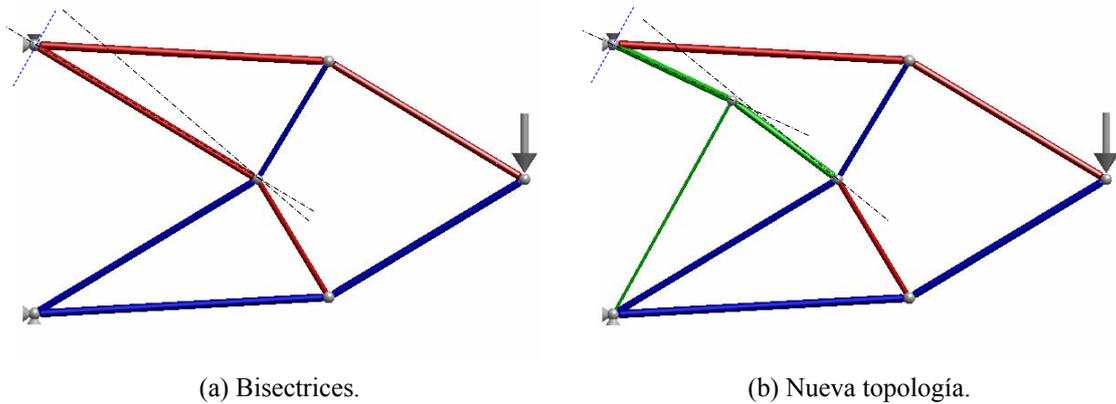


Figura 5.12 Mejora de la ortogonalidad al dividir una barra. En uno de los nudos de la barra no hay ninguna barra conectada con esfuerzo axial de diferente signo al suyo y en el otro hay una

En las Figs. 5.10 a 5.12 se aprecia que el ángulo que forman las barras en las nuevas topologías con las bisectrices es menor que el que forman con la topología anterior, es decir, ha mejorado la ortogonalidad.

5.6.2.2.3 División de dos barras que se cruzan

El procedimiento utilizado es una variante del anterior, la diferencia es que se considera la división de dos barras a la vez y el nuevo nudo es común. La zona correcta donde insertar el nuevo nudo será la zona común a las zonas de las dos barras. En este caso no hay que añadir ninguna nueva barra, ya que el número de grados de libertad aumenta en dos (en estructuras planas), se eliminan dos barras y se añaden cuatro, con lo que la nueva topología sigue siendo isostática.

En la Fig 5.13 se muestra un ejemplo donde se aprecia la mejora en la ortogonalidad de las barras al dividir dos barras que se cruzan en cuatro y mover el nuevo nudo en la dirección correcta.

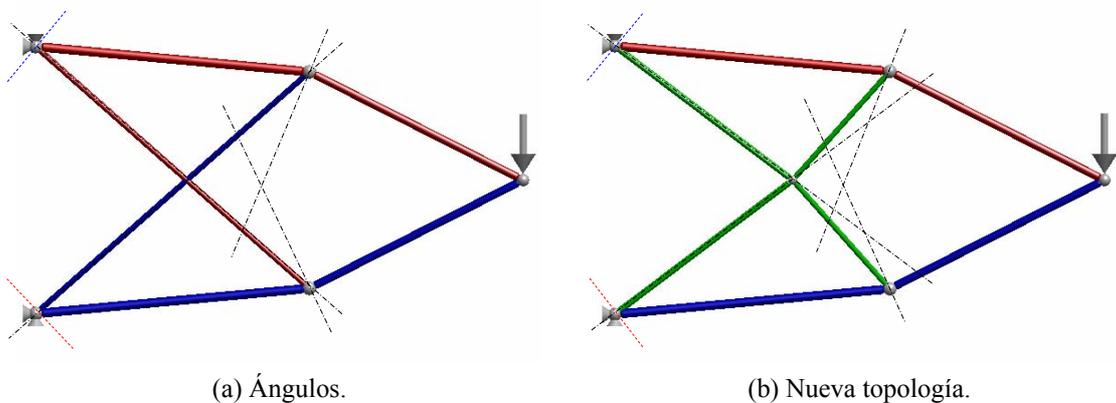


Figura 5.13 Mejora de la ortogonalidad al dividir dos barras que se cruzan

5.6.2.3 Elección de la posición del nuevo nudo

Ya se ha definido la zona hacia la que debe moverse el nuevo nudo para que mejore la ortogonalidad de la nueva topología. Falta definir la posición.

Para la elección de la posición del punto donde añadir un nuevo nudo, se prueban diferentes posiciones y se elige aquella que produce una menor función objetivo.

Para que la elección de la posición sea la correcta, se debería optimizar la topología y la geometría para cada posición probada. Como, normalmente, el número de posiciones probadas es alto, el coste computacional es elevado. Para limitar el coste, en el algoritmo propuesto, se ha optado por optimizar la topología de todas las posiciones probadas, pero optimizar únicamente la geometría de la topología elegida. Esto hace que la posición de los puntos donde probar a añadir el nuevo nudo sea determinante en la elección de la nueva topología.

A continuación se enumeran los criterios usados para la elección de la posición donde añadir el nuevo nudo para los dos casos descritos en el epígrafe anterior.

5.6.2.3.1 División de una barra

Al dividir una barra se toma como dirección donde situar el nuevo nudo la perpendicular a la barra a dividir por su punto medio, y como sentido el que mejora la ortogonalidad, tal como se ha definido anteriormente.

Para elegir la posición exacta del nuevo nudo se han desarrollado y probado cuatro criterios:

- 1 Tomando una aproximación polinómica cúbica que pasa por los dos extremos de la barra y cuyas pendientes en dichos extremos son las de las bisectrices (m_i y m_j), se elige el valor de la aproximación en el punto medio (Fig. 5.14).

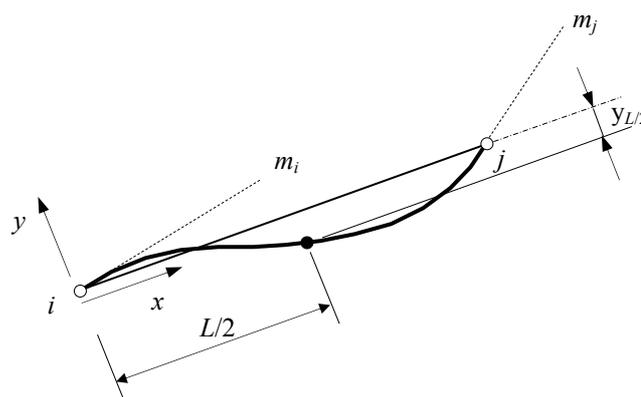


Figura 5.14 Valor medio para una aproximación polinómica cúbica

La aproximación polinómica cúbica es

$$y = L \left(m_i \frac{x}{L} - (2m_i + m_j) \left(\frac{x}{L} \right)^2 + (m_i + m_j) \left(\frac{x}{L} \right)^3 \right)$$

y el valor en el punto medio

$$y(x = L/2) = \frac{1}{8}(m_i - m_j)$$

- 2 El mismo criterio anterior pero eligiendo el punto de mayor distancia para la aproximación polinómica cúbica (Fig. 5.15).

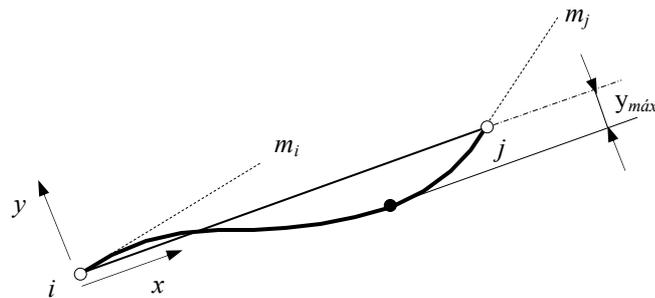


Figura 5.15 Valor máximo para una aproximación polinómica cúbica

- 3 La media de las distancias del primer criterio (la misma para todas las barras). Con este criterio se han obtenido mejores resultados que con los anteriores.
- 4 Una fracción de la longitud de la barra, aplicada en la perpendicular por el punto medio de la barra (la misma para todas las barras).

Con el último criterio se han obtenido buenos resultados (por defecto se usa el 1% de la longitud de la barra, valor para el que se obtienen buenos resultados).

5.6.2.3.2 División de dos barras que se cruzan

Al dividir dos barras que se cruzan se toma como dirección donde situar el nuevo nudo la bisectriz de las barras que se cruzan, en el sentido en el que mejora la ortogonalidad, ya definido anteriormente.

Se han probado los mismos criterios que en el epígrafe anterior, y con una fracción de la media de las longitudes de las barras que se cruzan (por defecto se usa el 1% de la media) se han obtenido buenos resultados.

5.6.3 Elección de las nuevas barras a añadir

Una vez determinada la posición del nuevo nudo, se añaden barras que unen este nudo con todos o parte de los nudos restantes.

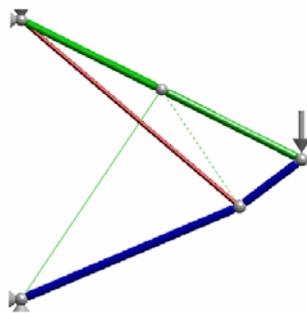
Para los casos particulares de división de una barra o división de dos barras que se cruzan se pueden considerar ciertas particularidades a la hora de añadir las barras.

5.6.3.1 División de una barra

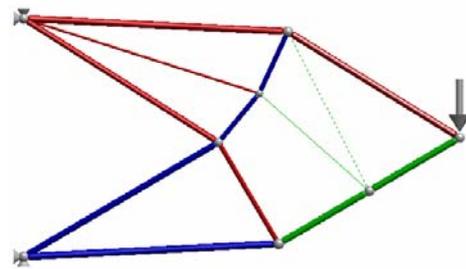
Para mantener la isostaticidad al dividir una barra hay que añadir una nueva barra. Para ello se une el nuevo nudo con alguno de los restantes.

La mayoría de los investigadores, Rule (1992), Bojczuk y Mróz (1998), recurren al nudo más cercano (barra más corta), pero éste no siempre es el mejor. Teniendo en cuenta el criterio de ortogonalidad, se podría pensar en unirlo con el nudo más cercano a la perpendicular, pero éste tampoco es siempre el mejor ya que el mejor nudo depende de la longitud de la barra, de que el otro nudo sea fijo o no, etc.

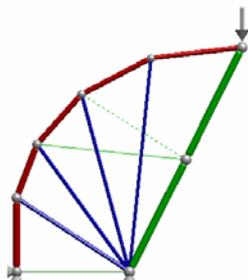
En la Fig. 5.16 se muestran varios ejemplos, con la mejor barra en línea continua y la más corta o la más ortogonal en línea discontinua, en los que puede verse que la mejor barra no es siempre la más cercana a la perpendicular o la más corta.



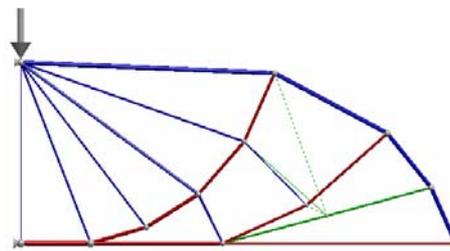
(a) La mejor barra es la más perpendicular pero no la más corta.



(b) La mejor barra es la más corta pero no la más perpendicular.



(c) La mejor barra no es la más corta y perpendicular.



(d) La mejor barra no es ni la más perpendicular ni la más corta.

Figura 5.16 Ejemplos donde la mejor barra a añadir no es la más perpendicular o la más corta

Hasta ahora no se ha encontrado un criterio sencillo que defina siempre la mejor barra a añadir, de ahí que, en el método propuesto se haya recurrido a añadir varias y, mediante una optimización de topología, decidir cuál es la mejor.

Para asegurar que no se va a descartar la barra correcta se pueden añadir todas las barras posibles uniendo el nuevo nudo con todos los demás (Fig. 5.17).

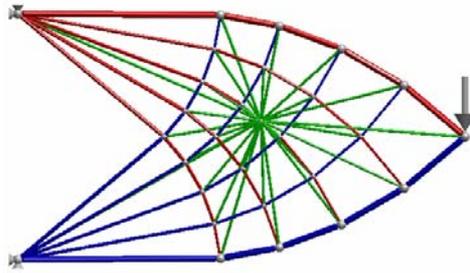


Figura 5.17 Universo estructural completo del nuevo nudo al dividir una barra

Si se tiene en cuenta que la nueva barra debe ser, aproximadamente, ortogonal a las dos en las que se divide la barra eliminada, el esfuerzo axial debe ser de signo opuesto, lo que descarta a las barras que están al mismo lado de la barra que el nuevo nudo. El nuevo universo puede verse en la Fig. 5.18.

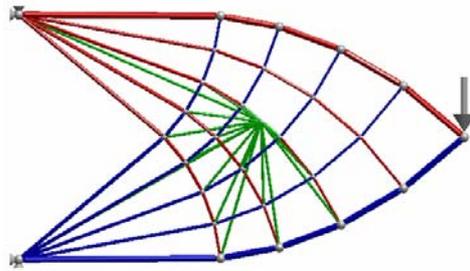


Figura 5.18 Universo estructural del lado opuesto del nuevo nudo al dividir una barra

Una vez obtenido el universo estructural (Fig. 5.18), se optimiza para obtener la topología óptima al dividir una barra. En la Fig. 5.19 se muestra la topología óptima obtenida.

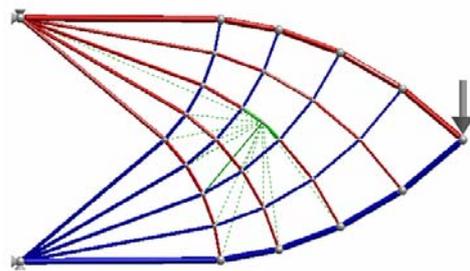


Figura 5.19 Topología óptima al dividir una barra

5.6.3.2 División de dos barras que se cruzan

Para elegir las nuevas barras a añadir, al dividir dos barras que se cruzan, se sigue el mismo procedimiento que al dividir una barra, pero utilizando un solo nudo (el punto de intersección de las barras) y todas las barras (las que se añadirían de forma independiente al dividir cada una de las dos barras que se cruzan).

En la Fig. 5.20 se muestran ejemplos de las nuevas topologías al dividir dos barras que se cruzan.

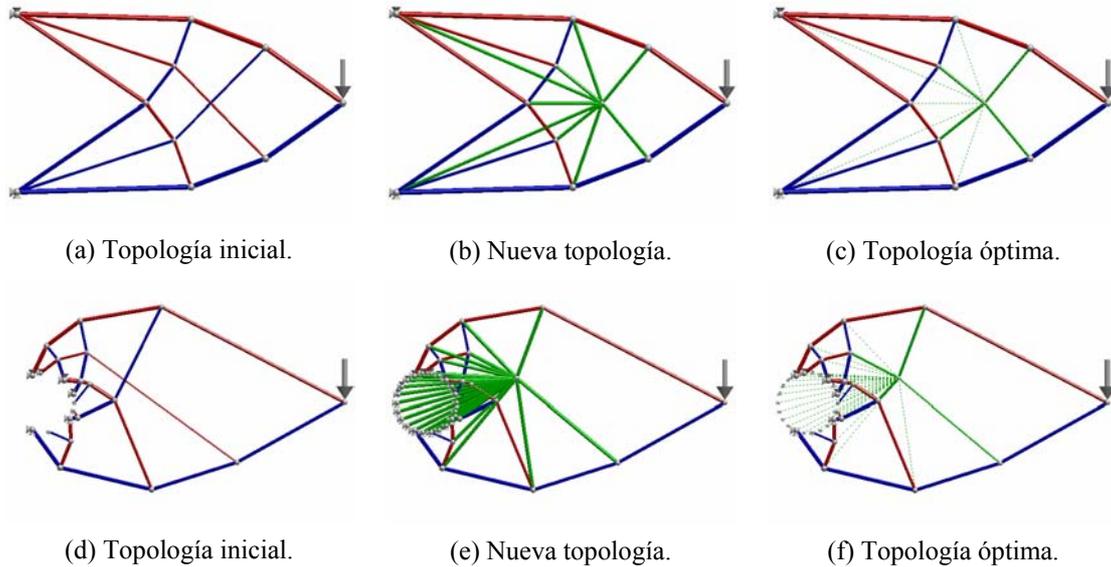


Figura 5.20 Topologías al dividir dos barras que se cruzan

5.6.3.3 Criterios prácticos para limitar el número de barras a añadir al nuevo nudo

Si se quiere limitar el coste de la optimización de topología se pueden aplicar algunos criterios prácticos que permiten limitar el número de barras a añadir con el nuevo nudo. Estos criterios deben usarse con cierta precaución para evitar eliminar la barra correcta.

5.6.3.3.1 Limitar el número de cruces de barras

En las estructuras reales es conveniente evitar el cruce de barras. Esto hace pensar en no permitir que los haya; pero hay ejemplos donde la barra óptima se cruza con alguna otra (Figs. 5.16a, 5.16c y 5.16d). Pero sí que parece razonable limitar el número máximo de cruces de barras. En la Fig. 5.21 se muestra el nuevo universo limitando el número máximo de cruces a uno.

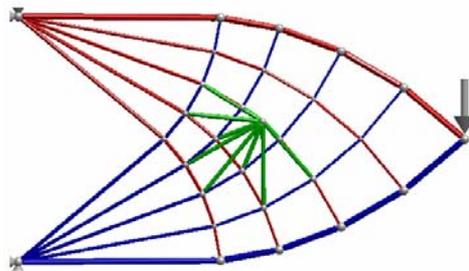


Figura 5.21 Universo estructural del nuevo nudo con un cruce

5.6.3.3.2 Limitar el número máximo de barras en exceso

Se puede limitar el número máximo de barras en exceso (grado de hiperestaticidad) que se van a añadir a la nueva topología. Para elegir las del universo que cumple el criterio de reducción anterior, hay que fijar un criterio de preferencia.

Teniendo en cuenta el criterio de ortogonalidad de las barras, parece lógico que las barras más cercanas a la perpendicular a la barra a eliminar tengan más posibilidades de formar parte de la topología óptima. El criterio elegido en el método propuesto es seleccionar las barras cuyos ángulos con la perpendicular a la barra a eliminar sean los más pequeños. En la Fig. 5.22 se muestra el nuevo universo limitando el grado de hiperestaticidad máximo a tres.

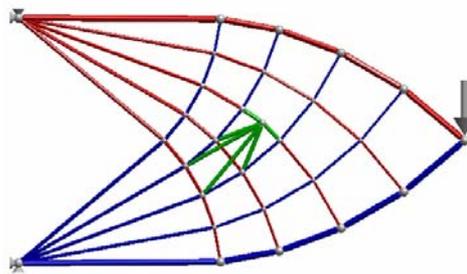
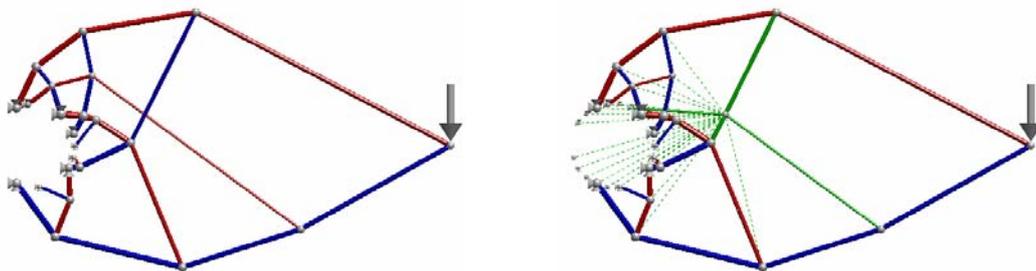


Figura 5.22 Universo estructural del nuevo nudo con un cruce y un grado de hiperestaticidad máximo de 3

En la Fig. 5.16 ya se han mostrado casos en los que un grado de hiperestaticidad cero no es suficiente. Hay situaciones extremas en las que alguna de las barras en las que se divide la barra original no forma parte de la topología óptima (Fig. 5.23); en estas situaciones el grado de hiperestaticidad máximo (dato) debe ser alto o no se debe limitar.



(a) Topología inicial.

(b) Topología óptima.

Figura 5.23 Topología en la que al dividir dos barras que se cruzan, alguna de las barras en las que se dividen las originales no forma parte de la topología óptima

5.6.4 Elección de la barra o cruce de barras a dividir

En los epígrafes anteriores, y dada una barra o un cruce de barras a dividir, se ha definido el proceso de crecimiento de la topología. Falta seleccionar qué barra o cruce de barras son los más adecuados en dicho proceso de crecimiento.

Hasta el momento no se ha encontrado un criterio sencillo, que sea válido siempre, para la elección de la mejor barra o cruce de barras a dividir.

Algunos autores, Rule (1992) por ejemplo, dividen la barra más larga, pero ésta no es siempre la mejor decisión. En la Fig. 5.24 se muestran dos ejemplos en los cuales la mejor barra a dividir no es la más larga.

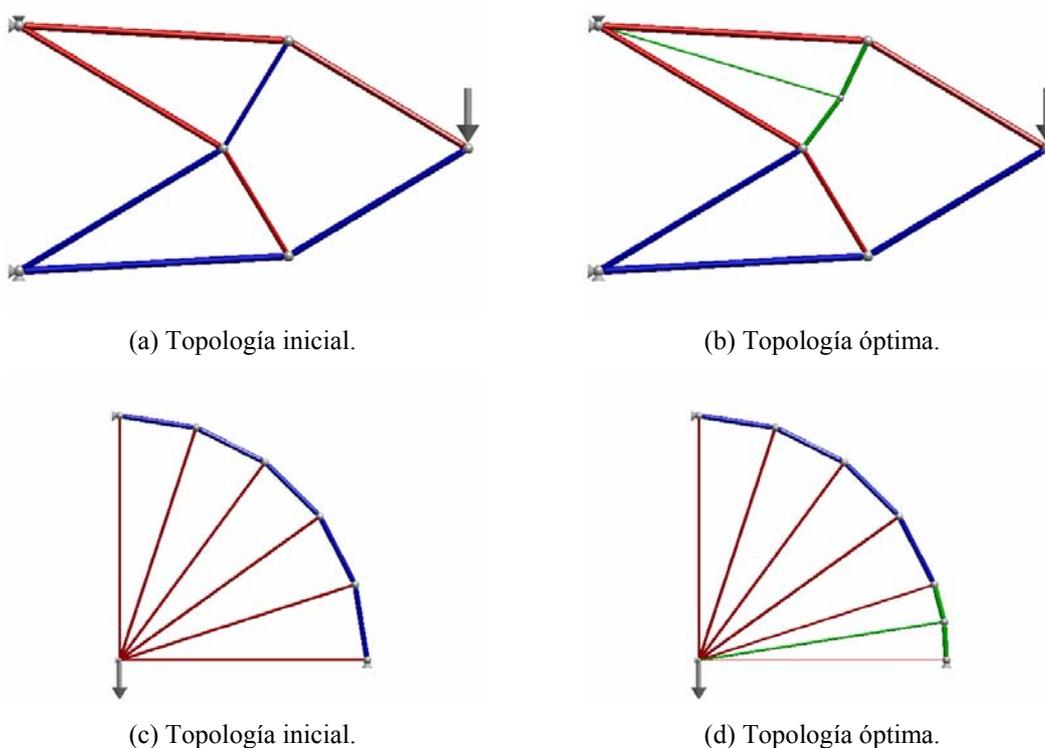


Figura 5.24 Ejemplos donde la mejor barra a dividir no es la más larga

En el método propuesto se prueban todas las barras y todos los cruces de barras, y se elige aquella con la que menor función objetivo se obtiene. En la Fig. 5.25 se muestran la topología inicial y la topología óptima cuando se añade un nuevo nudo al dividir una barra o un cruce de barras en las cuatro primeras iteraciones del ejemplo del voladizo de Michell. En la tercera iteración (Figs. 5.25e y 5.25f) la mejor opción es dividir un cruce de barras, en las otras tres la mejor opción es dividir una barra.

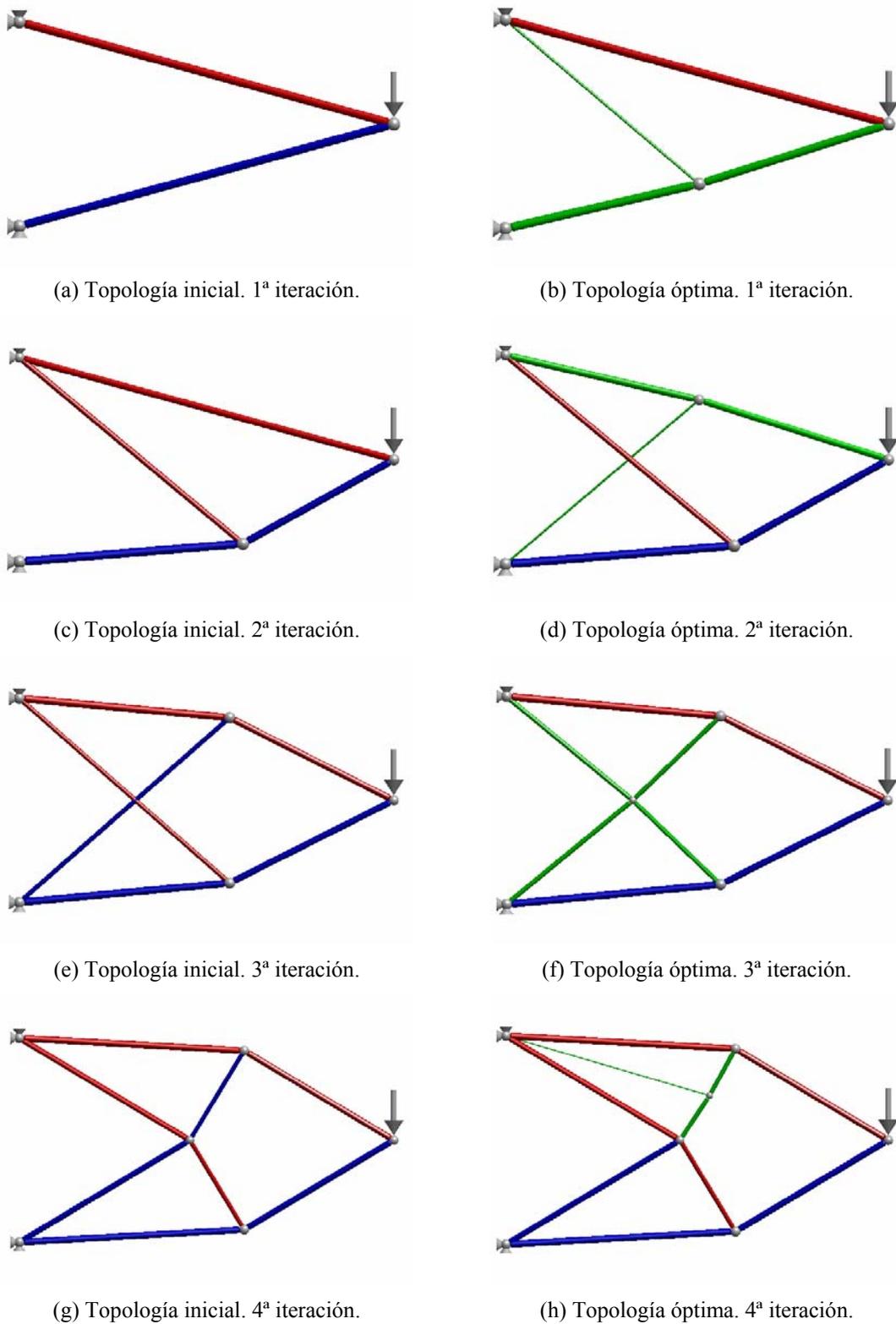


Figura 5.25 Mejor barra o cruce de barras a dividir para obtener la nueva topología óptima. Cuatro primeras iteraciones del voladizo de Michell

5.6.5 Resumen del proceso de crecimiento de la topología

El diagrama de flujo para el proceso de crecimiento de la topología expuesto en los epígrafes 5.6.2 a 5.6.4 se muestra en la Fig. 5.26.

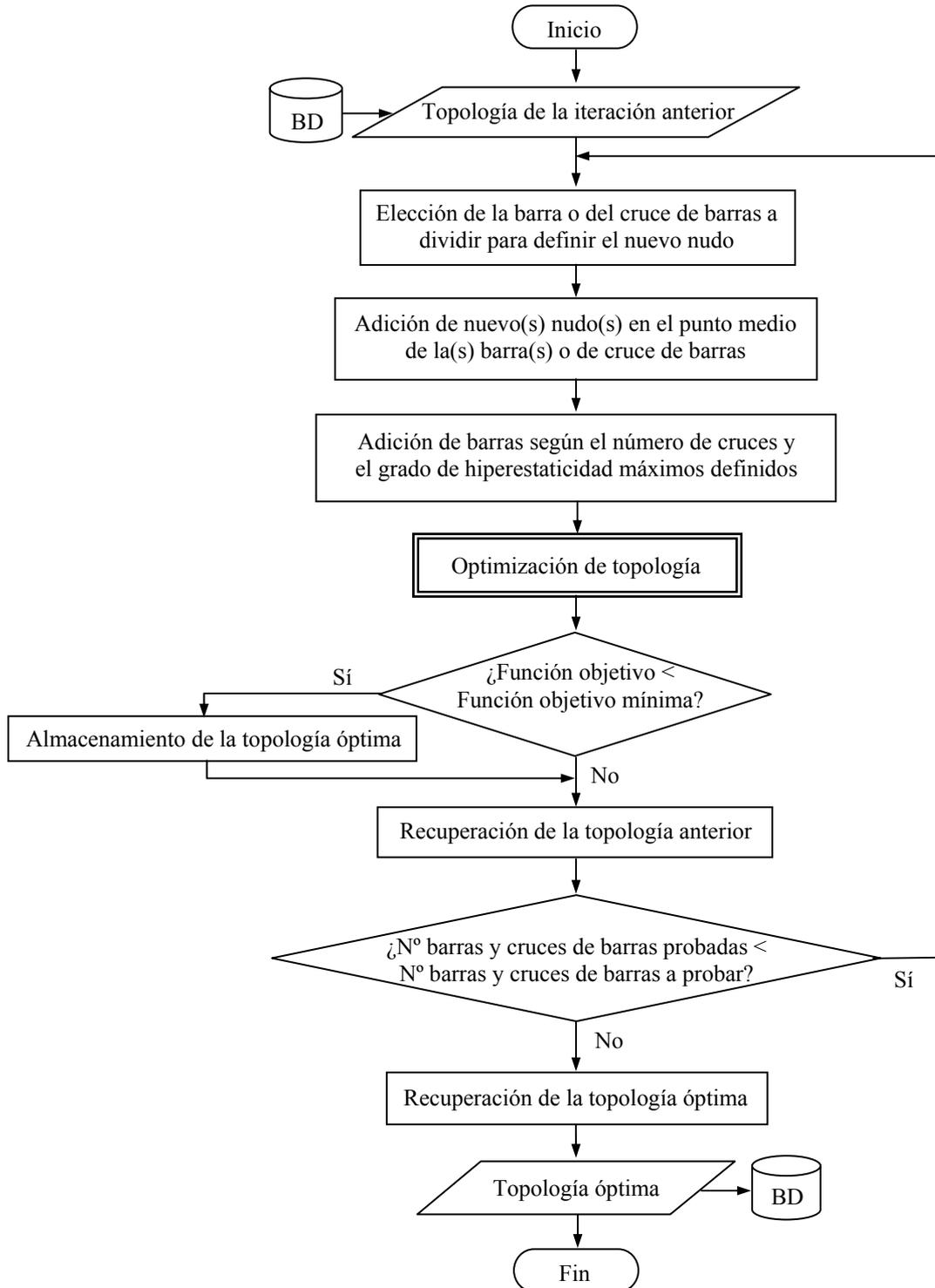


Figura 5.26 Diagrama de flujo para el proceso de crecimiento de la topología

5.7 COMPROBACIÓN DE LA TOPOLOGÍA ÓPTIMA EN LA NUEVA GEOMETRÍA

En cada iteración del proceso de optimización se optimiza la topología y a continuación se optimiza la geometría. Al moverse los nudos, como consecuencia de la optimización de geometría, puede ocurrir que la topología que se había escogido ya no sea la óptima, de ahí que sea necesario comprobar si optimizando de nuevo la topología se mejora el diseño óptimo.

En la Fig. 5.27 se muestra el diagrama de flujo para la comprobación de la topología óptima en la nueva geometría.

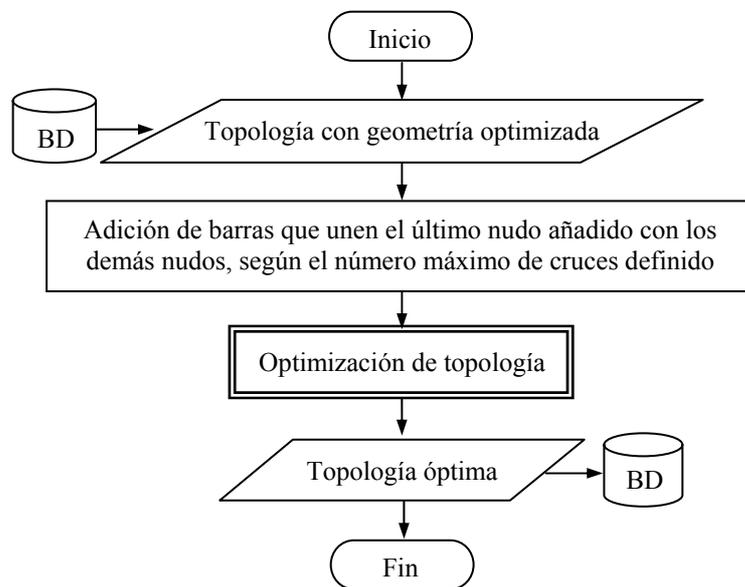


Figura 5.27 Diagrama de flujo para la comprobación de la topología óptima en la nueva geometría

Normalmente el nudo que más se mueve en la optimización de geometría, es el último nudo añadido, ya que la posición de los demás nudos se ha optimizado en iteraciones anteriores, y al añadir un nuevo nudo, la posición de los demás, normalmente, va a cambiar poco.

Por lo tanto, una buena estrategia para encontrar la nueva topología óptima (si es que ha cambiado) es generar un nuevo universo estructural, añadiendo barras que unan el último nudo añadido con los demás, teniendo en cuenta la limitación del número máximo de cruces. Si la topología óptima ha cambiado al optimizar la geometría, al optimizar esta última topología se obtendrá una topología diferente. En este caso, es aconsejable volver a optimizar la geometría.

En la Fig. 5.28 se puede ver un ejemplo donde cambia la topología óptima al optimizar la geometría. En la Fig. 5.28c se aprecia que la topología óptima al dividir una barra no une el nuevo nudo con el más cercano sino con el apoyo superior. Una vez optimizada la geometría (Fig. 5.28d) si se añaden barras que unen el último nudo con los demás (Fig. 5.28e) y se optimiza la topología (Fig. 5.28f), se observa que la topología óptima sí une el último nudo con el más cercano, es decir, la topología óptima ha cambiado al optimizar la geometría.

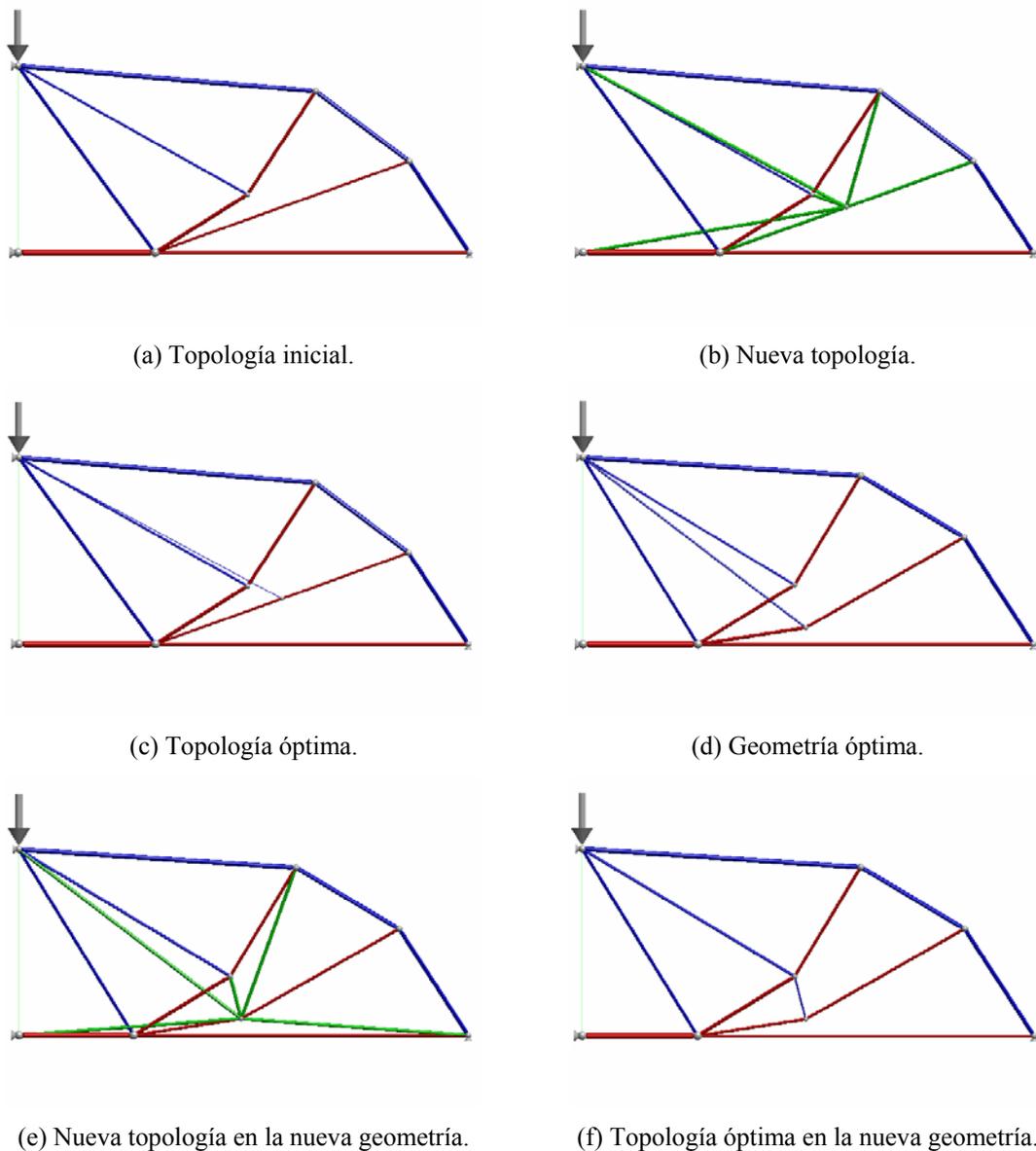


Figura 5.28 Cambio de la topología óptima al optimizar la geometría

5.8 EJEMPLOS

En este epígrafe se muestran las primeras iteraciones de la evolución del método propuesto para varios ejemplos.

5.8.1 Viga con apoyos fijos

En la Fig. 5.29 puede verse la evolución, en las tres primeras iteraciones, del diseño óptimo obtenido al aplicar el método propuesto a una viga con apoyos fijos (para más detalles ver el capítulo de ejemplos) con un número de cruces máximo de 2 (valor por defecto) y un grado de hiperestaticidad máximo de 2 (valor por defecto).

En la Fig. 5.29a se muestran los datos iniciales, en la 5.29b los nudos iniciales, y en la 5.29c la topología inicial, formada mediante el universo estructural a partir de todos los nudos. Una vez creada la topología inicial se realiza una optimización de topología (Fig. 5.29d) y geometría (Fig. 5.29e), puesto que el nudo superior tiene permitido el movimiento vertical. Con esto concluye la primera iteración, resultando la estructura de la Fig. 5.29e.

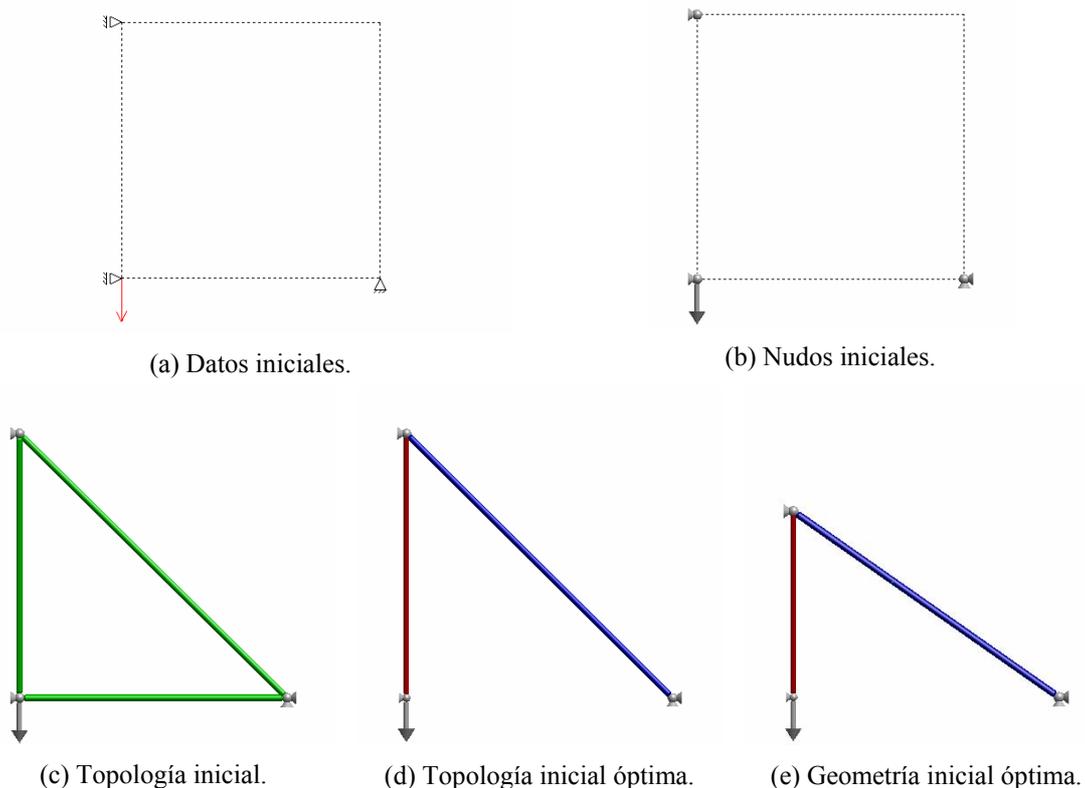


Figura 5.29 Evolución del diseño óptimo en las tres primeras iteraciones. Viga con apoyos fijos

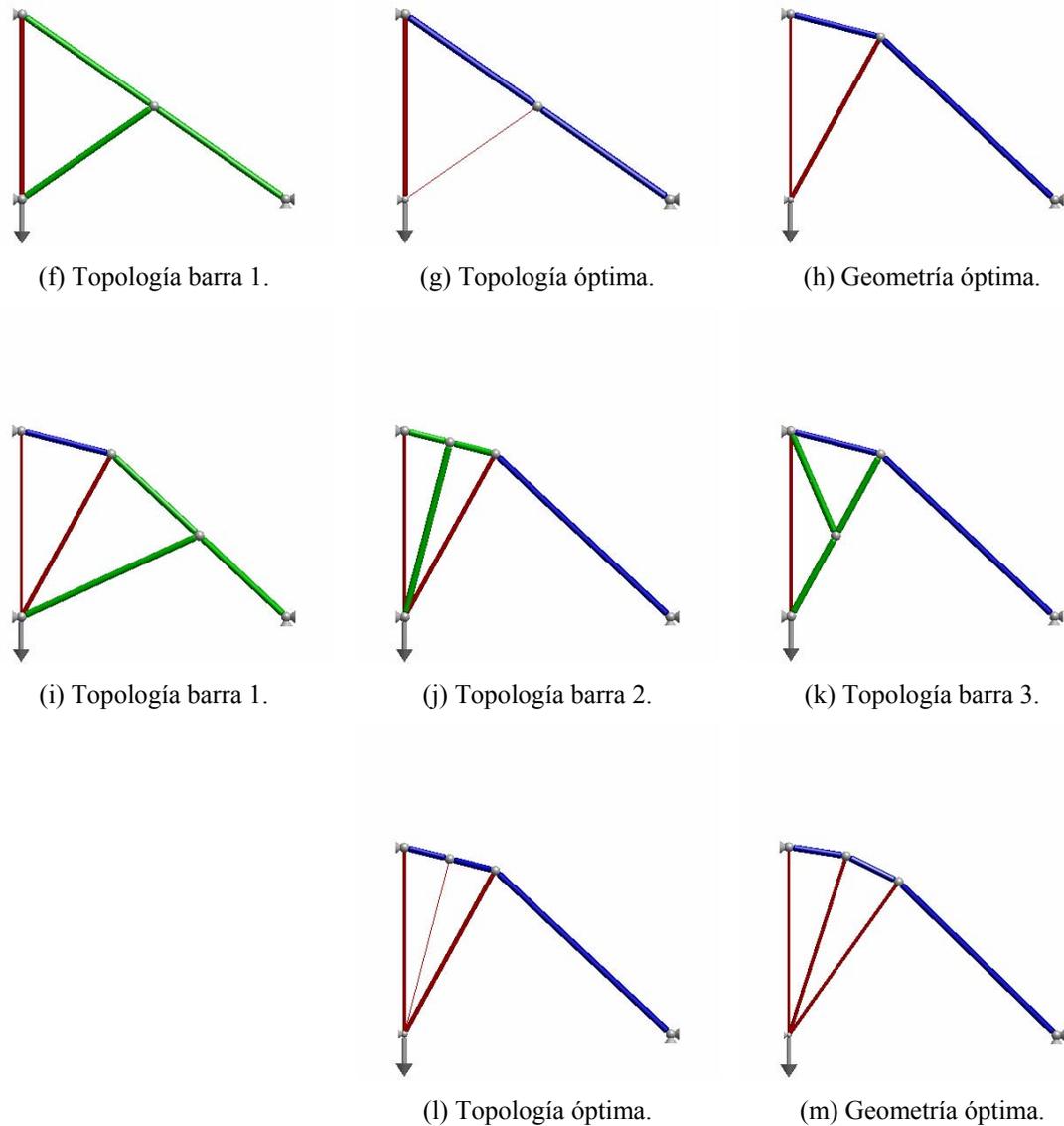


Figura 5.29 Evolución del diseño óptimo en las tres primeras iteraciones. Viga con apoyos fijos.
(Continuación)

En la segunda iteración se comienza probando a dividir cada una de las barras (Fig. 5.29f, la barra vertical no se divide ya que la dirección correcta hacia donde hay que mover el nudo intermedio es hacia la izquierda y la restricción de borde lo impide), se elige la mejor (Fig. 5.29g, la única) y se optimiza la geometría (Fig. 5.29h).

En la tercera iteración se repite el mismo proceso, se prueba a dividir cada una de las barras (Figs. 5.29i, 5.29j y 5.29k), se elige la mejor (Fig. 5.29l) y se optimiza la geometría (Fig. 5.29m).

5.8.2 Viga con disco circular rígido

En la Fig. 5.30 puede verse la evolución, en las tres primeras iteraciones, del diseño óptimo obtenido al aplicar el método propuesto a una viga con disco circular rígido (para más detalles ver el capítulo de ejemplos) con un número de cruces máximo de 2 (valor por defecto) y un grado de hiperestaticidad ilimitado.

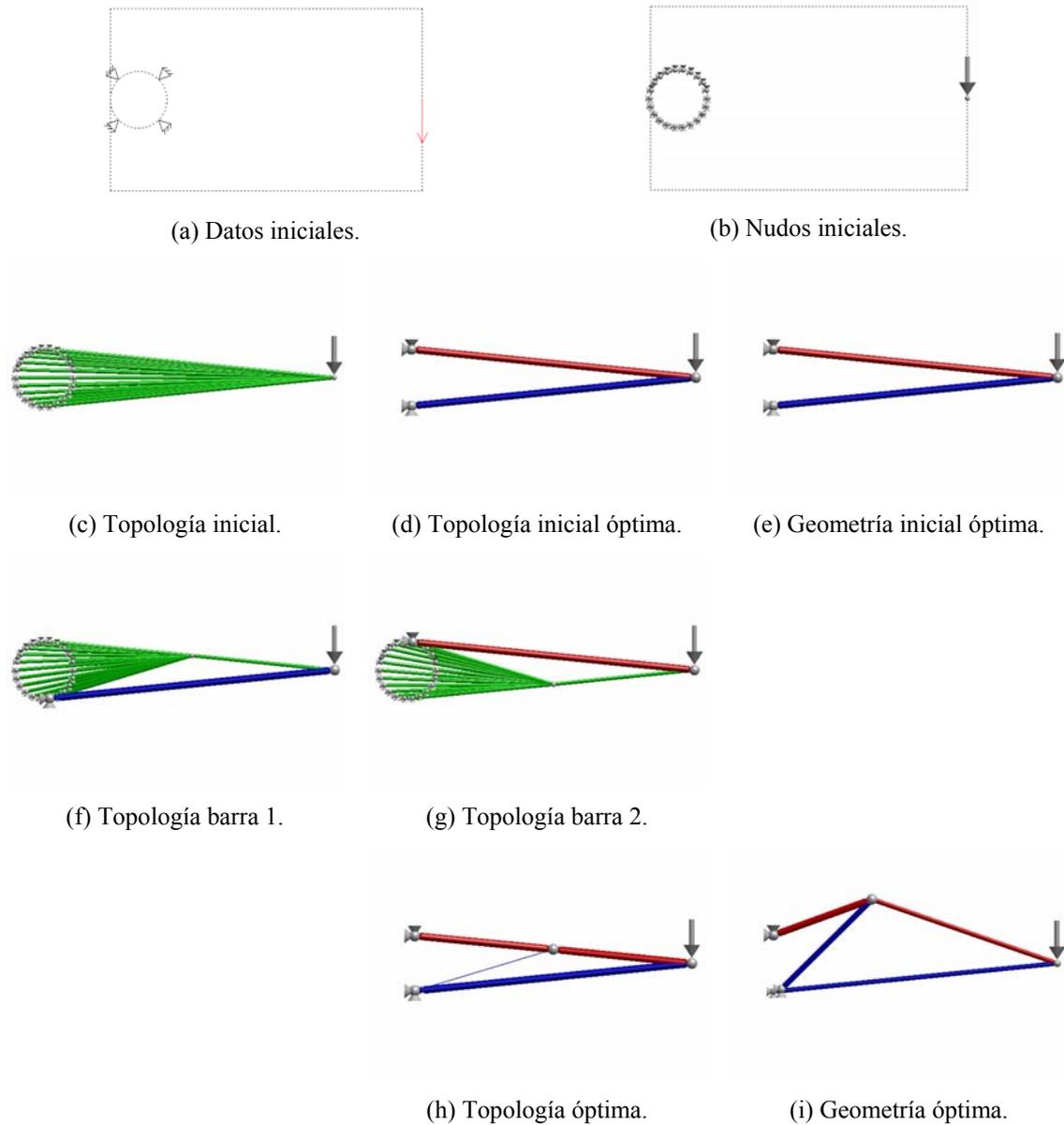


Figura 5.30 Evolución del diseño óptimo en las tres primeras iteraciones. Viga con disco circular rígido

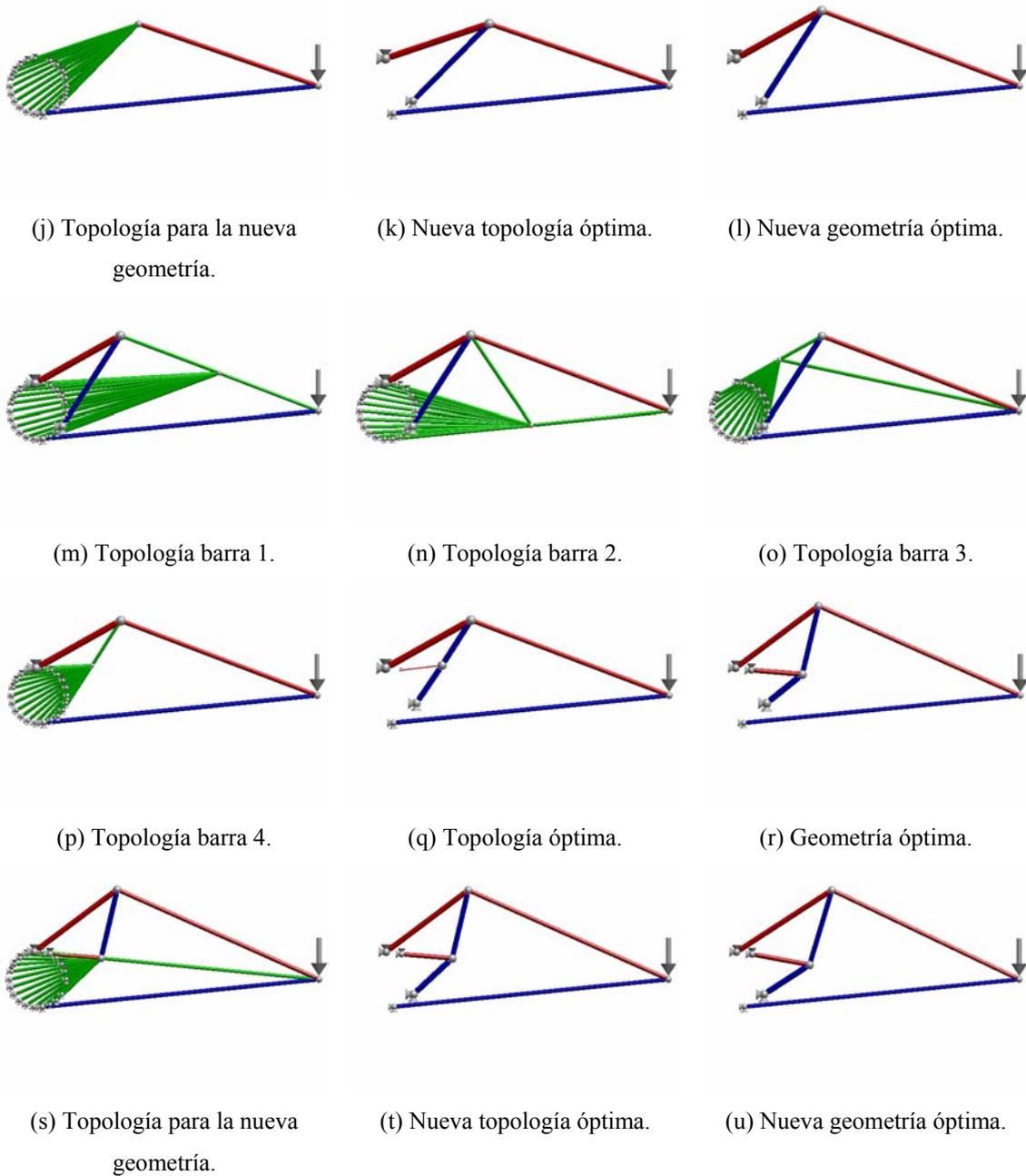


Figura 5.30 Evolución del diseño óptimo en las tres primeras iteraciones. Viga con disco circular rígido. (Continuación)

En la Fig. 5.30a se muestran los datos iniciales, en la 5.30b los nudos iniciales, en la 5.30c la topología inicial, correspondiente al universo estructural completo para todos los nudos. Una vez creado el universo estructural inicial se optimiza la topología (Fig. 5.30d) y la geometría (Fig. 5.30e). La geometría óptima coincide con la de la topología óptima, ya que, inicialmente, no hay ningún punto móvil. Con esto concluye la primera iteración.

En la segunda iteración se divide cada una de las barras (Figs. 5.30f y 5.30g), se elige la mejor (Fig. 5.30h) y se optimiza la geometría (Fig. 5.30i). En este ejemplo varía la topología óptima al optimizar la geometría (Figs. 5.30j y 5.30k); esto se detecta porque al añadir barras que unen el último nudo con los demás, teniendo en cuenta la limitación del número de cruces, y optimizando la topología, se obtiene una topología óptima diferente. Como la topología ha variado, se debe volver a optimizar la geometría (Fig. 5.30l), con la que termina la segunda iteración.

En la tercera iteración se divide cada una de las barras (Figs. 5.30m a 5.30p), se elige la mejor (Fig. 5.30q) y se optimiza la geometría (Fig. 5.30r). En este caso también varía la topología óptima al optimizar la geometría (Figs. 5.30s y 5.30t). Como la topología ha variado se vuelve a optimizar la geometría (Fig. 5.30u) con la que termina la tercera iteración.

Diseño óptimo simultáneo de topología y geometría de estructuras articuladas. Aplicación informática

6.1 INTRODUCCIÓN

Los distintos algoritmos desarrollados en el capítulo 5 se han implementado en una aplicación informática propia, a la que se ha denominado TTO, acrónimo de *Truss Topology Optimization*. Dicha aplicación se ha desarrollado con el objetivo de servir de herramienta para la realización de esta tesis.

TTO es una aplicación informática para el diseño óptimo simultáneo de topología y geometría de estructuras articuladas sometidas a un estado de cargas y con restricciones de diseño de tensión, pandeo de elementos, y esbeltez, ejecutable desde un PC con Windows 95/NT4 o superior. La aplicación también analiza problemas de elasticidad plana y de campos en régimen permanente, con vistas a una futura ampliación a la optimización de topología de problemas continuos.

Esta aplicación se ha escrito con el lenguaje de programación orientado a objetos C++, y en el entorno de programación Borland C++ Builder 5.0 para Windows.

El código correspondiente a la parte numérica es compatible ANSI/ISO Standard C++ (1998), el código gráfico es compatible OpenGL 1.0 para Windows, y el código de Windows es válido para Borland C++ Builder 5.0 para Windows.

En este capítulo se hace, en primer lugar, una descripción las declaraciones de las clases definidas para la implementación de TTO.

Seguidamente se hace una descripción general del funcionamiento de la aplicación desarrollada.

Se puede obtener más información sobre TTO en el manual de usuario del programa TTO (Martínez, 2003).

6.2 CLASES DEFINIDAS

6.2.1 Introducción

Un objeto es una estructura que, además de contener datos, contiene funciones que manejan esos datos. Antes de poder crear un objeto de un cierto tipo será necesario definir dicho tipo. A dicho tipo se le denomina clase, y a los datos y las funciones miembros de esa clase.

Una de las características importantes de la programación orientada a objetos es la reutilización de código, para la que se usa la herencia de clases.

Las funciones que deben ser definidas en las clases derivadas de una determinada clase base se denominan funciones virtuales puras, y a las clases base que las contienen, clases abstractas. No se pueden crear objetos de clases abstractas.

A continuación se va a hacer una breve descripción de las clases más significativas de las definidas en TTO, agrupadas según los archivos de cabecera donde están declaradas (en C++ las clases se suelen declarar en archivos con extensión *h* y definir en archivos con extensión *cpp*). Las clases abstractas son las que están en los óvalos con línea simple.

Para cada clase sólo se muestran los miembros (variables y funciones) más importantes.

6.2.2 Archivo: Principal.h

Este archivo incluye la clase *Principal*. Esta clase se encarga del manejo del entorno de Windows de la aplicación. En esta clase se crean objetos de las clases *Mef*, *Analisis* y *Optimizacion*. Estas tres clases hacen uso de las clases restantes.

6.2.3 Archivo: Mef.h

Este archivo incluye la clase *Mef*. Esta clase se encarga de la entrada de datos y de la generación del modelo de elementos finitos, y almacena los datos del modelo de diseño y del modelo de elementos finitos.

La declaración de la clase *Mef* es la siguiente:

```
class Mef
{
public:
    Mef();
    void leer(char* nombre);
    void mef();
private:
    Vector<Punto*> puntos;
    Vector<Linea*> lineas;
    Vector<Area*> areas;
    Vector<Volumen*> volumenes;
    Vector<Material*> materiales;
    Vector<Propiedad*> propiedades;
    Vector<Nodo*> nodos;
    Vector<Elemento*> elementos;
```

};

La clase *Mef* tiene 3 funciones miembro públicas (se puede acceder a ellas desde otros objetos) y 8 variables miembro privadas (sólo se puede acceder desde el propio objeto). La primera función, *Mef*, es el constructor (función que se llama automáticamente al crear el objeto) donde se inicializan los datos del objeto. La función *leer* lee los datos de un archivo de datos. La función *mef* genera el modelo de elementos finitos. Las variables tienen un nombre lo suficientemente explícito como para no precisar una explicación.

6.2.4 Archivo: Analisis.h

Este archivo incluye la clase *Analisis* y sus clases derivadas (Fig. 6.1). Esta clase se encarga del análisis del modelo de elementos finitos. Aunque están declaradas 5 clases derivadas, una para cada tipo de análisis (*Estatico*, *Modal*, *Armonico*, *Transitorio* y *Espectral*), sólo está definida la clase *Estatico* y sus derivadas (para análisis estático).

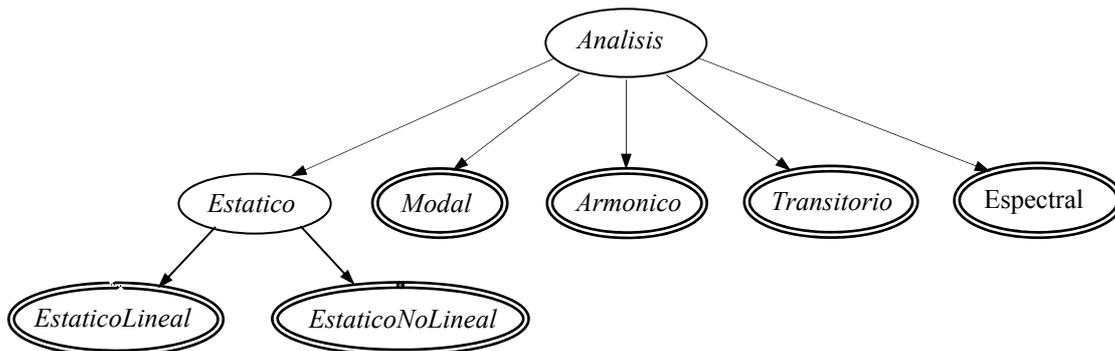


Figura 6.1 Clase base *Analisis* y sus clases derivadas

La declaración de la clase base *Analisis* y sus clases derivadas es la siguiente:

```

class Analisis
{
public:
    Analisis(Mef* me);
    virtual void analizar() = 0;
protected:
    Mef* mef;
};

class Estatico: public Analisis
{
public:
    Estatico(Mef* me);
};

class EstaticoLineal: public Estatico
{
public:
    EstaticoLineal(Mef *me);
    void analizar();
};
  
```

```

class EstaticoNoLineal: public Estatico
{
public:
    EstaticoNoLineal(Mef *me);
    void analizar();
};

class Modal: public Analisis
{
public:
    Modal(Mef* me);
    void analizar();
};

class Armonico: public Analisis
{
public:
    Armonico(Mef* me);
    void analizar();
};

class Transitorio: public Analisis
{
public:
    Transitorio(Mef* me);
    void analizar();
};

class Espectral: public Analisis
{
public:
    Espectral(Mef* me);
    void analizar();
};

```

La clase *Analisis* hace uso de la clase *Mef* (se le pasa la dirección del objeto al constructor) para tener acceso a los datos.

De la clase *Estatico* derivan las clases *EstaticoLineal* y *EstaticoNoLineal*, para análisis estático lineal y no lineal, respectivamente. El análisis no lineal considera no linealidad geométrica para el estudio del movimiento de mecanismos. Añadiendo un muelle a un mecanismo se puede analizar como una estructura, pero, debido a los grandes desplazamientos que se producen en los mecanismos, se debe considerar no linealidad geométrica.

La función *analizar* es virtual pura en la clase base *Analisis*, puesto que es específica para cada clase derivada de *Analisis*. Por ejemplo, para leer los datos del archivo 'PlacaAgujero.dat' y hacer un análisis estático lineal, el código que habría que escribir es el siguiente:

```

Mef* mef = new Mef;
mef->leer("PlacaAgujero.dat");
Analisis* analisis = new EstaticoLineal(mef);
analisis->analizar();

```

6.2.5 Archivo: Optimizacion.h

Este archivo incluye la clase *Optimizacion* y sus clases derivadas (Fig. 6.2). Esta clase se encarga de la optimización de topología y geometría de estructuras articuladas mediante métodos basados en el universo estructural o mediante el método de crecimiento propuesto.

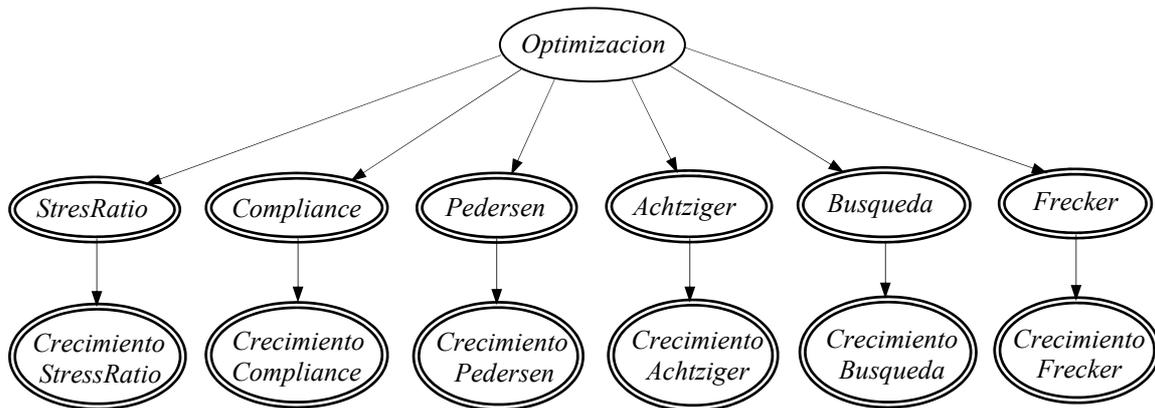


Figura 6.2 Clase base *Optimizacion* y sus clases derivadas

La declaración de la clase base *Optimizacion* y sus clases derivadas es la siguiente:

```

class Optimizacion
{
public:
    Optimizacion(Mef* me);
    virtual void optimizar();
protected:
    virtual void optimizarTopologia() = 0;
    bool optimizarGeometria();
    Mef* mef;
};

class StressRatio: public Optimizacion
{
public:
    StressRatio(Mef* me);
    void optimizarTopologia();
};

class Compliance: public Optimizacion
{
public:
    Compliance(Mef* me);
    void optimizarTopologia();
};

class Pedersen: public Optimizacion
{
public:
    Pedersen(Mef* me);
    void optimizarTopologia();
};
  
```

```
};

class Achtziger: public Optimizacion
{
public:
    Achtziger(Mef* me);
    void optimizarTopologia();
};

class Busqueda: public Optimizacion
{
public:
    Busqueda(Mef* me);
    void optimizarTopologia();
};

class Frecker: public Optimizacion
{
public:
    Frecker(Mef* me);
    void optimizarTopologia();
};

class CrecimientoStressRatio: public StressRatio
{
public:
    CrecimientoStressRatio(Mef* me);
    void optimizar();
};

class CrecimientoCompliance: public Compliance
{
public:
    CrecimientoCompliance(Mef* me);
    void optimizar();
};

class CrecimientoPedersen: public Pedersen
{
public:
    CrecimientoPedersen(Mef* me);
    void optimizar();
};

class CrecimientoAchtziger: public Achtziger
{
public:
    CrecimientoAchtziger(Mef* me);
    void optimizar();
};

class CrecimientoBusqueda: public Busqueda
{
public:
    CrecimientoBusqueda(Mef* me);
    void optimizar();
};

class CrecimientoFrecke: public Frecker
{
```

```
public:
    CrecimientoFrecker (Mef* me);
    void optimizar();
};
```

La clase *Optimizacion* hace uso de la clase *Mef* (se le pasa la dirección del objeto al constructor) para tener acceso a los datos, de la clase *Analisis* (se crea un objeto local) para los métodos que necesitan hacer análisis, y de la clase *Simplex* (se crea un objeto local) para los métodos que necesitan resolver un problema de programación lineal.

La función *optimizarTopologia* es virtual pura en la clase base *Optimizacion*, puesto que es específica para cada clase derivada de *Optimizacion*. Las clases derivadas de *Optimizacion* son las siguientes:

- *StressRatio*: optimiza la topología mediante el método *stress-ratio*.
- *Compliance*: optimiza la topología mediante el método de la *compliance*. Para ello se hace uso de la clase *SimplexSparse*.
- *Pedersen*: optimiza la topología mediante el método de Pedersen. Para ello se hace uso de la clase *SimplexModificado*.
- *Achtziger*: optimiza la topología mediante el método de Achtziger. Para ello se hace uso de la clase *SimplexSparse*.
- *Busqueda*: optimiza la topología mediante el método de búsqueda exhaustiva.
- *Frecker*: optimiza la topología mediante el método de Frecker. Para ello se hace uso de la subrutina de la librería IMSL, DNCONG, que resuelve un problema de programación no lineal general usando un algoritmo de programación cuadrática sucesiva y gradientes analíticos.

La función virtual (no pura) *optimizar*, de la clase base *Optimizacion*, se define en esta clase (de ahí que no sea pura) y llama a la función *optimizarTopologia*, es decir, optimiza la topología. Por ejemplo, para leer los datos del archivo ‘Michell.dat’ y hacer una optimización de topología mediante el método de la *compliance*, el código que habría que escribir es el siguiente:

```
Mef* mef = new Mef;
mef->leer("Michell.dat");
Optimizacion* optimizacion = new Compliance(mef);
optimizacion->optimizar();
```

Los métodos de crecimiento derivan de sus respectivos métodos del universo estructural porque para la optimización de topología en cada iteración se hace una optimización de topología, mediante los métodos universales, y una optimización de geometría (ver capítulo 5). Para ello se redefine la función *optimizar* en las clases de crecimiento. La función *optimizarGeometria* se define en la clase base *Optimizacion* porque es común a todos los métodos. Si el mismo ejemplo anterior se quisiera optimizar mediante el método de crecimiento, utilizando el método de la *compliance* en la etapa de optimización de topología en cada iteración, el código sería el siguiente:

```
Mef* mef = new Mef;
mef->leer("Michell.dat");
Optimizacion* optimizacion = new CrecimientoCompliance(mef);
optimizacion->optimizar();
```

Se observa que el código es el mismo, salvo que el objeto *optimizacion* se crea a partir de la clase *CrecimientoCompliance* en vez de a partir de la clase *Compliance*.

6.2.6 Archivo: Matriz.h

Este archivo incluye las clases *Vector*, *Matriz*, *MatrizSim*, *MatrizPerf* y *Matriz3*. Estas clases manejan vectores, matrices, matrices simétricas, matrices en perfil y matrices tridimensionales, respectivamente, sin el uso de objetos temporales (mediante metaplantillas).

6.2.7 Archivo: Expresion.h

Este archivo incluye la clase *Expresion*. Esta clase analiza gramaticalmente expresiones para su uso en la entrada de datos. Las expresiones admiten los operadores +, -, *, /, los paréntesis y las funciones siguientes: ‘abs’, ‘acos’, ‘asin’, ‘atan’, ‘atan2’, ‘cos’, ‘exp’, ‘int’, ‘log’, ‘pow’, ‘sgn’, ‘sin’, ‘sqr’, ‘sqrt’ y ‘tan’.

6.2.8 Archivo: Punto.h

Este archivo incluye la clase *Punto*. Esta clase se encarga de inicializar, mallar, imponer desplazamientos y aplicar cargas en los puntos.

La declaración de la clase *Punto* es la siguiente:

```
class Punto
{
public:
    Punto();
    void mallar();
    void mallarNodo();
    void imponerDesplazamiento();
    void aplicarCarga();
private:
    int          numero;
    int          tipoElemento;
    double       tamanoMalla;
    Material*    material;
    Propiedad*   propiedad;
    Vector<double> coordenadas;
    MatrizPerf<double> desplazamiento;
    MatrizPerf<bool> impuesto;
    MatrizPerf<double> carga;
    MatrizPerf<double> aceleracion;
    Nodo*        nodo;
    Elemento*    elemento;
};
```

La función *mallarNodo* es necesaria porque aunque no haya un elemento en el punto, si dicho punto pertenece a una línea, a un área o a un volumen donde haya elementos, habrá que crear un nodo.

6.2.9 Archivo: Linea.h

Este archivo incluye la clase base *Linea* y sus clases derivadas (Fig. 6.3). Esta clase se encarga de inicializar, mallar, imponer desplazamientos y aplicar cargas en las líneas.

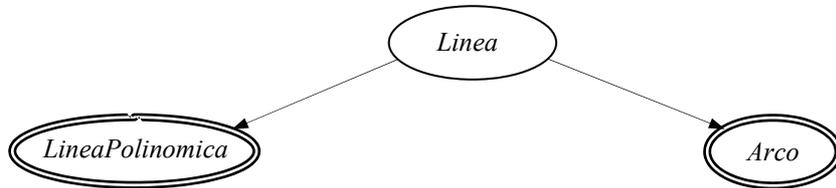


Figura 6.3 Clase base *Linea* y sus clases derivadas

La declaración de la clase base *Linea* y sus clases derivadas es la siguiente:

```

class Linea
{
public:
    Linea();
    void mallar();
    virtual void mallarNodos() = 0;
    void imponerDesplazamiento();
    void aplicarCarga();
protected:
    int                numero;
    int                tipoElemento;
    int                numDivisiones;
    Material*         material;
    Vector<Propiedad*> propiedades;
    Vector<Punto*>    puntos;
    MatrizPerf<double> desplazamiento;
    MatrizPerf<bool>  impuesto;
    Vector<double>    incrementoTemperatura;
    Vector<double>    faltaAjuste;
    MatrizPerf<double> aceleracion;
    Matriz<Vector<double>> > cargaPuntualGlobal;
    Matriz<Vector<double>> > cargaPuntualLocal;
    Matriz<Matriz<double>> > cargaDistribuidaGlobal;
    Matriz<Matriz<double>> > cargaDistribuidaLocal;
    Vector<Nodo*>     nodos;
    Vector<Elemento*> elementos;
};

class LineaPolinmica: public Linea
{
public:
    void mallarNodos();
};

class Arco: public Linea
{
public:
    void mallarNodos();
};
  
```

Las variables miembro son protegidas (sólo se puede acceder desde el propio objeto y desde sus derivados). Todas las funciones son comunes a todos los tipos de líneas excepto la función *mallarNodos* que depende del tipo de línea, de ahí que se defina como una función virtual pura en la clase base *Linea*.

Hay dos clases derivadas, *LineaPolinomica* y *Arco*, en las que únicamente hay que definir la función *mallarNodos*. Se puede añadir cualquier tipo de línea simplemente implementando la función *mallarNodos*, las demás funciones ya están definidas en la clase base.

6.2.10 Archivo: Area.h

Este archivo incluye la clase base *Area* y sus clases derivadas (Fig. 6.4). Esta clase se encarga de inicializar, mallar, imponer desplazamientos y aplicar cargas en las áreas.

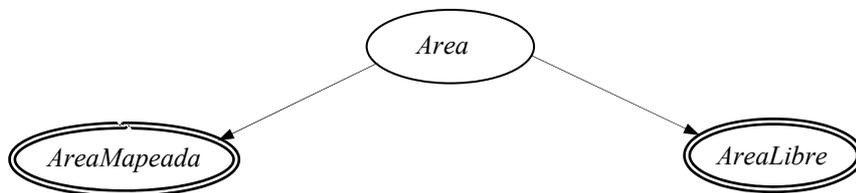


Figura 6.4 Clase base *Area* y sus clases derivadas

La declaración de la clase base *Area* y sus clases derivadas es la siguiente:

```

class Area
{
public:
    Area();
    void mallar();
    virtual void mallarNodos() = 0;
    void imponerDesplazamiento();
    void aplicarCarga();
protected:
    int                numero;
    int                tipoElemento;
    Material*          material;
    Vector<Propiedad*> propiedades;
    Vector<Linea*>     lineas;
    MatrizPerf<double> desplazamiento;
    MatrizPerf<bool>   impuesto;
    Vector<double>     incrementoTemperatura;
    MatrizPerf<double> aceleracion;
    MatrizPerf<double> cargaDistribuidaGlobal;
    MatrizPerf<double> cargaDistribuidaLocal;
    Vector<Nodo*>     nodos;
    Vector<Elemento*> elementos;
};

class AreaMapeada: public Area
{
public:
    void mallarNodos();
};
  
```

```
class AreaLibre: public Area
{
public:
    void mallarNodos();
};
```

La función *mallarNodos* es una función virtual pura, puesto que es específica para cada clase derivada de la clase base *Area*. Aunque esta función tiene el mismo nombre que la correspondiente de la clase *Linea*, no hay problemas de ambigüedad puesto que para acceder a cada miembro (si es público) hay que hacerlo a través del objeto al que pertenece. Por ejemplo, si *linea* y *area* son dos objetos (creados dinámicamente, con *new*) de tipo *Linea* y *Area*, respectivamente, para acceder a la función *mallarNodos* del objeto *linea* se haría con *linea->mallarNodos()*, y para el objeto *area*, con *area->mallarNodos()*.

Hay dos clases derivadas, *AreaMapeada* y *AreaLibre*, en las que únicamente hay que definir la función *mallarNodos*. La clase *AreaMapeada* utiliza un mallador mapeado para mallar los nodos, y la clase *AreaLibre* utiliza un mallador libre basado en la triangulación de Delaunay-Voronoi.

6.2.11 Archivo: Volumen.h

Este archivo incluye la clase base *Volumen* y sus clases derivadas (Fig. 6.5). Esta clase se encarga de inicializar, mallar, imponer desplazamientos y aplicar cargas en las volúmenes.

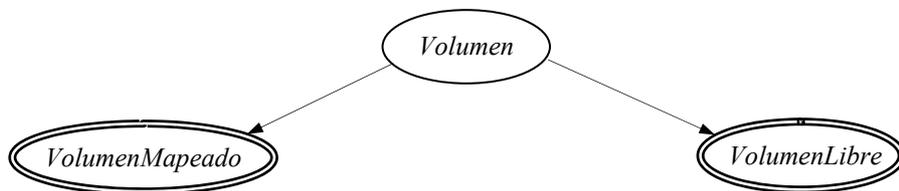


Figura 6.5 Clase base *Volumen* y sus clases derivadas

La declaración de la clase base *Volumen* y sus clases derivadas es la siguiente:

```
class Volumen
{
public:
    Volumen();
    void mallar();
    virtual void mallarNodos() = 0;
    void aplicarCarga();
protected:
    int                numero;
    int                tipoElemento;
    Material*          material;
    Vector<Propiedad*> propiedades;
    Vector<Area*>      areas;
    Vector<double>     incrementoTemperatura;
    MatrizPerf<double> aceleracion;
    Vector<Nodo*>     nodos;
    Vector<Elemento*> elementos;
```

```

};

class VolumenMapeado: public Volumen
{
public:
    void mallarNodos();
};

class VolumenLibre: public Volumen
{
public:
    void mallarNodos();
};

```

La función *mallarNodos* es una función virtual pura, puesto que es específica para cada clase derivada de la clase base *Volumen*.

Hay dos clases derivadas, *VolumenMapeado* y *VolumenLibre*, en las que únicamente hay que definir la función *mallarNodos*. La clase *VolumenMapeado* utiliza un mallador mapeado para mallar los nodos, y la clase *VolumenLibre* aún no está definida.

6.2.12 Archivo: Material.h

Este archivo incluye la clase *Material*. Esta clase se almacenan las propiedades de los materiales.

La declaración de la clase *Material* es la siguiente:

```

class Material
{
public:
    Material();
private:
    int    numero;
    double young;
    double poisson;
    double densidad;
    double tension;
    double dilatacion;
    double conductividad;
};

```

6.2.13 Archivo: Propiedad.h

Este archivo incluye la clase *Propiedad*. Esta clase se almacenan las propiedades de los elementos.

La declaración de la clase *Propiedad* es la siguiente:

```

class Propiedad
{
public:
    Propiedad();
private:
    int    numero;
    Vector<double> valores;
};

```

```
};
```

6.2.14 Archivo: Nodo.h

Este archivo incluye la clase *Nodo*. Esta clase se encarga de almacenar las variables relacionadas con los nodos.

La declaración de la clase *Nodo* es la siguiente:

```
class Nodo
{
public:
Nodo(int nDim, int nGdl, int nEst);
private:
Vector<double> coordenadas;
Matriz<double> desplazamiento;
Matriz<bool> impuesto;
Matriz<double> carga;
int numero;
bool activo;
};
```

La variable *numero* es utilizada por la clase *Numeración*, para la numeración y renumeración de los nodos. La variable *activo* es utilizada por la clase *Optimización* para activar o desactivar los nodos que formen parte o no del óptimo.

6.2.15 Archivo: Elemento.h

Este archivo incluye la clase base *Elemento* y sus clases derivadas (Fig. 6.6). Esta clase se encarga de almacenar y manejar las variables relacionadas con los elementos.

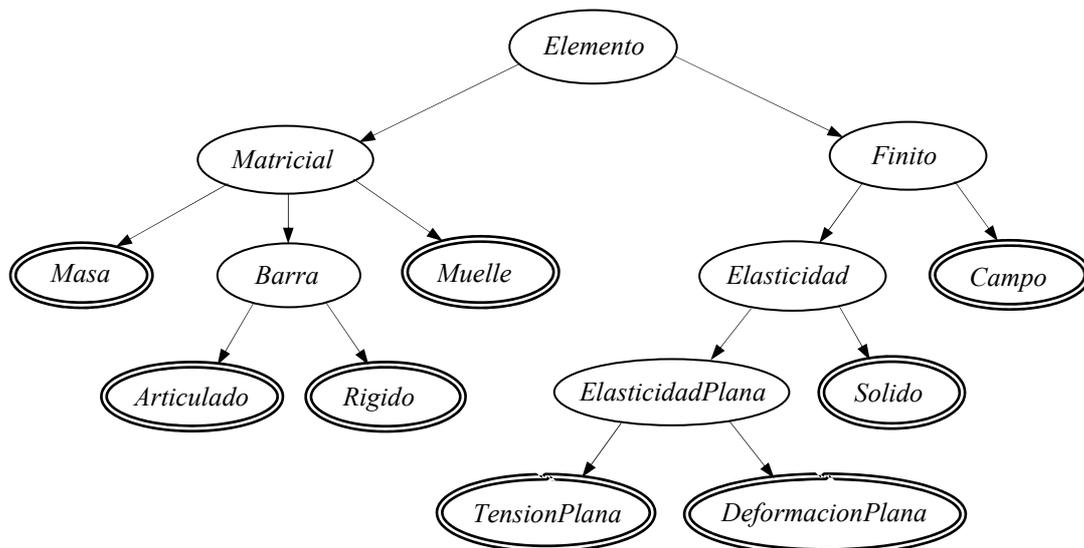


Figura 6.6 Clase base *Elemento* y sus clases derivadas

La declaración de la clase base *Elemento* y sus clases derivadas es la siguiente:

```
class Elemento
{
```

```

public:
    enum {masa, articulado, rigido, muelle,
          campo, tensionPlana, deformacionPlana, solido};
    Elemento(int nNod, int nDim, int nGdl, int nRel, int nEst);
    virtual void calcularRigidez() = 0;
    virtual void calcularCarga(int est) = 0;
    virtual void calcularTension(int est) = 0;
protected:
    Material*          material;
    Vector<double>     propiedades;
    Vector<Nodo*>     nodos;
    Matriz<double>    deformacionInitial;
    Matriz<double>    cargaVolumetrica;
    int               numero;
    bool              activo;
    MatrizSim<double> rigidez;
    Matriz<double>    carga;
    Matriz3<double>  deformacion;
    Matriz3<double>  tension;
};

class Matricial: public Elemento
{
public:
    Matricial(int nNod, int nDim, int nGdl, int nRel, int nEst);
};

class Masa: public Matricial
{
public:
    Masa(int nDim, int nEst);
    void calcularRigidez();
    void calcularCarga(int est);
    void calcularTension(int est);
};

class Barra: public Matricial
{
public:
    Barra(int nDim, int nGdl, int nRel, int nEst);
    void calcularTension(int est);
protected:
    Matriz3<double> cargaDistribuidaGlobal;
    Matriz3<double> cargaDistribuidaLocal;
};

class Articulado: public Barra
{
public:
    Articulado(int nDim, int nEst);
    void calcularRigidez();
    void calcularCarga(int est);
};

class Rigido: public Barra
{
public:
    Rigido(int nDim, int nEst);
    void calcularRigidez();
    void calcularCarga(int est);
};

```

```

};

class Muelle: public Matricial
{
public:
    Muelle(int nDim, int nGdl, int nEst);
    void calcularRigidez();
    void calcularCarga(int est);
    void calcularTension(int est);
};

class Finito: public Elemento
{
public:
    Finito(int nNod, int nDim, int nGdl, int nRel, int nEst);
    void calcularRigidez();
    void calcularCarga(int est);
    void calcularTension(int est);
protected:
    virtual void calcularGeometria(double r, double s, double t) = 0;
    virtual void calcularElasticidad() = 0;
    Matriz<double> geometria;
    MatrizSim<double> elasticidad;
    Forma forma;
};

class Campo: public Finito
{
public:
    Campo(int nNod, int nDim, int nEst);
private:
    void calcularGeometria(double r, double s, double t);
    void calcularElasticidad();
};

class Elasticidad: public Finito
{
public:
    Elasticidad(int nNod, int nDim, int nGdl, int nRel, int nEst);
};

class ElasticidadPlana: public Elasticidad
{
public:
    ElasticidadPlana(int nNod, int nEst);
protected:
    void calcularGeometria(double r, double s, double);
};

class TensionPlana: public ElasticidadPlana
{
public:
    TensionPlana(int nNod, int nEst);
private:
    void calcularElasticidad();
};

class DeformacionPlana: public ElasticidadPlana
{
public:

```

```

        DeformacionPlana(int nNod, int nEst);
private:
    void calcularElasticidad();
};

class Solido: public Elasticidad
{
public:
    Solido(int nNod, int nEst);
private:
    void calcularGeometria(double r, double s, double t);
    void calcularElasticidad();
};

```

La clase base *Elemento* tiene 3 funciones virtuales puras, *calcularRigidez*, *calcularCarga* y *calcularTension*. De *Elemento* se derivan las clases *Matricial* y *Finito*. La clase *Finito* tiene 2 funciones virtuales puras, *calcularGeometria* y *calcularElasticidad*.

Para una reutilización efectiva del código, si varias clases tienen alguna función con un código común, debe definirse en una clase de la que se deriven las clases anteriores. Por ejemplo, y empezando por abajo, las clases *TensionPlana* y *DeformacionPlana* sólo se diferencian por la matriz de elasticidad, por lo tanto, la función *calcularElasticidad* es la única que debe definirse en estas clases. La matriz de geometría es común a los dos problemas y específica de problemas de elasticidad plana, por lo tanto, la función *calcularGeometria* debe definirse en la clase *ElasticidadPlana*. La forma de calcular la matriz de rigidez, el vector de cargas y el vector tensión es la misma para todos los elementos continuos (según la formulación del MEF), luego, las funciones *calcularRigidez*, *calcularCarga* y *calcularTension* se definen en la clase *Finito*. De esta forma, añadir nuevos elementos es sencillo, sólo hay que derivar una clase de la que corresponda y definir las matrices específicas para el nuevo elemento, puesto que el código común ya está definido.

6.2.16 Archivo: Forma.h

Este archivo incluye la clase *Forma*. Esta clase se encarga del cálculo de las funciones de forma, de continuidad C_0 , y de sus derivadas, en coordenadas locales y globales.

La declaración de la clase *Forma* es la siguiente:

```

class Forma
{
public:
    Forma(int nNod);
    Forma(int nNod, int nDim);
    void calcularMatrices(double r);
    void calcularMatrices(double r, double s);
    void calcularMatrices(double r, double s, double t);
    void calcularMatrices(double r, const Matriz<double>& coord);
    void calcularMatrices(double r, double s,
                           const Matriz<double>& coord);
    void calcularMatrices(double r, double s, double t,

```

```

        const Matriz<double>& coord);
private:
    void funcion(double r);
    void funcion(double r, double s);
    void funcion(double r, double s, double t);
    void derivada(double r);
    void derivada(double r, double s);
    void derivada(double r, double s, double t);
    double detJacobiano;
    Vector<double> funcion;
    Matriz<double> derivada;
    Matriz<double> jacobiano;
    Matriz<double> invJacobiano;
    Matriz<double> derivadaGlobal;
};

```

Las funciones privadas *funcion* y *derivada* son de uso interno, para llenar las matrices.

6.2.17 Archivo: Gauss.h

Este archivo incluye la clase *Gauss*. Esta clase se encarga de la integración numérica mediante la cuadratura de Gauss de funciones en regiones unidimensionales, bidimensionales o tridimensionales en coordenadas normalizadas.

La declaración de la clase *Gauss* es la siguiente:

```

class Gauss
{
public:
    double integral(double funcion(double), int ord);
    double integral(double funcion(double, double), int ord);
    double integral(double funcion(double, double, double), int ord);
private:
    static MatrizSim<double> puntos;
    static MatrizSim<double> pesos;
};

```

Las funciones *integral* tienen dos parámetros, el primero es la función que se quiere integrar y el segundo el orden de integración.

Las variables estáticas *puntos* y *pesos* contienen las posiciones y los pesos de Gauss para integraciones de orden 1 a 10.

6.2.18 Archivo: Numeracion.h

Este archivo incluye la clase *Numeracion* y sus clases derivadas (Fig. 6.7). Esta clase se encarga de la numeración de los nodos.

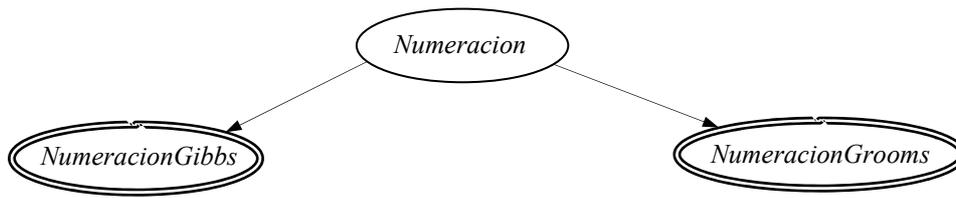


Figura 6.7 Clase base *Numeracion* y sus clases derivadas

La declaración de la clase base *Numeracion* y sus clases derivadas es la siguiente:

```

class Numeracion
{
public:
    Numeracion(const Vector<Nodo*>& nod,
              const Vector<Elemento*>& ele);
    void numerar();
    virtual void renumerar();
protected:
    Vector<Nodo*>    nodos;
    Vector<Elemento*> elementos;
};

class Gibbs: public Numeracion
{
public:
    Gibbs(const Vector<Nodo*>& nod, const Vector<Elemento*>& elem);
    void renumerar();
};

class Grooms: public Numeracion
{
public:
    Grooms(const Vector<Nodo*>& nod, const Vector<Elemento*>& elem);
    void renumerar();
};
  
```

Al constructor de la clase base *Numeracion* hay que pasarle los nodos y los elementos (la renumeración depende de estos).

En las clases derivadas, *Gibbs* y *Grooms* se define la función *renumerar*.

La clase *Gibbs* renumera los nodos por el método de Gibbs (Gibbs, 1976) que minimiza el perfil de la matriz de rigidez global, lo que disminuye la memoria necesaria para almacenarla y el coste computacional de factorizarla. Este método es el implementado en la clase *Mef*.

La clase *Grooms* renumera los nodos por el método de Grooms que tiene el mismo objetivo que el método de Gibbs, aunque es menos efectivo.

6.2.19 Archivo: Simplex.h

Este archivo incluye la clase *Simplex* y sus clases derivadas (Fig. 6.8). Esta clase se encarga de la resolución de diferentes tipos de problemas de programación lineal.

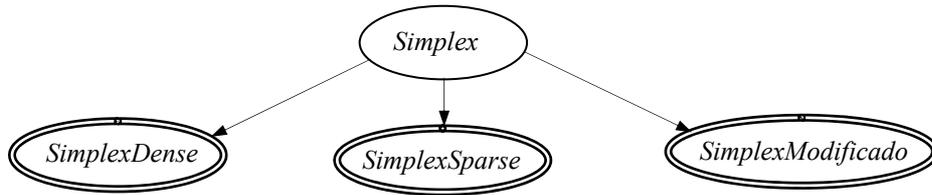


Figura 6.8 Clase base *Simplex* y sus clases derivadas

```

class Simplex
{
public:
    Simplex(int nVar, int nRest);
    virtual bool resolverSimplex() = 0;
};

class SimplexDense: public Simplex
{
public:
    SimplexDense(int nVar, int nRest);
    bool resolverSimplex();
};

class SimplexSparse: public Simplex
{
public:
    SimplexSparse(int nVar, int nRest);
    bool resolverSimplex();
};

class SimplexModified: public Simplex
{
public:
    SimplexModified(int nVar, int nRest);
    bool resolverSimplex();
};
  
```

De la clase base *Simplex* derivan las siguientes clases:

- *SimplexDense*: resuelve el problema de programación lineal mediante la subrutina de la librería IMSL, DDLPRS, que hace uso del algoritmo *simplex* revisado, y matrices densas (se usa la matriz completa).
- *SimplexSparse*: resuelve el problema de programación lineal mediante la subrutina de la IMSL, DSLPRS, que hace uso del algoritmo *simplex* revisado, y matrices *sparse* (poco densas, sólo se almacenan los elementos no nulos).
- *SimplexModificado*: resuelve el problema de programación lineal en restricciones pero no lineal en función objetivo, mediante el método *simplex* modificado (Pedersen, 1993). Al no ser estándar, se ha programado e implementado.

6.2.20 Archivo: OpenGL.h

Este archivo incluye la clase *OpenGL*. Esta clase se encarga de la representación gráfica del modelo de diseño de la estructura, del modelo de elementos finitos, de los diagramas de esfuerzos y de los mapas de tensiones. Para ello hace uso de la librería gráfica OpenGL para Windows.

6.3 APLICACIÓN INFORMÁTICA

6.3.1 Descripción general

TTO es una aplicación informática para el análisis de estructuras articuladas, de estructuras de nudos rígidos, de problemas de elasticidad plana y de problemas de campos en régimen permanente, y para el diseño óptimo simultáneo de topología y geometría de estructuras articuladas sometidas a un estado de cargas y con restricciones de diseño de tensión, pandeo de elementos, y esbeltez, ejecutable desde un PC con Windows 95/NT4 o superior.

En la Fig. 6.9 se muestra la pantalla principal de TTO. Ésta se autoconfigura (cambia el menú, la barra de botones y las subventanas) en función de los datos que hay cargados en memoria. En la Fig. 6.9 se muestra la ventana principal sin datos en memoria, en la Fig. 6.10 con un problema de barras articuladas, en la 6.11 con uno de barras de nudos rígidos, en la 6.12 con uno de elasticidad plana (tensión plana), en la 6.13 con uno de campos (transmisión de calor) y en la 6.14 con uno de optimización de topología y geometría de barras articuladas. Puede apreciarse que sólo son visibles los menús y los botones necesarios, en función del tipo de problema cargado en memoria. La subventana inferior derecha sólo aparece en los problemas de optimización.

La pantalla principal se divide en tres o cuatro subventanas:

- *Ventana de datos*: la ventana superior izquierda es un editor de textos donde se pueden cargar y modificar los datos de un archivo de datos (extensión *dat*) o teclearlos directamente.
- *Ventana de gráficos*: en la ventana superior derecha se representa el modelo (de diseño o de elementos finitos), los diagramas de esfuerzos (problemas de barras), o los mapas de resultados (problemas continuos). En problemas de optimización las barras, dibujadas en 3D, se colorean en función de la tensión (azul para el valor mínimo y rojo para el máximo).
- *Ventana de resultados*: en la subventana inferior izquierda se listan los resultados numéricos del análisis o del proceso de optimización.
- *Ventana de optimización*: la subventana inferior derecha, que sólo es visible en problemas de optimización, se muestra la evolución gráfica del proceso de optimización (función objetivo, variables mínima y máxima, y restricción más violada).

La entrada de datos es mediante texto, que es leído del editor de textos. Para cargar los datos del editor de textos en memoria hay que seleccionar el menú *Análisis->Actualizar*. Las opciones de menú más usadas tienen un botón de acceso directo.

El texto está formado por instrucciones y datos. Para ver la sintaxis de los archivos de datos, y las opciones de los menús y de los botones, ver el manual de usuario del programa TTO (Martínez, 2003).

6.3.2 Análisis

TTO realiza un análisis estático lineal del modelo de elementos finitos, mediante el método de los desplazamientos del MEF.

Para resolver el sistema de ecuaciones lineales algebraicas resultante se aplica el método de Cholevski, previa renumeración de los nodos para minimizar el perfil de la matriz de rigidez, y disminuir la memoria necesaria para almacenar dicha matriz y el coste computacional de su factorización.

En la Fig. 6.15 se muestra el mallado generado para un ejemplo de un diente de engranaje, y en la Fig. 6.16 (*Ver->Matriz de rigidez*) la matriz de rigidez sin renumeración de los nodos (la parte inferior) y con renumeración de los nodos (la parte superior).

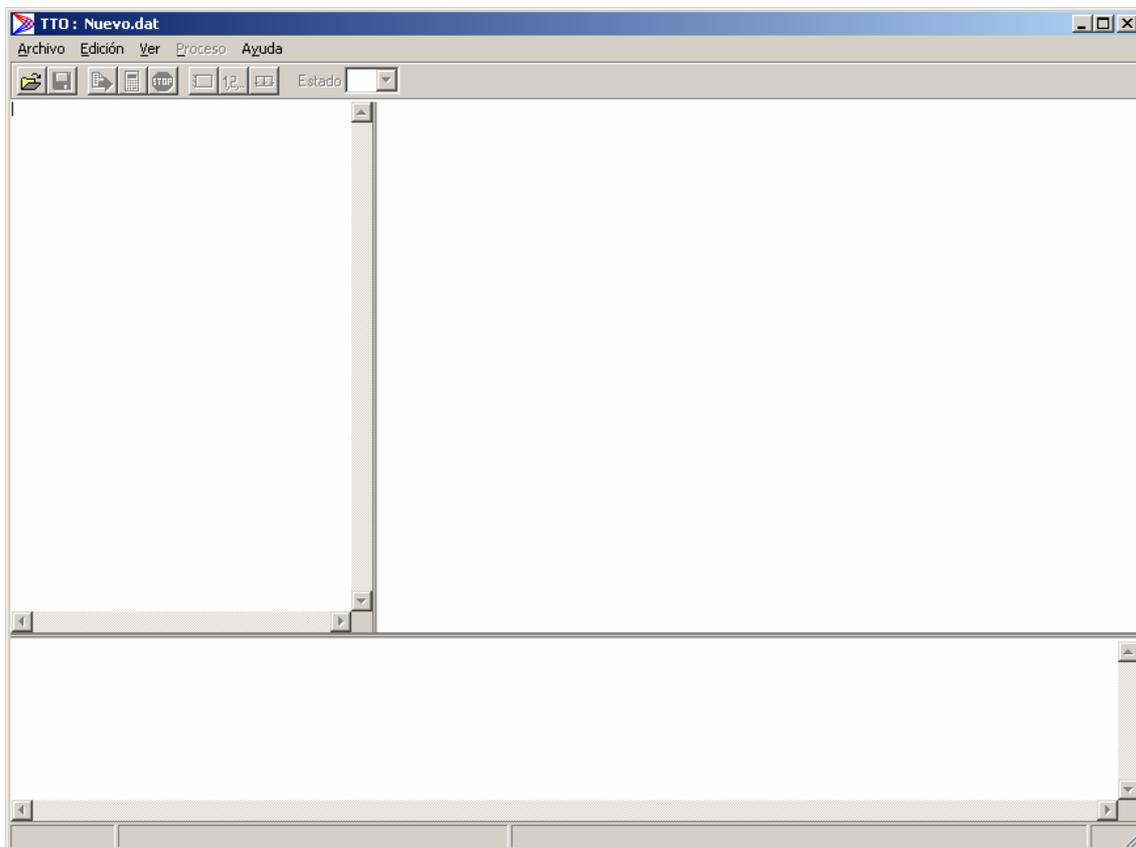


Figura 6.9 Pantalla principal de TTO sin datos

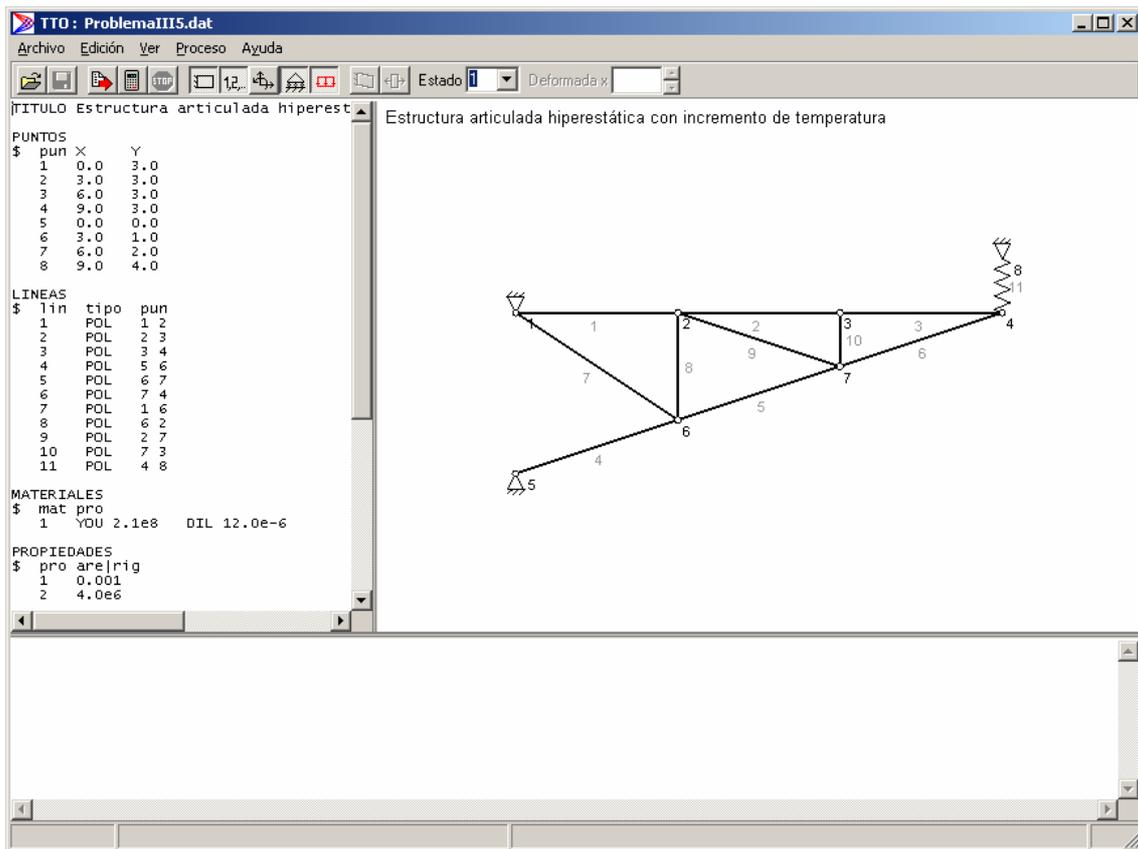


Figura 6.10 Pantalla principal de TTO con un problema de barras articuladas

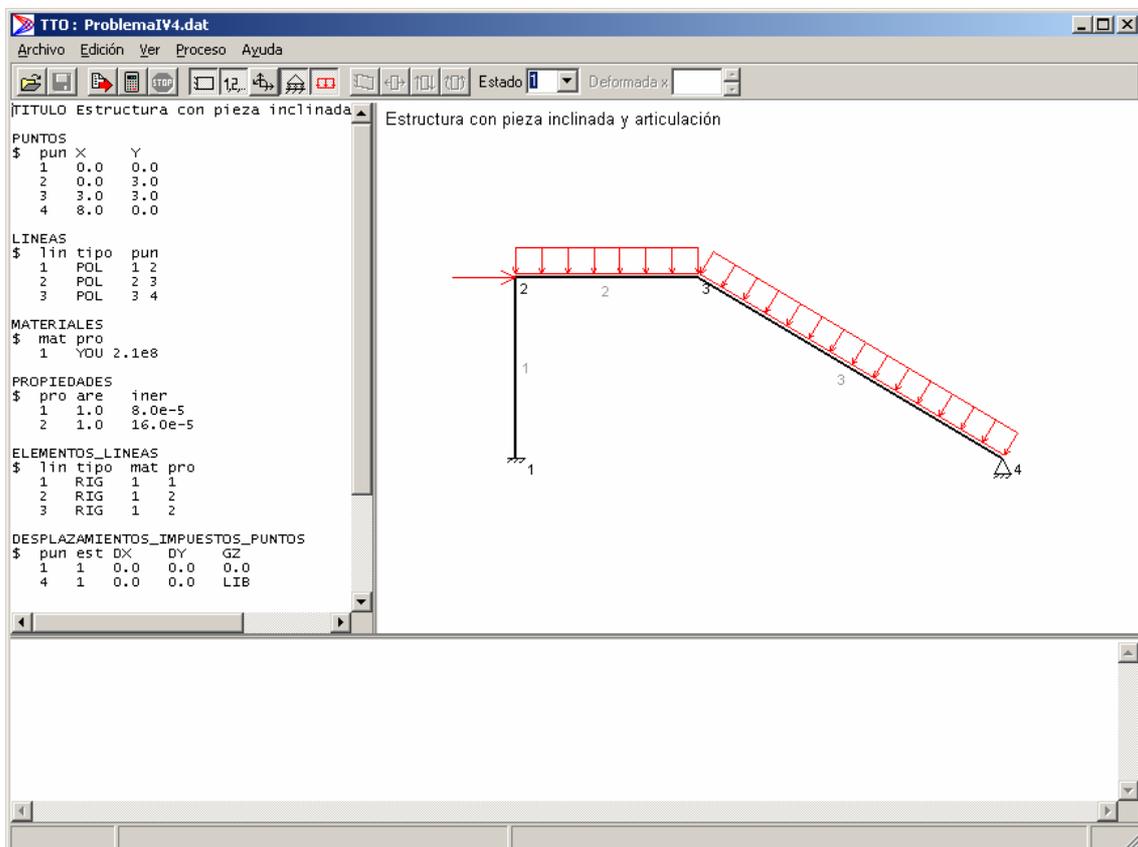


Figura 6.11 Pantalla principal de TTO con un problema de barras de nudos rígidos

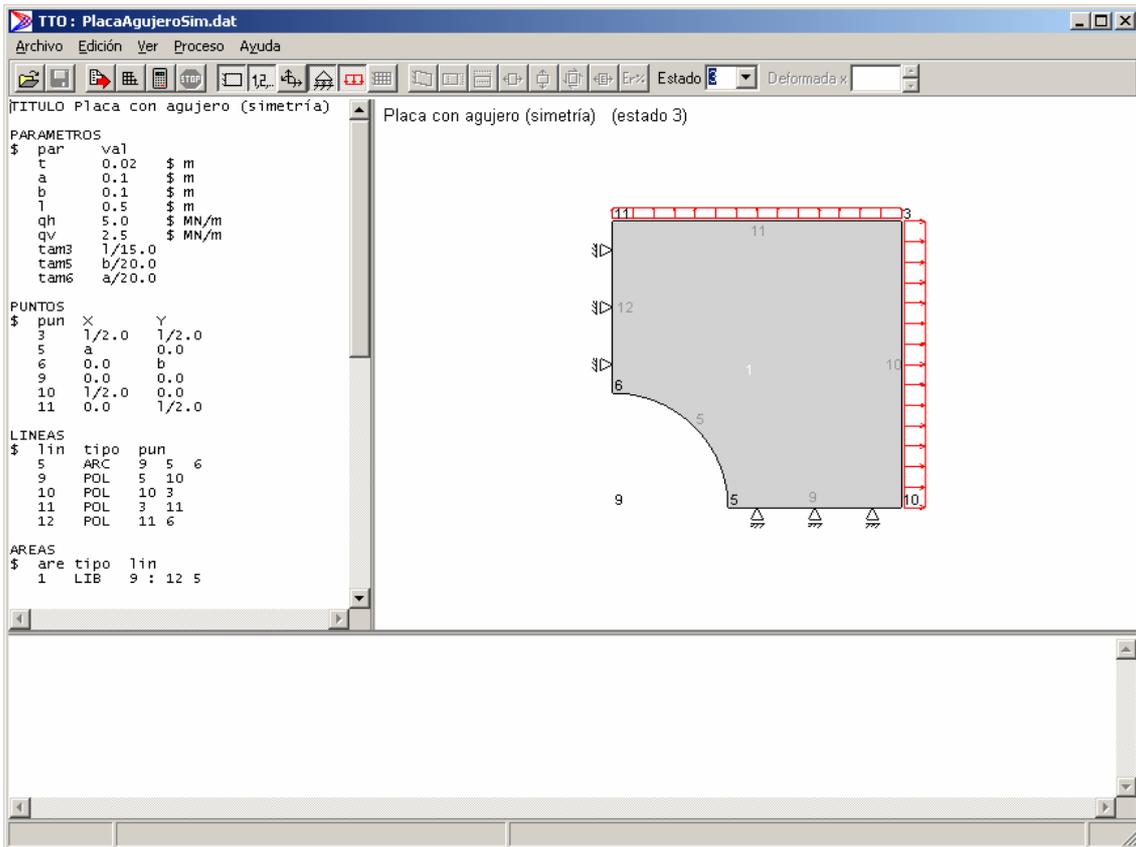


Figura 6.12 Pantalla principal de TTO con un problema de elasticidad plana

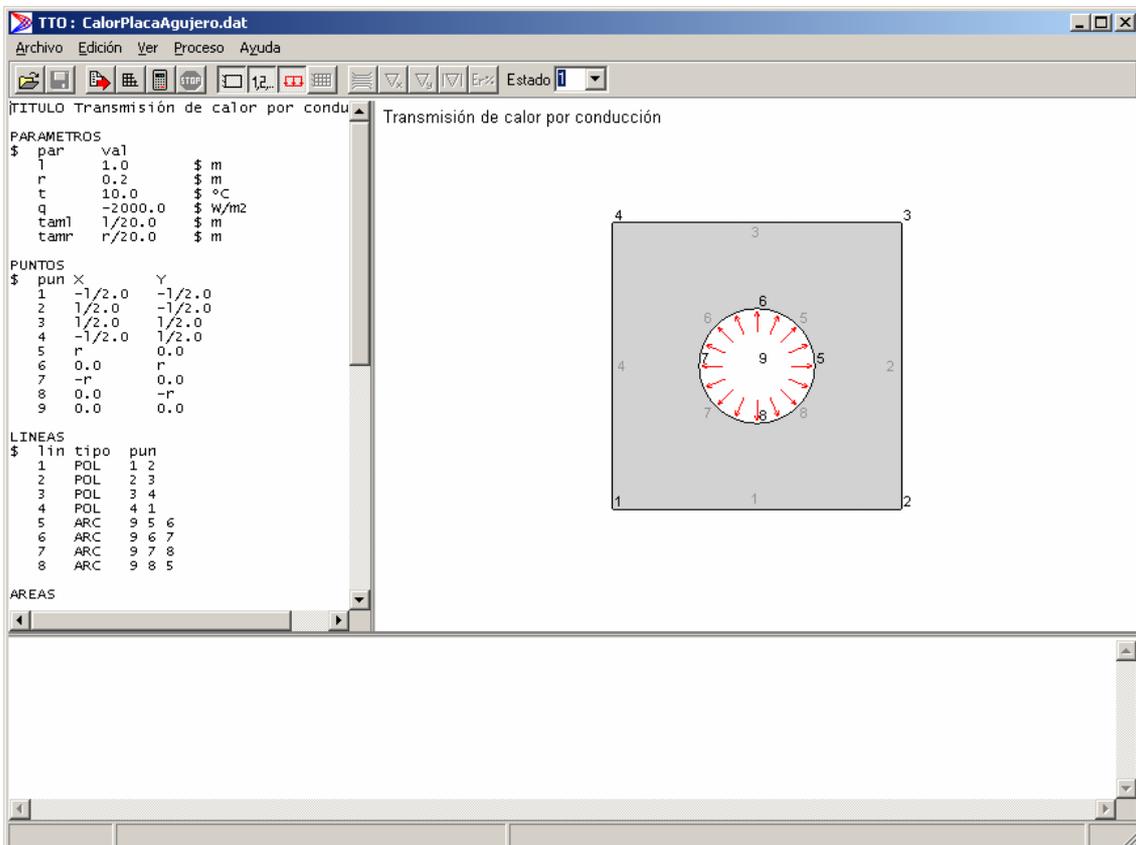


Figura 6.13 Pantalla principal de TTO con un problema de campos

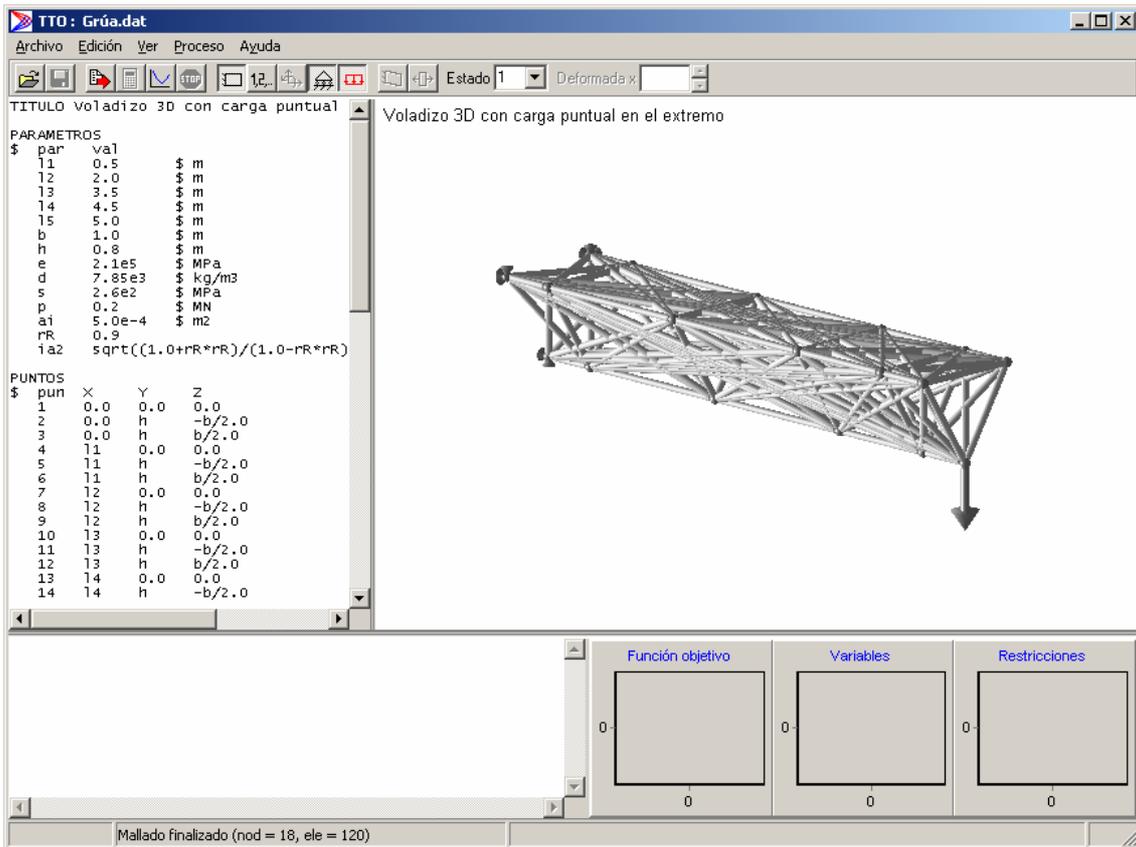


Figura 6.14 Pantalla principal de TTO con un problema de optimización de barras articuladas

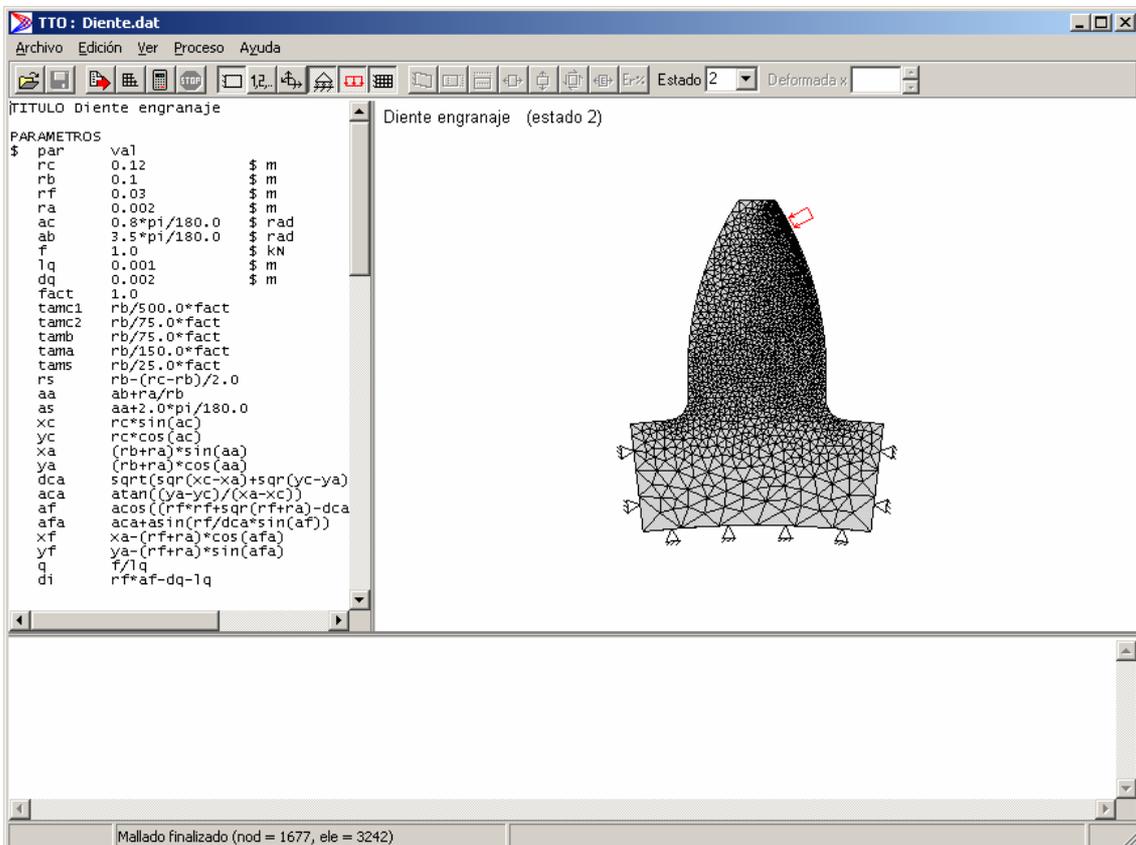


Figura 6.15 Mallado generado para un diente de engranaje

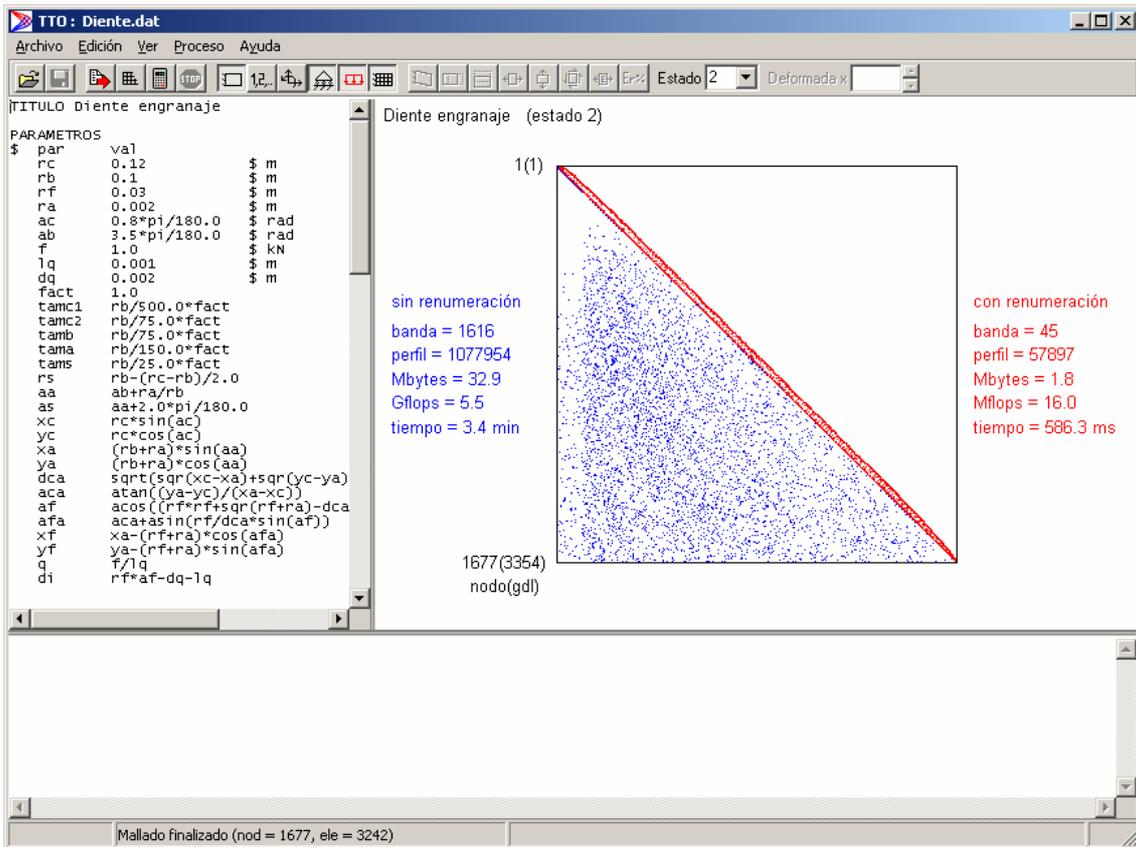


Figura 6.16 Matriz de rigidez sin y con renumeración de los nodos del problema de la Fig. 6.15

En la Fig. 6.16 se aprecia la gran diferencia de memoria necesaria para almacenar la matriz de rigidez (32,9 Mbytes sin renumeración y 1,8 Mbytes con renumeración) y del tiempo estimado necesario para la factorización de la matriz (3,4 minutos sin renumeración y 586 milisegundos con renumeración).

En la Fig. 6.17 se muestran los resultados del análisis (*Análisis*->*Analizar*) del problema de barras articuladas de la Fig. 6.10 (resultados numéricos, diagrama de esfuerzos axiales y deformada).

En la Fig. 6.18 se muestran los resultados del análisis del problema de barras de nudos rígidos de la Fig. 6.11 (resultados numéricos, diagramas de esfuerzos cortantes y momentos flectores, y deformada).

En la Fig. 6.19 se muestran los resultados del análisis del problema de elasticidad plana (tensión plana) de la Fig. 6.12 (resultados numéricos y mapa de la tensión de von Mises deformado).

En la Fig. 6.20 se muestran los resultados del análisis del problema de campos en régimen permanente (transmisión de calor por conducción en régimen permanente) de la Fig. 6.13 (resultados numéricos y mapa de temperaturas).

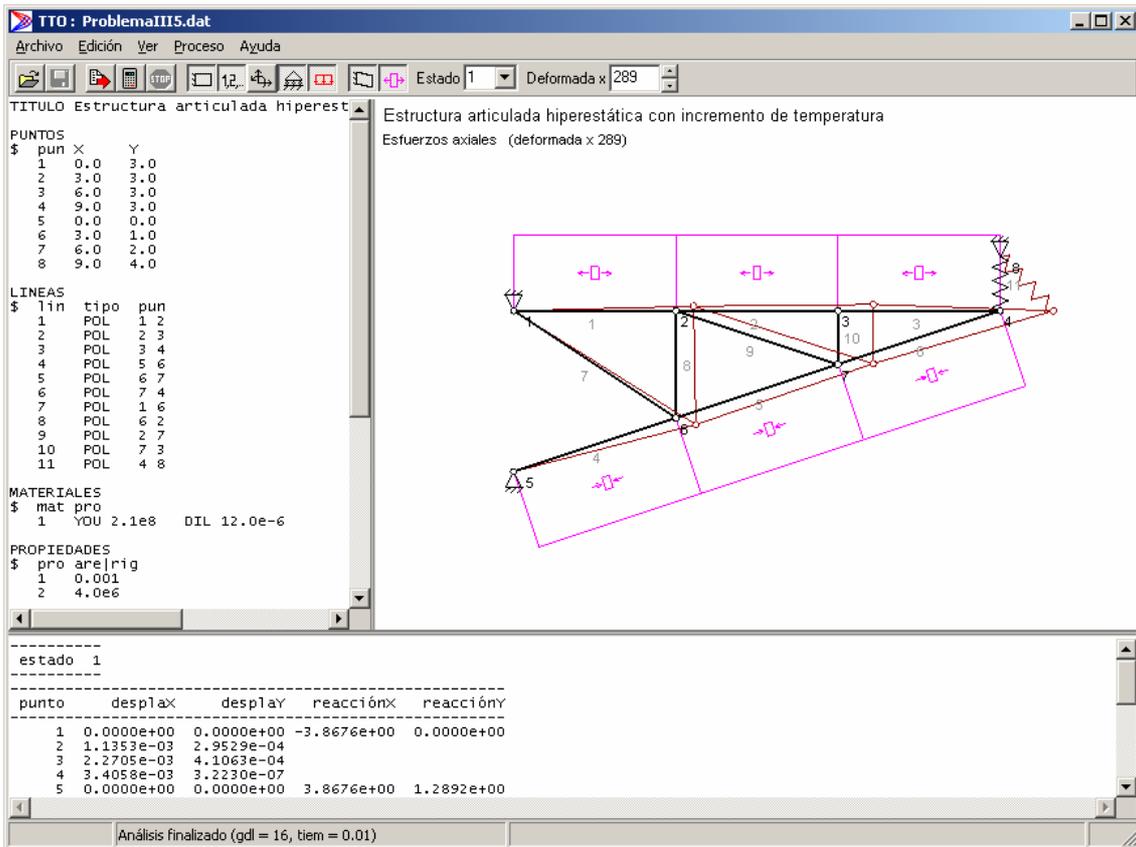


Figura 6.17 Resultados del análisis del problema de la Fig. 6.10

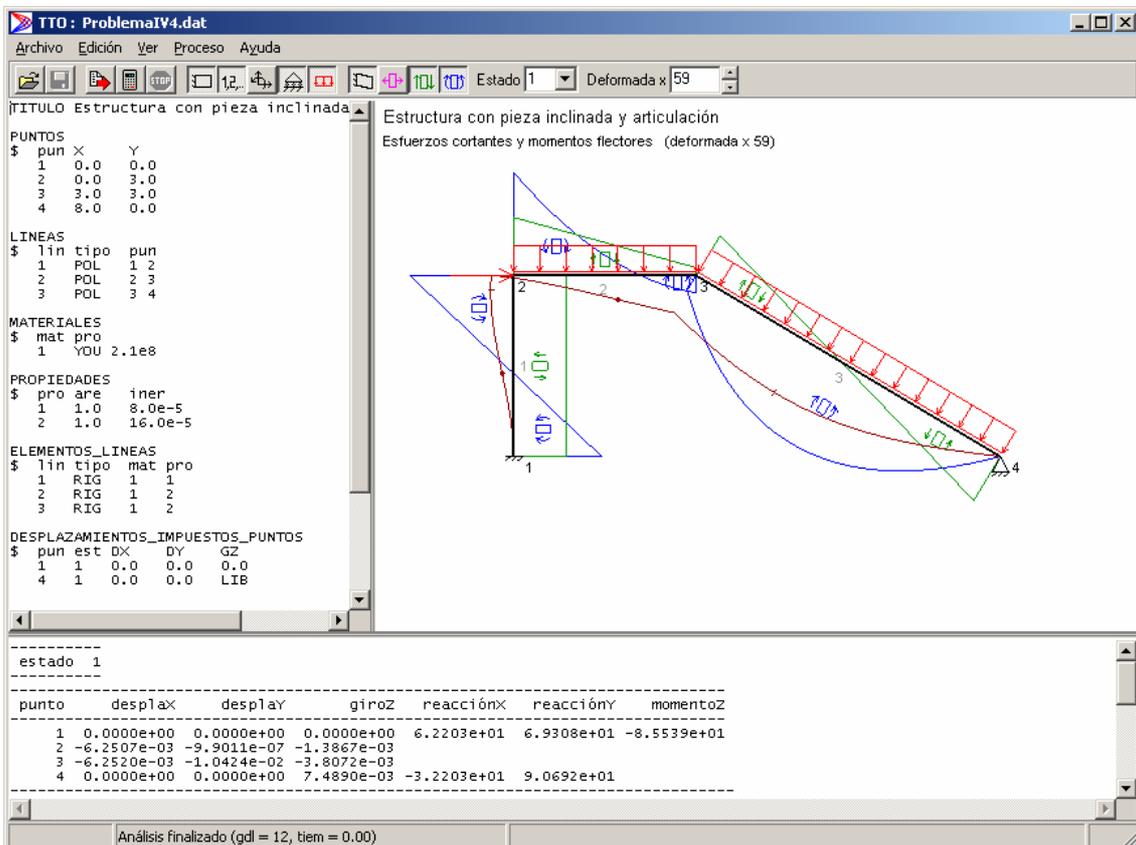


Figura 6.18 Resultados del análisis del problema de la Fig. 6.11

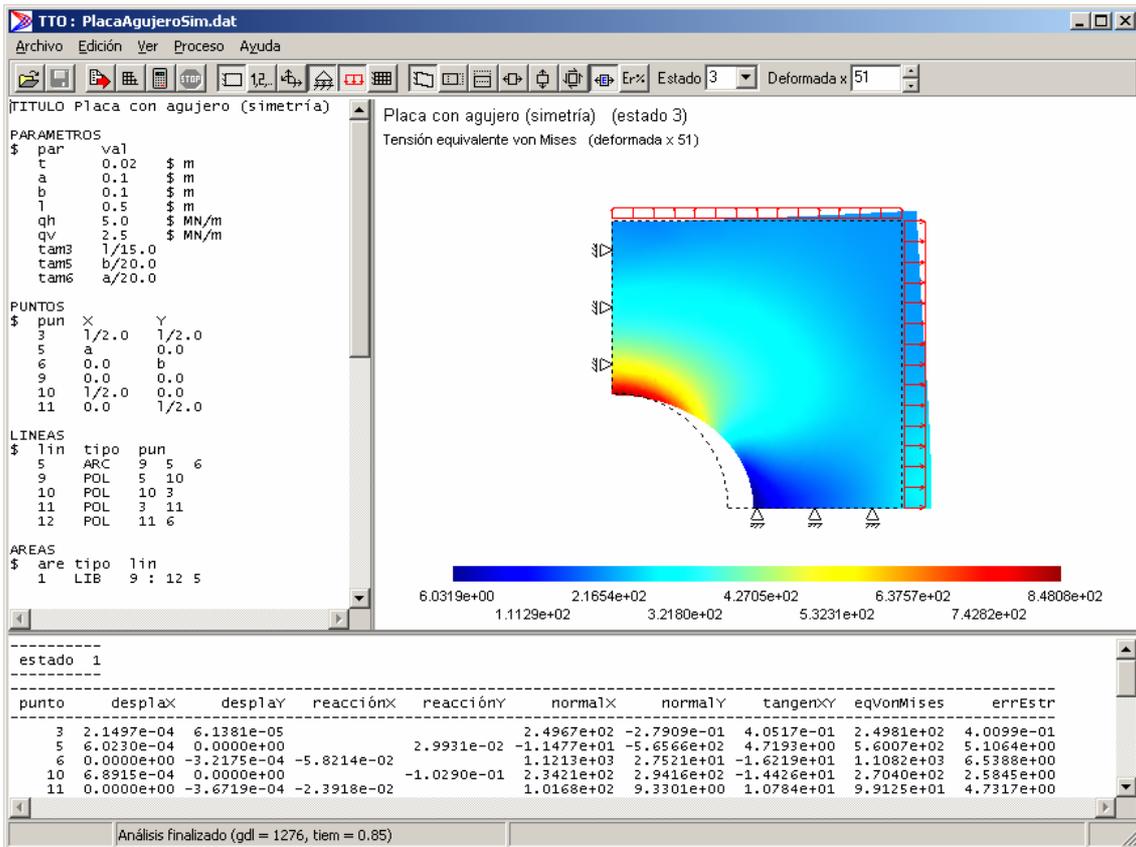


Figura 6.19 Resultados del análisis del problema de la Fig. 6.12

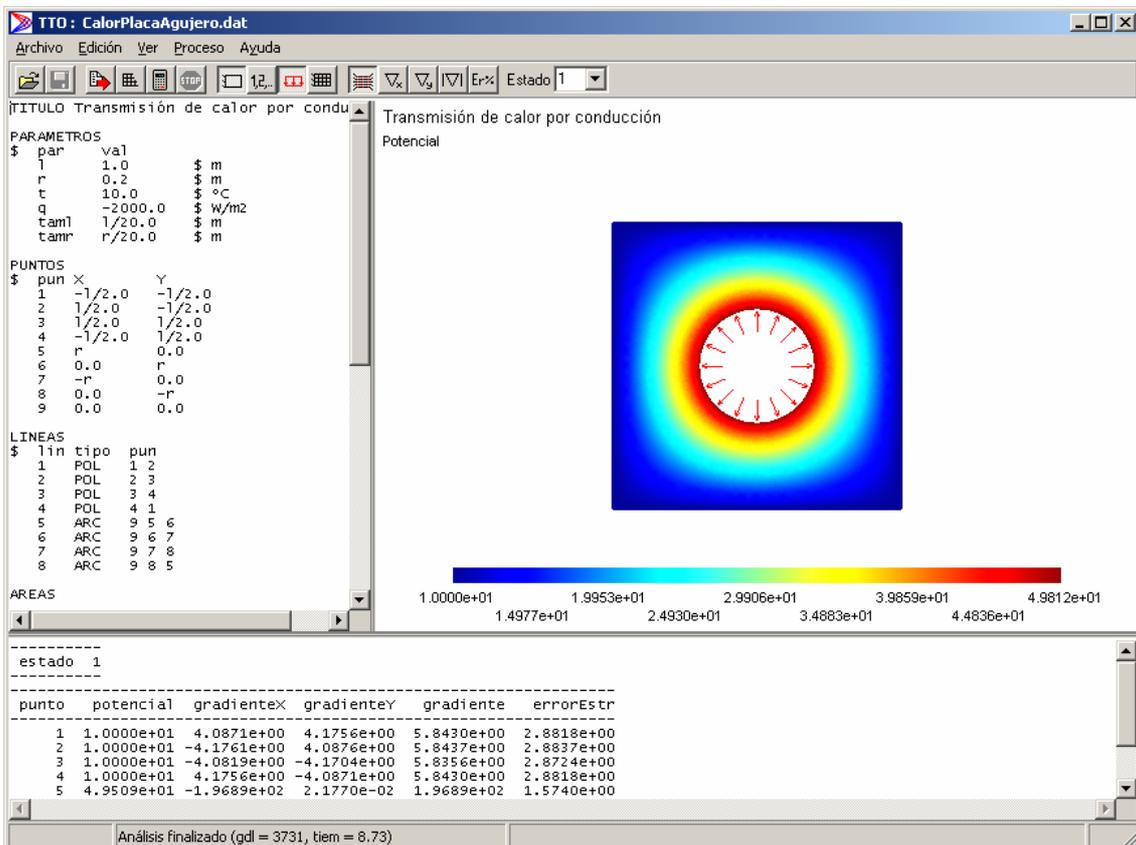


Figura 6.20 Resultados del análisis del problema de la Fig. 6.13

6.3.3 Optimización

TTO obtiene el diseño óptimo simultáneo de topología y geometría de estructuras articuladas sometidas a un estado de cargas y con restricciones de diseño de tensión, pandeo de elementos, y esbeltez.

Para los métodos basados en el universo estructural está implementada la optimización de estructuras tridimensionales (Figs. 6.21 y 6.22). Para el método de crecimiento propuesto sólo está implementada la optimización de estructuras bidimensionales.

En la Fig. 6.21 se muestra el diseño óptimo obtenido (*Optimización->Optimizar*) para el problema de la Fig. 6.14, sin optimización de geometría, y en la Fig. 6.22 el diseño obtenido con optimización de geometría.

Se puede guardar y recuperar (*Optimización->Recuperar*) la evolución del proceso de optimización mediante la caja de diálogo que se muestra en la Fig. 6.21. También se puede trazar (*Optimización->Trazar*) la evolución del proceso en cada iteración para ver con detalle la optimización de topología (puntos probados) y la de geometría (movimiento de los puntos).

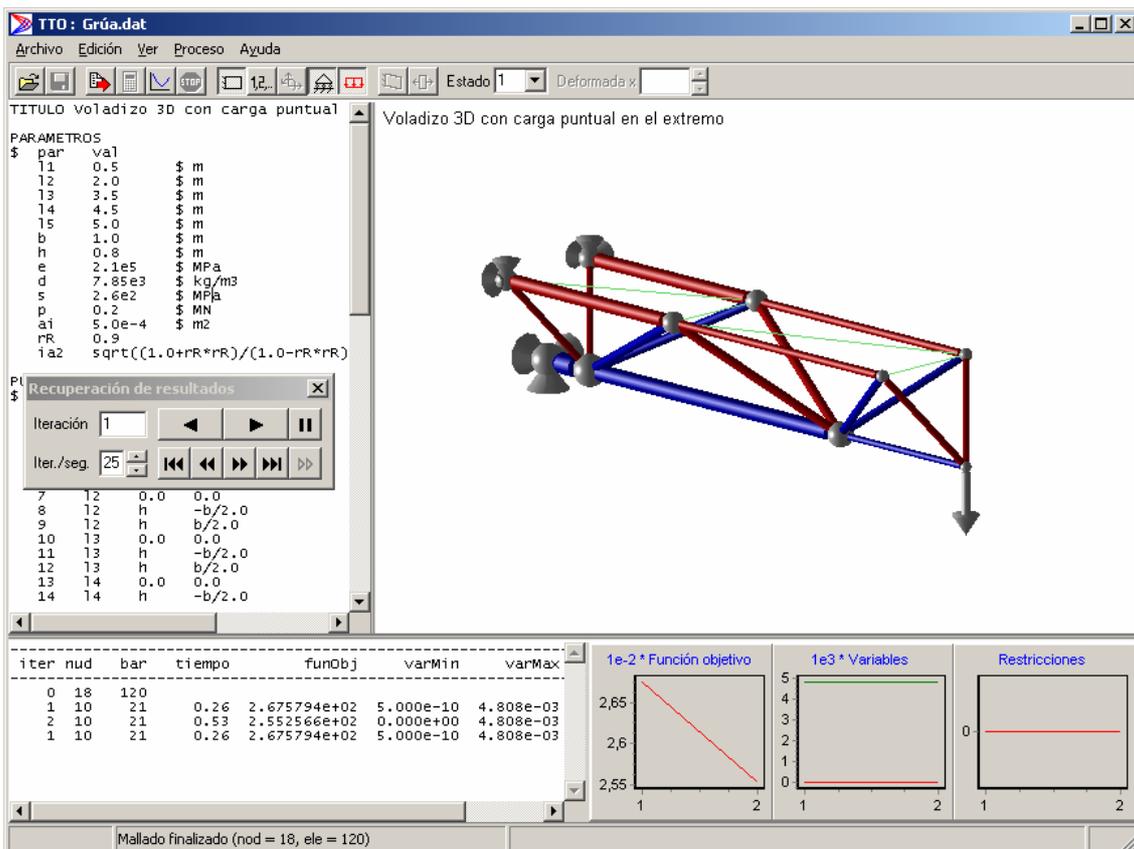


Figura 6.21 Resultados de la optimización del problema de la Fig. 6.14 sin optimización de geometría

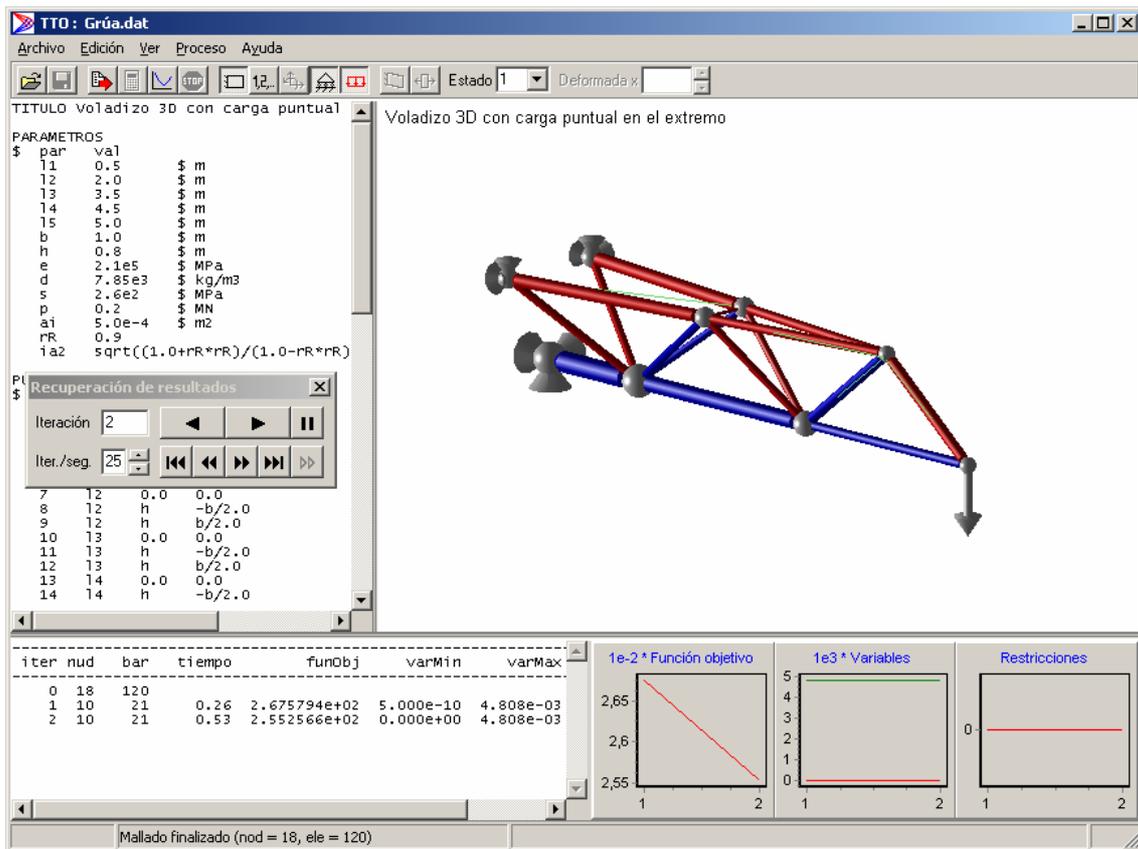


Figura 6.22 Resultados de la optimización del problema de la Fig. 6.14 con optimización de geometría

6.4 ORDENADOR UTILIZADO

Todos los resultados que se muestran en esta tesis han sido obtenidos utilizando un ordenador Pentium III a 1 GHz, con 256 Mbytes de memoria RAM. El sistema operativo utilizado es Windows 2000 Professional.

Ejemplos de aplicación del modelo de diseño óptimo propuesto

7.1 INTRODUCCIÓN

Con la implementación presentada en el capítulo 6 se han obtenido los diseños óptimos de varios ejemplos.

En una primera etapa, se han resuelto varios ejemplos para validar el método propuesto. Una vez que el método propuesto ha sido validado, se estudian posibles aplicaciones del mismo, tales como la determinación de topologías óptimas de estructuras de barras articuladas, su aplicación en el método de bielas y tirantes para estructuras de hormigón armado, o su aplicación para la optimización de topología y geometría de mecanismos.

Todos los ejemplos se han optimizado con el método de crecimiento propuesto (utilizando el método de la *compliance* para la etapa de optimización de topología si sólo hay restricciones de tensión, y el de búsqueda exhaustiva si hay restricciones de pandeo de elementos). Para comparar el método de crecimiento con los métodos universales, los ejemplos se han resuelto mediante el método de la *compliance*. Y para comparar los diferentes métodos del universo estructural entre sí, el voladizo de Michell se ha resuelto con todos los métodos basados en el universo estructural implementados, excepto el de búsqueda exhaustiva (*stress-ratio*, *compliance*, Pedersen y Achtziger).

7.2 EJEMPLOS DE VALIDACIÓN

7.2.1 Introducción

Para validar el método propuesto se han resuelto tres ejemplos cuya solución analítica es conocida (Rozvany, 1998): el voladizo de Michell, la viga de Messerschmidt-Bölkow-Blohm (MBB) y la viga con disco circular rígido. En estos ejemplos sólo se consideran restricciones de tensión.

7.2.2 Voladizo de Michell

El voladizo de Michell consiste en un voladizo con una carga puntual aplicada en el centro del extremo libre (Fig. 7.1a). El dominio de definición del voladizo es un rectángulo de longitud L y altura h . La solución analítica se muestra en la Fig. 7.1b. En la Fig. 7.1c se representan los valores óptimos de la masa adimensional $\bar{M} = M \frac{\sigma_e}{Ph\rho}$ en función de la longitud adimensional $\bar{L} = L/h$, siendo M la masa, σ_e el límite elástico, P la carga puntual, h la altura, ρ la densidad y L la longitud.

Los puntos A , D y J (Fig. 7.1c) se corresponden con longitudes adimensionales a partir de las cuales la topología óptima cambia. Se va a resolver este ejemplo para estas tres longitudes adimensionales.

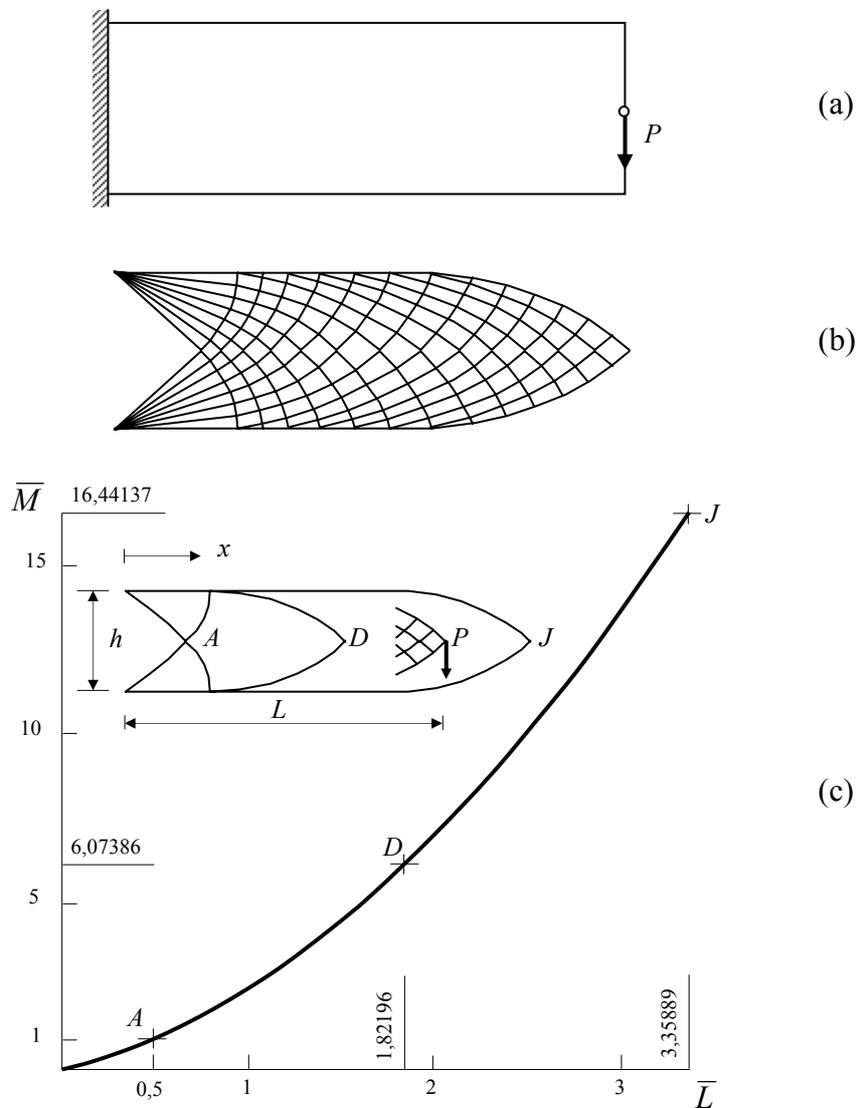


Figura 7.1 (a) Voladizo de Michell. (b) Solución analítica. (c) Valores óptimos de la masa (Rozvany, 1998)

Los datos del problema son: $h = 1 \text{ m}$; $\rho = 7850 \text{ kg/m}^3$; $\sigma_e = 260 \text{ MPa}$, y $P = \sigma_e/h\rho$ (para que no sea necesario adimensionalizar la masa).

En los universos estructurales la primera cifra se refiere al número de divisiones en dirección horizontal, la segunda al número de divisiones en dirección vertical y la tercera el grado de vecindad, que por defecto no se limita. El número de divisiones verticales debe ser par (la carga está aplicada en el punto medio) y el número de divisiones horizontal se ha elegido de forma que las celdas sean lo más cuadradas posible.

Para mejorar las topologías óptimas conseguidas con los métodos del universo estructural, se ha optimizado la geometría final, permitiendo el movimiento de los grados de libertad de todos los nudos, excepto los cargados.

Se comparan los óptimos obtenidos con los métodos del universo estructural, sin optimización de geometría y con optimización de geometría, con los obtenidos con el método propuesto.

El error cometido es la diferencia relativa de la función objetivo obtenida con respecto a la solución analítica con infinitos nudos. Los tiempos de cálculo, para todos los ejemplos que siguen, se refieren a los obtenidos con un Pentium III a 1 GHz.

Se ha establecido como límite un número máximo de nudos de la estructura óptima de 100 (aproximadamente) o un tiempo de cálculo máximo del orden de 1000 segundos en todos los ejemplos de validación.

7.2.2.1 Longitud adimensional $\bar{L} = 0,5$

La solución para este caso consiste en una estructura con tres nudos y las dos barras que unen el punto cargado con los dos puntos extremos de la sección del empotramiento.

En los métodos basados en el universo estructural es suficiente con 1x2 divisiones para obtener el óptimo y en el método de crecimiento propuesto es suficiente con el universo estructural inicial (no hay que añadir ningún nudo).

En la Fig. 7.2 se muestran los resultados obtenidos mediante todos los métodos del universo estructural implementados, y en la Fig. 7.3 los obtenidos mediante el método de crecimiento propuesto. En lo sucesivo, y a efectos de representación gráfica, se consideran barras tubulares cilíndricas con pared de espesor igual al 10% del radio. El color representa la tensión, variando desde el azul (mínimo) al rojo (máximo).

El error es nulo para todos los métodos, ya que todos obtienen la solución analítica. El tiempo de cálculo es demasiado pequeño como para que la comparativa entre los diferentes métodos sea fiable.

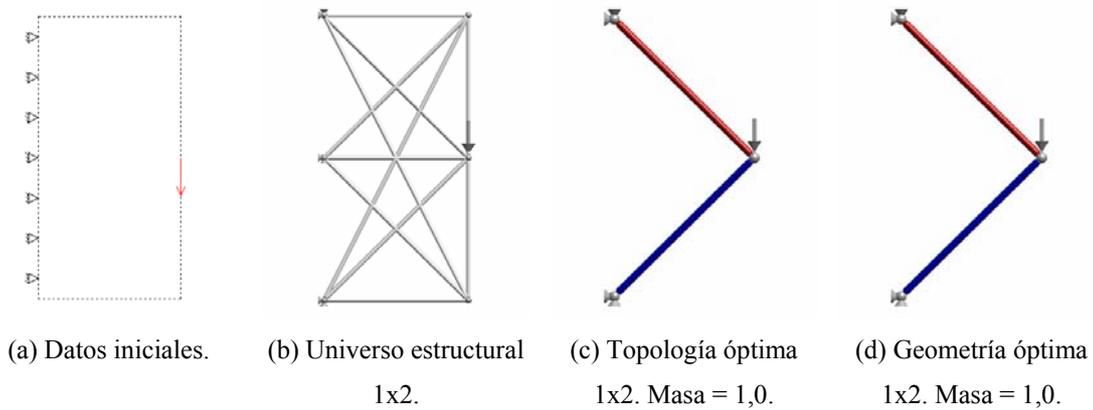


Figura 7.2 Voladizo de Michell ($\bar{L} = 0,5$). Diseños óptimos obtenidos mediante todos los métodos del universo estructural y optimización final de geometría

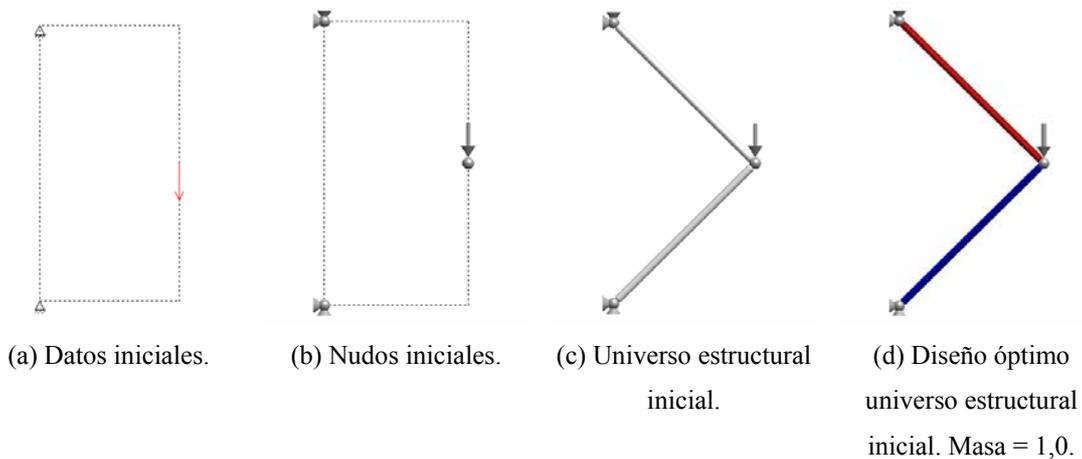


Figura 7.3 Voladizo de Michell ($\bar{L} = 0,5$). Diseño óptimo obtenido mediante el método de crecimiento propuesto

7.2.2.2 Longitud adimensional $\bar{L} = 1,82196$

Para longitudes adimensionales mayores que 0,5, la solución analítica está formada por infinitos nudos e infinitas barras, de ahí que no pueda alcanzarse el óptimo.

En la Fig. 7.4 se muestran los óptimos obtenidos mediante los métodos del universo estructural para diferentes universos iniciales, y en la Fig. 7.5 los obtenidos con el método de crecimiento propuesto para diferentes números de nudos añadidos.

En la primera columna de la Fig. 7.4 aparecen los universos estructurales iniciales, en la segunda las topologías óptimas obtenidas con los diferentes métodos implementados, y en la tercera los diseños óptimos obtenidos al optimizar la geometría de las topologías óptimas anteriores.

El método de búsqueda exhaustiva tiene un coste computacional muy alto, por ello sólo se aplica a los universos estructurales de 1x2 y 4x2x1.

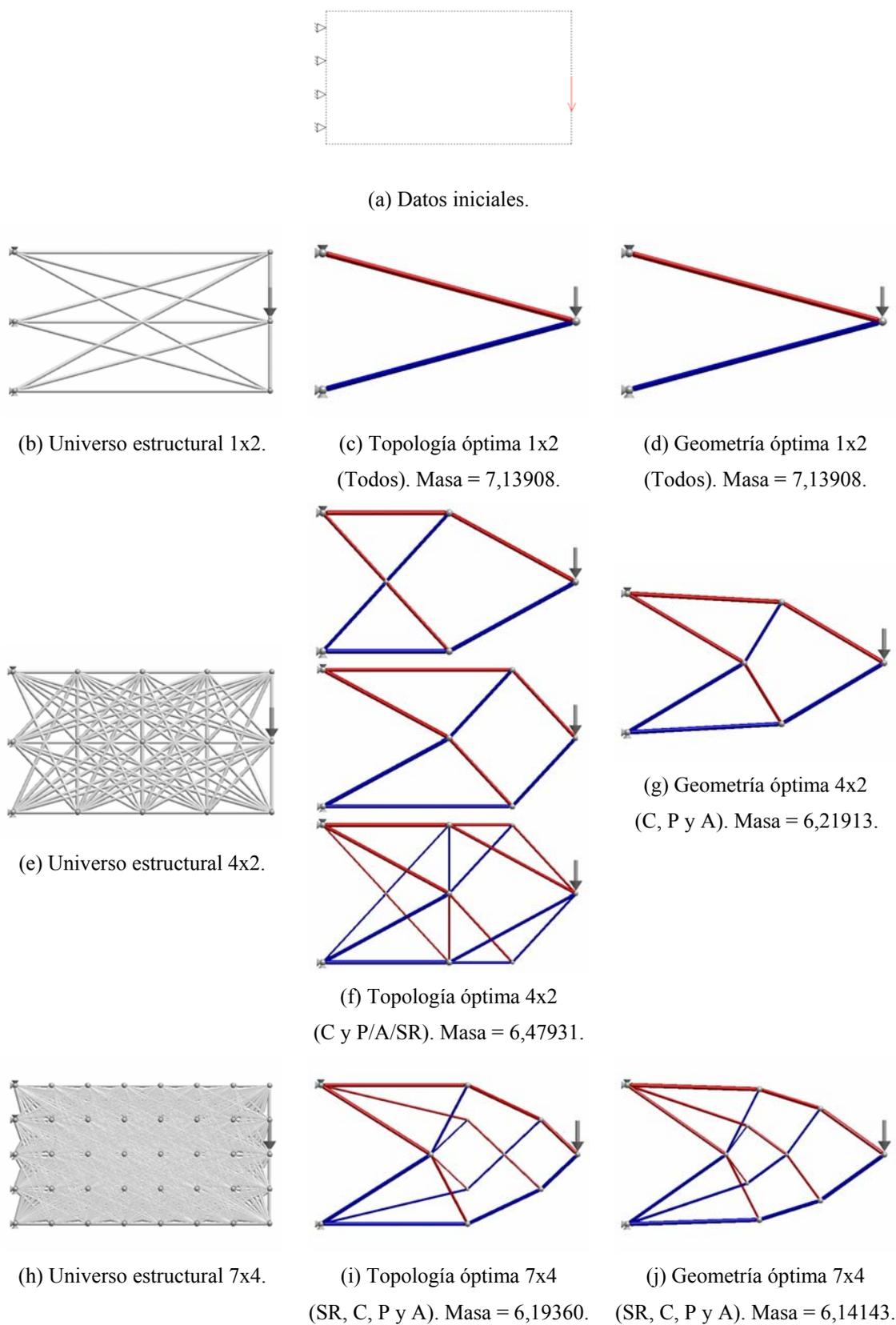


Figura 7.4 Voladizo de Michell ($\bar{L} = 1,82196$). Diseños óptimos obtenidos mediante métodos del universo estructural

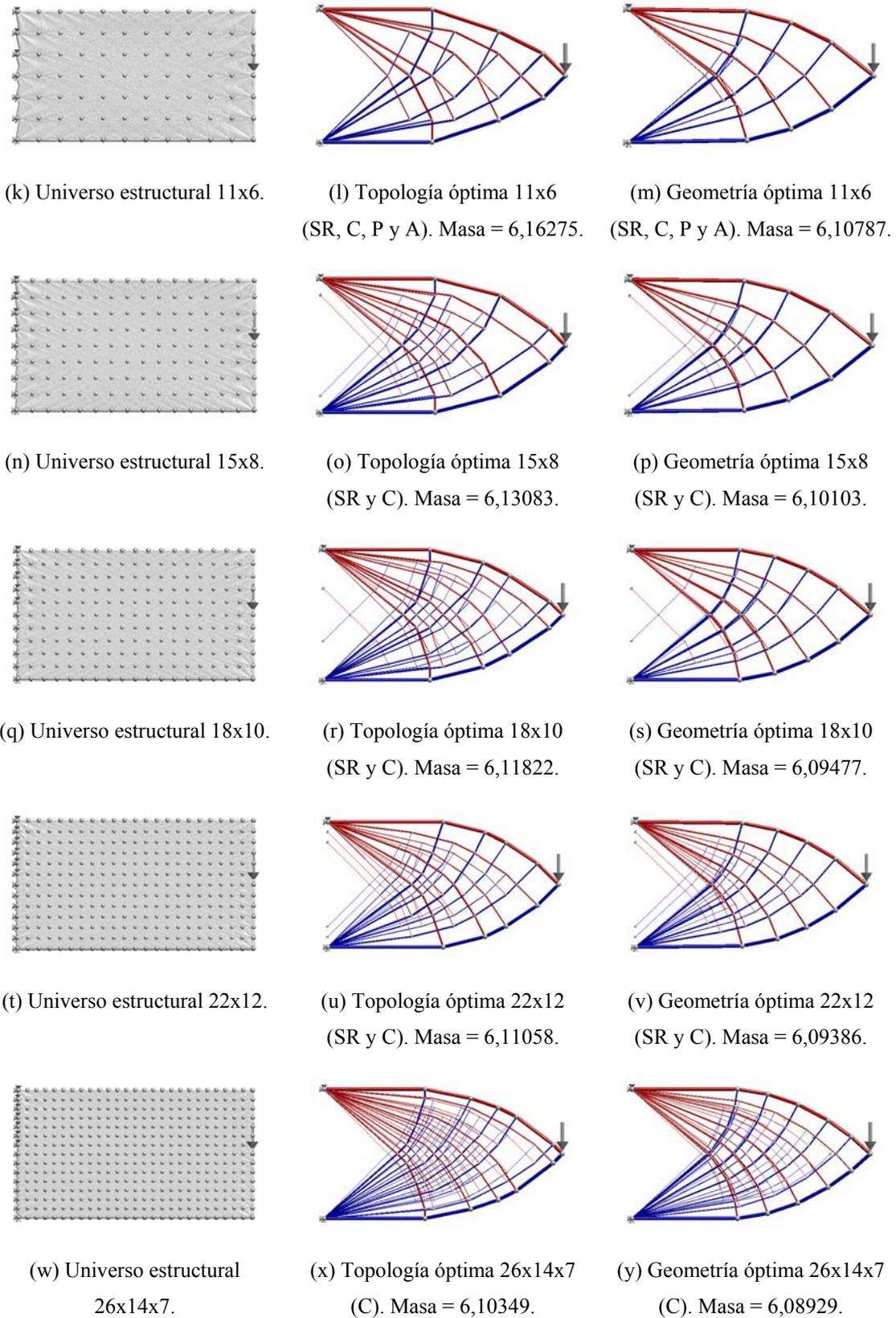


Figura 7.4 Voladizo de Michell ($\bar{L} = 1,82196$). Diseños óptimos obtenidos mediante métodos del universo estructural. (Continuación)

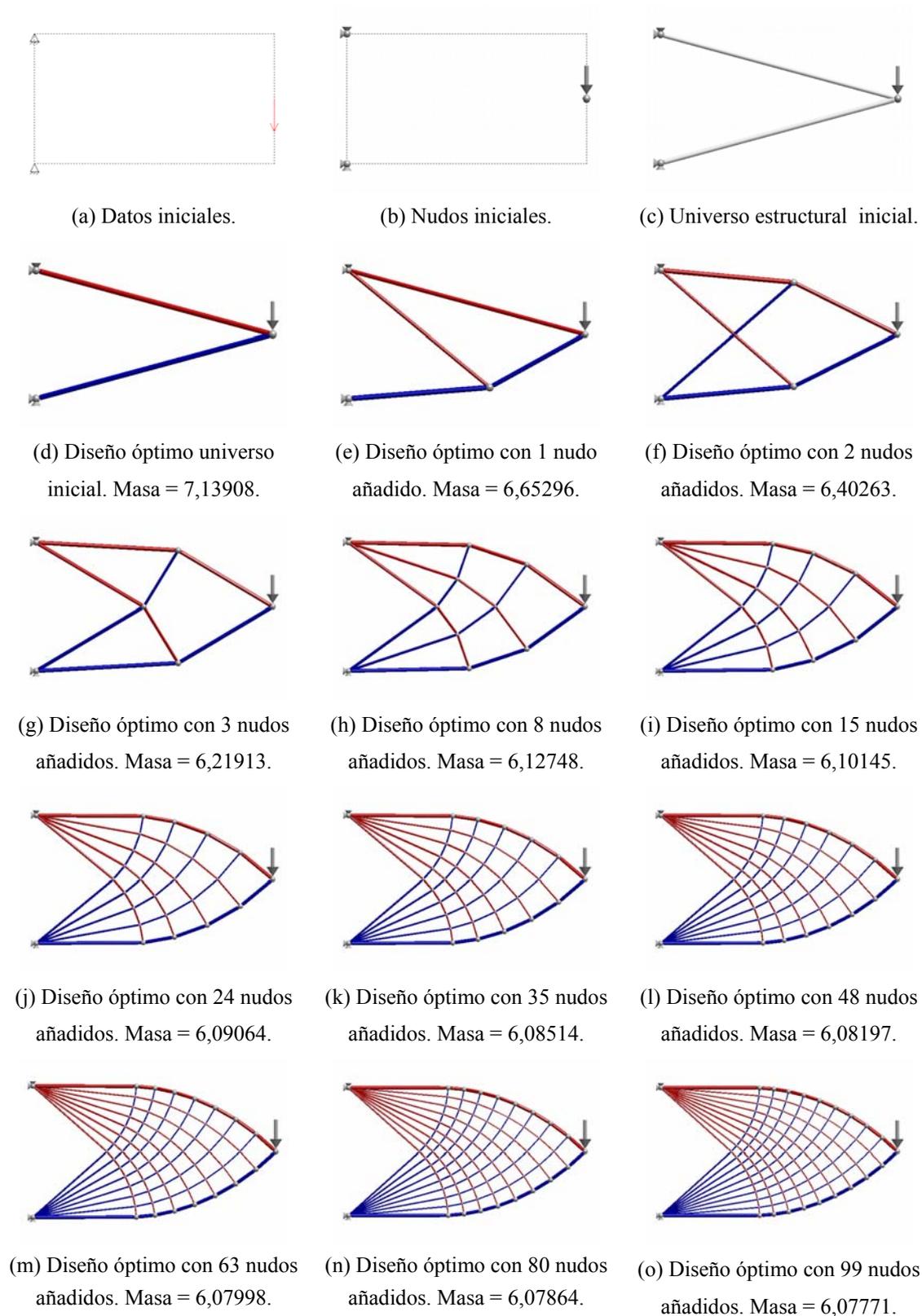


Figura 7.5 Voladizo de Michell ($\bar{L} = 1,82196$). Diseños óptimos obtenidos mediante el método de crecimiento propuesto

El algoritmo de resolución del *simplex* (DSLPRS, de la librería matemática IMSL[®]) utilizado en el método de la *compliance*, para el universo estructural 26x14, tiene problemas de convergencia, de ahí que se haya limitado el orden de vecindad a 7 en dirección vertical (universo estructural 26x14x7), lo que no impide que se alcance la misma solución que se obtendría con el universo estructural completo, puesto que la solución es simétrica.

Es relativamente frecuente, en los métodos basados en el universo estructural, que el óptimo no sea único, es decir, que haya varias topologías para las cuales la función objetivo mínima tenga el mismo valor. El universo con 4x2 divisiones (Fig. 7.4e) es un ejemplo de mínimo no único. Con los métodos de la *compliance* (C) y el de Pedersen (P) se consigue la misma topología óptima; con el método de Achtziger (A) se obtiene otra diferente, y con el método *stress-ratio* (SR) no se llega a una topología isostática, sino que se obtiene una mezcla de varias topologías isostáticas óptimas. El algoritmo propuesto no permite la optimización de estructuras articuladas hiperestáticas, de ahí que no aparezca la geometría óptima para este caso.

En las tablas 7.1 a 7.5 se muestran los resultados obtenidos mediante los métodos basados en el universo estructural: *stress-ratio*, *compliance*, Pedersen, Achtziger y búsqueda exhaustiva, respectivamente. En la tabla 7.6 se muestran los resultados conseguidos con el método de crecimiento propuesto.

Tabla 7.1 Voladizo de Michell ($\bar{L} = 1,82196$).

Diseños óptimos obtenidos mediante el método *stress-ratio*

div	nudos		barras		masa (adim)		error (%)		tiempo (s)	
	ini	fin	ini	fin	top	geo	top	geo	top	geo
1x2	6	3	11	2	7,13908	7,13908	17,538	17,538	0,03	0,03
4x2	15	9	72	18	6,47931	—	6,675	—	0,1	—
7x4	40	11	499	18	6,19360	6,14143	1,971	1,113	1,1	1,19
11x6	84	24	2171	44	6,16275	6,10787	1,463	0,560	129	129
15x8	144	43	6369	78	6,13083	6,10103	0,938	0,447	196	198
18x10	209	59	13322	110	6,11822	6,09477	0,730	0,344	1262	1265
22x12	299	66	27242	120	6,11058	6,09386	0,605	0,329	3543	3551
Solución analítica					6,07386	6,07386	0,000	0,000		

Tabla 7.2 Voladizo de Michell ($\bar{L} = 1,82196$).Diseños óptimos obtenidos mediante el método de la *compliance*

div	nudos		barras		masa (adim)		error (%)		tiempo (s)	
	ini	fin	ini	fin	top	geo	top	geo	top	geo
1x2	6	3	11	2	7,13908	7,13908	17,538	17,538	0,01	0,01
4x2	15	6	72	8	6,47931	6,21913	6,675	2,392	0,02	0,02
7x4	40	11	499	18	6,19360	6,14143	1,971	1,113	0,17	0,21
11x6	84	24	2171	44	6,16275	6,10787	1,463	0,560	6,01	6,32
15x8	144	43	6369	78	6,13083	6,10103	0,938	0,447	75,6	77,5
18x10	209	59	13322	110	6,11822	6,09477	0,730	0,344	491	494
22x12	299	66	27242	120	6,11058	6,09386	0,605	0,329	1867	1874
26x14x7	405	105	38158	206	6,10349	6,08929	0,488	0,254	5512	5523
Solución analítica					6,07386	6,07386	0,000	0,000		

Tabla 7.3 Voladizo de Michell ($\bar{L} = 1,82196$).

Diseños óptimos obtenidos mediante el método de Pedersen

div	nudos		barras		masa (adim)		error (%)		tiempo (s)	
	ini	fin	ini	fin	top	geo	top	geo	top	geo
1x2	6	3	11	2	7,13908	7,13908	17,538	17,538	0,01	0,01
4x2	15	6	72	8	6,47931	6,21913	6,675	2,392	0,02	0,02
7x4	40	11	499	18	6,19360	6,14143	1,971	1,113	8,57	8,6
11x6	84	24	2171	44	6,16275	6,10787	1,463	0,560	1234	1234
Solución analítica					6,07386	6,07386	0,000	0,000		

Tabla 7.4 Voladizo de Michell ($\bar{L} = 1,82196$).

Diseños óptimos obtenidos mediante el método de Achtziger

div	nudos		barras		masa (adim)		error (%)		tiempo (s)	
	ini	fin	ini	fin	top	geo	top	geo	top	geo
1x2	6	3	11	2	7,13908	7,13908	17,538	17,538	0,01	0,01
4x2	15	6	72	8	6,47931	6,21913	6,675	2,392	0,05	0,06
7x4	40	11	499	18	6,19360	6,14143	1,971	1,113	8,69	8,73
11x6	84	24	2171	44	6,16275	6,10787	1,463	0,560	300	300
Solución analítica					6,07386	6,07386	0,000	0,000		

Tabla 7.5 Voladizo de Michell ($\bar{L} = 1,82196$).

Diseños óptimos obtenidos mediante el método de búsqueda exhaustiva

div	nudos		barras		masa (adim)		error (%)		tiempo (s)	
	ini	fin	ini	fin	top	geo	top	geo	top	geo
1x2	6	3	11	2	7,13908	7,13908	17,538	17,538	0,01	0,01
4x2x1	15	9	36	12	6,97931	6,41081	14,907	5,548	3572	3572
Solución analítica					6,07386	6,07386	0,000	0,000		

Tabla 7.6 Voladizo de Michell ($\bar{L} = 1,82196$).

Diseños óptimos obtenidos mediante el método de crecimiento propuesto

añadidos	nudos	barras	masa (adim)	error (%)	tiempo (s)
0	3	2	7,13908	17,538	0,003
1	4	4	6,65296	9,534	0,005
2	5	6	6,40263	5,413	0,01
3	6	8	6,21913	2,392	0,02
8	11	18	6,12748	0,883	0,2
15	18	32	6,10145	0,454	1,03
24	27	50	6,09064	0,276	4,22
35	38	72	6,08514	0,186	15,3
48	51	98	6,08197	0,133	46,9
63	66	128	6,07998	0,101	129
80	83	162	6,07864	0,079	326
99	102	200	6,07771	0,063	761
Solución analítica			6,07386	0,000	

En las gráficas de la Fig. 7.6 se comparan la evolución del error (en %) y del tiempo de cálculo (en segundos), en función del número de nudos de la topología óptima obtenida para los diferentes métodos implementados y para diferentes universos estructurales (métodos basados en el universo estructural) sin optimización de geometría, y para diferente número de nudos añadidos (método de crecimiento propuesto). Debido a que los errores para las estructuras obtenidas con menos de 7 nudos son muy altos, no se han representado en estas gráficas. En las gráficas de la Fig. 7.7 se muestran las mismas gráficas con optimización de geometría.

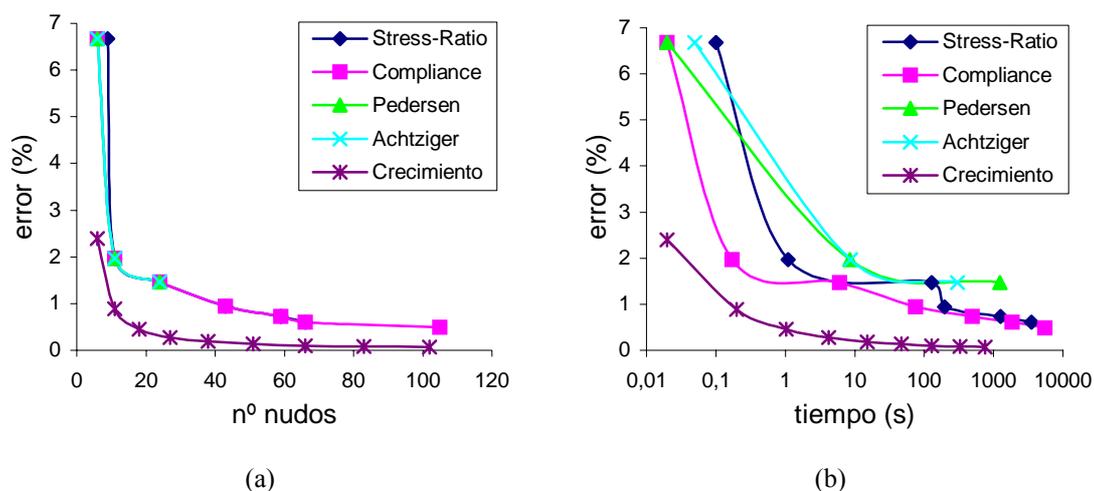


Figura 7.6 Voladizo de Michell ($\bar{L} = 1,82196$). Métodos del universo estructural (sin optimización de geometría) y de crecimiento propuesto. (a) Evolución error/nº nudos; (b) Evolución error/tiempo

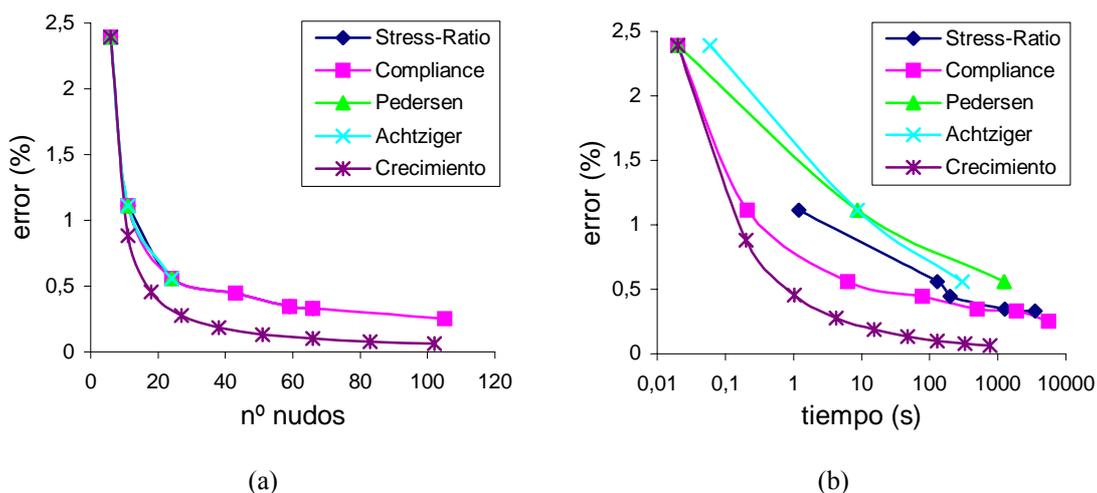


Figura 7.7 Voladizo de Michell ($\bar{L} = 1,82196$). Métodos del universo estructural (con optimización de geometría) y de crecimiento propuesto. (a) Evolución error/nº nudos; (b) Evolución error/tiempo

En las Figs. 7.6a y 7.7a puede apreciarse que para obtener un mismo error, con el método de crecimiento propuesto que con los del universo estructural, se necesitan menos nudos (incluso si se hace una optimización de geometría final en los métodos del universo estructural). También se observa que el tiempo de cálculo, para un error determinado, es significativamente menor en el método propuesto (Figs. 7.6b y 7.7b).

En las tablas 7.1 a 7.6 se observa que, por ejemplo, para obtener un error menor del 0,5 % se necesitan 43 nudos con los métodos del universo estructural (incluyendo una optimización de geometría final), mientras que para el método propuesto es suficiente con 18 nudos. Para este mismo caso, y para el método de la *compliance* (el

más rápido de los métodos universales implementados), el tiempo de cálculo es de 77,5 segundos, mientras que para el método propuesto se reduce a 1,03 segundos. Si en la optimización mediante los métodos del universo estructural no se efectúa una optimización de geometría final, el número de nudos necesario se eleva a 105 y el tiempo de cálculo a 8636 segundos (para el método de la *compliance*).

Otra ventaja de los diseños obtenidos con el método propuesto con respecto a los obtenidos con los métodos del universo estructural es la “calidad práctica” de los primeros. Los diseños obtenidos con el método de crecimiento tienen menos nudos, para el mismo error, y no tienen cruces de barras.

7.2.2.3 Longitud adimensional $\bar{L} = 3,35889$

Para la longitud adimensional correspondiente al punto J ($\bar{L} = 3,35889$) el problema se va a resolver mediante el método de la *compliance* (el más efectivo de los implementados cuando sólo hay restricciones de tensión), y mediante el método de crecimiento propuesto.

El método propuesto se va a aplicar, en primer lugar, añadiendo los nudos de uno en uno y, en segundo lugar, añadiendo los nudos de dos en dos, aprovechando la simetría que tiene la solución, como consecuencia de considerar sólo restricciones de tensión.

En la Fig. 7.8 se muestran los diseños óptimos obtenidos mediante el método de la *compliance* para diferentes universos estructurales, y en las Figs. 7.9 y 7.10 los obtenidos con el método de crecimiento para diferente número de nudos añadidos, la 7.9 añadiendo los nudos de uno en uno y la 7.10 añadiéndolos de dos en dos.

En la tabla 7.7 se muestran los resultados obtenidos mediante el método de la *compliance*. En la tabla 7.8 se muestran los resultados obtenidos con el método propuesto añadiendo los nudos de uno en uno y en la tabla 7.9 los resultados obtenidos añadiéndolos de dos en dos.

Para longitudes adimensionales pequeñas, la solución obtenida añadiendo los nudos de uno en uno o de dos en dos es la misma (por ejemplo para $\bar{L} = 1,82196$); sin embargo, para valores altos de la longitud adimensional, y para pocos nudos, las soluciones pueden ser diferentes, aunque al aumentar el número de nudos añadidos las soluciones convergen.

En el método de la *compliance*, para longitudes adimensionales grandes y universos estructurales pequeños, se consiguen buenos resultados, sobre todo si se hace una optimización de geometría final. El óptimo de la Fig. 7.8g coincide con el de la Fig. 7.10h, es decir, la topología obtenida, con optimización final de geometría, es la misma que la obtenida con el método de crecimiento propuesto. Sin embargo, al aumentar el tamaño de los universos estructurales, los resultados son claramente peores que los obtenidos con el método de crecimiento.

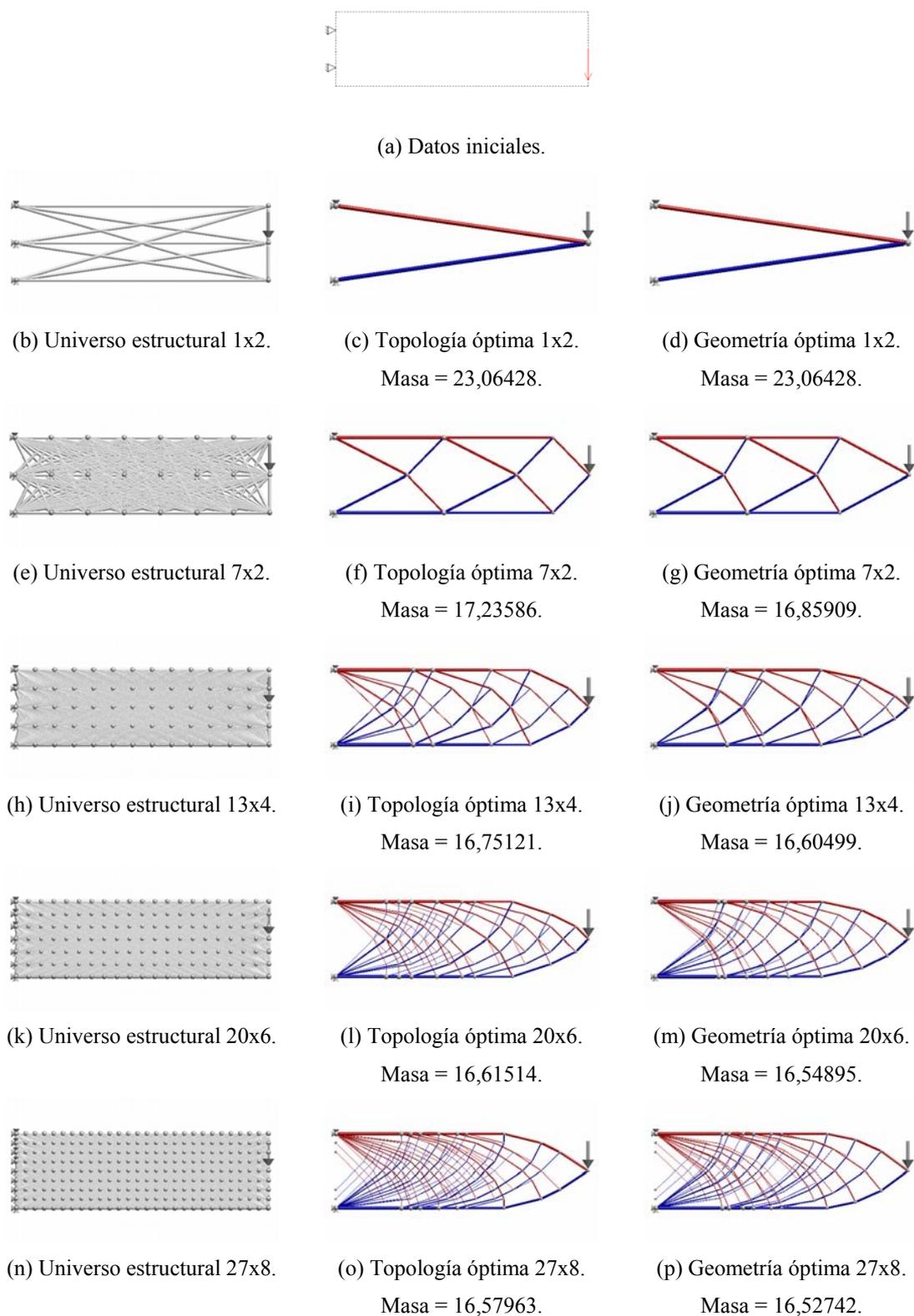


Figura 7.8 Voladizo de Michell ($\bar{L} = 3,35889$). Diseños óptimos obtenidos mediante el método de la *compliance*

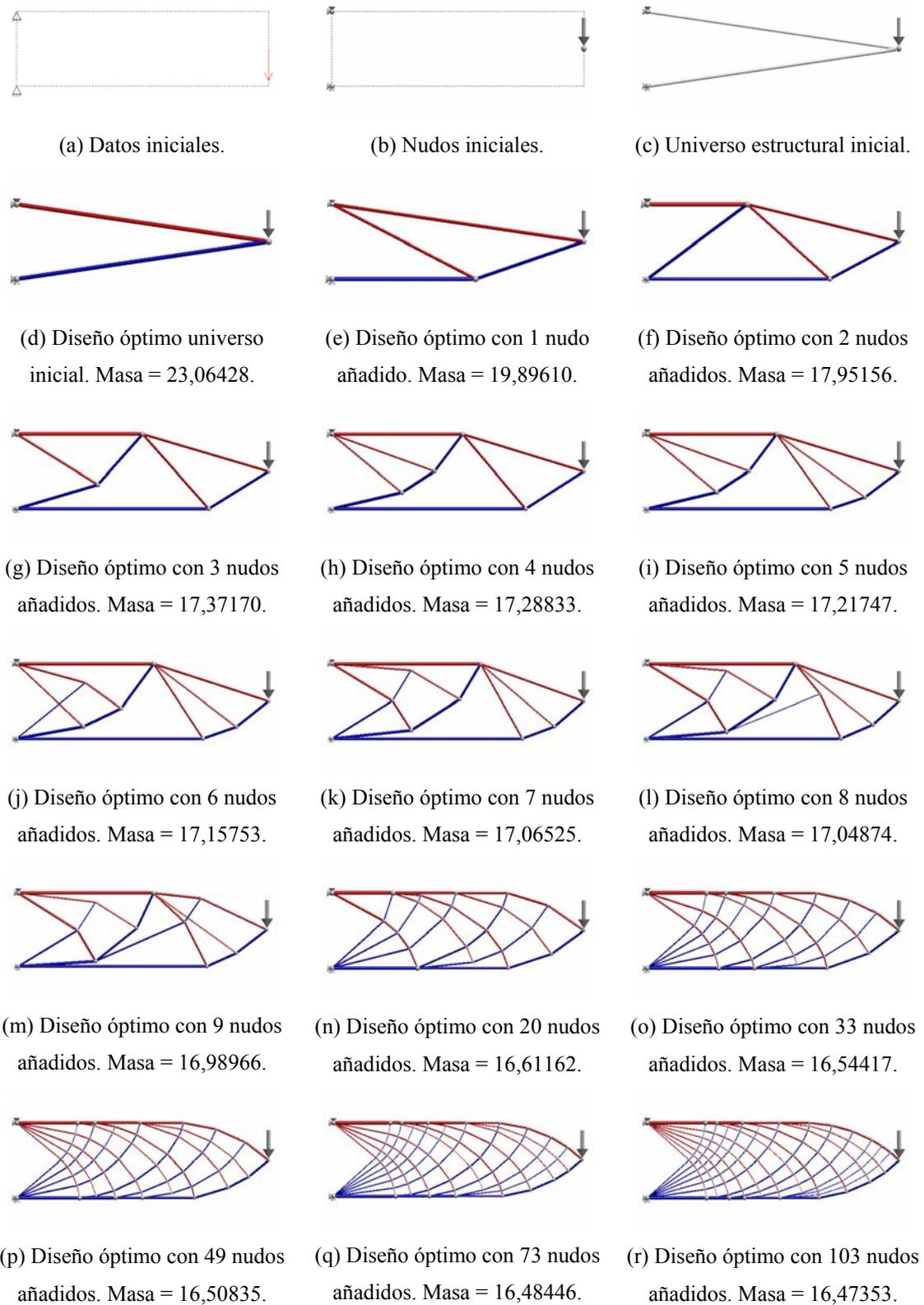


Figura 7.9 Voladizo de Michell ($\bar{L} = 3,35889$). Diseños óptimos obtenidos mediante el método de crecimiento propuesto, añadiendo los nudos de uno en uno

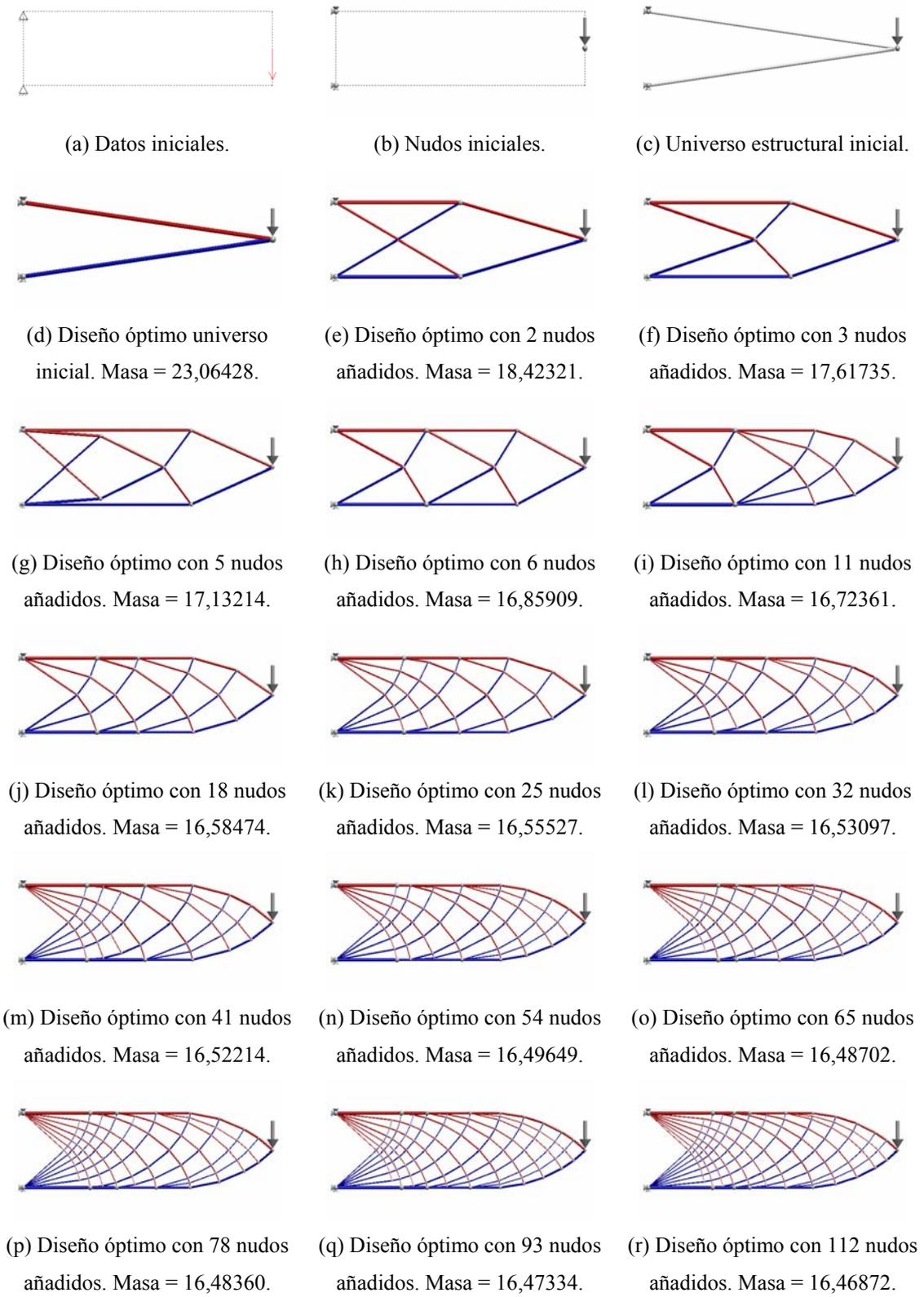


Figura 7.10 Voladizo de Michell ($\bar{L} = 3,35889$). Diseños óptimos obtenidos mediante el método de crecimiento propuesto, añadiendo los nudos de dos en dos

Tabla 7.7 Voladizo de Michell ($\bar{L} = 3,35889$).Diseños óptimos obtenidos mediante el método de la *compliance*

div	nudos		barras		masa (adim)		error (%)		tiempo (s)	
	ini	fin	ini	fin	top	geo	top	geo	top	geo
1x2	6	3	11	2	23,06428	23,06428	40,282	40,282	0,01	0,01
7x2	24	9	179	14	17,23586	16,85909	4,832	2,541	0,04	0,04
13x4	70	30	1497	56	16,75121	16,60499	1,885	0,995	2,92	3,66
20x6	147	70	6568	136	16,61514	16,54895	1,057	0,654	90,2	110
27x8	252	113	19369	214	16,57963	16,52742	0,841	0,523	1188	1217
Solución analítica					16,44137	16,44137	0,000	0,000		

Tabla 7.8 Voladizo de Michell ($\bar{L} = 3,35889$).

Diseños óptimos obtenidos mediante el método de crecimiento propuesto

(añadiendo los nudos de uno en uno y de dos en dos)

añadidos	nudos	barras	masa (adim)		error (%)		tiempo (s)	
			1 en 1	2 en 2	1 en 1	2 en 2	1 en 1	2 en 2
0	3	2	23,06428	23,06428	40,282	40,282	0,003	0,003
1	4	4	19,89610	—	21,012	—	0,005	—
2	5	6	17,95156	18,42321	9,185	12,054	0,01	0,01
3	6	8	17,37170	17,61735	5,658	7,153	0,02	0,02
4	7	10	17,28833	—	5,151	—	0,03	—
5	8	12	17,21747	17,13214	4,720	4,201	0,09	0,05
7	9	14	17,15753	16,85909	4,356	2,541	0,16	0,06
8	10	16	17,06525	—	3,795	—	0,2	—
9	11	18	17,04874	16,83453	3,694	2,391	0,26	0,09
10	12	20	16,98966	—	3,335	—	0,38	—
11	14	24	16,83633	16,72361	2,402	1,717	0,61	0,2
18	21	38	16,61976	16,58474	1,085	0,872	3,76	0,77
22	23	42	16,61162	16,58108	1,035	0,850	4,45	1,03
25	28	52	16,57108	16,55527	0,789	0,693	7,11	1,98
32	35	66	16,54417	16,53097	0,625	0,545	14,4	4,28
35	36	68	16,53477	—	0,568	—	15,2	—
41	44	84	16,51918	16,52214	0,473	0,491	29,9	9,42
51	52	100	16,50835	16,50934	0,407	0,413	55,9	17,7
54	57	110	16,50112	16,49649	0,363	0,335	79,6	24,8
65	68	132	16,48833	16,48702	0,286	0,278	166	51,9
75	76	148	16,48446	—	0,262	—	244	—
78	81	158	16,48379	16,48360	0,258	0,257	308	94,4
93	96	188	16,47638	16,47334	0,213	0,194	601	189
105	106	208	16,47353	16,47269	0,196	0,190	915	273
112	115	226	16,46856	16,46872	0,165	0,166	1302	405
Solución analítica			16,44137	16,44137	0,000	0,000		

En la Fig. 7.11 se representan las gráficas comparativas de la evolución del error (%) y del tiempo de cálculo (segundos), en función del número de nudos de la topología obtenida para diferentes universos estructurales (método de la *compliance*) o para diferente número de nudos añadidos (método de crecimiento propuesto). Debido a que los errores para las estructuras obtenidas con menos de 9 nudos son muy altos, no se han representado en estas gráficas.

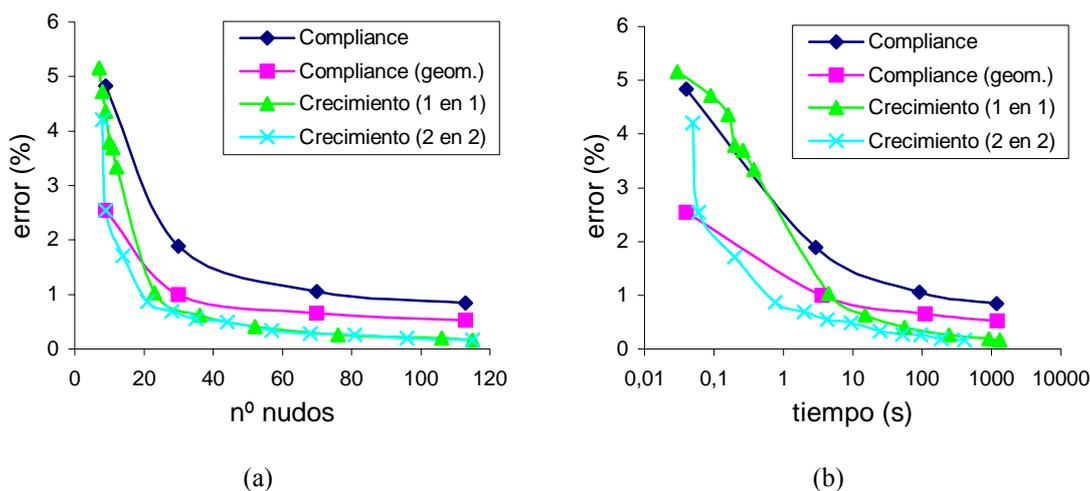


Figura 7.11 Voladizo de Michell ($\bar{L} = 3,35889$). Método de la *compliance* (sin y con optimización de geometría) y de crecimiento propuesto (añadiendo los nudos de uno en uno y de dos en dos).
(a) Evolución error/nº nudos; (b) Evolución error/tiempo

En la Fig. 7.11a puede apreciarse que para obtener un mismo error, con el método de crecimiento propuesto que con los del universo estructural, se necesitan menos nudos (incluso si se hace una optimización de geometría final). También se observa que el tiempo de cálculo, para un error determinado, es menor en el método propuesto (Figs. 7.11b), salvo para soluciones con pocos nudos donde el tiempo (menos de 5 segundos cuando se añaden los nudos de uno en uno) es poco significativo.

En las Figs. 7.9 y 7.10 se puede apreciar la diferencia al añadir los nudos, mediante el método propuesto, de uno en uno o de dos en dos. Para un número alto de nudos añadidos la solución tiende a ser simétrica, sin embargo, para un número bajo de nudos añadidos puede no serlo. Por ejemplo, para dos nudos añadidos la solución no simétrica (Fig. 7.9f) es mejor que la simétrica (Fig. 7.10e), pero esto hace que las topologías posteriores, cuando se añaden los nudos de uno en uno, cambien, por lo que puede ocurrir que el camino escogido, en el crecimiento de la topología, no sea el mejor.

En la Fig. 7.11a se observa que, a partir de un número determinado de nudos, es mejor la solución añadiendo los nudos de dos en dos que añadiéndolos de uno en uno, a pesar de que hay más variables, en la optimización de geometría, si no se imponen condiciones de simetría.

7.2.3 Viga de Messerschmidt-Bölkow-Blohm

La viga de Messerschmidt-Bölkow-Blohm (viga MBB, Fig. 7.12a) es atribuida a la empresa aeronáutica Messerschmidt-Bölkow-Blohm y concierne al piso del avión de pasajeros Airbus. El dominio de definición de la viga es un rectángulo de longitud $2L$ y altura h . La solución analítica, aplicando simetría, se muestra en la Fig. 7.12b. En la Fig. 7.12c se representan los valores óptimos de la masa adimensional $\bar{M} = M \frac{\sigma_e}{Ph\rho}$ en función de la longitud adimensional $\bar{L} = L/h$, siendo M la masa, σ_e el límite elástico, P la carga puntual, h la altura, ρ la densidad y L la longitud.

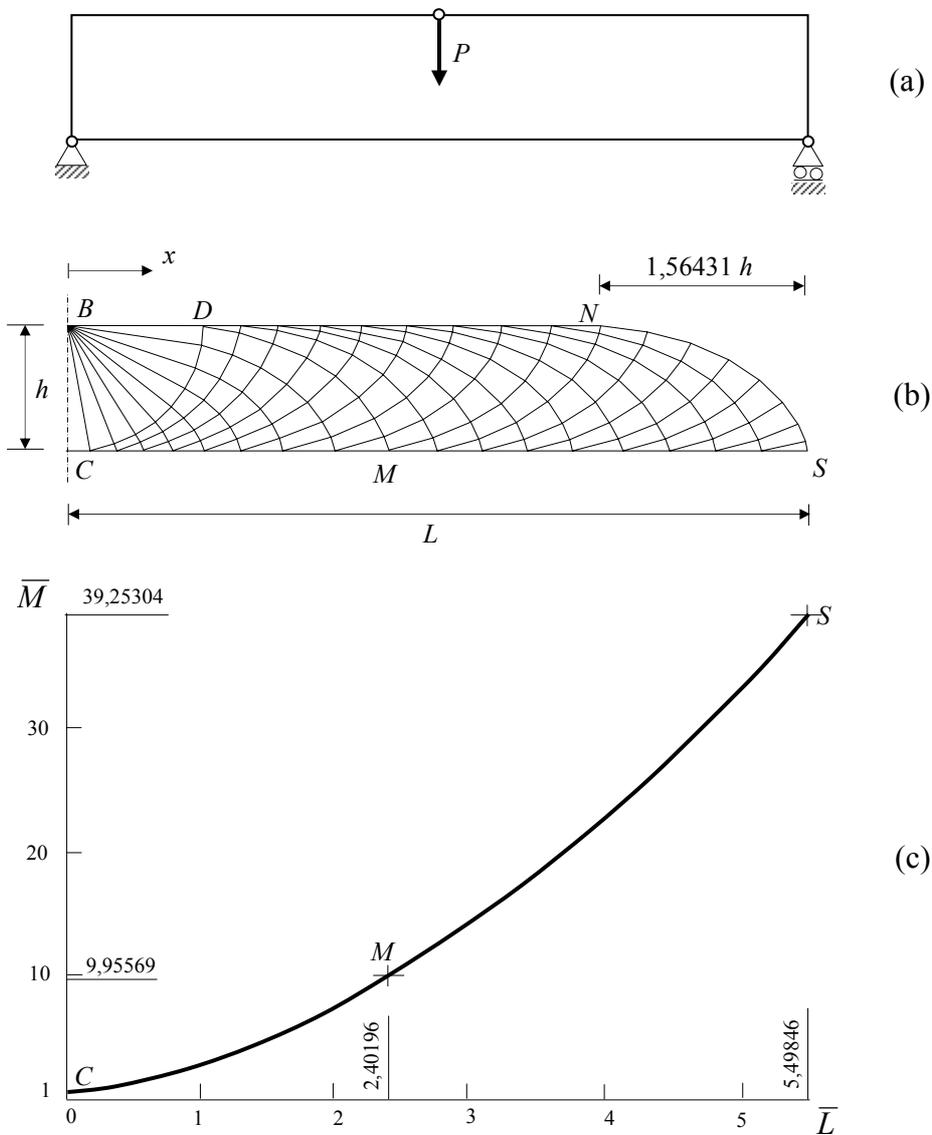


Figura 7.12 (a) Viga MBB. (b) Solución analítica. (c) Valores óptimos de la masa (Rozvany, 1998)

Los puntos M y S (Fig. 7.12c) se corresponden con longitudes adimensionales a partir de las cuales la topología cambia. Se va a resolver este ejemplo para estas dos longitudes adimensionales.

Los datos del problema son: $h = 1 \text{ m}$; $\rho = 7850 \text{ kg/m}^3$; $\sigma_e = 260 \text{ MPa}$, y $P = \sigma_e/h\rho$ (para que no sea necesario adimensionalizar la masa).

En los universos estructurales la primera cifra se refiere al número de divisiones en dirección horizontal y la segunda al número de divisiones en dirección vertical. El número de divisiones horizontal se ha elegido de forma que las celdas sean lo más cuadradas posible.

Se comparan los óptimos obtenidos con el método de la *compliance*, sin optimización de geometría y con optimización de geometría, con los obtenidos con el método propuesto.

7.2.3.1 Longitud adimensional $\bar{L} = 2,40196$

La solución analítica está formada por infinitos nudos e infinitas barras, de ahí que no pueda alcanzarse el óptimo.

En la Fig. 7.13 se muestran los óptimos obtenidos mediante el método de la *compliance* para diferentes universos iniciales, y en la Fig. 7.14 los obtenidos con el método de crecimiento propuesto para diferentes números de nudos añadidos.

En la primera columna de la Fig. 7.13 aparecen los universos estructurales iniciales, en la segunda las topologías óptimas obtenidas mediante el método de la *compliance*, y en la tercera los diseños óptimos obtenidos al optimizar la geometría de las topologías óptimas anteriores.

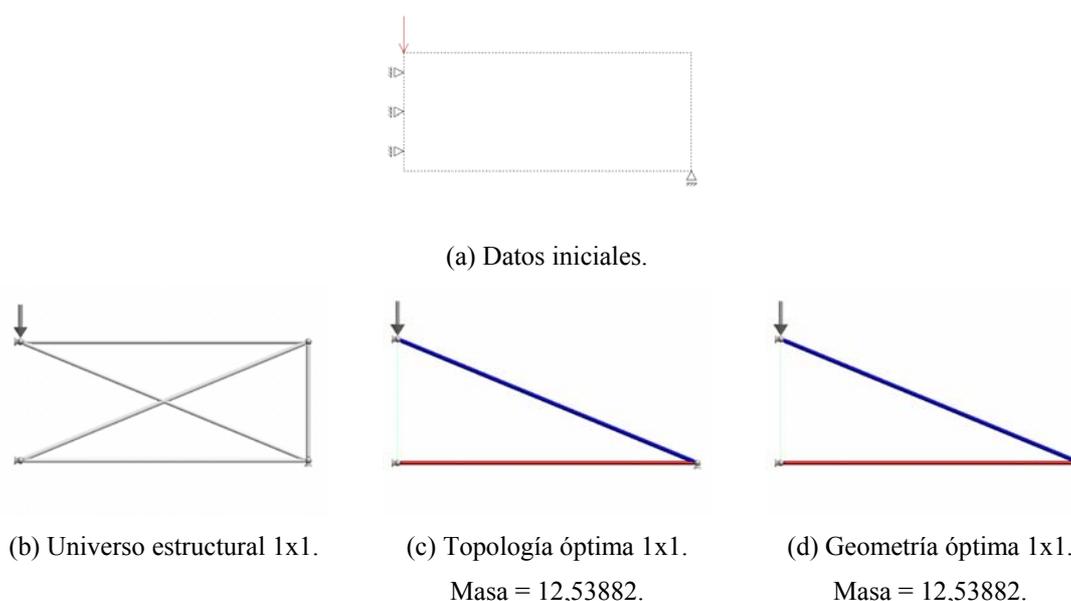


Figura 7.13 Viga MBB ($\bar{L} = 2,40196$). Diseños óptimos obtenidos mediante el método de la *compliance*

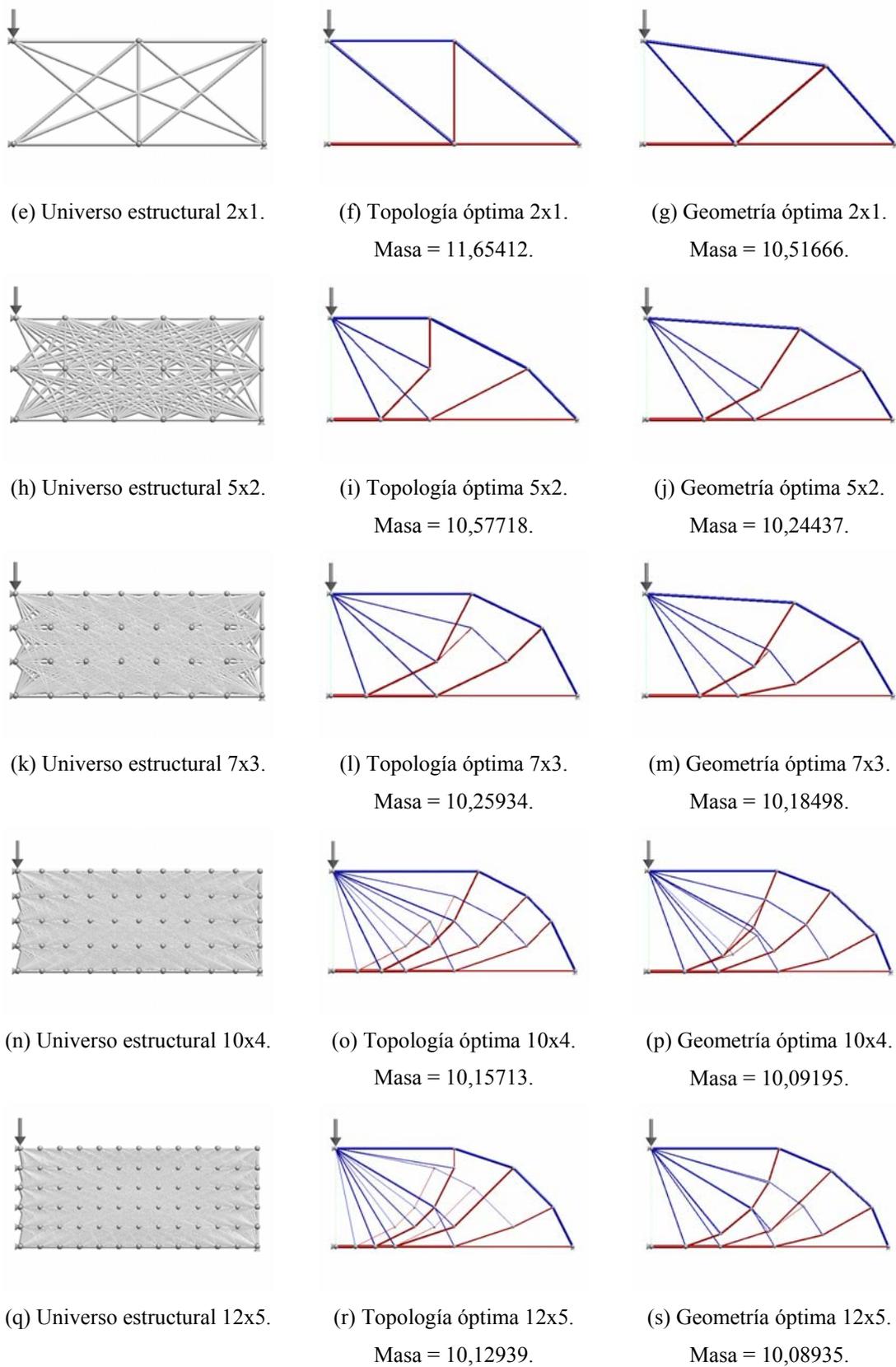


Figura 7.13 Viga MBB ($\bar{L} = 2,40196$). Diseños óptimos obtenidos mediante el método de la *compliance*. (Continuación)

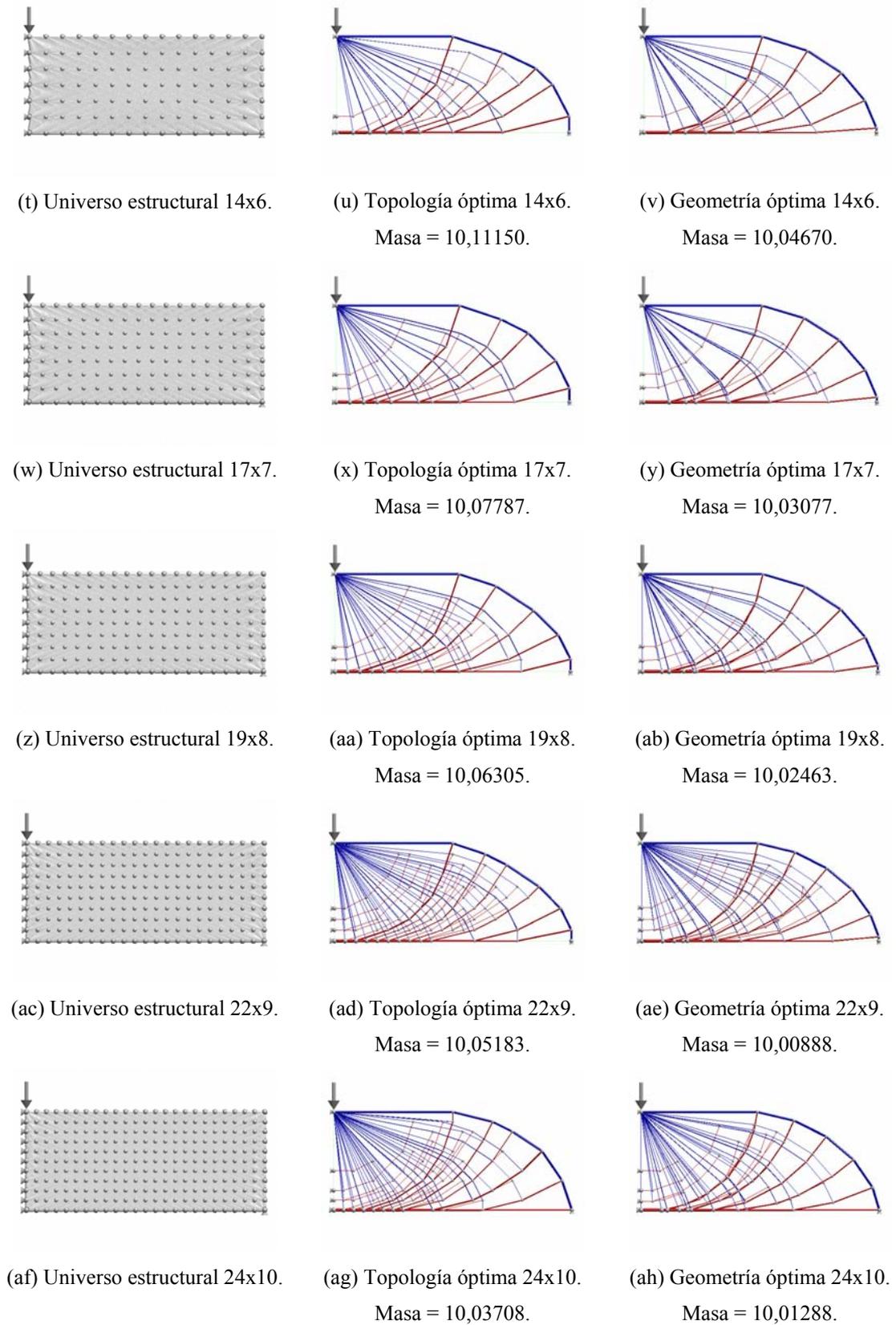


Figura 7.13 Viga MBB ($\bar{L} = 2,40196$). Diseños óptimos obtenidos mediante el método de la *compliance*. (Continuación)

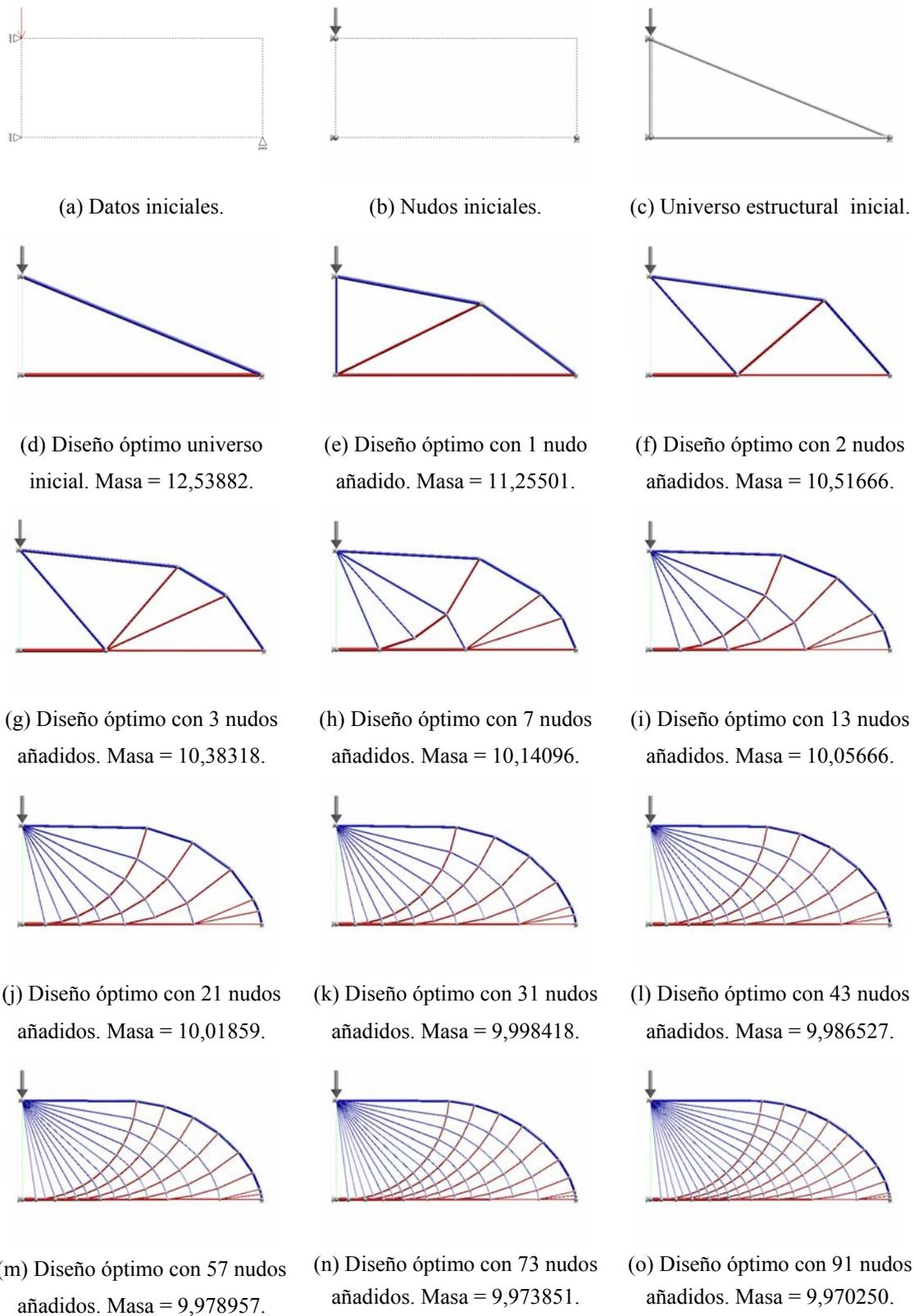


Figura 7.14 Viga MBB ($\bar{L} = 2,40196$). Diseños óptimos obtenidos mediante el método de crecimiento propuesto

En la tabla 7.9 se muestran los resultados obtenidos mediante el método de la *compliance*. En la tabla 7.10 se muestran los resultados conseguidos con el método de crecimiento propuesto.

Tabla 7.9 Viga MBB ($\bar{L} = 2,40196$).

Diseños óptimos obtenidos mediante el método de la *compliance*

div	nudos		barras		masa (adim)		error (%)		tiempo (s)	
	ini	fin	ini	fin	top	geo	top	geo	top	geo
1x1	4	3	6	3	12,53882	12,53882	25,946	25,946	0,005	0,005
2x1	6	5	13	7	11,65412	10,51666	17,060	5,635	0,015	0,02
5x2	18	8	105	13	10,57718	10,24437	6,243	2,900	0,02	0,03
7x3	32	10	326	17	10,25934	10,18498	3,050	2,303	0,09	0,13
10x4	55	18	934	33	10,15713	10,09195	2,023	1,369	0,54	0,71
12x5	78	20	1891	37	10,12939	10,08935	1,745	1,343	3,82	4
14x6	105	40	3380	76	10,11150	10,04670	1,565	0,914	22,5	23,5
17x7	144	44	6370	83	10,07787	10,03077	1,227	0,754	78,1	79
19x8	180	60	9933	115	10,06305	10,02463	1,078	0,692	171	202
22x9	230	93	16191	180	10,05183	10,00888	0,966	0,534	469	476
24x10	275	83	23030	160	10,03708	10,01288	0,818	0,574	1279	1284
Solución analítica					9,95569	9,95569	0,000	0,000		

Tabla 7.10 Viga MBB ($\bar{L} = 2,40196$).

Diseños óptimos obtenidos mediante el método de crecimiento propuesto

añadidos	nudos	barras	masa (adim)	error (%)	tiempo (s)
0	3	3	12,53882	25,946	0,01
1	4	5	11,25501	13,051	0,02
2	5	7	10,51666	5,635	0,03
3	6	9	10,38318	4,294	0,06
7	10	17	10,14096	1,861	0,22
13	16	29	10,05666	1,014	1,05
21	24	45	10,01859	0,632	4,22
31	34	65	9,99842	0,429	14,6
43	46	89	9,98653	0,310	45,2
57	60	117	9,97896	0,234	125
73	76	149	9,97385	0,182	324
91	94	185	9,97025	0,146	782
Solución analítica			9,95569	0,000	

En las gráficas de la Fig. 7.15 se comparan la evolución del error (%) y del tiempo de cálculo (segundos), en función del número de nudos de la topología óptima obtenida para diferentes universos estructurales (método de la *compliance*), y para diferente número de nudos añadidos (método de crecimiento propuesto). Debido a que los errores

para las estructuras obtenidas con menos de 6 nudos son muy altos, no se han representado en estas gráficas.

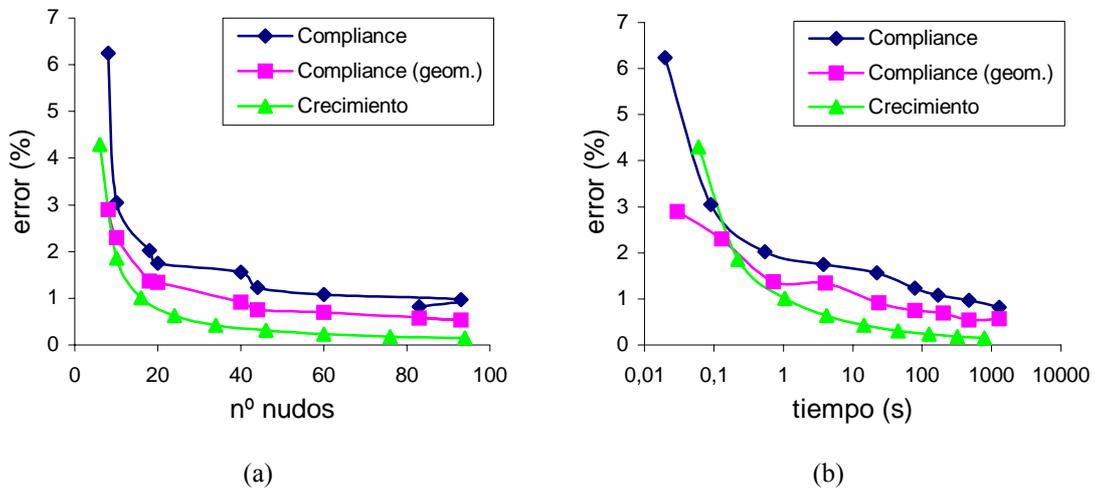


Figura 7.15 Viga MBB ($\bar{L} = 2,40196$). Método de la *compliance* (sin y con optimización de geometría) y de crecimiento propuesto.
(a) Evolución error/nº nudos; (b) Evolución error/tiempo

Con el método propuesto se obtienen mejores diseños (con menos nudos y sin cruces de barras) y con un tiempo de cálculo menor que con el método de la *compliance* (salvo para diseños con pocos nudos, donde los tiempos son poco significativos).

7.2.3.2 Longitud adimensional $\bar{L} = 5,49846$

En la Fig. 7.16 se muestran los óptimos obtenidos mediante el método de la *compliance* para diferentes universos iniciales, y en la Fig. 7.17 los obtenidos con el método de crecimiento propuesto para diferentes números de nudos añadidos.

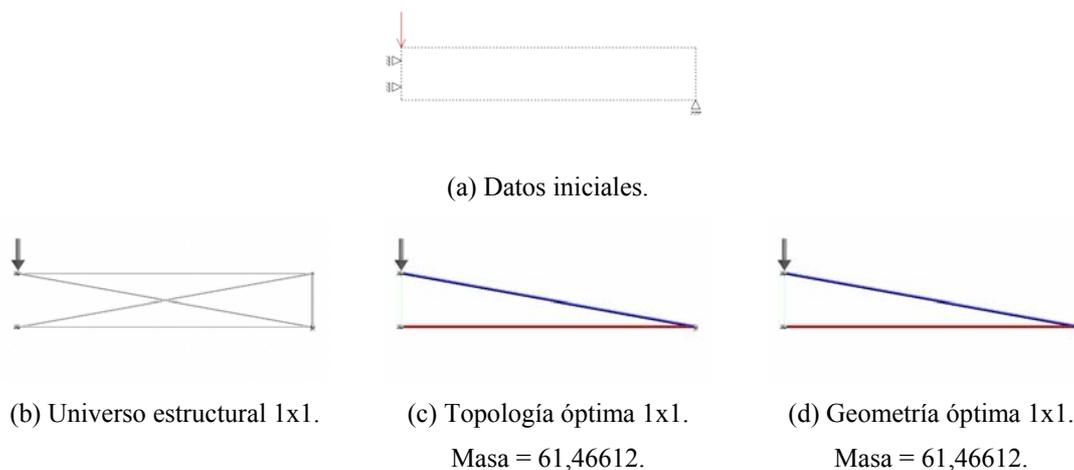


Figura 7.16 Viga MBB ($\bar{L} = 5,49846$). Diseños óptimos obtenidos mediante el método de la *compliance*

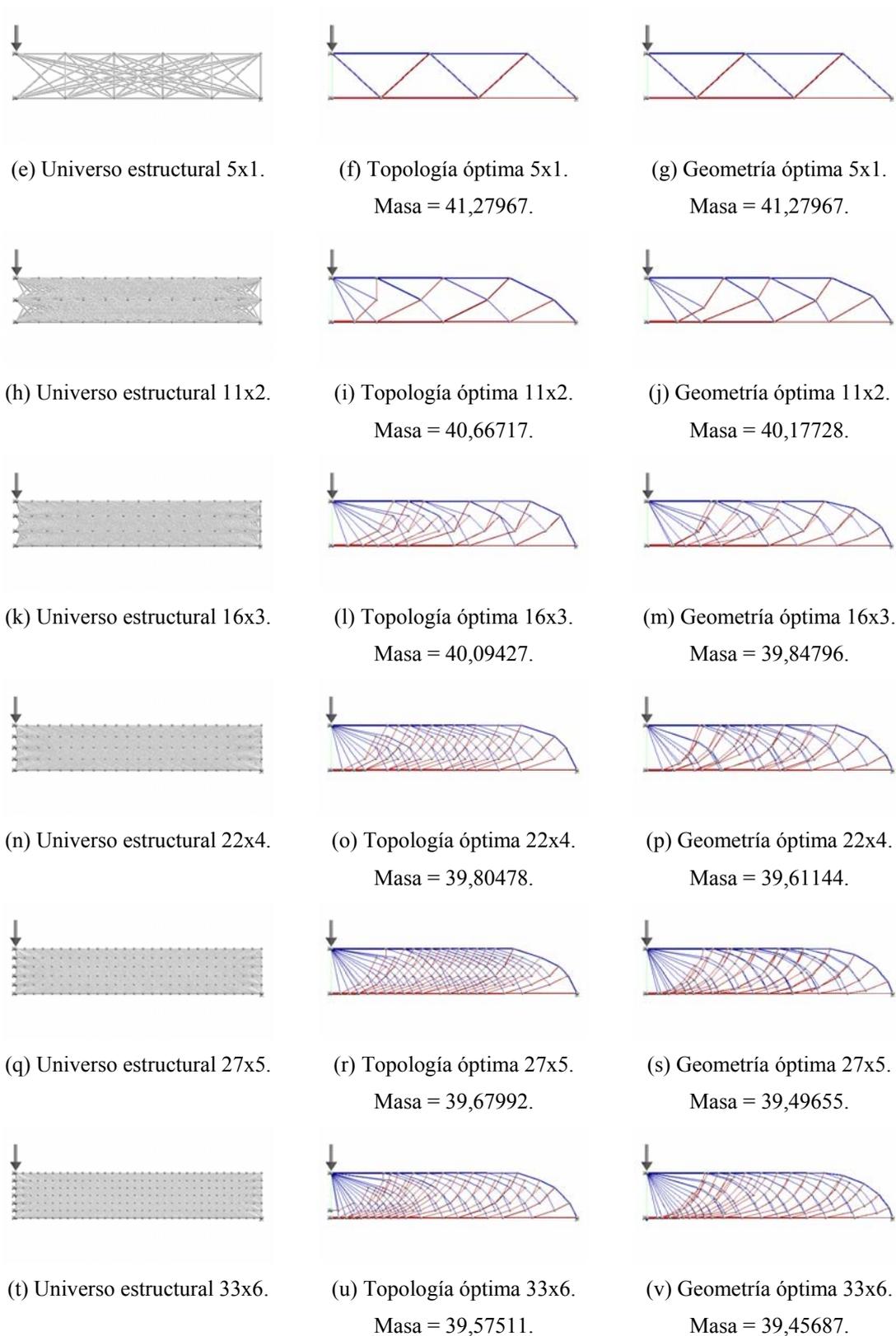


Figura 7.16 Viga MBB ($\bar{L} = 5,49846$). Diseños óptimos obtenidos mediante el método de la *compliance*. (Continuación)

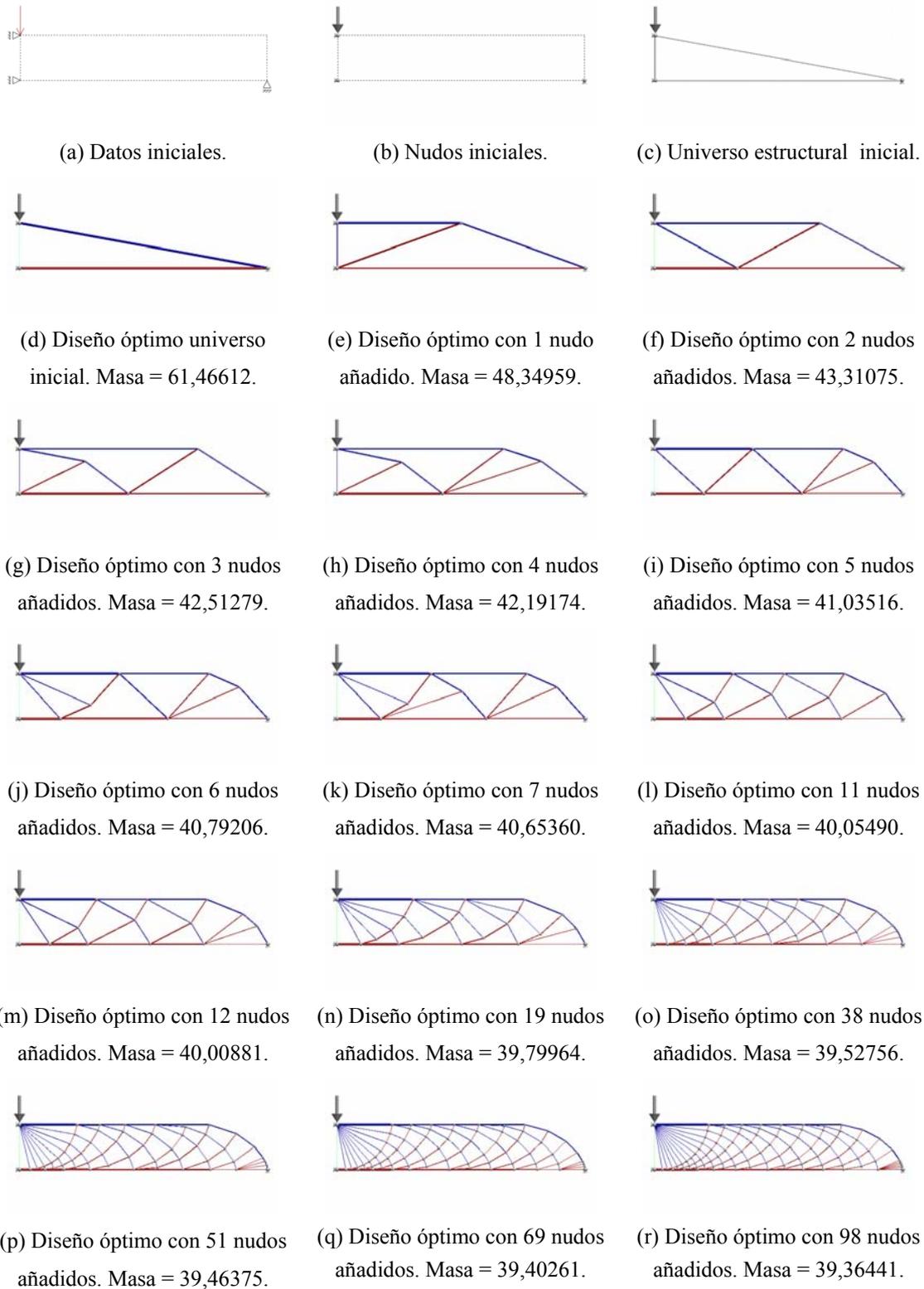


Figura 7.17 Viga MBB ($\bar{L} = 5,49846$). Diseños óptimos obtenidos mediante el método de crecimiento propuesto

En la primera columna de la Fig. 7.16 aparecen los universos estructurales iniciales, en la segunda las topologías óptimas obtenidas mediante el método de la *compliance*, y en la tercera los diseños óptimos obtenidos al optimizar la geometría de las topologías óptimas anteriores.

En la tabla 7.11 se muestran los resultados obtenidos mediante el método de la *compliance*. En la tabla 7.12 se muestran los resultados conseguidos con el método de crecimiento propuesto.

Tabla 7.11 Viga MBB ($\bar{L} = 5,49846$).

Diseños óptimos obtenidos mediante el método de la *compliance*

div	nudos		barras		masa (adim)		error (%)		tiempo (s)	
	ini	fin	ini	fin	top	geo	top	geo	top	geo
1x1	4	3	5	3	61,46612	61,46612	56,589	56,589	0,01	0,01
5x1	12	7	45	11	41,27967	41,27967	5,163	5,163	0,02	0,02
11x2	36	14	391	25	40,66717	40,17728	3,603	2,355	0,14	0,21
16x3	68	35	1408	67	40,09427	39,84796	2,143	1,516	1,79	3,07
22x4	115	73	3982	143	39,80478	39,61144	1,406	0,913	27,5	34,4
27x5	168	111	8621	219	39,67992	39,49655	1,088	0,620	171	193
33x6	238	114	17099	224	39,57511	39,45687	0,820	0,519	720	781
Solución analítica					39,25304	39,25304	0,000	0,000		

Tabla 7.12 Viga MBB ($\bar{L} = 5,49846$).

Diseños óptimos obtenidos mediante el método de crecimiento propuesto

añadidos	nudos	barras	masa (adim)	error (%)	tiempo (s)
0	3	3	61,46612	56,589	0,005
1	4	5	48,34959	23,174	0,01
2	5	7	43,31075	10,337	0,012
3	6	9	42,51279	8,304	0,03
4	7	11	42,19174	7,487	0,05
5	8	13	41,03516	4,540	0,06
6	9	15	40,79206	3,921	0,1
7	10	17	40,65360	3,568	0,13
11	14	25	40,05490	2,043	0,51
12	15	27	40,00881	1,925	0,64
19	22	41	39,79964	1,393	2,62
38	41	79	39,52756	0,699	22,9
51	54	105	39,46375	0,537	73,8
69	72	141	39,40261	0,381	267
98	101	199	39,36441	0,284	931
Solución analítica			39,25304	0,000	

En las gráficas de la Fig. 7.18 se comparan la evolución del error (%) y del tiempo de cálculo (segundos), en función del número de nudos de la topología óptima obtenida para diferentes universos estructurales (método de la *compliance*), y para diferente número de nudos añadidos (método de crecimiento propuesto). Debido a que los errores para las estructuras obtenidas con menos de 10 nudos son muy altos, no se han representado en estas gráficas.

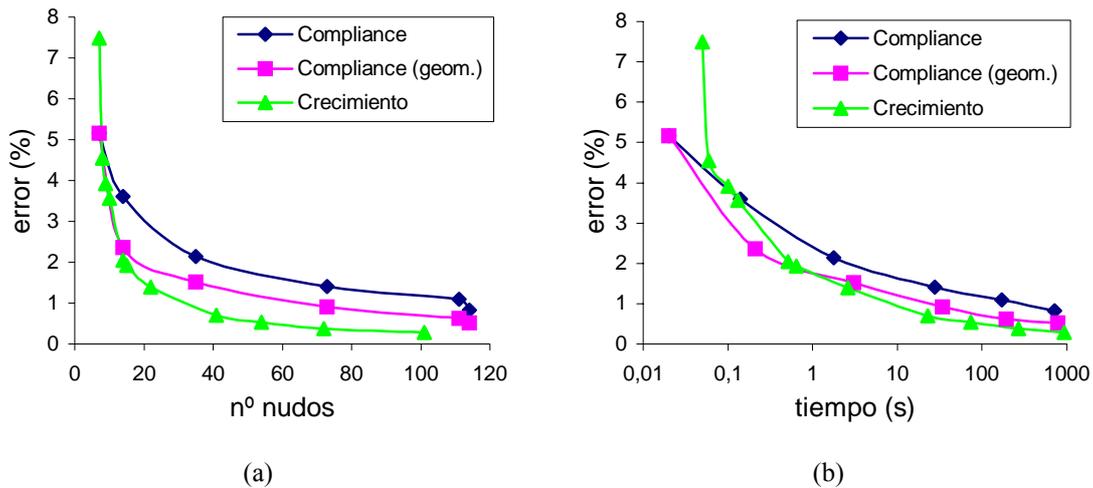


Figura 7.18 Viga MBB ($\bar{L} = 5,49846$). Método de la *compliance* (sin y con optimización de geometría) y de crecimiento propuesto.
(a) Evolución error/nº nudos; (b) Evolución error/tiempo

Las conclusiones que se obtienen de las tablas 7.11 y 7.12 y de la Fig. 7.18 son similares a las obtenidas en el ejemplo del voladizo de Michell, es decir, con el método propuesto se obtienen mejores diseños (con menos nudos y sin cruces de barras) y con un tiempo de cálculo menor que con el método de la *compliance* (salvo para diseños con pocos nudos, donde los tiempos son poco significativos).

Los diseños óptimos obtenidos con el método de la *compliance* tienen unas topologías complicadas. Como consecuencia de esto, el algoritmo de optimización de geometría tiene muchas dificultades para obtener el óptimo (de hecho, en muchas ocasiones el algoritmo no encuentra una dirección de búsqueda satisfactoria).

En la Fig. 7.18 se aprecia que, si al diseño óptimo obtenido mediante el método de la *compliance*, no se le efectúa una optimización final de la geometría, las diferencias, con respecto a los diseños obtenidos con el método de crecimiento propuesto, son significativamente mayores.

7.2.4 Viga con disco circular rígido

La viga con disco circular rígido es la equivalente, aplicando simetría, a una viga biempotrada con un disco central al que se le aplica un momento torsor (Fig. 7.19a). El dominio de definición de la viga es un rectángulo de longitud $2L$ y de altura no inferior a $L \frac{\sqrt{2}}{e^{\pi/4}}$ (altura de la solución analítica cuando no hay restricciones de borde activas).

La solución analítica se muestra en la Fig. 7.19b. El valor óptimo de la masa adimensional es $\bar{M} = M \frac{\sigma_e}{M_t \rho} = \ln\left(\frac{L}{r}\right)$, siendo M la masa, σ_e el límite elástico, M_t el momento torsor, ρ la densidad, L la longitud y r el radio del disco.

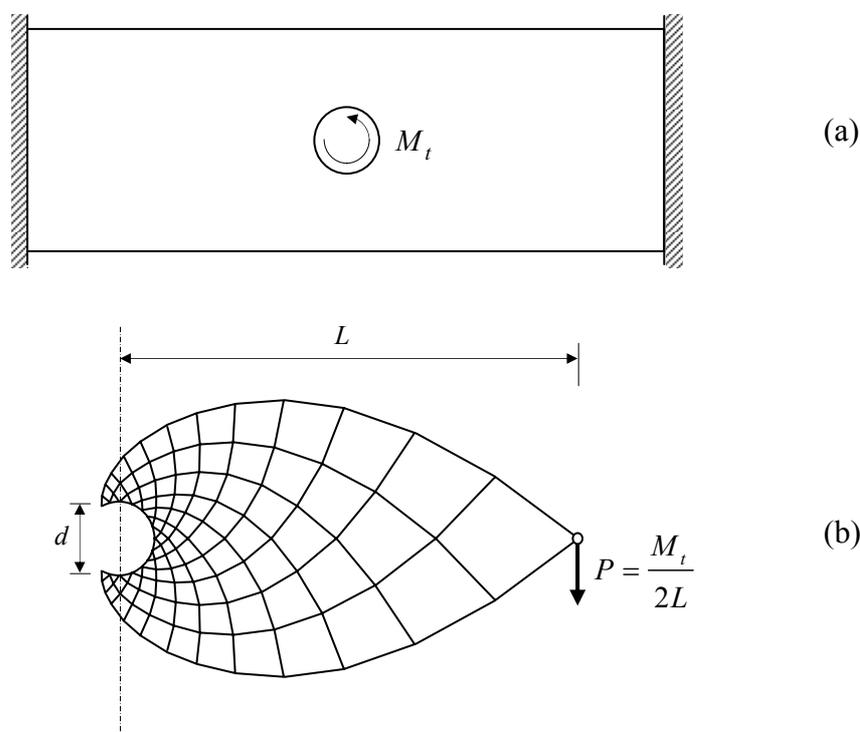


Figura 7.19 (a) Viga con disco circular rígido. (b) Solución analítica (Rozvany, 1998)

Los datos del problema son: $L = 1,5 \text{ m}$; $r = 0,15 \text{ m}$; $\rho = 7850 \text{ kg/m}^3$; $\sigma_e = 260 \text{ MPa}$, y $M_t = \sigma_e / \rho$ (para que no sea necesario adimensionalizar la masa).

Cuando, como en este caso, el dominio de definición de la estructura tiene una forma compleja (un rectángulo con un agujero circular) hay que dividir dicho dominio en varias superficies cuadrangulares para poder mapearlas y crear el universo estructural, lo que supone una tarea laboriosa. En la Fig. 7.20 se muestran las 10 superficies en que se ha dividido el dominio. En los universos estructurales la primera cifra se refiere al número de divisiones en dirección horizontal de las superficies rectangulares y la segunda al número de divisiones en las superficies y direcciones restantes.

Para el método de crecimiento propuesto el disco se ha discretizado en 22 nudos.

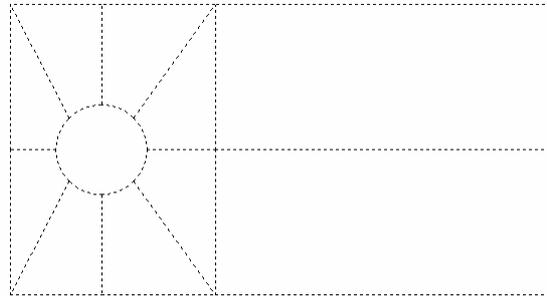


Figura 7.20 Viga con disco circular rígido. Superficies a mapear para crear el universo estructural

Se comparan los óptimos obtenidos con el método de la *compliance*, sin optimización de geometría y con optimización de geometría, con los obtenidos con el método propuesto.

La solución analítica está formada por infinitos nudos e infinitas barras, de ahí que no pueda alcanzarse el óptimo.

En la Fig. 7.21 se muestran los óptimos obtenidos mediante el método de la *compliance* para diferentes universos iniciales, y en la Fig. 7.22 los obtenidos con el método de crecimiento propuesto para diferentes números de nudos añadidos.

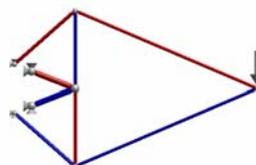
En la primera columna de la Fig. 7.21 aparecen los universos estructurales iniciales, en la segunda las topologías óptimas obtenidas mediante el método de la *compliance*, y en la tercera los diseños óptimos obtenidos al optimizar la geometría de las topologías óptimas anteriores.



(a) Datos iniciales.

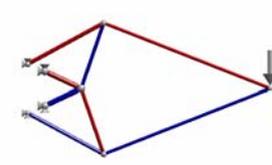


(b) Universo estructural 1x1.



(c) Topología óptima 1x1.

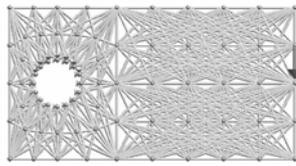
Masa = 2,85548.



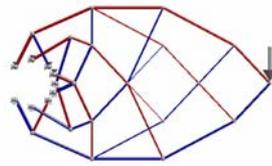
(d) Geometría óptima 1x1.

Masa = 2,73706.

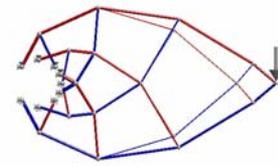
Figura 7.21 Viga con disco circular rígido. Diseños óptimos obtenidos mediante el método de la *compliance*



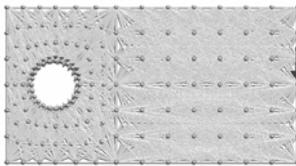
(e) Universo estructural 2x5.



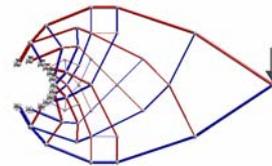
(f) Topología óptima 2x5.
Masa = 2,52015.



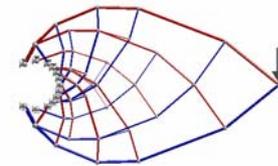
(g) Geometría óptima 2x5.
Masa = 2,42889.



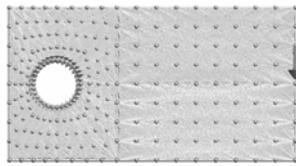
(h) Universo estructural 3x7.



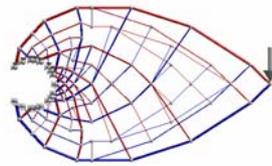
(i) Topología óptima 3x7.
Masa = 2,44051.



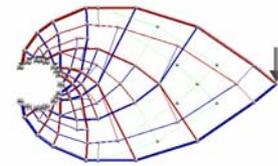
(j) Geometría óptima 3x7.
Masa = 2,38550.



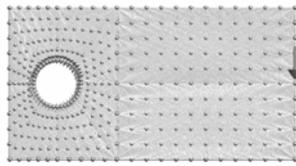
(k) Universo estructural 4x9.



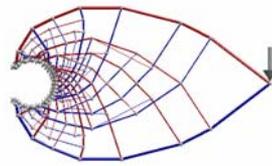
(l) Topología óptima 4x9.
Masa = 2,39270.



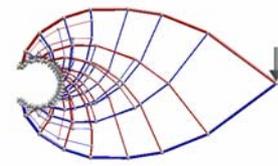
(m) Geometría óptima 4x9.
Masa = 2,37019.



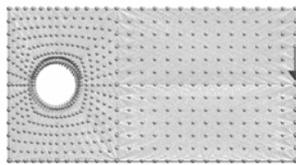
(n) Universo estructural 5x12.



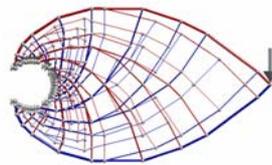
(o) Topología óptima 5x12.
Masa = 2,37371.



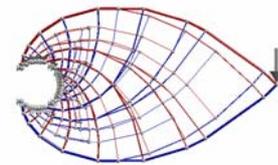
(p) Geometría óptima 5x12.
Masa = 2,34600.



(q) Universo estructural 6x14.



(r) Topología óptima 6x14.
Masa = 2,35547.



(s) Geometría óptima 6x14.
Masa = 2,33848.

Figura 7.21 Viga con disco circular rígido. Diseños óptimos obtenidos mediante el método de la *compliance*. (Continuación)

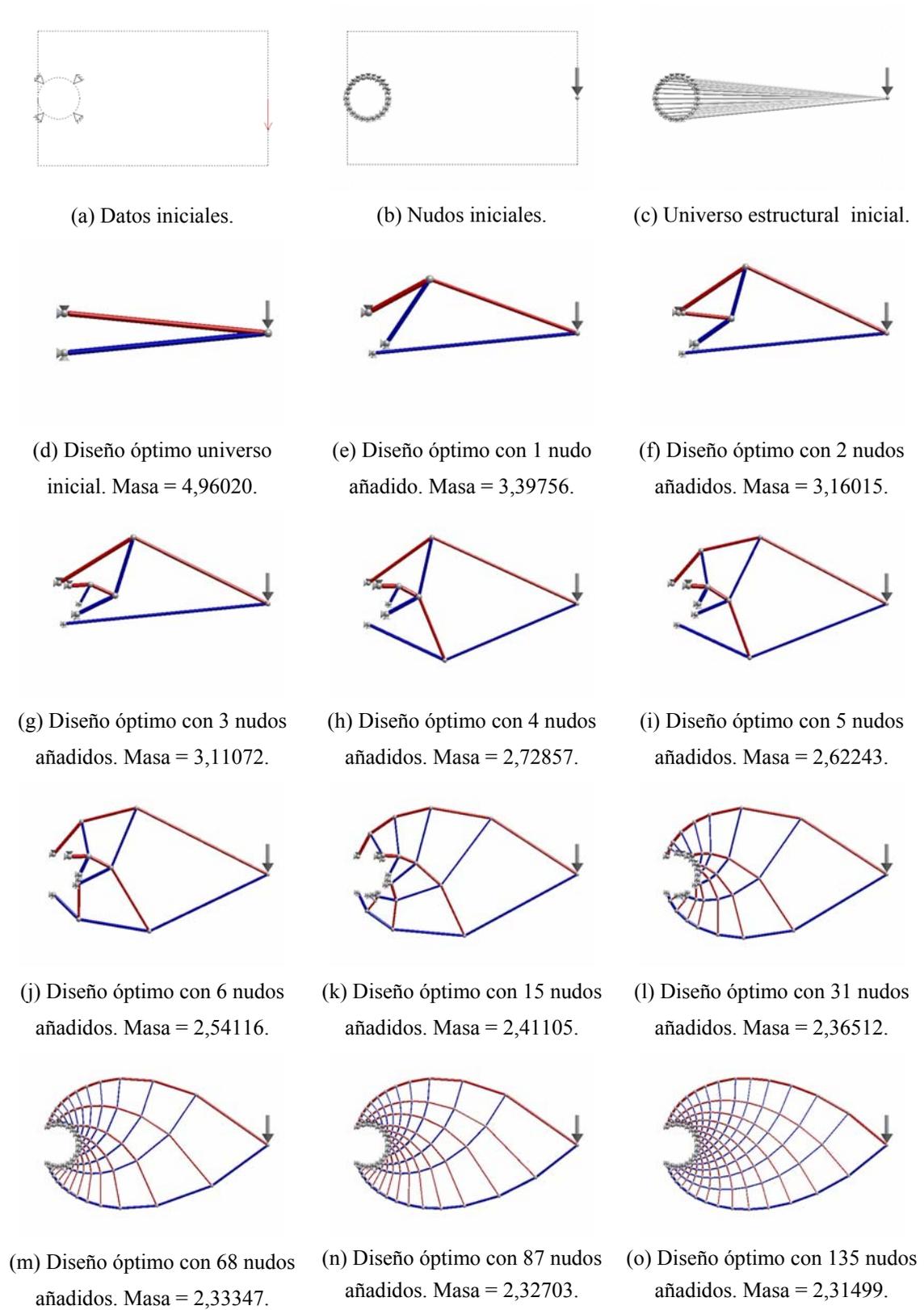


Figura 7.22 Viga con disco circular rígido. Diseños óptimos obtenidos mediante el método de crecimiento propuesto

En la tabla 7.13 se muestran los resultados obtenidos mediante el método de la *compliance*. En la tabla 7.14 se muestran los resultados conseguidos con el método de crecimiento propuesto.

Tabla 7.13 Viga con disco circular rígido.
Diseños óptimos obtenidos mediante el método de la *compliance*

div	nudos		barras		masa (adim)		error (%)		tiempo (s)	
	ini	fin	ini	fin	top	geo	top	geo	top	geo
1x1	19	8	41	8	2,85548	2,73706	24,012	18,869	0,015	0,02
2x5	73	27	393	36	2,52015	2,42889	9,449	5,485	0,19	0,35
3x7	145	49	1279	72	2,44051	2,38550	5,990	3,601	3,33	4,39
4x9	241	85	3073	136	2,39270	2,37019	3,914	2,936	35,3	38,5
5x12	372	105	7024	164	2,37371	2,34600	3,089	1,885	285	301
6x14	518	160	12622	266	2,35547	2,33848	2,297	1,559	1379	1441
Solución analítica					2,30259	2,30259	0,000	0,000		

Tabla 7.14 Viga con disco circular rígido.
Diseños óptimos obtenidos mediante el método de crecimiento propuesto

añadidos	nudos	barras	masa (adim)	error (%)	tiempo (s)
0	3	2	4,96020	115,419	0,005
1	5	4	3,39756	47,554	0,04
2	7	6	3,16015	37,243	0,08
3	9	8	3,11072	35,097	0,12
4	10	10	2,72857	18,500	0,19
5	11	12	2,62243	13,891	0,28
6	13	14	2,54116	10,361	0,41
15	26	32	2,41105	4,710	2,04
31	46	64	2,36512	2,716	15
68	86	136	2,33347	1,341	193
87	105	176	2,32703	1,062	430
135	153	272	2,31499	0,539	2356
Solución analítica			2,30259	0,000	

En las gráficas de la Fig. 7.23 se comparan la evolución del error (%) y del tiempo de cálculo (segundos), en función del número de nudos de la topología óptima obtenida para diferentes universos estructurales (método de la *compliance*), y para diferente número de nudos añadidos (método de crecimiento propuesto). Debido a que los errores para las estructuras obtenidas con menos de 8 nudos son muy altos, no se han representado en estas gráficas.

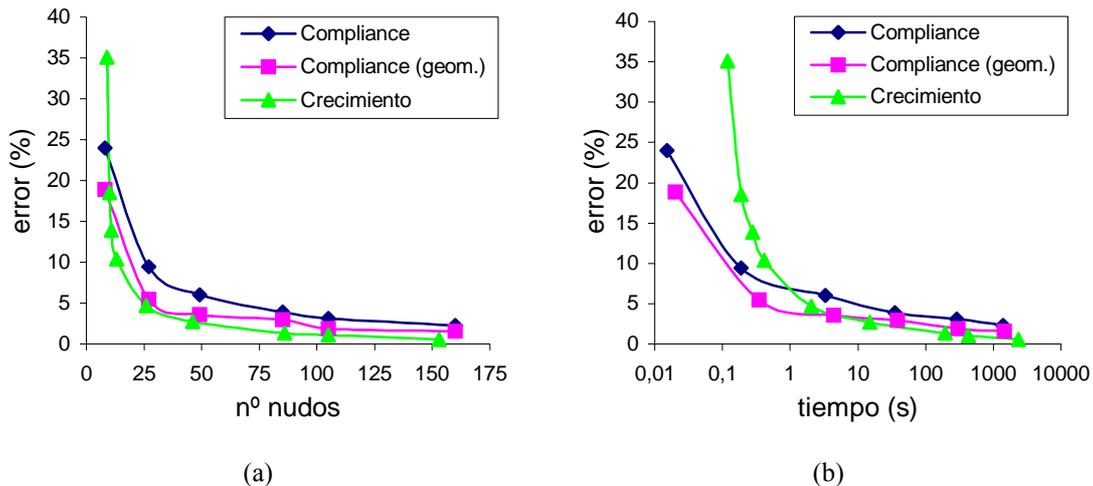


Figura 7.23 Viga con disco circular rígido. Método de la *compliance* (sin y con optimización de geometría) y de crecimiento propuesto.
(a) Evolución error/nº nudos; (b) Evolución error/tiempo

Las conclusiones que se obtienen de las tablas 7.13 y 7.14 y de la Fig. 7.23 son similares a las obtenidas en el ejemplo del voladizo de Michell, es decir, con el método propuesto se obtienen mejores diseños (con menos nudos y sin cruces de barras) y con un tiempo de cálculo menor que con el método de la *compliance* (salvo para diseños con pocos nudos, donde los tiempos son poco significativos).

Debido a la complejidad de las topologías de los diseños óptimos obtenidos mediante el método de la *compliance*, el algoritmo de optimización de geometría falla con la mayoría de los universos estructurales (a partir del 4x9). En estos casos, se representa el diseño obtenido hasta el momento en el que el algoritmo declara “dirección de búsqueda insatisfactoria”.

7.3 APLICACIONES

7.3.1 Introducción

La optimización de topología y geometría de estructuras articuladas también es aplicable al método de bielas y tirantes para el hormigón armado, y a la optimización de topología y geometría de mecanismos.

7.3.2 Estructuras de barras articuladas

7.3.2.1 Introducción

Se va a resolver el ejemplo del voladizo de Michell con carga variable (en ángulo).

7.3.2.2 Voladizo de Michell con carga variable

Para ver la influencia del ángulo de aplicación de la carga en el diseño óptimo, se va a resolver el voladizo de Michell para diferentes ángulos de aplicación de la carga, diferentes longitudes adimensionales, sin pandeo y con pandeo de elementos (Fig. 7.24).

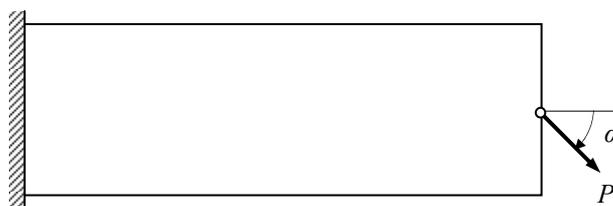


Figura 7.24 Voladizo de Michell con carga variable

Los datos son los mismos que los del voladizo de Michell (apartado 6.2.2).

Los diferentes ángulos probados son: $\alpha = 0^\circ$, $\alpha = 15^\circ$, $\alpha = 30^\circ$, $\alpha = 45^\circ$, $\alpha = 60^\circ$, $\alpha = 75^\circ$, $\alpha = 90^\circ$, y las longitudes adimensionales: $\bar{L} = 0,5$ y $\bar{L} = 1,82196$.

Este ejemplo se va a resolver con el método de crecimiento propuesto.

En las Figs. 7.25 a 7.31 se muestran los diseños óptimos obtenidos para $\bar{L} = 0,5$, sin considerar pandeo de elementos y para diferentes ángulos. En la Fig. 7.32 se muestra la evolución de la masa adimensional y del diseño óptimo en función del ángulo.

En las Figs. 7.33 a 7.39 se muestran los diseños óptimos obtenidos para $\bar{L} = 1,82196$, sin considerar pandeo de elementos y para diferentes ángulos. En la Fig. 7.40 se muestra la evolución de la masa adimensional y del diseño óptimo en función del ángulo.

En las Figs. 7.41 a 7.47 se muestran los diseños óptimos obtenidos para $\bar{L} = 0,5$, considerando pandeo de elementos y para diferentes ángulos. En la Fig. 7.48 se muestra la evolución de la masa adimensional y del diseño óptimo en función del ángulo.

En las Figs. 7.49 a 7.55 se muestran los diseños óptimos obtenidos para $\bar{L} = 1,82196$, considerando pandeo de elementos y para diferentes ángulos. En la Fig. 7.56 se muestra la evolución de la masa adimensional y del diseño óptimo en función del ángulo.

Cuando la línea de aplicación de la carga cruza el empotramiento la solución es una barra sometida a tracción a la que se le añade una barra de área casi nula para evitar la inestabilidad (Figs. 7.25c a 7.28c, 7.33c a 7.34c, 7.41c a 7.44c y 7.49c a 7.50c).

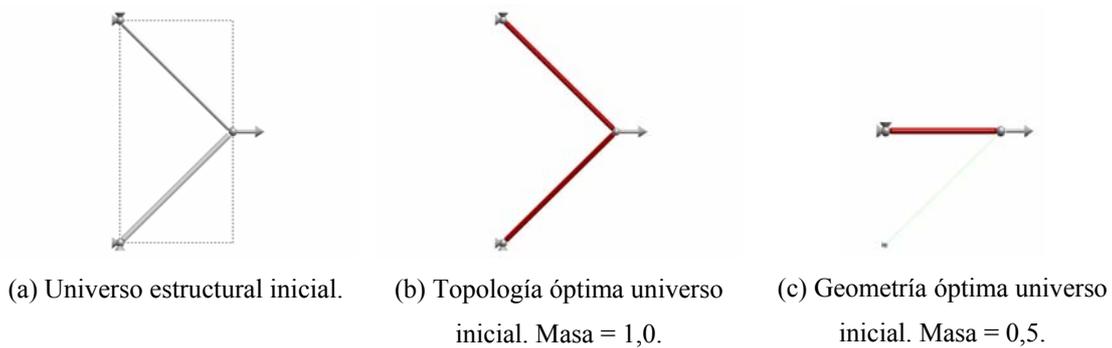


Figura 7.25 Voladizo de Michell ($\bar{L} = 0,5$). Diseño óptimo: $\alpha = 0^\circ$ (sin pandeo)

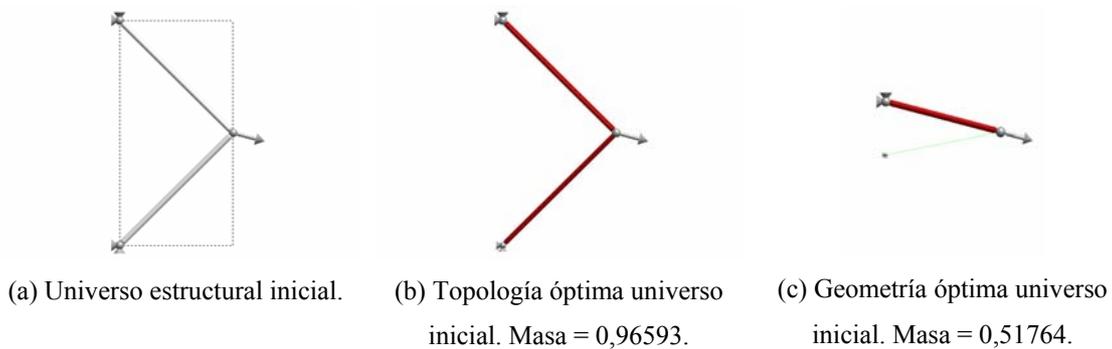


Figura 7.26 Voladizo de Michell ($\bar{L} = 0,5$). Diseño óptimo: $\alpha = 15^\circ$ (sin pandeo)

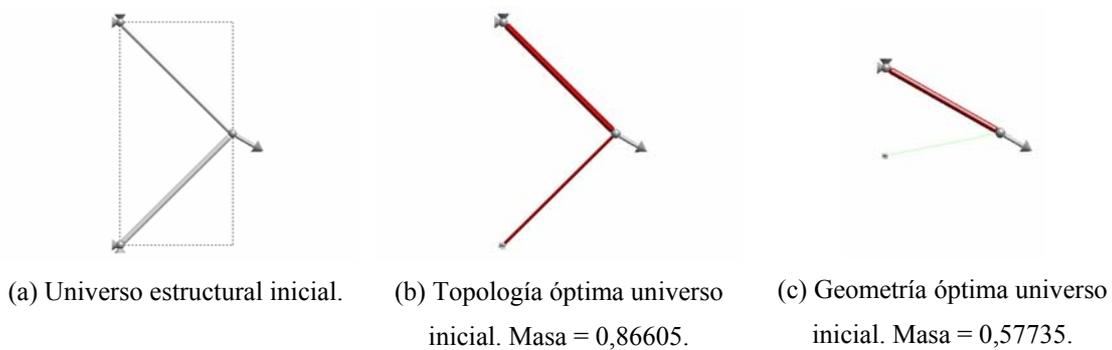


Figura 7.27 Voladizo de Michell ($\bar{L} = 0,5$). Diseño óptimo: $\alpha = 30^\circ$ (sin pandeo)

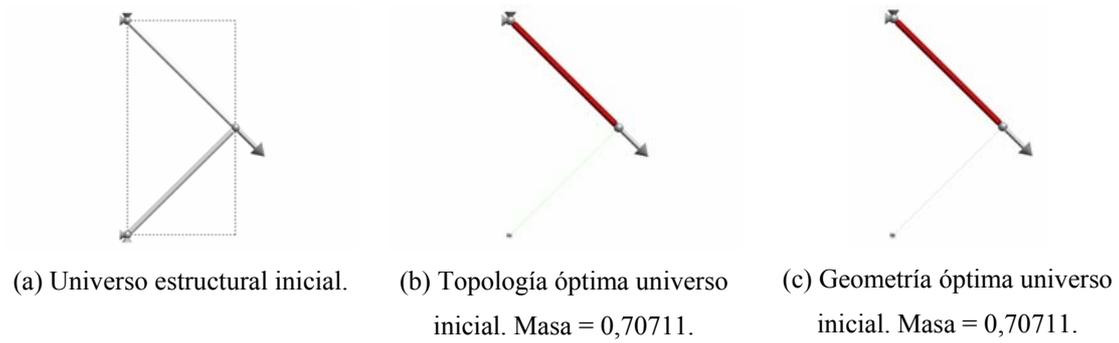


Figura 7.28 Voladizo de Michell ($\bar{L} = 0,5$). Diseño óptimo: $\alpha = 45^\circ$ (sin pandeo)

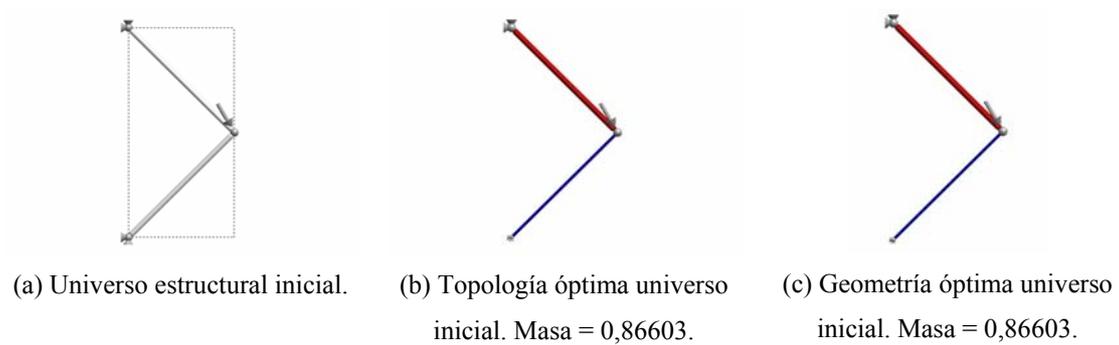


Figura 7.29 Voladizo de Michell ($\bar{L} = 0,5$). Diseño óptimo: $\alpha = 60^\circ$ (sin pandeo)

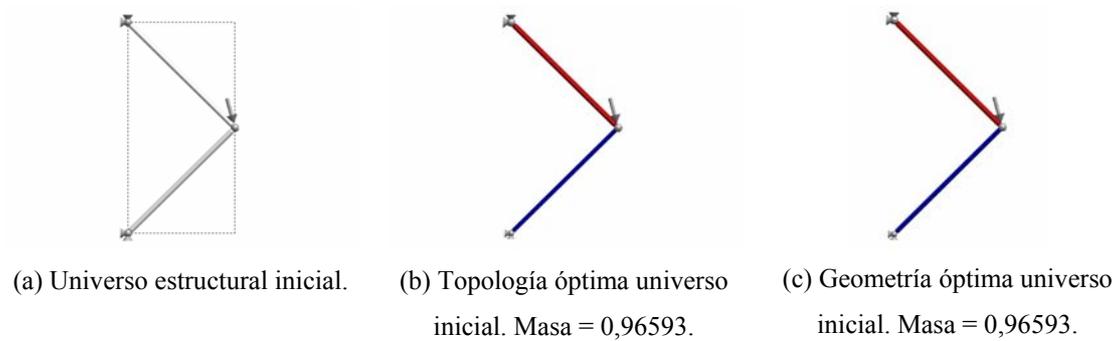


Figura 7.30 Voladizo de Michell ($\bar{L} = 0,5$). Diseño óptimo: $\alpha = 75^\circ$ (sin pandeo)

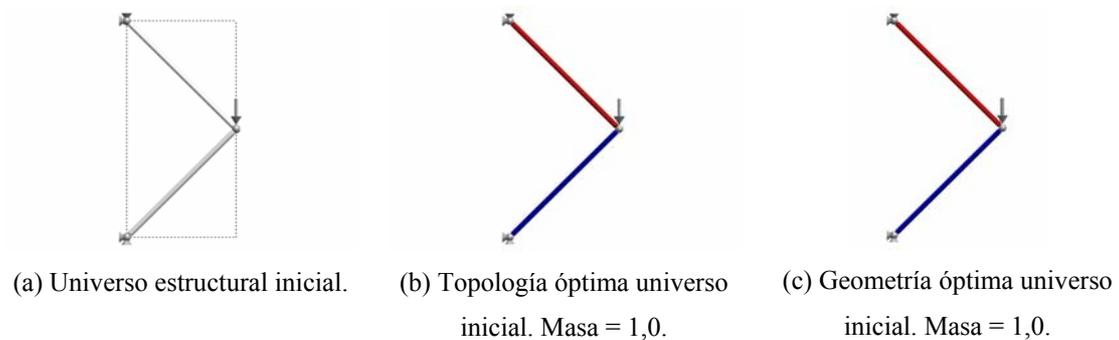


Figura 7.31 Voladizo de Michell ($\bar{L} = 0,5$). Diseño óptimo: $\alpha = 90^\circ$ (sin pandeo)

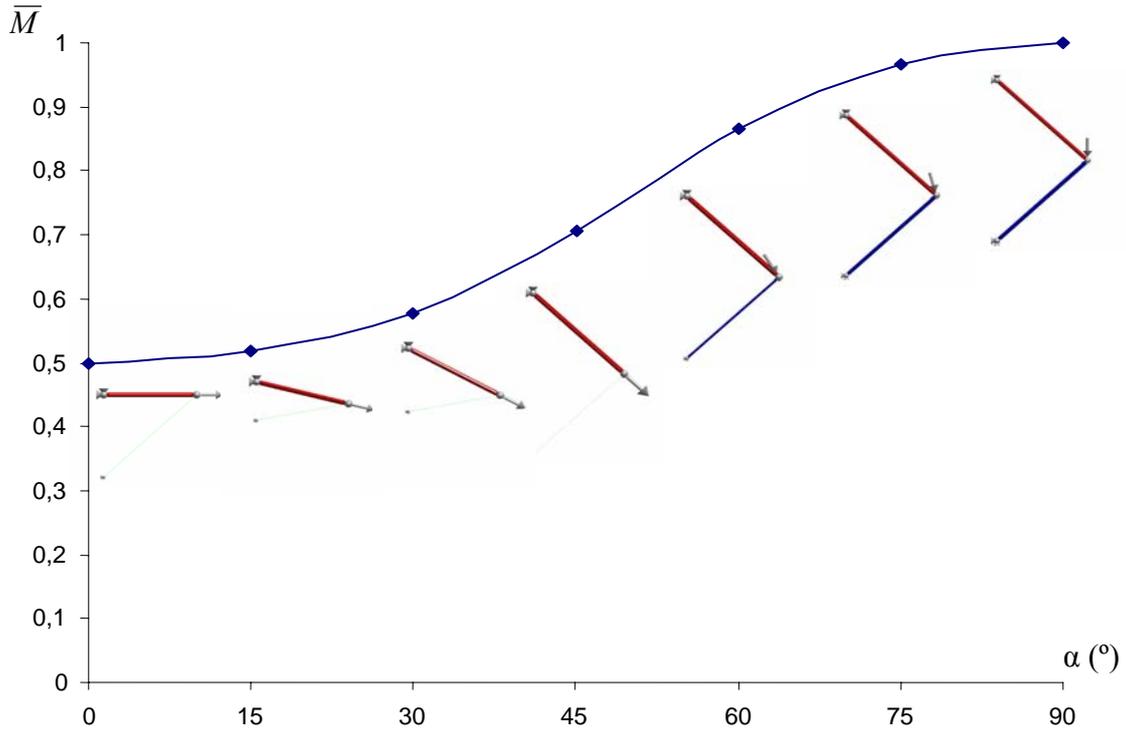


Figura 7.32 Voladizo de Michell ($\bar{L} = 0,5$). Evolución masa adimensional/ángulo (sin pandeo)

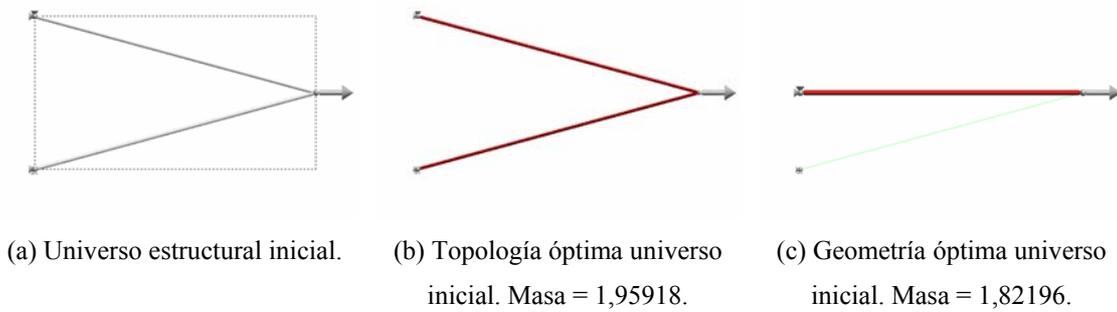


Figura 7.33 Voladizo de Michell ($\bar{L} = 1,82196$). Diseño óptimo: $\alpha = 0^\circ$ (sin pandeo)

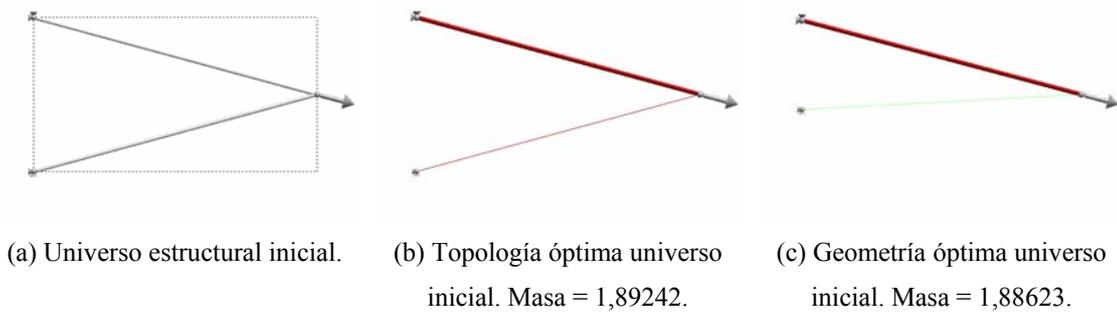


Figura 7.34 Voladizo de Michell ($\bar{L} = 1,82196$). Diseño óptimo: $\alpha = 15^\circ$ (sin pandeo)

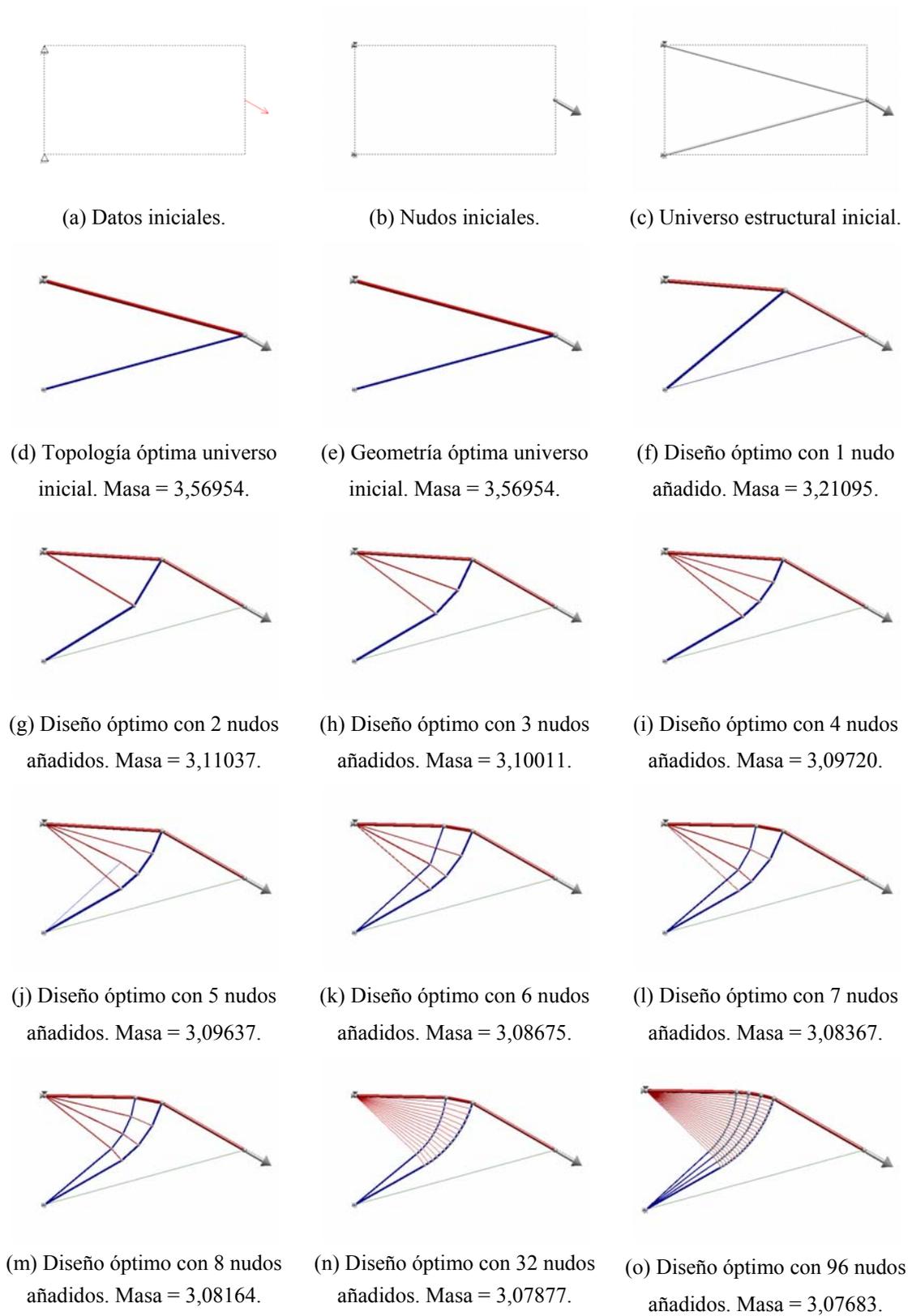


Figura 7.35 Voladizo de Michell ($\bar{L} = 1,82196$). Diseño óptimo: $\alpha = 30^\circ$ (sin pandeo)

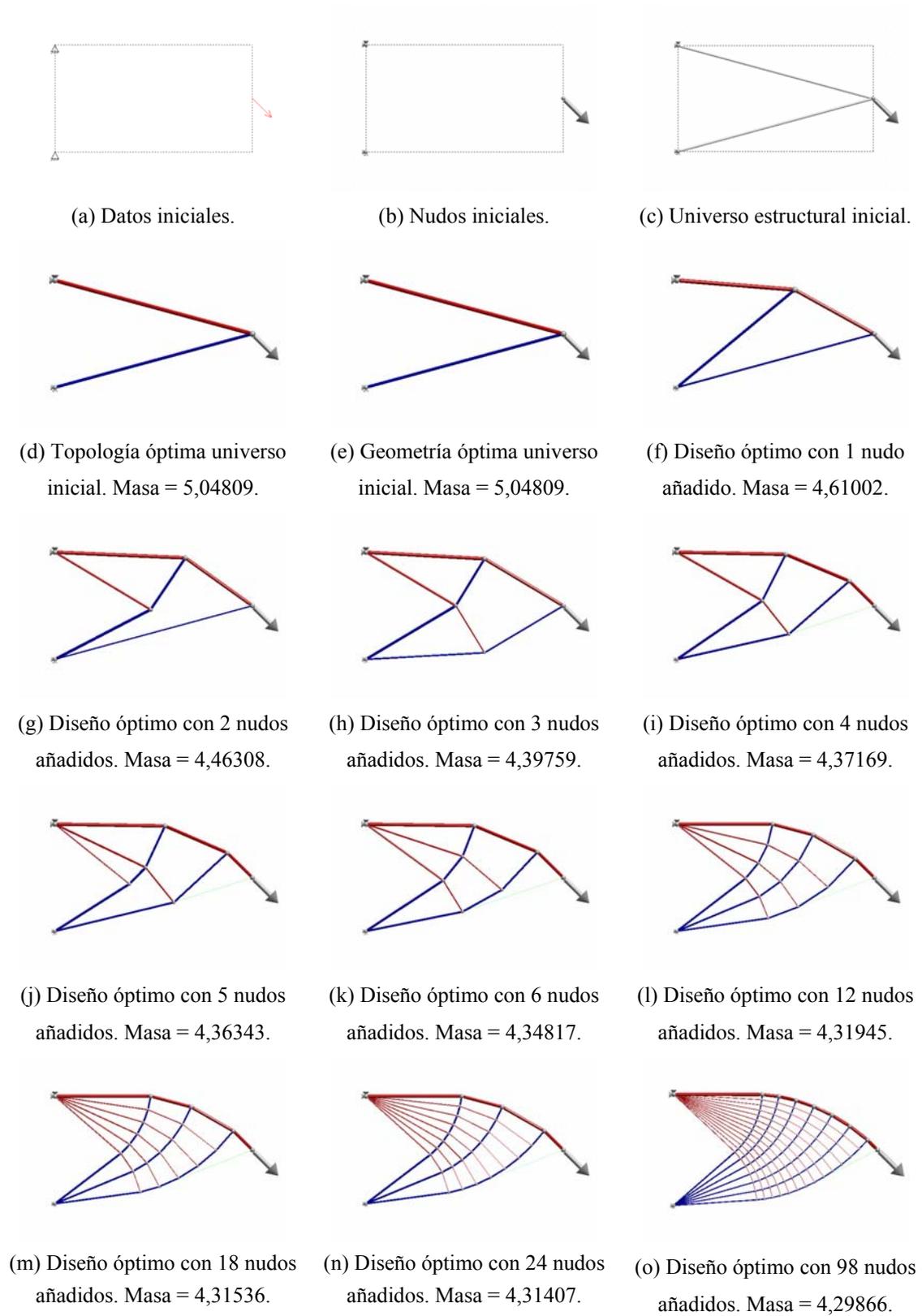


Figura 7.36 Voladizo de Michell ($\bar{L} = 1,82196$). Diseño óptimo: $\alpha = 45^\circ$ (sin pandeo)

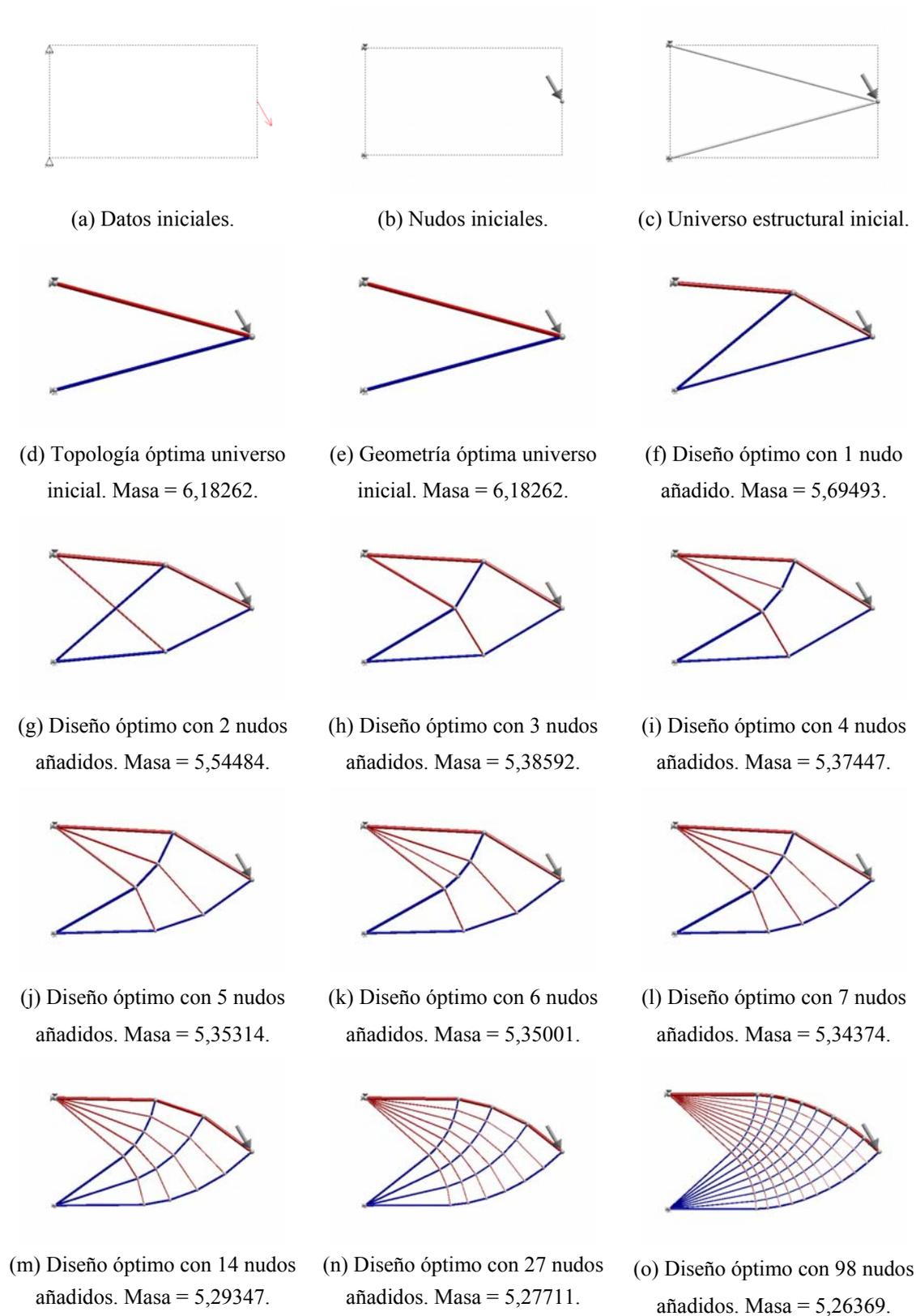


Figura 7.37 Voladizo de Michell ($\bar{L} = 1,82196$). Diseño óptimo: $\alpha = 60^\circ$ (sin pandeo)

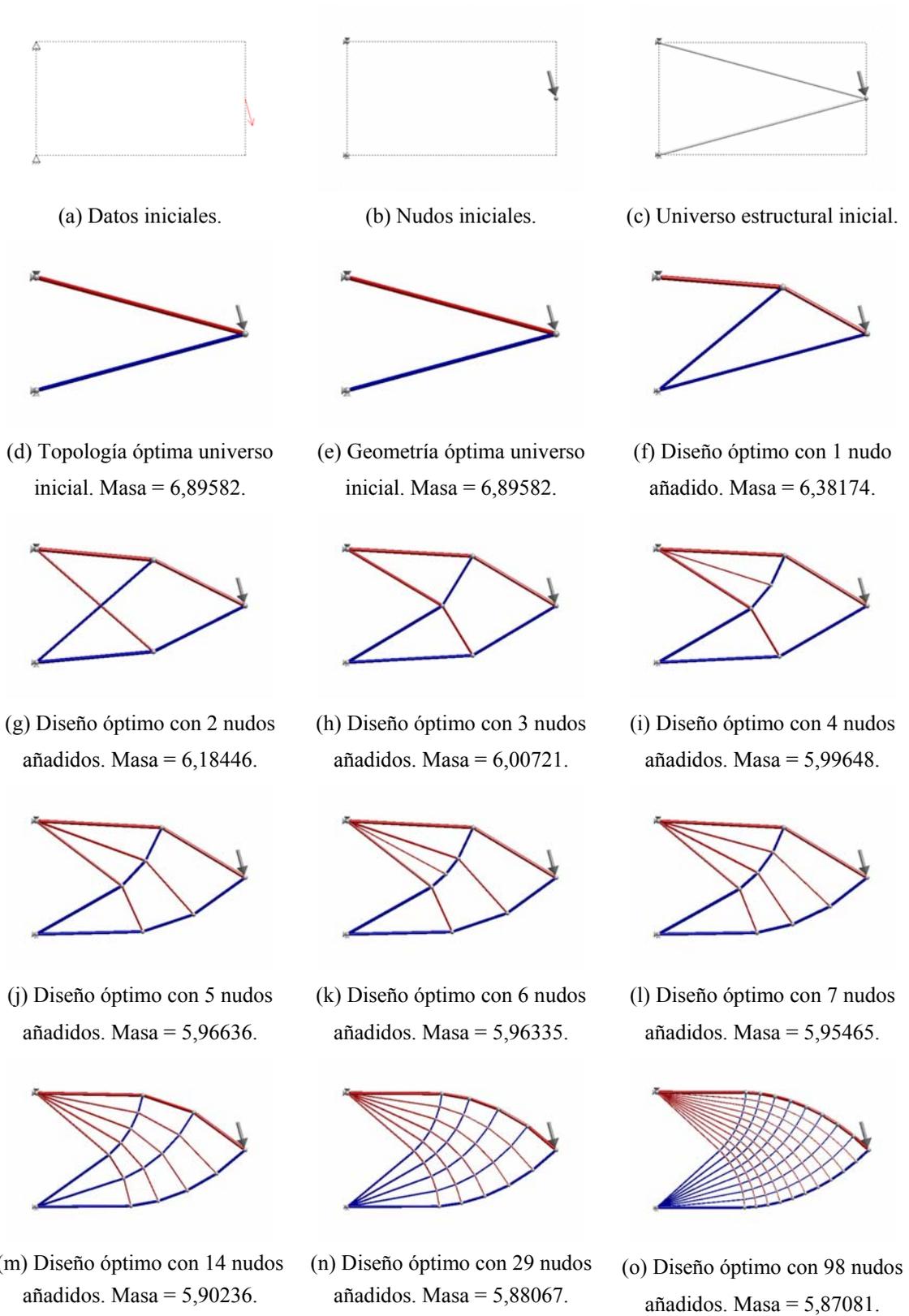


Figura 7.38 Voladizo de Michell ($\bar{L} = 1,82196$). Diseño óptimo: $\alpha = 75^\circ$ (sin pandeo)

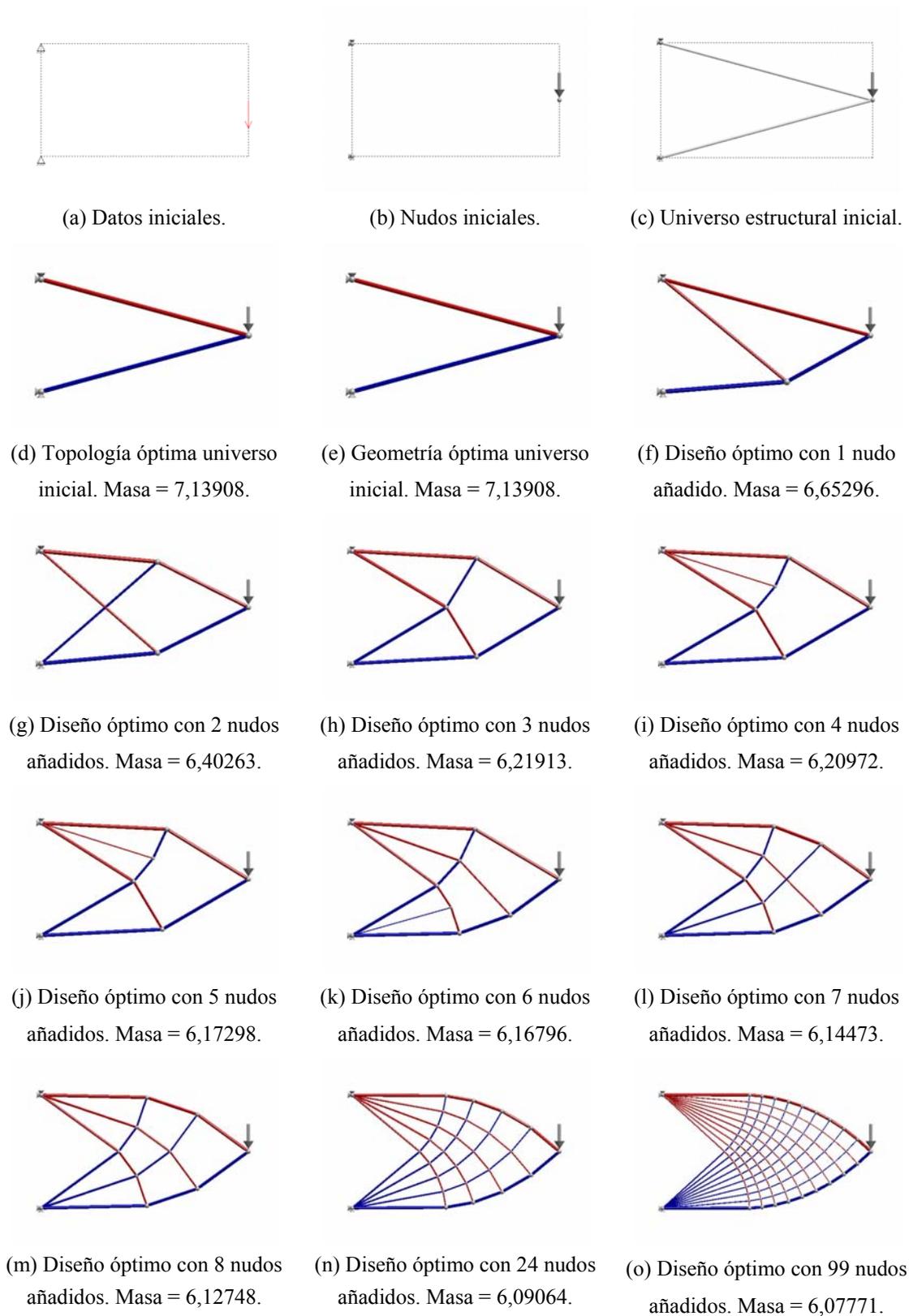


Figura 7.39 Voladizo de Michell ($\bar{L} = 1,82196$). Diseño óptimo: $\alpha = 90^\circ$ (sin pandeo)

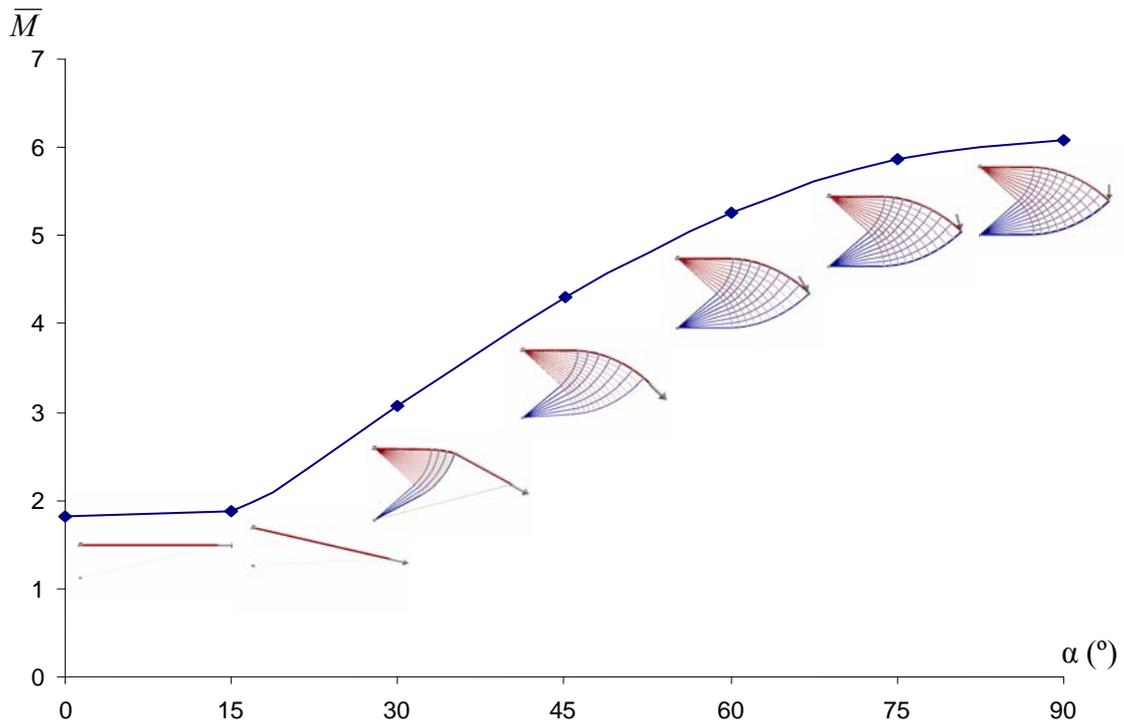


Figura 7.40 Voladizo de Michell ($\bar{L} = 1,82196$). Evolución masa adimensional/ángulo (sin pandeo)

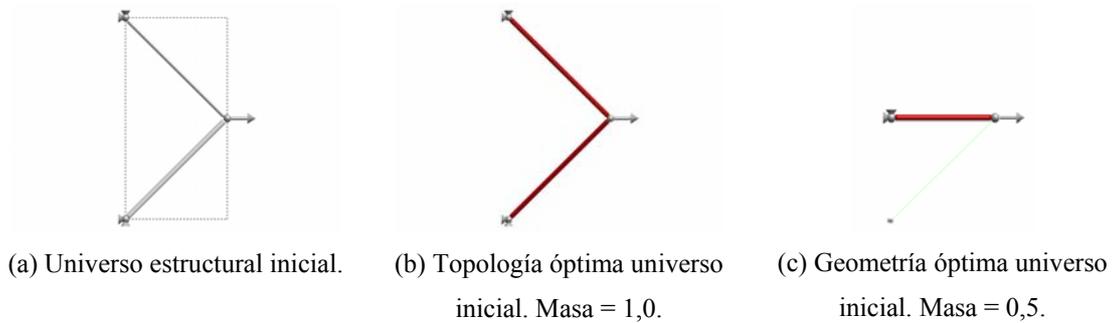


Figura 7.41 Voladizo de Michell ($\bar{L} = 0,5$). Diseño óptimo: $\alpha = 0^\circ$ (con pandeo)

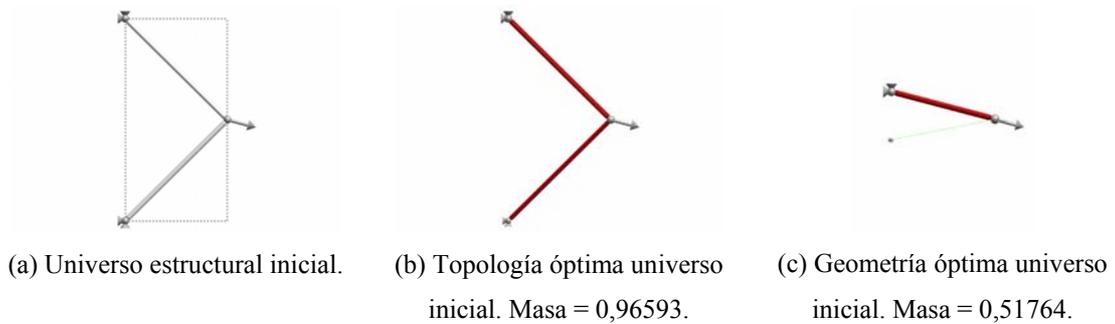


Figura 7.42 Voladizo de Michell ($\bar{L} = 0,5$). Diseño óptimo: $\alpha = 15^\circ$ (con pandeo)

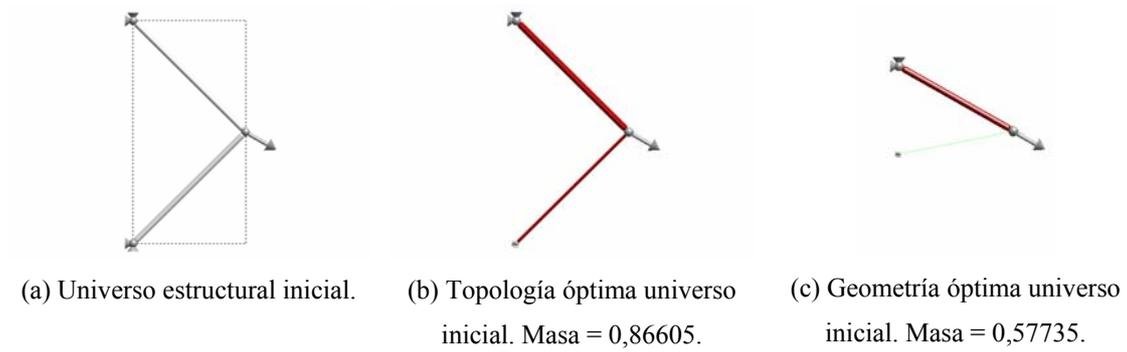


Figura 7.43 Voladizo de Michell ($\bar{L} = 0,5$). Diseño óptimo: $\alpha = 30^\circ$ (con pandeo)

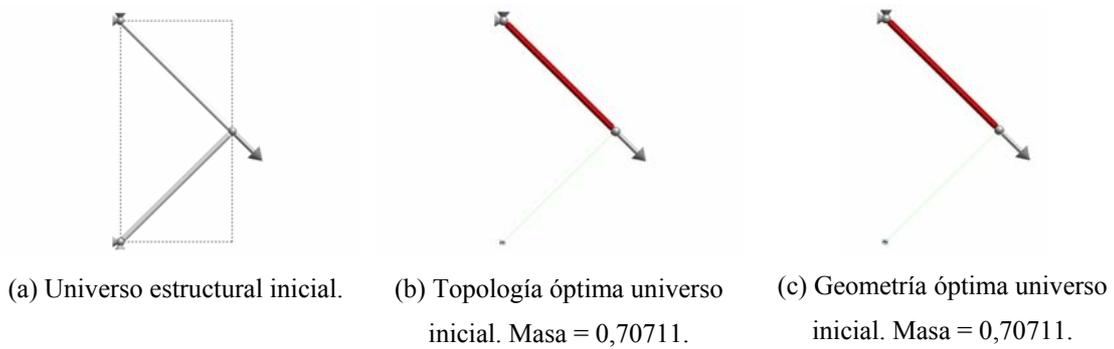


Figura 7.44 Voladizo de Michell ($\bar{L} = 0,5$). Diseño óptimo: $\alpha = 45^\circ$ (con pandeo)

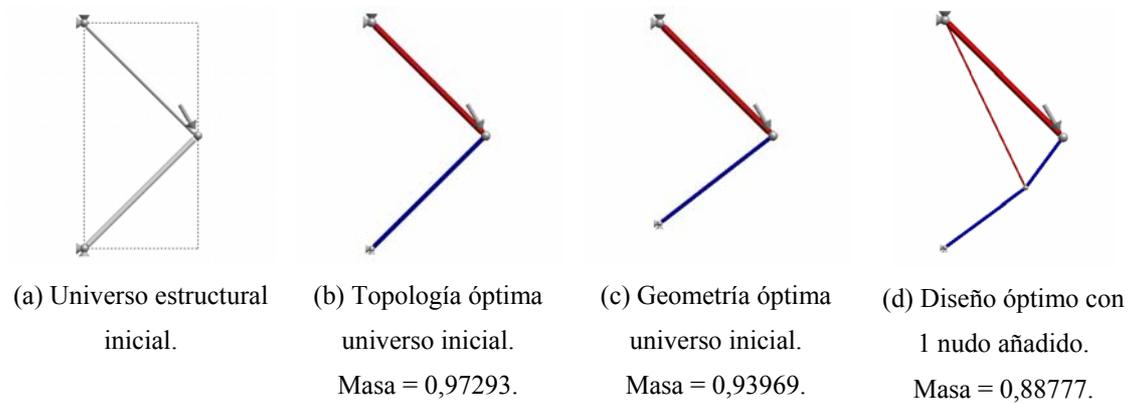


Figura 7.45 Voladizo de Michell ($\bar{L} = 0,5$). Diseño óptimo: $\alpha = 60^\circ$ (con pandeo)

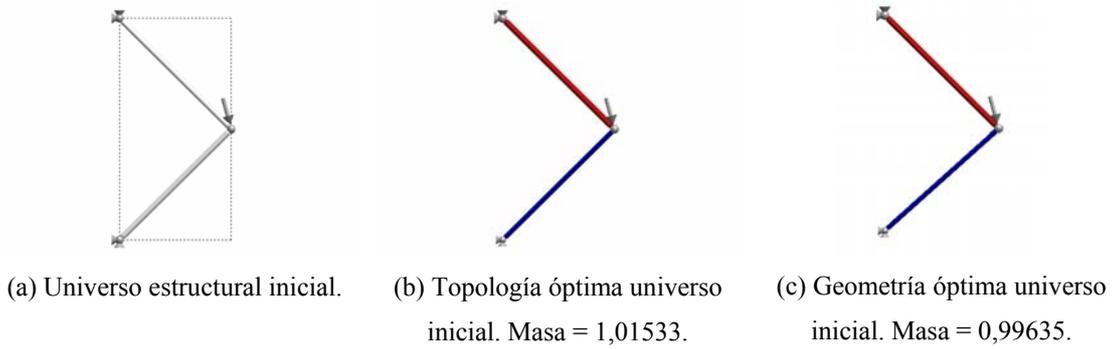


Figura 7.46 Voladizo de Michell ($\bar{L} = 0,5$). Diseño óptimo: $\alpha = 75^\circ$ (con pandeo)

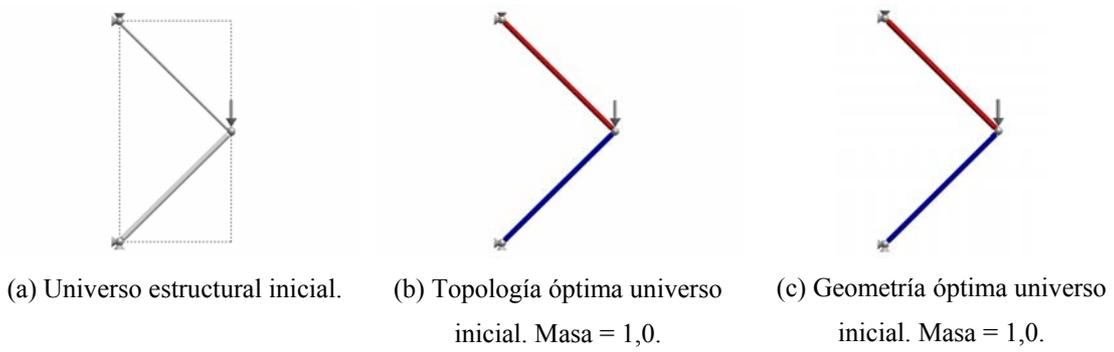


Figura 7.47 Voladizo de Michell ($\bar{L} = 0,5$). Diseño óptimo: $\alpha = 90^\circ$ (con pandeo)

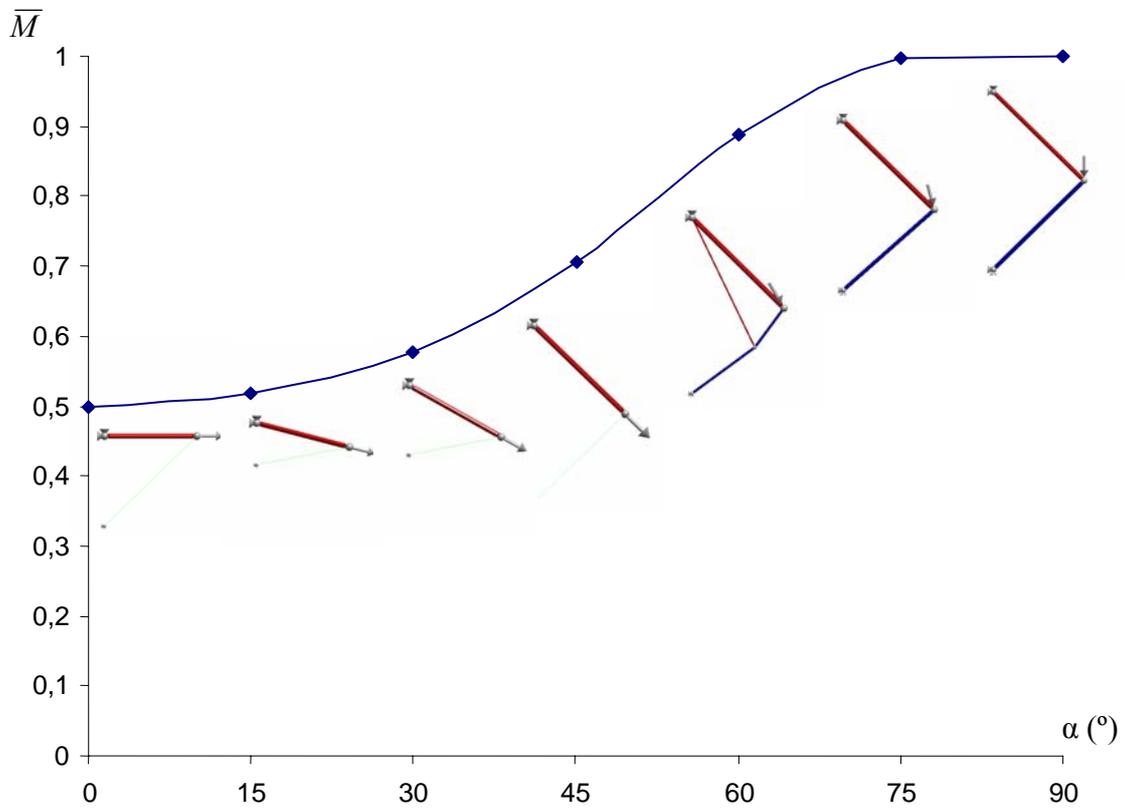


Figura 7.48 Voladizo de Michell ($\bar{L} = 0,5$). Evolución masa adimensional/ángulo (con pandeo)

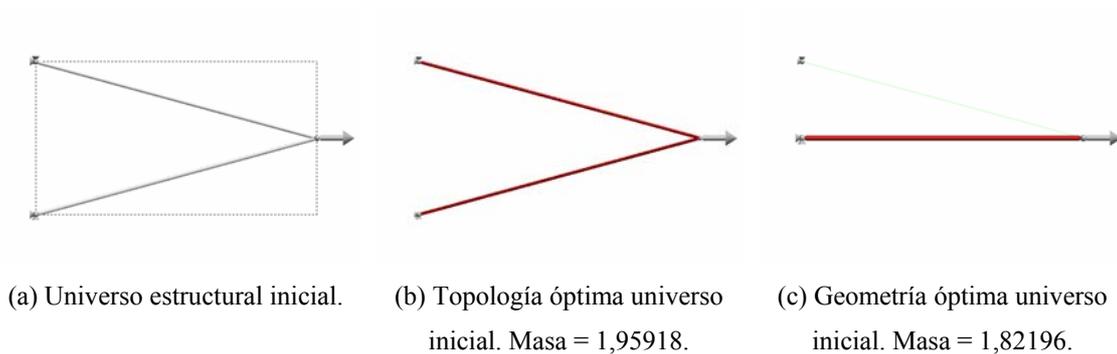


Figura 7.49 Voladizo de Michell ($\bar{L} = 1,82196$). Diseño óptimo: $\alpha = 0^\circ$ (con pandeo)

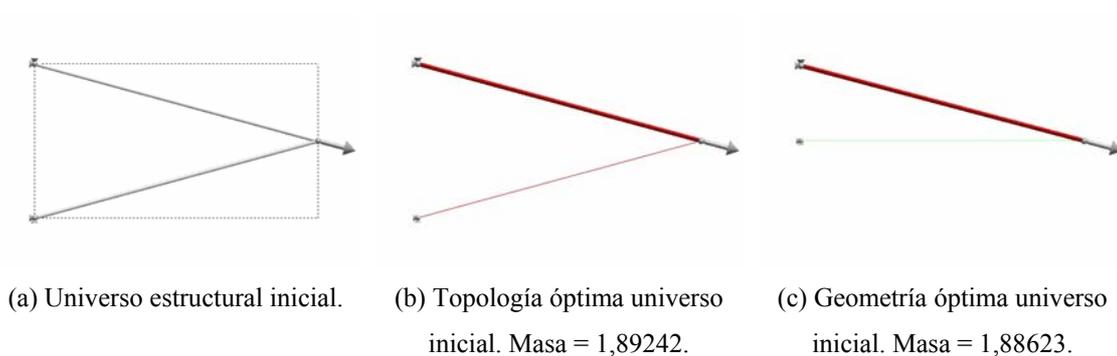


Figura 7.50 Voladizo de Michell ($\bar{L} = 1,82196$). Diseño óptimo: $\alpha = 15^\circ$ (con pandeo)

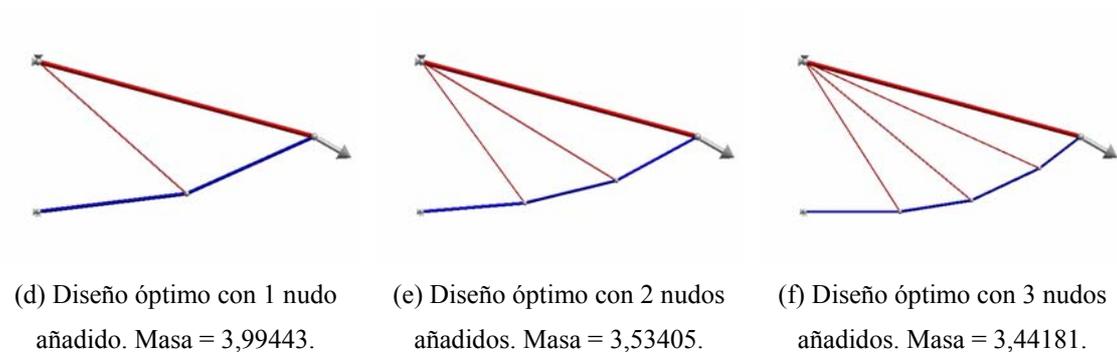
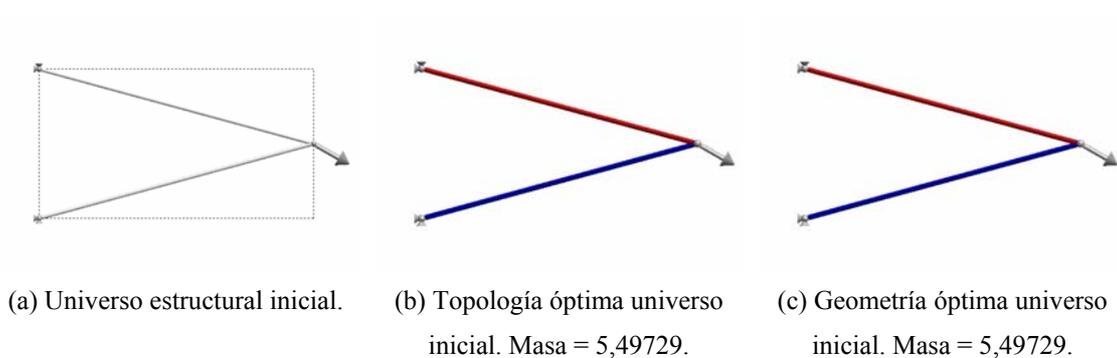


Figura 7.51 Voladizo de Michell ($\bar{L} = 1,82196$). Diseño óptimo: $\alpha = 30^\circ$ (con pandeo)

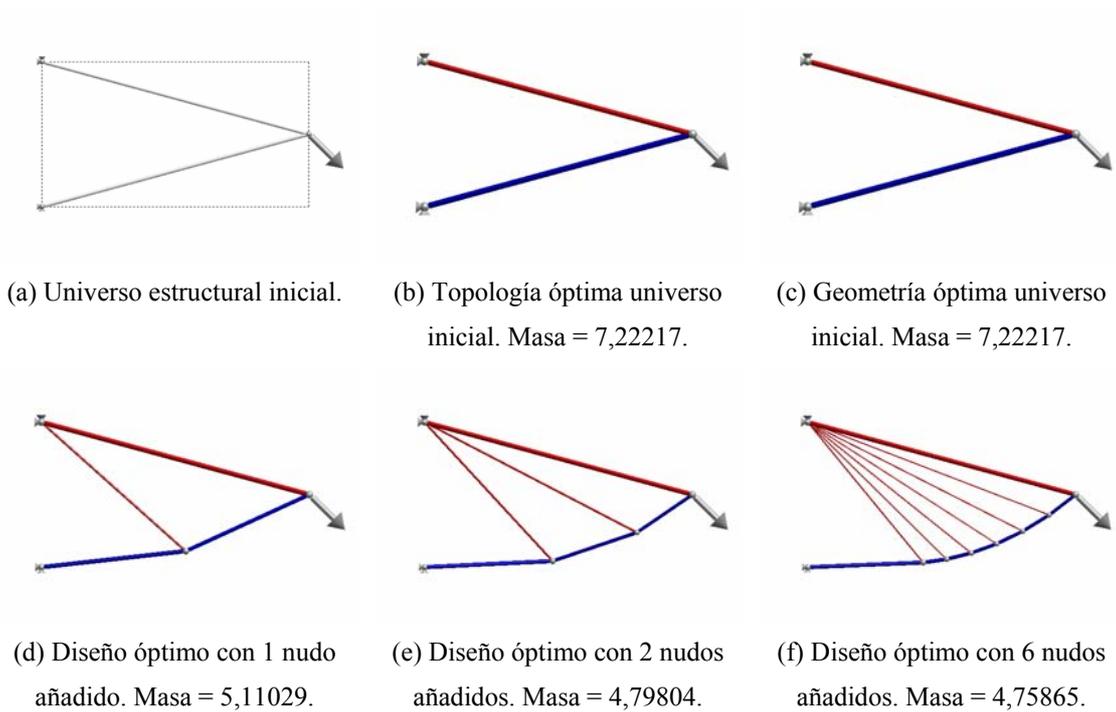


Figura 7.52 Voladizo de Michell ($\bar{L} = 1,82196$). Diseño óptimo: $\alpha = 45^\circ$ (con pandeo)

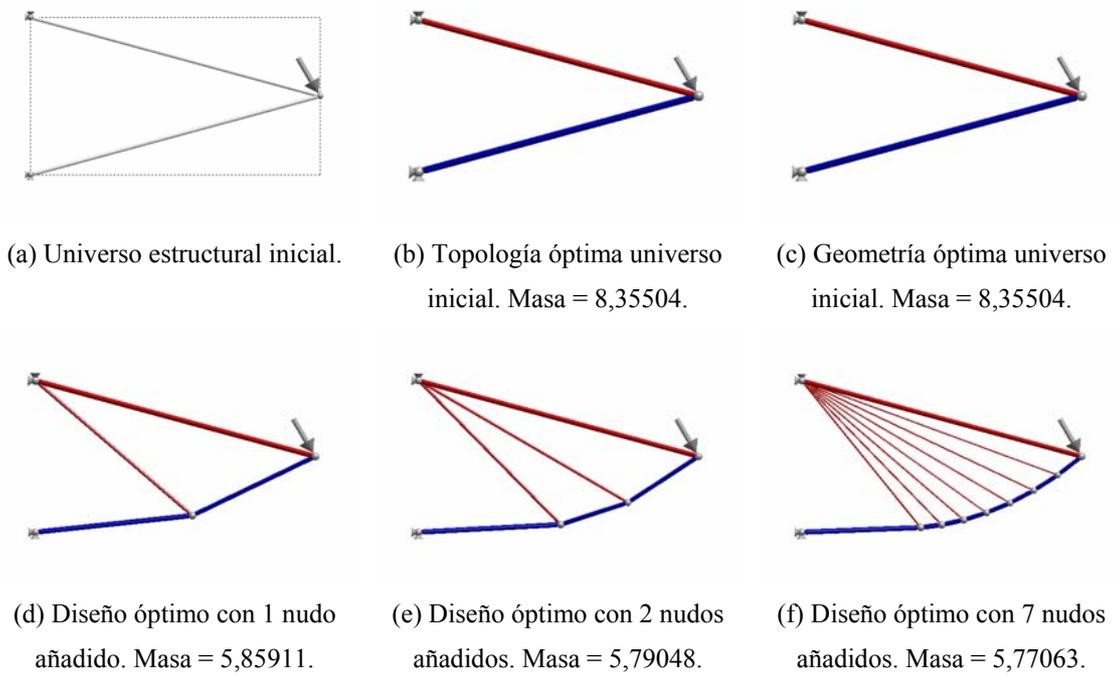


Figura 7.53 Voladizo de Michell ($\bar{L} = 1,82196$). Diseño óptimo: $\alpha = 60^\circ$ (con pandeo)

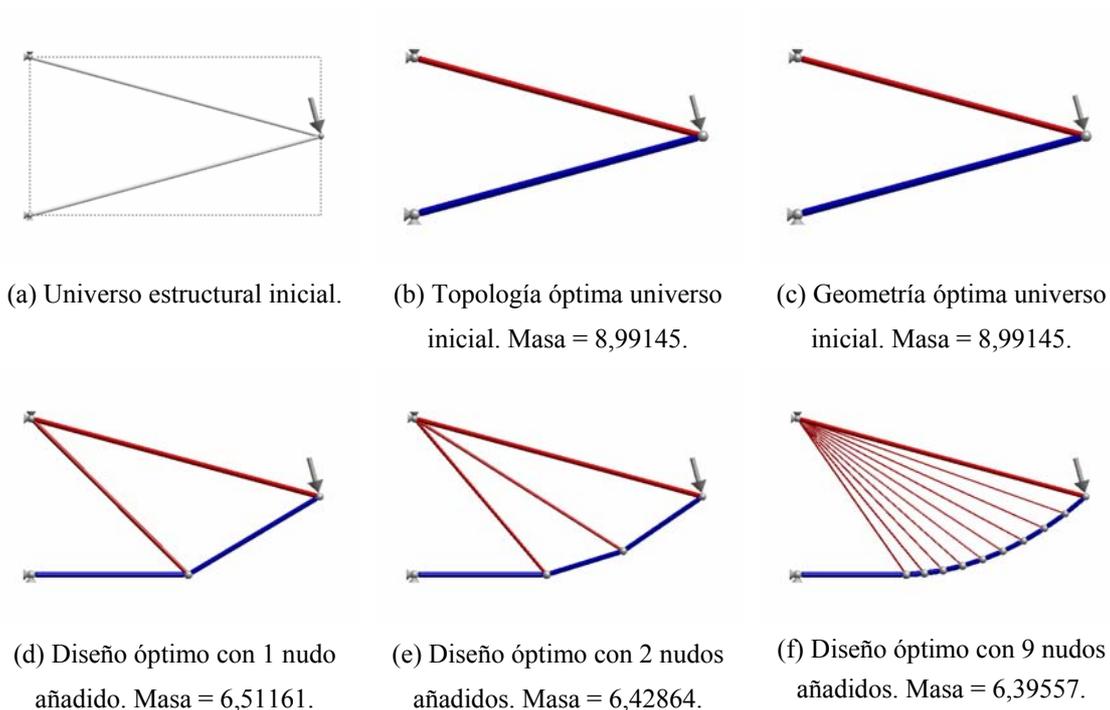


Figura 7.54 Voladizo de Michell ($\bar{L} = 1,82196$). Diseño óptimo: $\alpha = 75^\circ$ (con pandeo)

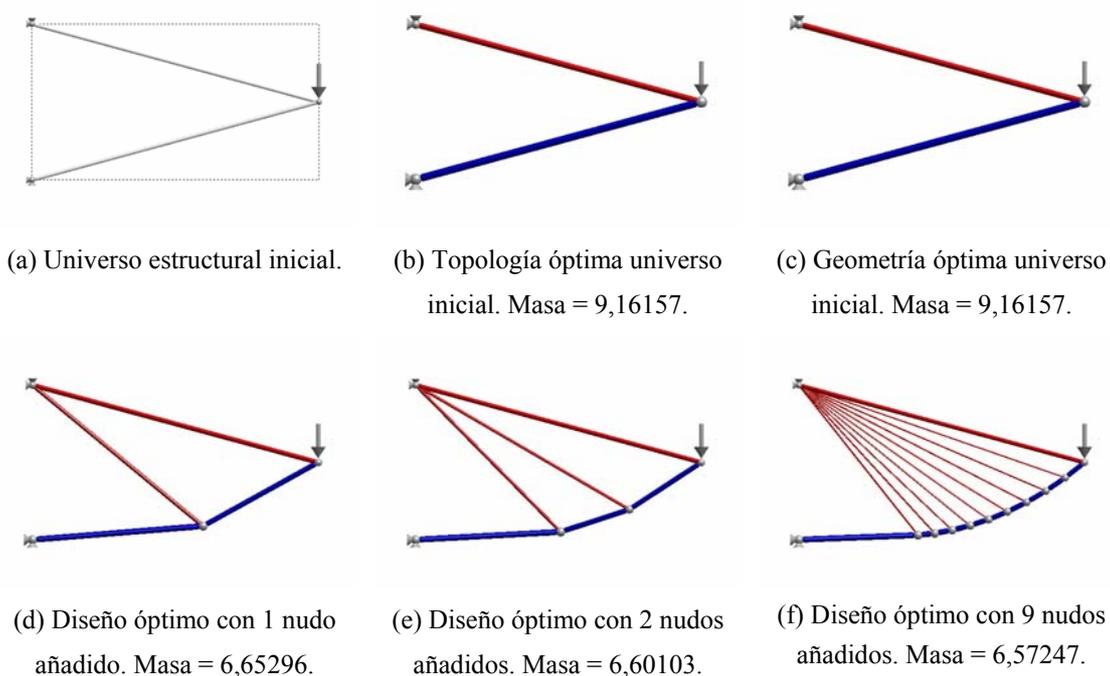


Figura 7.55 Voladizo de Michell ($\bar{L} = 1,82196$). Diseño óptimo: $\alpha = 90^\circ$ (con pandeo)

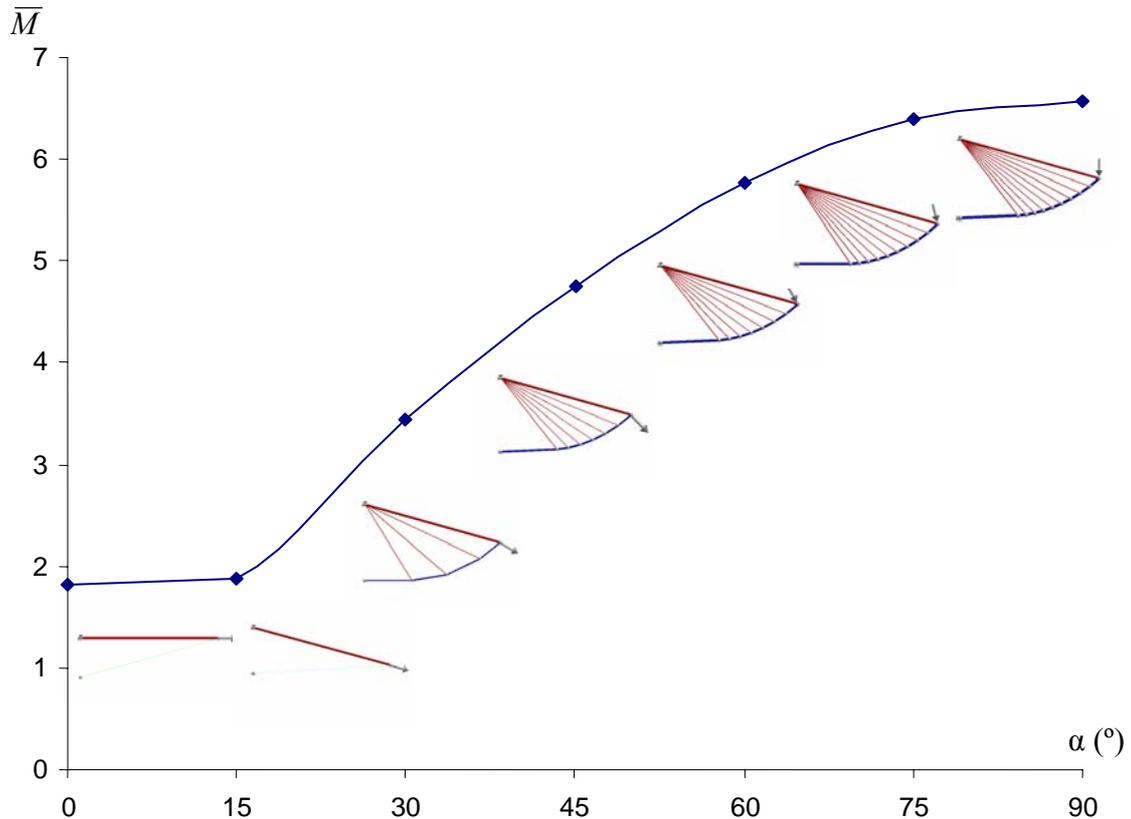


Figura 7.56 Voladizo de Michell ($\bar{L} = 1,82196$). Evolución masa adimensional/ángulo (con pandeo)

No se puede asegurar que los diseños óptimos obtenidos, para los casos en los que se considera pandeo de elementos, sean los correctos puesto que, aunque en la etapa de optimización de topología se utiliza el método de búsqueda exhaustiva (lo que asegura que la topología elegida es la mejor), después de optimizar la geometría puede ocurrir que la topología elegida ya no sea la óptima.

En estos diseños llega un momento en que no se encuentra ninguna posición donde añadir nuevos nudos de forma que disminuya la función objetivo. En cambio, si no se consideran restricciones de pandeo, el diseño óptimo tiene infinitos nudos, salvo cuando la línea de aplicación de la carga pase entre los dos apoyos, en cuyo caso, el diseño óptimo tiene dos barras (una que absorbe la carga y otra que evita la singularidad de la matriz de rigidez).

7.3.3 Método de bielas y tirantes para el hormigón armado

7.3.3.1 Introducción

El método de bielas y tirantes consiste en modelizar cada región D (de discontinuidad) de una estructura de hormigón armado sustituyéndola por un elemento (normalmente plano) constituido por barras isostáticas, que representan el comportamiento de dicha región. Las barras comprimidas se denominan bielas y representan los campos de compresiones del hormigón. Las barras traccionadas se denominan tirantes y representan las fuerzas de tracción de las armaduras.

Cuando la región D pertenece a una zona de la estructura, el modelo debe equilibrar las solicitaciones exteriores existentes en el contorno de dicha región. Cuando, por el contrario, la región D constituye por sí sola una estructura (discontinuidad generalizada), el modelo debe equilibrar las fuerzas exteriores y las reacciones de apoyo.

Al asimilar la región D a una estructura articulada, se puede aplicar la optimización de topología y geometría de estructuras articuladas al método de bielas y tirantes para el dimensionado de las armaduras.

7.3.3.2 Ménsula corta

Se va a resolver el ejemplo de la ménsula corta de la Fig. 7.57 (Adebar, Kuchma y Collins, 1990). Una ménsula corta es una región D , de tipo estático (carga concentrada), y geométrico (cambio brusco en la geometría del elemento).

La ménsula corta está diseñada a partir de un pilar de 350 mm x 350 mm para soportar la reacción de una viga que está apoyada sobre una placa soporte. En la Fig. 7.57a se muestran las dimensiones de la ménsula corta.

Para que los resultados puedan compararse con los del artículo de referencia, se ha adoptado para el acero un límite elástico $f_y = 420$ MPa y una resistencia de cálculo $f_{yd} = 315$ MPa, y para el hormigón una resistencia característica $f_{yk} = 35$ MPa y una resistencia de cálculo $f_{yd} = 16,7$ MPa.

La carga vertical mayorada es de 250 kN y la horizontal de 50 kN y se asume que el punto de aplicación está desplazado hacia el exterior 25 mm respecto del centro de la placa de apoyo para permitir excentricidades (Fig. 7.57b). El recubrimiento adoptado en la zona de la placa de apoyo y en la cara izquierda del pilar, es 50 mm. Adebar determina que la distancia mínima del punto D al exterior del pilar es 39,5 mm.

La función objetivo es la masa del acero, cuya densidad es $\rho = 7850$ kg/m³.

En la Fig. 7.57c se muestra el modelo para la optimización de topología y geometría, eliminando del dominio de existencia de la ménsula las zonas de recubrimiento, y aplicando las condiciones de contorno.

Este ejemplo se va a resolver con el método propuesto en esta tesis y a comparar con el diseño obtenido por Adebar.

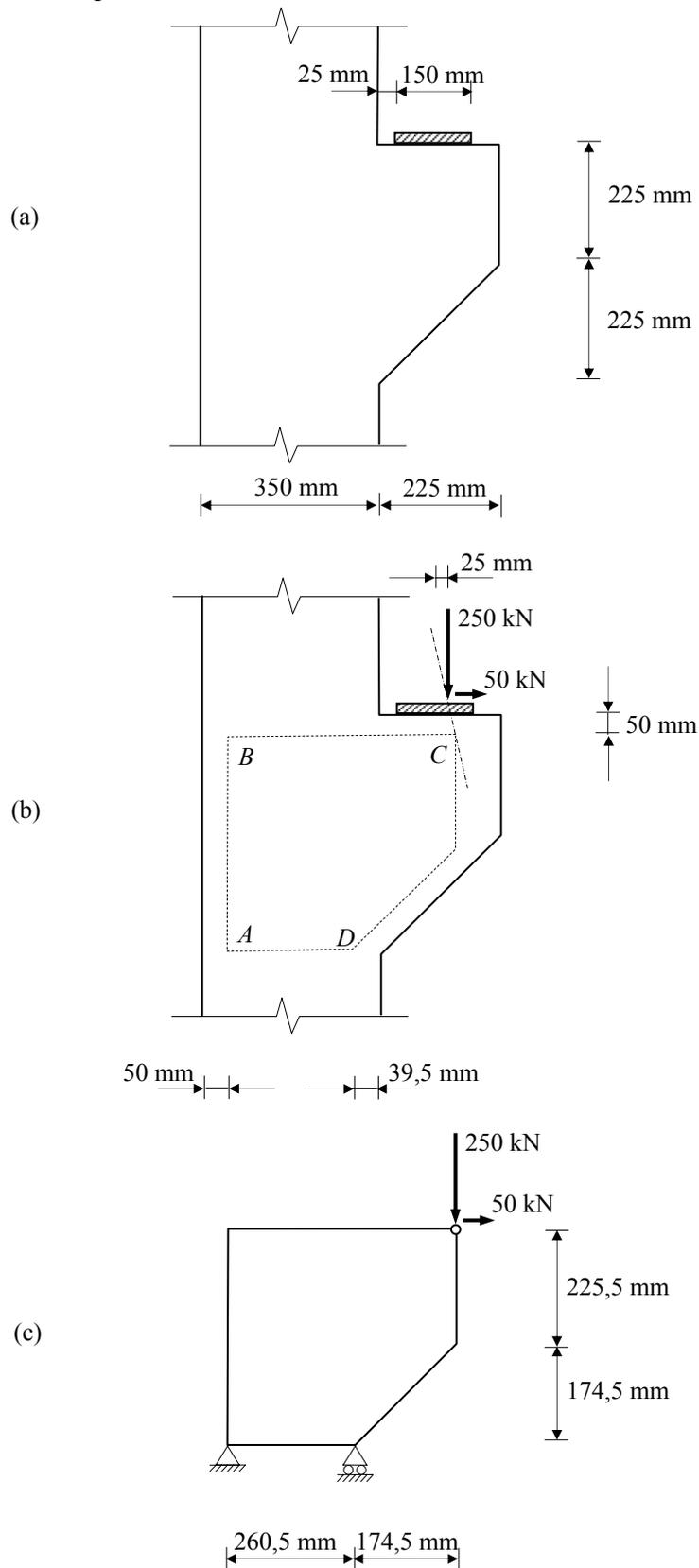


Figura 7.57 Ménsula corta (Adebar, Kuchma y Collins, 1990).

En la Fig. 7.58 se muestran los datos (Fig. 7.58a), la topología de cinco barras adoptada por Adebar (Fig. 7.58b) y el diseño obtenido por Adebar (Fig. 7.58c), que sería el mismo que se obtendría optimizando la topología de la estructura articulada de la Fig. 7.58b (en realidad sería una optimización únicamente de propiedades ya que la estructura es isostática).

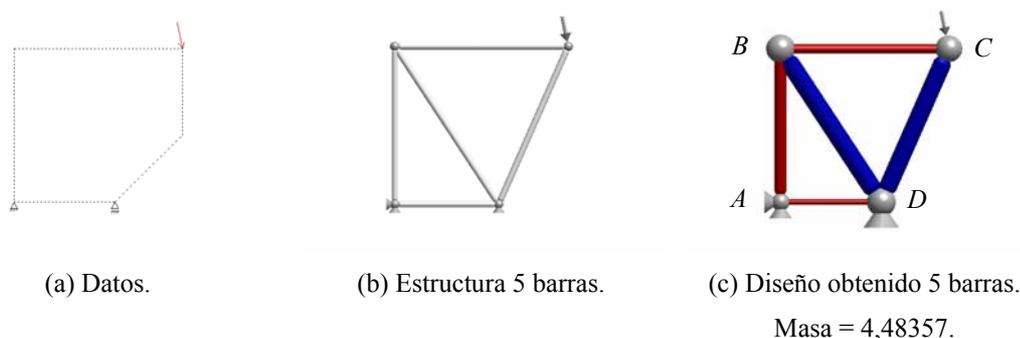


Figura 7.58 Ménsula corta. Diseño obtenido por Adebar

En la Fig. 7.58c las secciones transversales de las barras se dibujan a escala real. Las secciones de las bielas se consideran rectangulares (con una anchura de 350 mm). Las secciones de los tirantes se consideran circulares, aunque en la práctica se dividen en varios redondos repartidos a lo largo de la profundidad del pilar.

Las áreas de las secciones transversales de los tirantes obtenidas por Adebar son: $A_{AB} = 7,754 \text{ cm}^2$, $A_{BC} = 5,050 \text{ cm}^2$ y $A_{AD} = 1,587 \text{ cm}^2$. Como en la práctica los tirantes AB y BC se obtienen a partir de una misma barra de sección constante doblada por B , el área de la sección transversal de esta barra debe ser la de la barra AB (la de mayor sección). Con lo que el peso de la armadura será 5,40695 kg.

En la Fig. 7.59 se muestran los diseños óptimos obtenidos mediante el método propuesto para diferentes números de nudos añadidos.

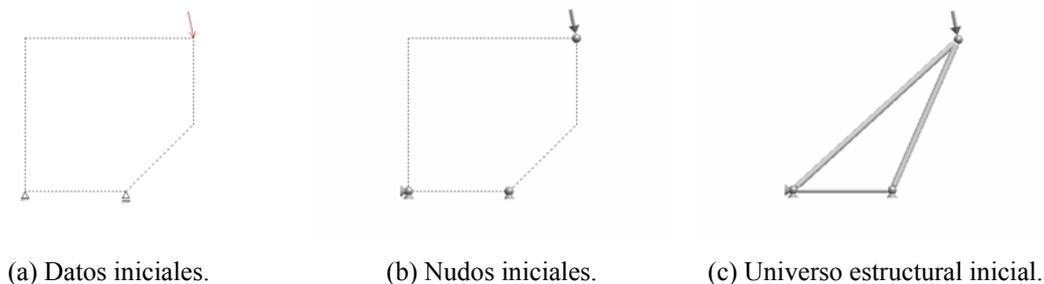


Figura 7.59 Ménsula corta. Diseños óptimos obtenidos mediante el método de crecimiento propuesto

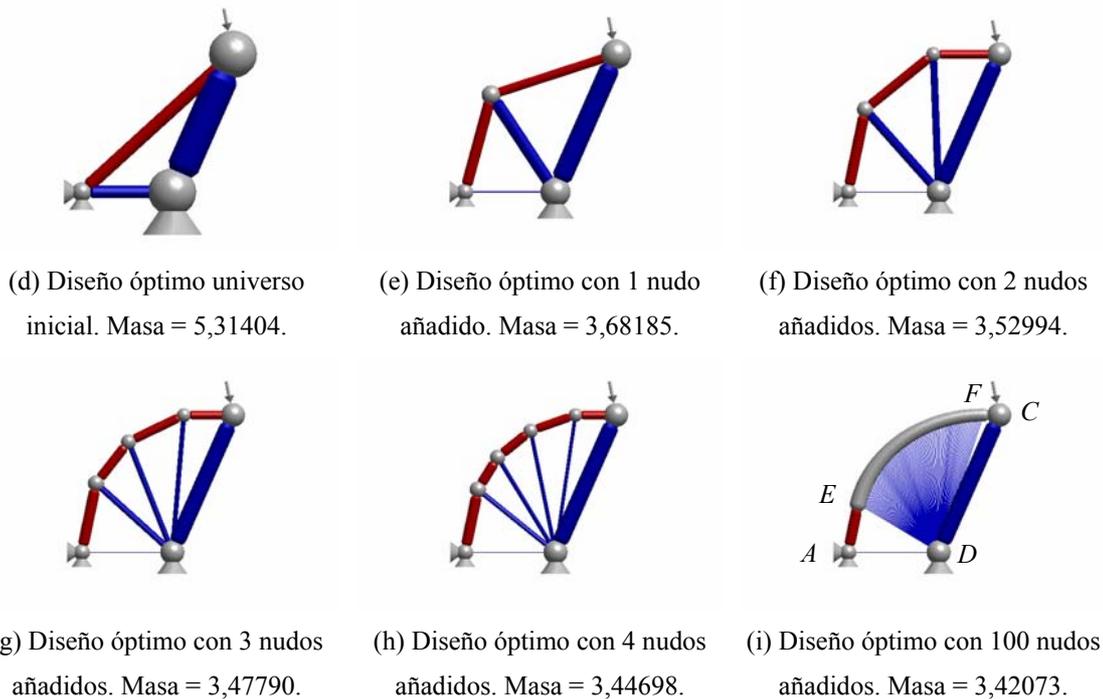


Figura 7.59 Ménsula corta. Diseños óptimos obtenidos mediante el método de crecimiento propuesto. (Continuación)

La topología del diseño obtenido con el método de crecimiento con 1 nudo añadido es la misma que la adoptada por Adebar. Optimizando la geometría del diseño obtenido por Adebar se obtendría la misma solución de la Fig. 7.59e.

Todos los tirantes de los diseños obtenidos están unidos uno a continuación de otro, con lo que la armadura se puede formar con una barra convenientemente doblada.

En el diseño de la Fig. 7.59i el valor máximo de las áreas de las secciones transversales de los tirantes (tirante AE) es $A_{m\acute{a}x} = 7,915 \text{ cm}^2$ y el valor mínimo (tirante FC) es $A_{m\acute{i}n} = 5,050 \text{ cm}^2$. Una barra de sección constante debería tener un área igual al valor máximo de las áreas de los tirantes, con lo que el peso de la armadura sería 4,15167 kg, un 23,3 % menor que el peso del diseño de Adebar.

El diseño obtenido no necesita el tirante horizontal inferior que aparece en el diseño de Adebar (Fig. 7.58c).

La biela BD del diseño de la Fig. 7.58c está sometida a un esfuerzo de compresión de 291,5 kN, que absorbe el hormigón próximo a esa diagonal. En cambio, en el diseño de la Fig. 7.59i, el esfuerzo está mucho más repartido, como muestran las dimensiones de las bielas, mucho más pequeñas que las del diseño anterior.

7.3.4 Optimización de topología y geometría de mecanismos

7.3.4.1 Introducción

Se va a hacer una aplicación en el campo de los mecanismos. Frecker (1999) aplica la optimización de topología de estructuras articuladas a la optimización de topología de mecanismos haciendo uso del universo estructural.

El objetivo es obtener la topología óptima de forma que, al aplicar unas fuerzas exteriores, ciertos puntos se desplacen en una dirección determinada y que el mecanismo tenga la suficiente rigidez para transmitir el movimiento. Se define como ganancia geométrica a la relación entre los desplazamientos obtenidos y los desplazamientos de los puntos de aplicación de las fuerzas exteriores.

7.3.4.2 Método de Frecker

Para el diseño óptimo de mecanismos, Frecker (1999) plantea dos estados de cargas (Fig. 7.60): el primer estado de cargas incluye las cargas exteriores y cargas virtuales en los puntos donde se quiere provocar un desplazamiento (en la dirección y sentido de los desplazamientos), y el segundo estado de cargas fija los puntos de aplicación de las cargas exteriores y aplica unas cargas en los puntos de desplazamiento deseado, opuestas a las del primer estado. El segundo estado representa la resistencia del mecanismo.

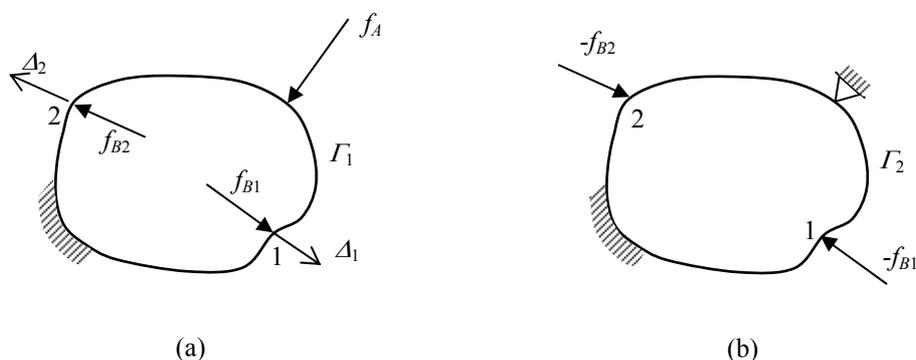


Figura 7.60 Mecanismo. (a) Diseño a flexibilidad; (b) Diseño a rigidez

Para optimizar la topología del mecanismo hay que maximizar la flexibilidad de la estructura del primer estado, que equivale a maximizar la energía potencial mutua, que es el producto del vector de desplazamientos debidos a las cargas exteriores por el vector de fuerzas virtuales (Shield y Prager, 1970). Pero esto no es suficiente ya que, en este caso, el algoritmo de optimización tendería a hacer nulas las áreas de las barras (flexibilidad infinita) y, además, el mecanismo debe tener cierta rigidez para poder transmitir esfuerzos. Por lo tanto, también hay que maximizar la rigidez de la estructura del segundo estado, que equivale a minimizar la energía de deformación (*compliance*).

Para la resolución de este problema, Frecker plantea dos posibilidades: la primera es maximizar la diferencia ponderada de la energía potencial mutua del primer estado y la *compliance* del segundo (minimizar la *compliance* equivale a maximizar su opuesta), y la segunda es maximizar el cociente entre la energía potencial mutua del primer estado dividida por la *compliance* del segundo (minimizar la *compliance* equivale a maximizar su inversa). En el presente trabajo se ha implementado el segundo criterio.

Para evitar la singularidad de la matriz de rigidez del mecanismo resultante, Frecker añade, para todas las barras, restricciones de área mínima, y para que las áreas no se hagan infinitas, añade restricciones de área máxima y de masa máxima del mecanismo.

El problema de optimización de topología de mecanismos es el siguiente

$$\begin{aligned} \text{maximizar :} & \quad \frac{\int_{\Gamma_1} \mathbf{f}_B \mathbf{u}_A d\Gamma}{\int_{\Gamma_2} -\mathbf{f}_B \mathbf{u}_B d\Gamma} \\ \text{sujeito a :} & \quad A_{\min} \leq A_i \leq A_{\max} \quad i = 1, 2, \dots, m \\ & \quad M \leq M_{\max} \end{aligned} \quad (7.1)$$

siendo:

- \mathbf{f}_B Vector de fuerzas virtuales.
- \mathbf{u}_A Vector de desplazamientos debidos a las cargas exteriores.
- \mathbf{u}_B Vector de desplazamientos debidos a las cargas virtuales.
- Γ_1 Contorno del primer estado de cargas.
- Γ_2 Contorno del segundo estado de cargas.
- A_i Área de la barra i .
- A_{\min} Área mínima de las barras.
- A_{\max} Área máxima de las barras.
- M Masa del mecanismo.
- M_{\max} Valor máximo de la masa del mecanismo.

Al existir restricciones de área mínima para las barras, no hay optimización de topología (no pueden desaparecer al hacerse el área nula). Frecker no comenta cómo ha solucionado este problema.

Frecker utiliza un método de programación lineal secuencial (SLP) para la resolución del problema. Justifica lo no utilización de métodos de programación cuadrática secuencial (SQP), cuya convergencia es más rápida, a causa de lo especial de la función objetivo. La función objetivo es un cociente de dos funciones convexas, y Frecker dice que puede no ser adecuadamente aproximado por una función cuadrática en una aproximación SQP.

7.3.4.3 Método de crecimiento propuesto

En el método de crecimiento propuesto se eliminan las restricciones de área mínima y se añade un muelle que evita la singularidad de la matriz de rigidez al convertir el mecanismo en una estructura articulada.

Para la optimización de la topología en cada iteración, se utiliza el método de búsqueda exhaustiva. Para calcular la función objetivo, en cada una de las topologías isostáticas consideradas, se resuelve el problema (7.1) mediante un método SQP, que para un número reducido de variables no tiene los problemas que comenta Frecker.

En las estructuras articuladas, para la definición de los nuevos nudos en el crecimiento de la topología, se sigue el método basado en las direcciones principales de tensión. Este método no parece el más adecuado para problemas de mecanismos, por lo que se ha optado por el método general basado en los puntos de una rejilla en el dominio de definición del mecanismo.

7.3.4.4 Mecanismo extractor de disquetes

Se va a resolver el ejemplo del mecanismo extractor de disquetes (Fig. 7.61) propuesto por Frecker (1999). Se desea que al aplicar una fuerza horizontal en el punto A el punto 1 se desplace hacia arriba y el punto 2 hacia la izquierda.

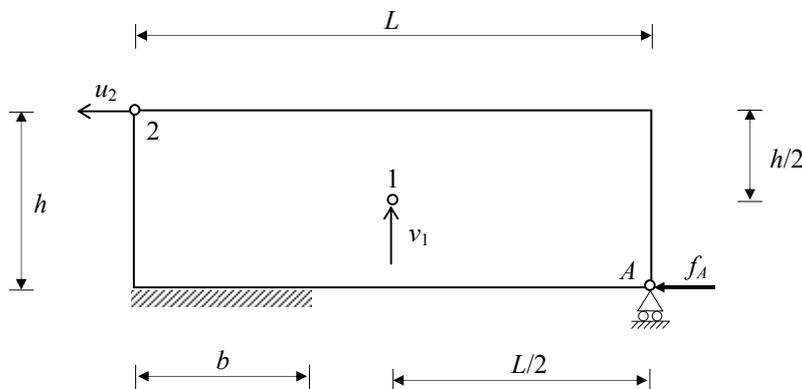


Figura 7.61 Mecanismo extractor de disquetes (Frecker, 1999)

El primer estado de cargas (Fig. 7.62a) incluye la fuerza en el punto A y dos fuerzas virtuales en los puntos 1 y 2 (en la dirección y sentido de los desplazamientos deseados), y el segundo estado de cargas (Fig. 7.62b) fija el punto A y aplica unas fuerzas en los puntos 1 y 2 opuestas a las fuerzas virtuales del primer estado de cargas.

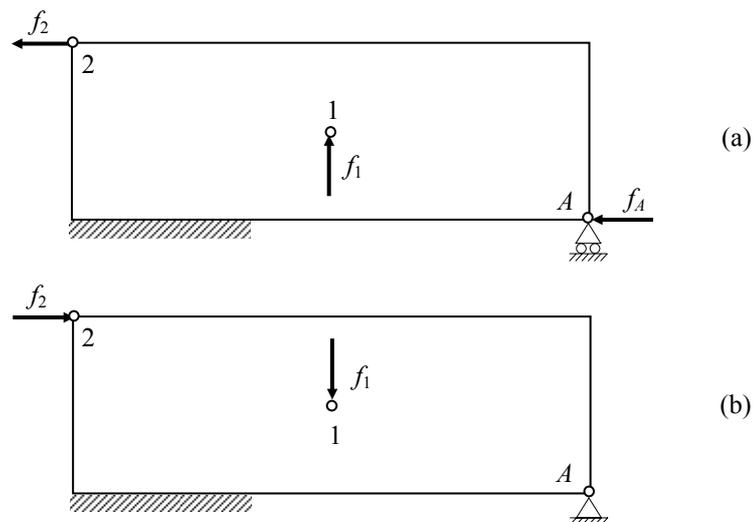


Figura 7.62 Mecanismo extractor de disquetes. Estados de cargas

El muelle que se añade en el algoritmo implementado se coloca horizontalmente, fijo en un extremo y conectado al nudo A por el otro extremo.

Las ganancias geométricas de los mecanismos resultantes, para los datos de la Fig. 7.61, son independientes de los valores numéricos de estos datos, pero para obtener la función objetivo sí son necesarios. Frecker no dice qué valores ha utilizado, así que se han elegido los siguientes: longitud $L = 120$ mm, altura $h = 70$ mm, anchura del apoyo $b = 40$ mm, módulo de Young $E = 210$ kN/mm², densidad $\rho = 7850$ kg/m³, rigidez del muelle $k = 0,1$ kN/mm, desplazamiento horizontal del punto A (para el cálculo de la ganancia geométrica) $u_A = 10$ mm, carga puntual en A $f_A = u_A k = 1$ kN, carga virtual 1 $f_1 = f_A = 1$ kN, carga virtual 2 $f_2 = f_A = 1$ kN, área máxima $A_{m\acute{a}x} = 10$ mm², y masa máxima $M_{m\acute{a}x} = 0,01$ kg. Se consideran barras con sección transversal circular (a efectos de representación gráfica).

El algoritmo de Frecker supone linealidad geométrica. Como los mecanismos tienen una no linealidad geométrica alta, además de las ganancias geométricas lineales, se han determinado las ganancias geométricas no lineales en los diseños óptimos obtenidos. Para ello, se ha realizado un análisis no lineal del mecanismo obtenido, aplicando la fuerza exterior de forma incremental y modificando la geometría en cada paso.

En la Fig. 7.63 se muestra el diseño óptimo obtenido por Frecker, los dos estados de cargas (Figs. 7.63a y 7.63b), el universo estructural utilizado (6x4, Fig. 7.63c), y la topología óptima (Fig. 7.63d) con las ganancias geométricas conseguidas (la ganancia no lineal es la correspondiente a un desplazamiento del punto A de 10 mm). Las deformadas se dibujan, sin amplificación, con línea continua para la geometría lineal, y con línea discontinua para la geometría no lineal.

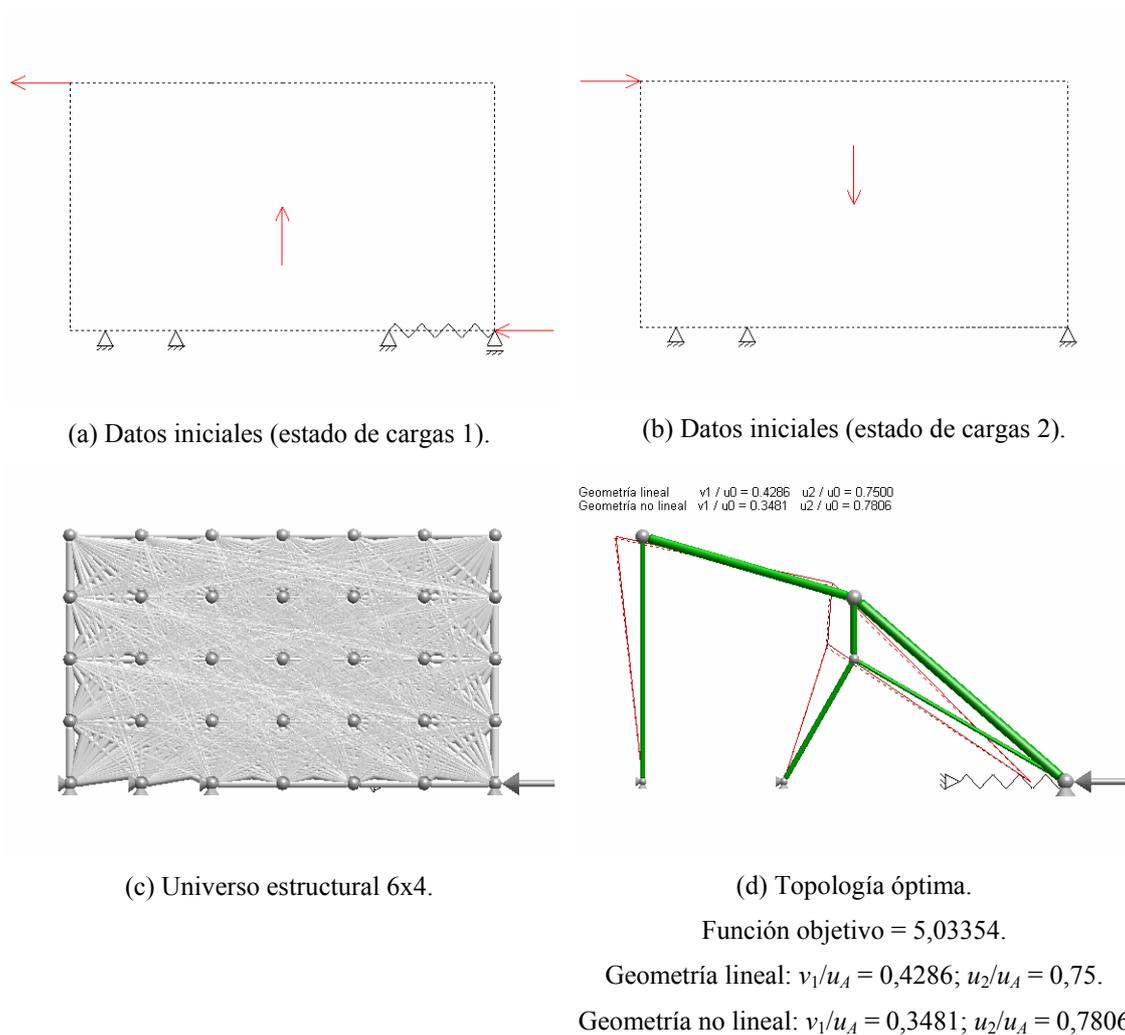


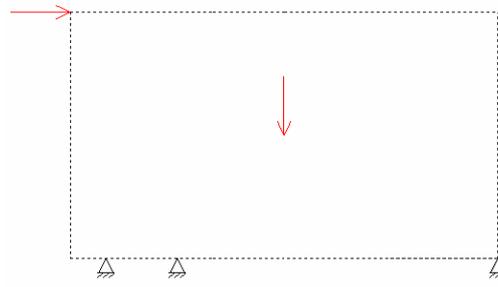
Figura 7.63 Mecanismo extractor de disquetes. Diseño óptimo obtenido por Frecker

Para la optimización del mecanismo extractor de disquetes mediante el método de crecimiento se ha elegido una rejilla de 14x14 puntos. En la Fig. 7.64 se muestra el diseño óptimo obtenido, los dos estados de cargas (Figs. 7.64a y 7.64b), los nudos iniciales (Fig. 7.64c), el diseño óptimo sin nudos añadidos con sus deformadas (Fig. 7.64d), el diseño óptimo con un nudo añadido (Fig. 7.64e), y el diseño óptimo con dos nudos añadidos (Fig. 7.64f).

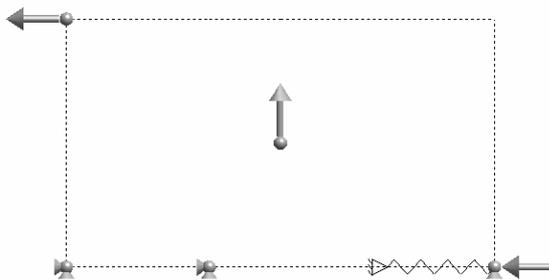
Puede apreciarse que la topología del diseño óptimo obtenido con un nudo añadido es la misma que la obtenida por Frecker, pero al efectuarse una optimización de geometría, la función objetivo del diseño óptimo correspondiente al método de crecimiento es mayor.



(a) Datos iniciales (estado de cargas 1).

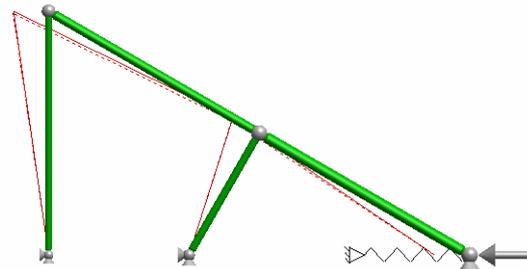


(b) Datos iniciales (estado de cargas 2).



(c) Nudos iniciales.

Geometría lineal $v_1 / u_0 = 0.4286$ $u_2 / u_0 = 1.0000$
 Geometría no lineal $v_1 / u_0 = 0.3481$ $u_2 / u_0 = 1.0094$



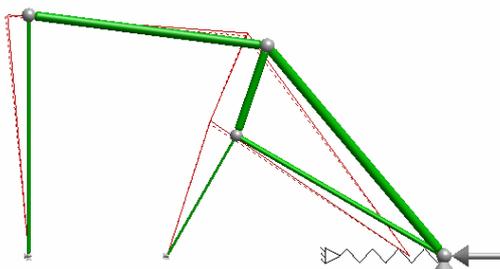
(d) Diseño óptimo sin nudos añadidos.

Función objetivo = 3,64388.

Geometría lineal: $v_1/u_A = 0,4286$; $u_2/u_A = 1,0$.

Geometría no lineal: $v_1/u_A = 0,3481$; $u_2/u_A = 1,0094$

Geometría lineal $v_1 / u_0 = 0.4286$ $u_2 / u_0 = 0.6111$
 Geometría no lineal $v_1 / u_0 = 0.3481$ $u_2 / u_0 = 0.6659$



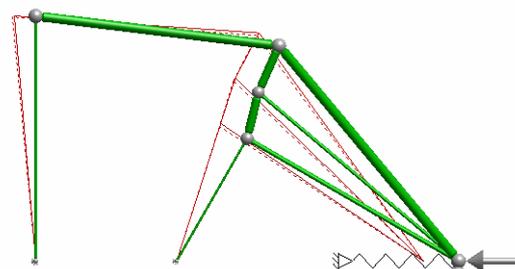
(e) Diseño óptimo con un nudo añadido.

Función objetivo = 5,50846.

Geometría lineal: $v_1/u_A = 0,4286$; $u_2/u_A = 0,6111$.

Geometría no lineal: $v_1/u_A = 0,3481$; $u_2/u_A = 0,6659$

Geometría lineal $v_1 / u_0 = 0.4286$ $u_2 / u_0 = 0.6054$
 Geometría no lineal $v_1 / u_0 = 0.3481$ $u_2 / u_0 = 0.6611$



(f) Diseño óptimo con dos nudos añadidos.

Función objetivo = 5,52823.

Geometría lineal: $v_1/u_A = 0,4286$; $u_2/u_A = 0,6054$.

Geometría no lineal: $v_1/u_A = 0,3481$; $u_2/u_A = 0,6611$

Figura 7.64 Mecanismo extractor de disquetes. Diseño óptimo obtenido mediante el método de crecimiento propuesto

En este ejemplo, la ganancia geométrica del desplazamiento vertical del punto 1 se mantiene al aumentar el número de nudos añadidos, pero la ganancia del desplazamiento horizontal del punto 2 disminuye, es decir, la flexibilidad del primer estado de cargas no sólo no aumenta si no que disminuye. Esto se explica, porque la rigidez correspondiente al segundo estado de cargas aumenta en mayor medida que disminuye la flexibilidad del primero, con lo que la función objetivo aumenta. Como es lógico, la función objetivo sí aumenta al añadir nuevos nudos.

El tiempo de cálculo para el método de crecimiento propuesto sin nudos añadidos es de 0,01 segundos, con un nudo añadido (topología igual a la obtenida por Frecker) es de 8 segundos y con dos nudos añadidos 1 minuto. Frecker comenta que el algoritmo SLP tarda 31 minutos en resolver el problema, pero no dice el ordenador utilizado (1999).

Se aprecia que los diseños óptimos obtenidos mediante el método de crecimiento propuesto son mejores que los obtenidos mediante el método de Frecker, y, probablemente, el coste computacional sea significativamente inferior.

Conclusiones y trabajos futuros

8.1 INTRODUCCIÓN

En este capítulo se hace un resumen del trabajo de investigación realizado y descrito en esta tesis, se exponen las conclusiones del mismo y se proponen trabajos futuros como continuación de la línea de investigación seguida.

8.2 TRABAJOS REALIZADOS

Durante el desarrollo de esta tesis se han realizado los siguientes trabajos:

- Revisión bibliográfica de artículos de revistas, libros y congresos, relacionados con el diseño óptimo de topología y geometría de estructuras articuladas, centrada, fundamentalmente, en los últimos 10 años.
- Se han analizado, e implementado, los siguientes métodos del universo estructural: *stress-ratio*, de la *compliance*, de Pedersen (1993), de Achtziger (1999a y 1999b), y de búsqueda exhaustiva.
- Se ha implementado un generador de universos estructurales, que utiliza, para la generación de los nudos en el dominio de diseño, generadores de malla mapeados similares a los de los programas de elementos finitos.
- Se han comparado los métodos del universo estructural implementados, teniendo en cuenta, principalmente, el tipo de restricciones admitidas y el coste computacional.
- Se han analizado los siguientes métodos de crecimiento: de Rule (1994), de McKeown (1998), y de Bojczuk y Mróz (1998 y 1999).
- Se ha propuesto, e implementado, un método novedoso, basado en técnicas de crecimiento, para el diseño óptimo simultáneo de topología y geometría de estructuras articuladas, sometidas a un estado de cargas y con restricciones de diseño de tensión, pandeo de elementos, esbeltez y área mínima. El método consiste en un proceso iterativo a dos niveles: un primer nivel en el que se añade un nuevo nudo a la topología anterior, y se realiza una optimización de topología

con métodos del universo estructural, y un segundo nivel en el que se realiza una optimización de geometría. Este método proporciona una secuencia de diseños óptimos, en la que disminuye la función objetivo y aumenta el tamaño de la estructura al ir aumentando el número de nudos y barras de la misma.

- Se han resuelto varios ejemplos para validar el método de crecimiento propuesto, así como para compararlo con los métodos del universo estructural implementados.
- Se han resuelto algunos ejemplos de diseño óptimo de topología en otros campos, tales como el método de bielas y tirantes para estructuras de hormigón armado, y la optimización de topología y geometría de mecanismos.
- Todas las implementaciones se han agrupado en una aplicación informática propia (TTO), desarrollada en el lenguaje de programación orientado a objetos C++. La aplicación también analiza problemas de elasticidad plana y de campos en régimen permanente, con vistas a una futura ampliación a la optimización de topología de problemas continuos.

8.3 CONCLUSIONES

Del trabajo desarrollado en esta tesis se extraen las siguientes conclusiones:

- A pesar de la complejidad del diseño óptimo simultáneo de topología y geometría de estructuras articuladas, se ha obtenido un método de optimización suficientemente robusto para las aplicaciones consideradas.
- En el método propuesto no es necesario definir una topología inicial, es suficiente con definir las condiciones de contorno de desplazamiento, las cargas y las propiedades del material.
- Con el método de crecimiento propuesto se obtienen diseños óptimos con menos nudos, y sin cruces de barras, que con los métodos del universo estructural. Esto supone estructuras más adecuadas para su construcción.
- El método propuesto es más adecuado para el diseño óptimo de topología que los métodos del universo estructural, ya que, para mejorar un diseño óptimo, basta con añadirle nuevos nudos, mientras que, en los métodos del universo estructural hay que definir un nuevo universo estructural y reiniciar el proceso de optimización.
- El coste computacional del método propuesto es inferior al coste de los métodos del universo estructural.
- Para la definición de los nuevos nudos en el crecimiento de la topología, los mejores resultados se obtienen con el método basado en las direcciones principales de tensión.

- Para la optimización de topología en cada iteración, los mejores resultados, cuando sólo se consideran restricciones de tensión, se obtienen con el método de la *compliance*.

8.4 TRABAJOS FUTUROS

Como continuación del trabajo desarrollado en esta tesis, se proponen las siguientes líneas de trabajos futuros:

- Buscar otras estrategias para añadir nuevos nudos, sobre todo para problemas con restricciones de pandeo de elementos y para otros campos de aplicación del método propuesto.
- Considerar otros tipos de restricciones (desplazamiento, estabilidad global, etc.).
- Considerar varios estados de cargas.
- Ampliar a estructuras tridimensionales.
- Analizar en profundidad la estructura matemática del problema de diseño óptimo abordado, para encontrar bases matemáticas que justifiquen y aumenten la robustez del método de optimización propuesto.
- Aplicar el método a otros tipos de problemas de optimización de topología.

Referencias bibliográficas

- Achtziger, W. (1993). "Minimax compliance truss topology subject to multiple loading." *Topology Design of Structures*, Bendsoe, M.P., and Mota, C.A., ed., Kluwer Academic Publishers, Dordrecht, 43-54.
- Achtziger, W. (1996). "Truss topology optimization including bar properties different for tension and compression." *Structural Optimization*, 12, 63-74.
- Achtziger, W. (1999a). "Local stability of trusses in the context of topology optimization. Part I: Exact modelling." *Structural Optimization*, 17, 235-246.
- Achtziger, W. (1999b). "Local stability of trusses in the context of topology optimization. Part II: A numerical approach." *Structural Optimization*, 17, 247-258.
- Adebar, P.E., Kuchma, D.A., and Collins, M.P. (1990) "Strut-and-tie models for the design of pile caps: an experimental study." *ACI Structural Journal*, 87 (1), 81-92.
- Adeli, H. (1994). *Advances in design optimization*, Chapman & Hall, London.
- Adeli, H., and Park, H.S. (1995). "Optimization of space structures by neural dynamics." *Neural Networks*, 8 (5), 769-781.
- Ali, M.A., and White, R.N. (2001). "Automatic generation of truss model for optimal design of reinforced concrete structures." *ACI Structural Journal*, 98 (4), 431-442.
- Azid, I.A., Kwan, A.S.K., and Seetharamu, K.N. (2002). "An evolutionary approach for layout optimization of a three-dimensional truss." *Structural Optimization*, 24, 333-337.
- Beckers, M., and Fleury, C. (1997). "A primal-dual approach in truss topology optimization." *Computers & Structures*, 64 (1-4), 77-88.
- Belegundu, A.D. (1982). "A study of mathematical programming methods for structural optimization." M. S. Thesis, University of Iowa.
- Bendsoe, M.P., and Kikuchi, N. (1992). "Topology and layout optimization of discrete structures." *Structural optimization: status and promise*, Kamat, M.P., ed., American Institute of Aeronautics and Astronautics, Inc., Washington, 517-547.

- Bendsoe, M.P., and Ben-Tal, A. (1993). "Truss topology optimization by a displacements based optimality criterion approach." *Topology Design of Structures*, Rozvany, G.I.N., ed., Kluwer Academic Publishers, Dordrecht, 139-155.
- Bendsoe, M.P. (1995). *Optimization of structural topology, shape, and material*, Springer, Berlin.
- Bendsoe, M.P., and Sigmund, O. (2003). *Topology optimization. Theory, methods and applications*, Springer, Berlin.
- Bennage, W.A., and Dhingra, A.K. (1995). "Optimization of truss topology using tabu search." *International Journal for Numerical Methods in Engineering*, 38, 4035-4052.
- Ben-Tal, A., Kocvara, M., and Zowe, J. (1993a). "Two nonsmooth approaches to simultaneous geometry and topology design of trusses." *Topology Design of Structures*, Bendsoe, M.P., and Mota, C.A., ed., Kluwer Academic Publishers, Dordrecht, 31-42.
- Ben-Tal, A., and Nemirovskii, A. (1993b). "An interior point algorithm for truss topology design." *Topology Design of Structures*, Bendsoe, M.P., and Mota, C.A., ed., Kluwer Academic Publishers, Dordrecht, 55-69.
- Birker, T. (1995). "Generalized Michell structures – exact least-weight truss layouts for combined stress and displacement constraints: Part II – Analytical solutions within a two-bar topology." *Structural Optimization*, 9, 214-219.
- Bojczuk, D., and Mróz, Z. (1998). "Optimal design of trusses with account for topology variation." *Mechanics of Structures and Machines*, 26 (1), 21-40.
- Bojczuk, D., and Mróz, Z. (1999). "Optimal topology and configuration design of trusses with stress and buckling constraints." *Structural Optimization*, 17, 25-35.
- Borland® (2000). *C++ Builder*, Scott Valley, California.
- Chai, S., Shi, L.S., and Sun, H.C. (1999). "An application of relative difference quotient algorithm to topology optimization of truss structures with discrete variables." *Structural Optimization*, 18, 48-55.
- Cheng, G. (1995). "Some aspects of truss topology optimization." *Structural Optimization*, 10, 173-179.
- Cheng, G.D., and Guo, X. (1997). "ε-relaxed approach in structural topology optimization." *Structural Optimization*, 13, 258-266.
- Cox, H.L. (1965). *The design of structures for least weight*. Pergamon Press, Oxford.
- Díaz, A.R., and Belding, B. (1993). "On optimum truss layout by a homogeneization method." *Journal of Mechanical Design*, 115, 367-373.
- Farkas, J., Jármai, K., Achtziger, W., and Rozvany, G.I.N. (2000). "Local stability of trusses in the context of topology optimization, Parts I and II." *Structural Optimization*, 19, 249-251.

- Frecker, M., Kikuchi, N., and Kota, S. (1997). "Topology optimization of compliant mechanisms with multiple outputs." *Structural Optimization*, 17, 269-278.
- Gellatly, R.M., and Berke, L. (1973). *Optimum structural design: theory and applications*. Gallager and Zienkiewicz Eds., New York.
- Gibbs, N.E., Poole, W.G., and Stockmeyer, P.K. (1976). "An algorithm for reducing the bandwidth and profile of a sparse matrix." *SIAM J. Numer. Anal.*, 13, 236-250.
- Gill, P., Murray, W., and Wright, M. (1981). *Practical optimization*. Ed. Academic Press, Orlando.
- Han, S.P. (1977). "A globally convergent method for nonlinear programming." *J. Optimization Theory Appl.*, 22, 297-309.
- Harasaki, H., and Arora, J.S. (2001). "A new class of evolutionary methods based on the concept of transferred force for structural design." *Structural Optimization*, 22, 35-56.
- Harasaki, H., and Arora, J.S. (2002). "Topology design based on transferred and potential transferred forces." *Structural Optimization*, 23, 372-381.
- Hemp, W.S. (1973). *Optimum structures*. Clarendon Press, Oxford.
- IMSL® (1997). *Visual Numerics, Inc.*, Houston, Texas.
- Kawamura, H., Osada, S., and Ohmori, H. (2000). "Genetic creation of 3D-dimensional truss structures." *Journal of the International Association for Shell and Spatial Structures*, 41 (134), 163-175.
- Kawamura, H., Ohmori, H., and Kito, N. (2002). "Truss topology optimization by a modified genetic algorithm." *Structural Optimization*, 23, 467-472.
- Kirsch, U. (1993a). "Fundamental properties of optimal topologies." *Topology Design of Structures*, Bendsoe, M.P., and Mota, C.A., ed., Kluwer Academic Publishers, Dordrecht, 3-18.
- Kirsch, U., and Rozvany, G.I.N. (1993b). "Design considerations in the optimization of structural topologies." *Topology Design of Structures*, Rozvany, G.I.N., ed., Kluwer Academic Publishers, Dordrecht, 121-138.
- Kirsch, U. (1996). "Integration of reduction and expansion processes in layout optimization." *Structural Optimization*, 11, 13-18.
- Kocvara, M. (1997). "Topology optimization with displacement constraints: a bilevel programming approach." *Structural Optimization*, 14, 256-263.
- Lewinski, T., Zhou M., and Rozvany G.I.N. (1994a). "Extended exact solutions for least-weight truss layouts – Part I: Cantilever with a horizontal axis of symmetry." *International Journal of Mechanical Sciences*, 36 (5), 375-398.
- Lewinski, T., Zhou M., and Rozvany G.I.N. (1994b). "Extended exact least-weight truss layouts – Part II: Unsymmetric cantilevers." *International Journal of Mechanical Sciences*, 36 (5), 399-419.

- Martí, P., Company, P.P., y Sanchis, M. (1985). "DISSENY. Un sistema interactivo para el diseño de estructuras basado en técnicas de optimización." *Anales de Ingeniería Mecánica*, 3, 285-290.
- Martínez, P., y Martí, P. (2002). "Un algoritmo de crecimiento para el diseño óptimo de topología y geometría de estructuras articuladas." *V Congreso Métodos Numéricos en Ingeniería*, Madrid, 379.
- Martínez, P. (2003). *Manual de usuario del programa TTO*, Cartagena.
- McKeown, J.J. (1998). "Growing optimal pin-jointed frames." *Structural Optimization*, 15, 92-100.
- Michell, A.G.M. (1904). "The limits of economy of material in framed structures". *Phil. Mag.*, 8, 589-597.
- Nakanishi, Y., and Nakagiri, S. (1996). "Optimization of truss topology using boundary cycle (derivation of design variables to avoid inexpedient structure)." *JSME International Journal*, 39 (3), 415-421.
- Oberndorfer, J.M., Achtziger, W., and Hörnlein, H.R.E.M. (1996). "Two approaches for truss topology optimization: a comparison for practical use." *Structural Optimization*, 11, 137-144.
- Ohsaki, M. (1995). "Genetic algorithm for topology optimization of trusses." *Computers & Structures*, 57 (2), 219-225.
- Ohsaki, M. (1998). "Simultaneous optimization of topology and geometry of a regular plane truss." *Computers & Structures*, 66 (1), 69-77.
- OpenGL® (1993). *Silicon Graphics, Inc.*, Mountain View, California.
- Pedersen, P. (1993). "Topology optimization of three-dimensional trusses." *Topology Design of Structures*, Bendsoe, M.P., and Mota, C.A., ed., Kluwer Academic Publishers, Dordrecht, 19-30.
- Pedersen, P. (1994). "Modified simplex optimization program." *Communications in Numerical Methods in Engineering*, 10, 303-312.
- Powell, M.J.D. (1978). "A fast algorithm for nonlinear constrained optimization calculations." *Lecture Notes in Mathematics*, 630.
- Pshenichny, B.N. (1970). "Algorithms for the general problem of mathematical programming." *Kibernetika*, 5, 120-125.
- Rajan, S.D., and Bhatti, M.A. (1986). "A computer-aided structural analysis and dynamic design language, Part II. Database management system." *Computers and Structures*, 22 (2), 205-212.
- Rajan, S.D. (1995). "Sizing, shape and topology design optimization of trusses using genetic algorithm." *Journal of Structural Engineering*, 121 (10), 1480-1487.

- Rajeev, S., and Krishnamoorthy, C.S. (1997). "Genetic algorithms-based methodologies for design optimization of trusses." *Journal of Structural Engineering*, 123 (3), 350-358.
- Reddy, G., and Cagan, J. (1995). "An improved shape annealing algorithm for truss topology generation." *Journal of Mechanical Design*, 117, 315-321.
- Rosko, P. (1995). "Three-dimensional topology design of structures using crystal models." *Computers & Structures*, 55 (6), 1077-1083.
- Rozvany, G.I.N., and Birker, T. (1994). "On singular topologies in exact layout optimization." *Structural Optimization*, 8, 228-235.
- Rozvany, G.I.N., and Birker, T. (1995). "Generalized Michell structures – exact least-weight truss layouts for combined stress and displacement constraints: Part I – General theory for plane trusses." *Structural Optimization*, 9, 178-188.
- Rozvany, G.I.N. (1996). "Difficulties in truss topology optimization with stress, local buckling and system stability constraints." *Structural Optimization*, 11, 213-217.
- Rozvany, G.I.N. (1998). "Exact analytical solutions for some popular benchmark problems in topology optimization." *Structural Optimization*, 15, 42-48.
- Rozvany, G.I.N. (2001a). "Topology optimization in structural mechanics." *Structural Optimization*, 21, 89.
- Rozvany, G.I.N. (2001b). "Aims, scope, methods, history and unified terminology of computer-aided topology optimization in structural mechanics." *Structural Optimization*, 21, 90-108.
- Rozvany, G.I.N. (2001c). "Stress ratio and compliance based methods in topology optimization – a critical review." *Structural Optimization*, 21, 109-119.
- Rule, W.K. (1994). "Automatic truss design by optimized growth." *Journal of Structural Engineering*, 120 (10), 3063-3070.
- Sankaranarayanan, S., Haftka, R.T., and Kapania, R.K. (1993). "Truss topology optimization with stress and displacement constraints." *Topology Design of Structures*, Bendsoe, M.P., and Mota, C.A., ed., Kluwer Academic Publishers, Dordrecht, 71-78.
- Sankaranarayanan, S., Haftka, R.T., and Kapania, R.K. (1994). "Truss topology optimization with simultaneous analysis and design." *AIAA Journal*, 32 (2), 420-424.
- Schittkowski, K. (1983). "On the convergence of a sequential quadratic programming method with an augmented lagrangian line search function." *Math. Operationsforsch u Statist. Ser. Optimization*, 14 (2), 197-216.
- Shield, R.T., and Prager, W. (1970). "Optimal structural design for given deflection." *J. Appl. Math. Phys.*, 21, 513-523.
- Smith, O. da Silva (1996). "An interactive system for truss topology design." *Advances in Engineering Software*, 27, 167-178.

- Smith, O. da Silva (1997). "Topology optimization of trusses with local stability constraints and multiple loading conditions – a heuristic approach." *Structural Optimization*, 13, 155-166.
- Wilson, R.B. (1963). "A simplicial algorithm for concave programming." Ph. D. Thesis, Harvard University, Boston.
- Zhou, M., and Rozvany, G.I.N. (1992). "DCOC: an optimality criteria method for large systems. Part I: theory." *Structural Optimization*, 5, 12-25.
- Zhou, M., and Rozvany, G.I.N. (1993). "DCOC: an optimality criteria method for large systems. Part II: algorithm." *Structural Optimization*, 6, 250-262.
- Zhou, M., and Haftka, R.T. (1995). "A comparison of optimality criteria methods for stress and displacement constraints." *Computer methods in applied mechanics and engineering*, 124, 253-271.
- Zhou, M. (1996a). "Difficulties in truss topology optimization with stress and local buckling constraints." *Structural Optimization*, 11, 134-136.
- Zhou, M., and Rozvany, G.I.N. (1996b). "An improved approximation technique for the DCOC method of sizing optimization." *Computers & Structures*, 60 (5), 763-769.
- Zhou, M., and Rozvany, G.I.N. (2001a). "On the validity of ESO type methods in topology optimization." *Structural Optimization*, 21, 80-83.