

Optimización en el Diseño de Torres

TESIS DOCTORAL
por **Salvador Merino Córdoba**



Directores : **Dr. Manuel Ojeda Aciego** (Universidad de Málaga)
T.U. del Dpto. de Matemática Aplicada
Dr. Isidro Ladrón de Guevara López (Universidad de Málaga)
T.U. del Dpto. de Expresión Gráfica, Diseño y Proyectos



UNIVERSIDAD
DE MÁLAGA

AUTOR: Salvador Merino Córdoba

 <http://orcid.org/0000-0002-8361-1047>

EDITA: Publicaciones y Divulgación Científica. Universidad de Málaga



Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-SinObraDerivada 4.0 Internacional:

<http://creativecommons.org/licenses/by-nc-nd/4.0/legalcode>

Cualquier parte de esta obra se puede reproducir sin autorización pero con el reconocimiento y atribución de los autores.

No se puede hacer uso comercial de la obra y no se puede alterar, transformar o hacer obras derivadas.

Esta Tesis Doctoral está depositada en el Repositorio Institucional de la Universidad de Málaga (RIUMA): riuma.uma.es



D. Isidro Ladrón de Guevara López, Profesor Titular de Universidad del Dpto. de Expresión Gráfica, Diseño y Proyectos de la Escuela Universitaria Politécnica de la Universidad de Málaga y D. Manuel Ojeda Aciego, Profesor Titular de Universidad del Dpto. de Matemática Aplicada de la Escuela Técnica Superior de Ingeniería Informática de la Universidad de Málaga,

CERTIFICAN:

Que D. Salvador Merino Córdoba, Licenciado de Grado en Ciencias (Matemáticas), ha realizado en el Dpto. de Expresión Gráfica, Diseño y Proyectos de la Escuela Universitaria Politécnica de la Universidad de Málaga, bajo nuestra dirección, el trabajo de investigación correspondiente a su Tesis Doctoral titulado:

Optimización en el Diseño de Torres

Revisado el presente trabajo, estimamos que puede ser presentado al tribunal que ha de juzgarlo. Y para que conste a efectos de lo establecido en el artículo octavo del Real Decreto 778/1998, autorizamos la presentación de este trabajo en la Universidad de Málaga.

Málaga, a 26 de Junio de 2003

Dr. Isidro Ladrón de Guevara López

Dr. Manuel Ojeda Aciego



TESIS DOCTORAL

OPTIMIZACIÓN EN EL DISEÑO DE TORRES

Autor : Salvador Merino Córdoba

Profesor de la Escuela Técnica de Ingenieros de Telecomunicaciones

Departamento de Matemática Aplicada

Universidad de Málaga

*Esta Tesis está dedicada a todos aquellos que entregan su vida
y sus conocimientos para mejorar nuestro Mundo*

Uno no puede escapar a la sensación de que esas fórmulas matemáticas poseen una existencia independiente y una inteligencia propia, que son más listas que nosotros, más listas incluso que sus descubridores, que podemos obtener de ellas más de lo que originalmente había sido puesto en ellas

Heinrich Herz

En relación a las opiniones de los santos acerca de esos asuntos de la naturaleza (geometría) , responderé sencillamente que en teología sólo es válido el peso de la Autoridad, pero en filosofía sólo cuenta el peso de la Razón. Pues, en efecto, santo fue Lactancio, que negó la redondez de la Tierra ; santo fue Agustín, que admitió la redondez pero negó la existencia de las antípodas. Sagrado es el Santo Oficio de nuestros días, que admite la pequeñez de la Tierra pero niega su movimiento : pero para mí, más sagrado que todo eso es la Verdad, cuando yo, con todo el respeto a los doctores de la Iglesia, demuestro a partir de la filosofía que la Tierra es redonda, habitada en toda su redondez por antípodas, de una pequeñez casi insignificante, y que se mueve velozmente entre las estrellas.

Discurso de Johannes Kepler en su Introducción a la Nueva Astronomía

INDICE DE SIGLAS

AENOR	Asociación Española de Normalización y Certificación
AG	Algoritmo Genético
ASCE	American Society of Civil Engineers
BOE	Boletín Oficial del Estado
CJT	Datos pertenecientes a un conjunto de modelos de torres
COM	Datos comunes a todos los modelos de torres
EDS	Every Day Stresses
MEF	Método de los Elementos Finitos
PAR	Datos particulares de un modelo de torres
RAT	Reglamento de Alta Tensión
RBT	Reglamento de Baja tensión
RCE	Reglamento sobre Centrales Eléctricas
TMR	Tiempo Medio de Recuperación
TTE	Torres de Transmisión Eléctrica

INDICE DE CONTENIDOS

1. LA INGENIERÍA DE LAS TORRES	1-1
1.1 OBJETIVO	1-1
1.2 REFERENCIAS HISTÓRICAS.....	1-1
1.3 FASES EN LA CONSTRUCCIÓN DE UNA LÍNEA.....	1-2
1.4 ESTUDIOS PREVIOS	1-5
1.5 ESTADO ACTUAL	1-6
1.6 PROBLEMAS A TRATAR	1-7
1.7 REFERENCIAS BIBLIOGRÁFICAS.....	1-9
2. MINIMIZACIÓN DE DATOS DE ENTRADA AL DISEÑO	2-1
2.1 ESQUEMA CENTRAL	2-2
2.2 DATOS BÁSICOS PARA EL SISTEMA	2-4
2.3 TIPOS DE TORRES Y CÓDIGOS GEOMÉTRICOS	2-6
2.3.1 Codificación de caras en una TTE (COM).....	2-7
2.3.2 Tipos de Paneles (COM)	2-9
2.3.3 Tipos de Rellenos (COM)	2-10
2.3.4 Tipos de Recuadros (CJT)	2-11
2.3.5 Tipos de Crucetas (PAR).....	2-12
2.4 ENTRADA DE DATOS DE LAS TORRES PIRAMIDALES.....	2-12
2.4.1 Datos básicos de un proyecto (COM).....	2-12
2.4.2 Limitaciones numéricas de la estructura (CJT).....	2-13
2.4.3 Geometría de las Caras (CJT).....	2-14
2.4.4 Rellenos de paneles (CJT)	2-15
2.4.5 Niveles de extensión (CJT)	2-15
2.4.6 Rellenos de extensión (CJT)	2-16
2.4.7 Limitaciones de diseño (CJT)	2-17
2.4.8 Escalada y Juntas (CJT).....	2-18
2.4.9 Geometría de Crucetas (PAR).....	2-20
2.4.10 Rellenos de crucetas (PAR)	2-20
2.4.11 Condiciones de Carga (PAR)	2-22
2.5 ENTRADA DE DATOS DE LAS TORRES DE VENTANA O DELTA.....	2-23
2.5.1 Datos de Geometría de la Cabeza (PAR)	2-23
2.5.2 Condiciones de Carga (PAR)	2-24
2.6 ENTRADA DE DATOS DE LAS TORRES CARA DE GATO.....	2-25
2.6.1 Datos de Geometría de la Cabeza (PAR)	2-25
2.6.2 Condiciones de Carga (PAR)	2-26
2.7 ENTRADA DE DATOS DE LAS TORRES ATIRANTADAS.....	2-27
2.8 APORTACIONES REALIZADAS	2-28
2.9 REFERENCIAS BIBLIOGRÁFICAS.....	2-28
3. SISTEMA ESPACIAL DE CÁLCULO.....	3-1
3.1 REPASO HISTÓRICO.....	3-1
3.2 FORMALIZACIÓN DEL MÉTODO DE LOS ELEMENTOS FINITOS.....	3-4
3.2.1 Notación clásica de Timoshenko	3-4
3.2.2 Notación diferencial de Castigliano.....	3-7
3.3 USO DEL MEF EN EL DISEÑO DE TORRES.....	3-8
3.3.1 Numeración de Nudos.....	3-9
3.3.2 Construcción de la Matriz	3-10
3.3.3 Resolución del Sistema de Ecuaciones	3-15
3.3.4 Ejemplo de Gauss por Bloques.....	3-16

3.4	MODELIZACIÓN Y CÁLCULO DE TTE	3-18
3.4.1	<i>Cálculo de Puntos Básicos de la Torre</i>	3-19
3.4.2	<i>Desarrollo Panel a Panel</i>	3-22
3.4.3	<i>Barras de Cálculo y Barras Secundarias</i>	3-23
3.4.4	<i>Compresión de Datos</i>	3-24
3.4.5	<i>Nudos, Barras y Cargas</i>	3-26
3.4.6	<i>Cono de Desplazamientos</i>	3-26
3.4.7	<i>Fuerzas, Reacciones y Precálculo</i>	3-28
3.4.8	<i>Viento sobre la Estructura</i>	3-29
3.4.9	<i>Modificación de Materiales</i>	3-31
3.4.10	<i>Cargas externas</i>	3-31
3.4.11	<i>Salida y Presentación de Datos</i>	3-31
3.5	ESQUEMA	3-33
3.6	APORTACIONES REALIZADAS.....	3-35
3.7	REFERENCIAS BIBLIOGRÁFICAS	3-35
4.	ALGORITMOS GENÉTICOS.....	4-1
4.1	INTRODUCCIÓN.....	4-1
4.2	FORMULACIÓN TRADICIONAL	4-3
4.2.1	<i>Selección</i>	4-3
4.2.2	<i>Cruzamiento</i>	4-5
4.2.3	<i>Mutación</i>	4-6
4.3	SOFTWARE DESARROLLADO	4-7
4.4	APLICACIÓN A LA MINIMIZACIÓN DE ANCHO DE BANDA	4-7
4.4.1	<i>Caras cúbicas</i>	4-8
4.4.2	<i>Caras tetraédricas</i>	4-9
4.4.3	<i>Sección de torre</i>	4-10
4.5	FORMULACIÓN DINÁMICA	4-10
4.5.1	<i>Caras cúbicas</i>	4-11
4.5.2	<i>Caras tetraédricas</i>	4-11
4.5.3	<i>Sección de torre</i>	4-12
4.6	DISEÑO ÓPTIMO DE ESTRUCTURAS.....	4-12
4.7	APORTACIONES REALIZADAS.....	4-14
4.8	REFERENCIAS BIBLIOGRÁFICAS.....	4-14
5.	MINIMIZACIÓN DE DATOS EN FÁBRICA	5-1
5.1	NOMENCLATURA EN LA FABRICACIÓN DE TTE.....	5-2
5.2	TIPOS DE JUNTAS Y DESPLAZAMIENTO SOBRE EL LÍMITE.....	5-2
5.3	DISTANCIAS DE SEGURIDAD.....	5-4
5.4	CÁLCULO DEL DESARROLLO PLANAR DE LA ESTRUCTURA.....	5-6
5.5	CÁLCULO DE LOS GRAMILES EN LOS MONTANTES Y DIAGONALES	5-7
5.6	ESTRUCTURAS DE DATOS.....	5-9
5.6.1	<i>Estructura de un Taladro</i>	5-10
5.6.2	<i>Estructura de un Corte</i>	5-11
5.6.3	<i>Estructura de una Barra</i>	5-12
5.7	MODELOS DE RECUADROS Y CRUCETAS.....	5-15
5.7.1	<i>Modelos de Recuadros</i>	5-15
5.7.2	<i>Modelos de Crucetas</i>	5-20
5.8	COMPRESIÓN DE DATOS.....	5-23
5.9	INTERFERENCIAS ESPACIALES.....	5-23
5.9.1	<i>Proceso algorítmico de análisis espacial de barras</i>	5-25
5.10	CREACIÓN DE PELDAÑOS (ESCALA)	5-28
5.11	CLASIFICACIÓN DE MATERIAL (CÓDIGO DE FABRICACIÓN)	5-28
5.12	LONGITUDES Y SUMA DE TORNILLOS	5-29
5.13	ALGORITMO DE FABRICACIÓN	5-30
5.14	DISEÑO DE CHAPAS (CARTELAS)	5-31
5.14.1	<i>Cartela Rectangular</i>	5-31
5.14.2	<i>Cartela Periférica</i>	5-32
5.15	APORTACIONES REALIZADAS.....	5-34
5.16	REFERENCIAS BIBLIOGRÁFICAS	5-34

6.	ANÁLISIS DE RESIDUO MÍNIMO	6-1
6.1	PLANTEAMIENTO	6-1
6.1.1	<i>Organización industrial</i>	6-2
6.2	MODELIZACIÓN	6-4
6.2.1	<i>Enunciado y restricciones</i>	6-5
6.3	MÉTODO GREEDY	6-6
6.3.1	<i>Almacenamiento óptimo en discos</i>	6-7
6.3.2	<i>Problema de la mochila</i>	6-12
6.4	ALGORITMOS PARA EL PROBLEMA DEL RESIDUO MÍNIMO	6-14
6.4.1	<i>Definiciones previas</i>	6-15
6.4.2	<i>Algoritmos para la resolución del problema de mínimo residuo</i>	6-15
6.4.3	<i>Clasificación de los algoritmos</i>	6-22
6.4.4	<i>Mixtura algorítmica y aplicaciones</i>	6-23
6.5	ANÁLISIS DE LOS ALGORITMOS PROPUESTOS	6-25
6.5.1	<i>Tiempos de ejecución</i>	6-25
6.5.2	<i>Tiempos de ejecución para algoritmos mixtos</i>	6-28
6.5.3	<i>Comparación y elección de algoritmos</i>	6-29
6.6	INTRODUCCIÓN AL ANÁLISIS ESPACIAL	6-32
6.7	LIMITACIONES	6-32
6.8	MODELIZACIÓN MATEMÁTICA	6-33
6.8.1	<i>Definición de rectángulo</i>	6-33
6.8.2	<i>Propiedades de los rectángulos</i>	6-34
6.8.3	<i>Medidas de rectángulos</i>	6-34
6.9	ESTRATEGIAS DE ALMACENAMIENTO	6-35
6.9.1	<i>Tiempo de ejecución del algoritmo espacial</i>	6-37
6.10	EJEMPLO DE APLICACIÓN	6-38
6.11	APORTACIONES REALIZADAS	6-39
6.12	REFERENCIAS BIBLIOGRÁFICAS	6-39
7.	CÁLCULO INVERSO. DIAGRAMAS DE UTILIZACIÓN	7-1
7.1	PARÁMETROS DE ENTORNO	7-2
7.2	DISEÑO DE ECUACIONES LNS	7-3
7.3	FACTORES DE REPERCUSIÓN Y ENTRADA DE DATOS	7-4
7.4	PROCEDIMIENTO DE CÁLCULO	7-6
7.5	REPRESENTACIÓN DE LAS ECUACIONES SOBRE EL DIAGRAMA	7-9
7.6	HIPERRESTRUCTURAS	7-10
7.7	TORRES DE COMUNICACIONES	7-12
7.7.1	<i>Estructuras multiusos</i>	7-13
7.7.2	<i>Fuerzas de viento sobre la estructura</i>	7-13
7.7.3	<i>Fuerzas de viento sobre las antenas</i>	7-15
7.7.4	<i>Hipótesis de cargas a aplicar</i>	7-20
7.7.5	<i>Cálculo de fuerzas sobre la estructura</i>	7-21
7.8	APORTACIONES REALIZADAS	7-29
7.9	REFERENCIAS BIBLIOGRÁFICAS	7-29
8.	APORTACIONES, APLICACIONES Y TRABAJO FUTURO	8-1
8.1	RESUMEN FINAL DE APORTACIONES	8-1
8.2	APLICACIONES DE LA TEORÍA DE TORRES	8-3
8.2.1	<i>Simulación de planta industrial</i>	8-4
8.2.2	<i>Cálculo de Torre Delta</i>	8-5
8.2.3	<i>Método de los Elementos Finitos</i>	8-5
8.2.4	<i>Colocación de sensores contra incendios</i>	8-6
8.2.5	<i>Algoritmos de minimización de residuos</i>	8-6
8.2.6	<i>Torres de ocio</i>	8-7
8.3	PATENTES DESARROLLADAS	8-9
8.4	TRABAJO FUTURO	8-10
8.4.1	<i>Proceso de simulación virtual</i>	8-12
8.4.2	<i>Aplicaciones en costes, tiempos y seguridad</i>	8-12

9.	ANEXO I. TEORÍA ALGORÍTMICA.....	9-1
9.1.1	<i>Eficiencia de los algoritmos.....</i>	<i>9-1</i>
9.1.2	<i>¿Porqué son necesarios algoritmos eficientes?.....</i>	<i>9-3</i>
9.1.3	<i>Análisis de Algoritmos.....</i>	<i>9-4</i>
10.	ANEXO II. NORMATIVA ASCE.....	10-1
10.1	DISEÑO DE PIEZAS	10-1
10.2	MATERIALES USADOS	10-1
10.3	SECCIONES MÍNIMAS.....	10-1
10.4	CÁLCULO DE LA ESBELTEZ	10-1
10.5	PROPIEDADES DE LOS MATERIALES.....	10-2
10.6	COMPRESIÓN PERMISIBLE	10-2
10.7	MIEMBROS A COMPRESIÓN : ANGULARES.....	10-3
10.7.1	<i>Máximo coeficiente w/t</i>	<i>10-3</i>
10.7.2	<i>Esfuerzos permitidos a compresión</i>	<i>10-3</i>
10.7.3	<i>Determinación de F_{cr}.....</i>	<i>10-4</i>
10.7.4	<i>Longitud efectiva.....</i>	<i>10-4</i>
10.8	TENSIÓN PERMISIBLE.....	10-6
10.9	CRÍTICA A LA NORMA ASCE	10-6
11.	ANEXO III. MINIMIZACIÓN DE ANCHO DE BANDA	11-1
11.1	APLICACIÓN A LAS ESTRUCTURAS ESPACIALES.....	11-1
11.1.1	<i>Programación de permutaciones.....</i>	<i>11-2</i>
11.1.2	<i>Análisis de soluciones.....</i>	<i>11-4</i>
11.1.3	<i>Problema mediante pirámides.....</i>	<i>11-4</i>
11.2	APLICACIÓN A LAS TORRES.....	11-8
12.	ANEXO IV. SISTEMA DE GESTIÓN DE MATERIALES.....	12-1
12.1	PRESENTACIÓN.....	12-1
12.2	EL PROGRAMA MATERIAL	12-1
12.2.1	<i>Archivos Necesarios</i>	<i>12-2</i>
12.3	PANTALLA DE TRABAJO	12-3
12.3.1	<i>Opciones del Menú</i>	<i>12-6</i>
12.3.2	<i>Inicio del Programa.....</i>	<i>12-8</i>
12.4	GESTIÓN DE ESQUEMAS DE BARRAS Y CARTELAS.....	12-9
12.4.1	<i>Visualización.....</i>	<i>12-9</i>
12.4.2	<i>Modificación.....</i>	<i>12-11</i>
12.4.3	<i>Adición de elementos</i>	<i>12-13</i>
12.4.4	<i>Eliminar elementos</i>	<i>12-18</i>
12.4.5	<i>Otras modificaciones</i>	<i>12-19</i>
12.5	LISTA DE MATERIALES	12-19
12.6	CONFIGURACIÓN	12-21
12.7	DOCUMENTACIÓN.....	12-23
13.	BIBLIOGRAFÍA	13-1
14.	SOFTWARE UTILIZADO.....	14-1
15.	ÍNDICE ALFABÉTICO.....	15-1
16.	TABLA DE ILUSTRACIONES.....	16-1

1. La Ingeniería de las Torres

1.1 Objetivo

En este trabajo se presenta la formalización de las técnicas de optimización sobre un sistema de cálculo, diseño y fabricación de torres metálicas de transmisión eléctrica y telecomunicaciones. Su desarrollo e implementación combina distintas herramientas matemáticas tales como el método de los elementos finitos, algoritmos de optimización o la resolución de grandes sistemas de ecuaciones, así como las correspondientes técnicas de ingeniería del software.

1.2 Referencias históricas

Para poder entender mejor el campo de trabajo en el que nos moveremos, presentamos un breve recorrido histórico :

Desde que en 1880 Volta construyese la primera pila eléctrica, se observó la enorme importancia que este tipo de energía podría tener para el uso industrial y doméstico. En el siglo XIX, el consumo de electricidad era muy bajo y su transmisión se realizaba a través de conductores de cobre apoyados sobre postes de madera. Con el aumento de la demanda eléctrica y la disminución de nuestros bosques a escala mundial se planteó la necesidad de usar otros materiales en el diseño tanto de los conductores (el cobre experimentó una tremenda subida con su uso en todo tipo de aparatos eléctricos) como de los apoyos que sustentaban a los mismos. Para los primeros se pasó a los conductores de aluminio-acero que usamos actualmente pero, debido a la menor conductividad eléctrica, se necesitan mayores voltajes y cables más gruesos para conseguir una menor pérdida de energía, la cual se produce principalmente en forma de calor (efecto Joule), de ahí que fuera también necesario crear estructuras que tuvieran mayor resistencia al peso de estos nuevos cables.

Los primeros que pensaron en el uso de una estructura metálica para transmisión de electricidad entre las fuentes de producción y los usuarios finales fueron algunas empresas suizas al intentar aprovechar los postes metálicos que sobraban de los

ferrocarriles italianos, a principios del siglo XX. Desde su origen debieron competir con los apoyos construidos en hormigón con los que aún hoy en día coexisten.

Desde el punto de vista económico, el apoyo de hormigón tiene la ventaja de ser más barato, pero su duración en el tiempo es menor y su peso bastante mayor, lo que lo hace más difícil de colocar en zonas montañosas (p.ej. España). Por ello se ha tendido a la construcción de torres de barras de acero galvanizadas¹ y atornilladas entre sí, que permiten ser transportadas por piezas y montadas in situ.

En la actualidad es por todos conocida la importancia estratégica que tiene para cada país la producción, el transporte y la distribución de la energía eléctrica. Sin ir más lejos, en los últimos años hemos tenido en España una Ley de Ordenación del Sistema Eléctrico Nacional (1994) [BO94] y un Protocolo Eléctrico (1997) [BO97]. De la primera surgió la unificación y nacionalización del transporte eléctrico por encima de 200 Kilovoltios con la creación de la compañía Red Eléctrica de España y, de la segunda, la diversificación de los operadores de distribución de energía en cada región y la mayor fusión de compañías eléctricas de nuestra historia en la compañía Endesa. Esto ha hecho posible que las empresas de este sector estén actualmente muy interesadas en todos los procedimientos de mejora de la calidad y disminución de costes y por ello han contribuido de forma activa en el desarrollo de este proyecto.

1.3 Fases en la construcción de una línea

Para entender de forma clara en qué puntos queremos mejorar la producción eléctrica, vamos a recorrer cada una de las fases en la construcción de una línea de transmisión y hacer hincapié en aquellas que hemos tratado :

Pre-proyecto de una línea : en primer lugar son normalmente organismos públicos (Estado, Ayuntamientos, Diputaciones,...) por parte de los consumidores, o compañías hidroeléctricas por parte de los productores, los que se plantean la necesidad de construir una línea eléctrica. A ellos los llamaremos clientes. A partir de aquí se plantea también la forma de financiar la misma, mediante aportaciones públicas o privadas, fondos de ayuda al desarrollo, Banco Mundial, etc. Con ello se realiza unas especificaciones técnicas por un consulting especializado o por un

¹ Recubierto de una capa protectora de zinc

organismo técnico independiente y se marca unos plazos (habitualmente cortos) para la apertura de un concurso público (normalmente a nivel internacional).

Realización del proyecto para concurso : aquí empiezan los resultados de esta tesis (*primera parte, diseño y cálculo de las estructuras*). Para competir en estos concursos es necesario tomar el tipo de torres elegido por el cliente [Ki02], parametrizarla e introducirla en un modelo matemático, junto con sus diferentes alturas [Gi95], cargas de cables, condiciones de viento, nieve, terremotos, etc., e ir buscando aquella estructura que, cumpliendo con todos los requisitos técnicos, tenga un mínimo de peso o un menor coste en el material necesario para su construcción (ambas cosas suelen ir en paralelo) [IM90].

Nuestra aportación para esta fase ha sido el desarrollo de utilidades que, a partir del mínimo número de datos necesario, puedan ayudar en el diseño y cálculo de las estructuras y tengan flexibilidad para modificar y recalcular los resultados de una forma rápida y precisa. Entre estas utilidades cabe destacar el hecho de haber realizado una implementación específica del método de los elementos finitos, en lugar de hacer uso de los paquetes de cálculo comerciales, para así poder acelerar los procesos de cálculo en el campo de las torres.

Concesión de obra : aquella empresa o grupo adjudicatario del concurso deberá ahora realizar los planos de construcción, detalles de fábrica, diseño de cada pieza, minimización del material a usar y plan de erección de las torres y fundaciones en el suelo (*segunda parte, fabricación*). A partir de todo ello, construirá un prototipo elegido por el cliente de cada una de las diferentes torres que entren en juego.

Respecto a esta fase, nuestra aportación ha consistido en crear tanto las bibliotecas de técnicas geométricas necesarias (cálculo de interferencias, distancias, diseño de sólidos en el espacio, estudio de cortes, etc.) como el sistema que a partir de éstas define los planos de fabricación, estudia cada una de las piezas que componen las torres, clasifica estos materiales, ve cómo poder realizar un mejor aprovechamiento de los mismos y expone todos estos resultados para facilitar en lo posible la fabricación de las estructuras. En este sentido hemos de resaltar el hecho de haber creado una base de datos única donde cada registro es una pieza diferente que contiene todas sus referencias estructurales (panel, sección, nomenclatura, coordenadas 2D y 3D), características físicas (tipo de angular o chapa, calidad, dimensiones, etc.) y

manipulaciones posibles (taladros, doblados, soldaduras, etc.) lo que nos permite, desde un punto de vista práctico:

1. Desarrollar directamente los planos de fabricación
2. Crear las ordenes de trabajo de cada pieza y las listas de materiales de los cuerpos o secciones de la torre
3. El estudio de las posibles interferencias espaciales entre las diferentes piezas durante el montaje.
4. Minimizar los residuos de materiales en la fábrica.

Ensayo de prototipos : cada uno de los prototipos se lleva a la estación de ensayos (*tercera parte, estación de ensayos y resistencia de torres*). En ella se comprueba que su resistencia cumple el 100% de lo exigido durante aproximadamente 30 segundos (lo cual suele estar muy por encima de las condiciones normales de la torre en el campo conocidas como *EDS = Every Day Stresses*). Una vez probado, se continúa aplicando carga de forma proporcional en cada punto de la torre. En caso de superar el 115% de lo exigido se suele rechazar la estructura por estar sobredimensionada [Na00]. Si cae dentro de dicho intervalo se da por válida y se aprueban los diferentes prototipos [AD00].

Debido a los enormes costes que tiene el ensayo de prototipos a escala real, esta ha sido una de las fases que constituirá una línea de trabajo futuro prioritaria. Tengamos en cuenta que el estudio espacial mediante computadoras del montaje de las estructuras, su resistencia local y global, el estudio de incidencias y debilidades y su comportamiento ante fenómenos ajenos al estudio formal de líneas (terremotos, vibraciones de industrias cercanas o carreteras, corrosión salina o mineral, impacto ambiental, etc.) nos dan una mayor seguridad de cálculo y aminora el número de errores y accidentes posteriores. Las repercusiones que tendrá la generación de un simulador de estaciones de ensayo se ha convertido en un nuevo reto a desarrollar a partir de ahora y contará con la colaboración de ilustres investigadores para su consecución y las más modernas herramientas CAE (*Computer Aided Engineering*).

Realización de la obra y tendido de la línea : ya superados todos los controles se pasa a construir las torres en serie en la fábrica, se realiza el posicionamiento de las mismas en el campo siguiendo el perfil del terreno, y se izan los cables por donde posteriormente se transmitirá la electricidad.

Aprovechamiento de torres ya ensayadas : en muchos casos es posible que los cables necesarios para una línea y la zona por la que discurre se asemejen a las condiciones de algunas otras torres ya fabricadas. En estos casos es conveniente comprobar si las mismas pueden resistir en el nuevo entorno (*cuarta y última fase de la tesis, cálculo inverso*). Con ello nos ahorramos todos los costes de ensayos y los tiempos requeridos para los mismos [Mo88].

Esta fase hizo necesario el desarrollo de técnicas informáticas que permitieran verificar las estructuras ya existentes. Los resultados que existían en este campo eran bastante teóricos y hacían excesivo uso del tanteo y de la experiencia a la hora de elegir las piezas que pudieran poner en peligro una antigua estructura en un nuevo entorno. Para ello, y haciendo uso de las herramientas que ya poseíamos, se pudo generar el estudio de todas y cada una de las barras de la torre y así, mediante un algoritmo de búsqueda encontrar aquellas condiciones y piezas que limitarían la utilización de las estructuras [IS01].

1.4 Estudios previos

A la hora de buscar bibliografía, programas o textos especializados sobre el diseño de estas torres metálicas, nos encontramos con que la mayoría hacen referencia al cálculo de los conductores [Mo88], las normativas sobre cálculo de fuerzas [AS88] y colocación de apoyos en el campo (p.ej. Reglamento de Alta Tensión o más comúnmente RAT) [RAT, RBT, RCE], cálculos eléctricos, presentación de proyectos, uso de modelos de torres del mercado [IM90], etc. Como podemos observar, tratan sobre el entorno de las torres pero en pocos casos hacen referencia al diseño de las mismas [Be73], el método recomendable de cálculo estructural a usar [Ru66], la optimización del peso de la estructura [IS01], la modelización, los mejores detalles de fabricación [IS97], la presentación de los planos, el control de la producción, la minimización de residuos [Me91, MT90], las normas de montaje, etc., es decir, todo lo que supone la erección de las torres en el campo y el montaje de la línea eléctrica. Por ello nos propusimos crear un único sistema que, secuencialmente, nos permitiera, partiendo de las especificaciones técnicas de un proyecto, llevar a cabo todas las actividades anteriores, terminando con el ensayo de las estructuras y su construcción final.

El camino no ha sido fácil. En principio nos encontramos con la dificultad de que cada diseñador seguía unas pautas propias, en base a su experiencia y a su nivel de conocimientos, y rechazaba la idea de crear un sistema experto² que contuviese los modelos más generalizados de uso. Por otra parte se trata de un campo de la ingeniería donde existe una enorme competencia entre las empresas para la consecución de un proyecto. Tengamos en cuenta que la creación de una estructura bien diseñada y ensayada conduce a la fabricación en serie de todos los apoyos que conforman una línea de transmisión eléctrica, lo cual equivale a ingentes cantidades de dinero.

Además de ello, contábamos con la gran diversidad de procedimientos que hay en este campo :

- *Cálculo* : Unos diseñadores utilizan unos métodos de cálculo diferentes a otros, aunque se impone hoy en día el uso del *método de los elementos finitos* [Oñ92, ZT94]. Los programas de elementos finitos comerciales tienen la limitación de no permitir una manipulación sencilla de los datos en caso de necesitar un cambio de dimensiones físicas sobre el modelo original (provocan la necesidad de recalcular todas las coordenadas espaciales) de ahí que se haya llevado a cabo su implementación [AR86, IS01].
- *Diseño* : Respecto al diseño de planos para fábrica cada cual usa un sistema CAD (*Computer Aided Design*) diferente que, además, no se puede interconectar de forma natural con los datos previos de cálculo.
- *Fabricación*: Por último, a partir del diseño se deben producir las planillas de materiales donde cada barra, chapa, o tornillo debe ser representado independientemente, para su fabricación y posterior ensamblaje. Tampoco existía una conexión natural entre el plano y la pieza y todo ello implicaba repeticiones innecesarias en la representación de los datos.

1.5 Estado actual

Debido a las dificultades ya expuestas, nuestros planteamientos, desde un principio, han sido siempre el máximo aprovechamiento de los datos y la anulación de

² Los sistemas expertos son programas que reproducen el proceso intelectual de un experto humano en un campo particular, pudiendo mejorar su productividad, ahorrar tiempo y dinero, conservar sus valiosos conocimientos y difundirlos más fácilmente.

duplicidades, como indican las *normas del correcto diseño del software (Normas 12000 AENOR)*. Como resultado se ha obtenido el sistema ISI³ de diseño, cálculo y fabricación de torres [IS97].

Aunque al inicio se pensó como una herramienta para torres de transmisión eléctrica, después ha sido utilizada exitosamente para todo tipo de torres (vigilancia de bosques, telefonía móvil, antenas parabólicas de televisión, teleféricos, transmisión de radio, comunicación y radares de aeropuertos, energía eólica, puentes, campanarios, etc.) debido a la similitud intrínseca de sus estructuras [Ca01].

En el presente trabajo vamos a hacer especial hincapié en la formalización de este sistema, sus innovaciones tecnológicas y científicas, los procedimientos matemáticos que contiene y cómo se han ido planteando y venciendo los diferentes interrogantes.

Con el trabajo desarrollado creemos que se da una aportación científica y tecnológica importante para sectores menos desarrollados en la implantación de software industrial, además de haber generalizado el uso de modelos rentables que permitan la expansión de nuestro mercado hacia todos los países del mundo.

De este sistema se han beneficiado todas las compañías eléctricas españolas y algunas extranjeras, encontrándose actualmente en periodo de uso por parte de las mismas. Por ello nos gustaría agradecerles su confianza e interés y las diferentes opiniones realizadas que nos ayudaron a confluír hoy en día en este sistema. También queremos agradecer a la *Estación de Ensayos Europea (CIET-Aenor)* sus aportaciones técnicas y humanas [Na00].

1.6 Problemas a tratar

En el siguiente esquema sintetizamos los problemas que hemos ido encontrando en el desarrollo del presente trabajo y a los que hemos ido dando solución:

³ Siglas de la compañía Ingeniería de Sistemas Interactivos, propietaria del software y con la que ha colaborado el autor desde el año 1988 en la realización y aplicación de dicho sistema.

1) Datos de entrada

- a) Desarrollo del diagrama central de procesos de un sistema experto en torres
- b) Clasificación de los modelos de torres más comúnmente usadas
- c) Codificación de los parámetros necesarios para desarrollar las estructuras

2) Sistema espacial de cálculo

- a) Numeración de nudos para la minimización de ancho de banda en la matriz de rigidez
- b) Elección del método de resolución del sistema de ecuaciones
- c) Representación tridimensional de las estructuras en el ordenador
- d) Codificación de nudos, barras y cargas. Desarrollo por paneles
- e) Desarrollo de algoritmos de compresión de datos superfluos
- f) Análisis numérico de la estabilidad estructural e implementación del mismo
- g) Algoritmo recursivo para el análisis de empujes de viento sobre la estructura
- h) Elección de materiales que aseguren el mínimo peso dentro de la resistencia impuesta
- i) Presentación de datos acorde a las especificaciones de este sector industrial

3) Algoritmos genéticos

- a) Implementación de la metodología de algoritmos genéticos
- b) Aplicación a la minimización del ancho de banda
- c) Aplicación al diseño óptimo de estructuras

4) Minimización de datos en fábrica

- a) Análisis de ubicación de la línea imaginaria de cálculo sobre la pieza real de fabricación
- b) Algoritmos para el cálculo y unificación automática de gramiles
- c) Implementación de las estructuras de barra, chapa, taladro, corte y doblado
- d) Modelización de diseños de fabricación
- e) Estudio de las interferencias espaciales
- f) Clasificación de materiales y creación automática de chapas.

5) Análisis de Residuo Mínimo

- a) Programación lineal del problema de residuos
- b) Análisis de algoritmos, implementación y clasificación

- c) Análisis plano y espacial del problema de residuos
- d) Modelización matemática y desarrollo de algoritmos de minimización

6) Cálculo Inverso y Diagramas de Utilización

- a) Establecimiento de la metodología del cálculo inverso
- b) Concreción de parámetros y formulación de las ecuaciones LNS
- c) Desarrollo de las hiperestructuras
- d) Aplicación de la metodología en el campo de las torres de telecomunicaciones

1.7 Referencias Bibliográficas

[AD00] Atwater, P.; DeHaan, J.; Romero, L. “Utilities Field Test Safety Grounds”. Transmission & Distribution World. Denver. USA. 2000

[AS88] American Society of Civil Engineers. “Guide for design of steel transmission towers”. Edit. ASCE. New York. EE.UU. 1988.

[BO94] BOE 313/1994. “Ley 40/1994 de 30 de diciembre de 1994, de ordenación del sistema eléctrico nacional”. España. 1994

[BO97] BOE 285/1997. “Ley 54/1997 de 28 de Noviembre de 1997, del Sector Eléctrico”. España. 1997

[Ca01] Cadavid, J.D. “Construcción de torres para equipos de conversión de energía eólica”. Revista de Aerodinámica 2001. Dpto. de Ingeniería Mecánica- Universidad de los Andes. Colombia. 2001

[FR84] Ford, B.; Rault, J.C.; Thomasset, F. “Tools, Methods and Languages for scientific and engineering computation”. Elsevier. Oxford. Inglaterra. 1984.

[Gi95] Gipe, P. “Wind turbine tower trends”. John Wiley & Sons USA. 1995

[IM90] IMEDEXSA (Industrias Mecánicas de Extremadura S.A.) . “Torres metálicas para líneas eléctricas de media tensión”. Manual. Cáceres. España. 1990.

[IS97] ISI (Ingeniería de Sistemas Interactivos Ltda.) . “Computer integrated transmission towers system”. Manual. Málaga. España. 1997.

[Ki02] Kipp, V.W. “Tower identification”. Site Management & Technology. Wisconsin. USA. 2002

[Mo88] Moreno Clemente, J. “Cálculo de líneas eléctricas aéreas de alta tensión”. Manual. Málaga. España. 1988.

[MO01] Merino, S. y Ojeda, M. “Preliminares sobre la automatización del diseño de torres”. Congreso ACIIC2001. Cádiz. España. 2001

[Na00] Navarro, J.A. “Estaciones de Ensayo. Historia y Funcionamiento”. Documentación diversa y entrevista personal con el responsable de ensayos durante 30 años del Centro Europeo de Investigación y Ensayo de Torres. Madrid. 2000

[RAT] Reglamento de líneas eléctricas aéreas de Alta Tensión

[RBT] Reglamento electrotécnico para baja tensión

[RCE] Reglamento sobre centrales eléctricas, subestaciones y centros de transformación

2. Minimización de datos de entrada al diseño

Durante muchos años el estudio de diseño de los apoyos para líneas de alta tensión o, más comúnmente, *torres de transmisión eléctrica (TTE)*, ha estado copado por técnicos que, de alguna u otra manera, han tratado de crear los modelos más económicos y de mejor montaje que fueran posible, y así generar nuevas ideas que



Ilustración 2-1. Estación de Ensayos de Arak (Irán)

permitieran disponer de un servicio eléctrico mejor y de menor coste. De estas ideas básicas surgieron los principios de modelizar [Ga98] y realizar el procedimiento más rápido para el cálculo de las TTE y su posterior diseño y montaje.

Al ser este campo uno de los pocos donde los propios modelos o prototipos de las TTE se ensayan a escala real, se presupone una mayor preocupación por la calidad y cuidados en el diseño, de manera que el propio análisis del ensayo nos lleve a conocer mejor y más de cerca los problemas estructurales que se presentan.

De todas las posibles geometrías con las que podemos contar, trataremos en este trabajo de intentar simplificar el campo y así ceñirnos a las más ligeras y económicas (entendiéndose esto

como aquellas que usan el mínimo de material sin disminuir la resistencia) [AA99].

La modelización implica el tratar de convertir una estructura real en un ente matemático definido por localizaciones en el espacio (posteriormente llamadas *nudos*) y relaciones entre ellos (que serán las *barras*) bajo unas ciertas condiciones iniciales (las fuerzas sobre los nudos o *cargas*) [CO98]. La mejor disposición de las posiciones espaciales y la forma de escoger estas relaciones serán las características intrínsecas de lo que consideraremos como un correcto diseño [AH88], siempre que éstas se adecuen a las limitaciones impuestas por las normativas del proyecto (léase como *especificaciones técnicas*).

Además de estas limitaciones geométricas y de seguridad tenemos otras que consideraremos de vital importancia, aunque estén más relacionadas con el campo económico o el estudio de mercado. Nos referimos a la existencia o no de los materiales que necesitaremos para nuestras estructuras y a las características de las fábricas que van a realizar estos trabajos[RM02].

Con todo ello nos disponemos a desarrollar el método para el cálculo y diseño de las torres mediante un *sistema interactivo*. Con este nombre denominamos a todo aquel procedimiento que realiza una labor determinada a través de las preguntas de la computadora y las respuestas del usuario. Al ser la tarea central, en nuestro caso, el entorno de las TTE, requerirá del conocimiento de diversas disciplinas relacionadas, como son :

- Cálculo Vectorial
- Esquemas de proyecto
- Preparación de documentación para concurso
- Realización de planos de fábrica
- Creación de órdenes de trabajo (planillas y listados de materiales)
- Organización de la planta industrial

Todas estas fases necesitarán diferentes modelos, algoritmos y pautas de diseño a los que el usuario debe responder en base a las cuestiones sobre las que le interroga el propio sistema [Ma00]. Presentemos por tanto cual es el esquema central de procesos del sistema de cálculo de torres, la teoría necesaria relativa al análisis algorítmico y la terminología que usaremos en el campo de las TTE.

2.1 Esquema Central

Este sería el esquema de trabajo de un *Sistema Interactivo de Cálculo, Diseño y Fabricación de Torres de Transmisión Eléctrica* [IS97]:

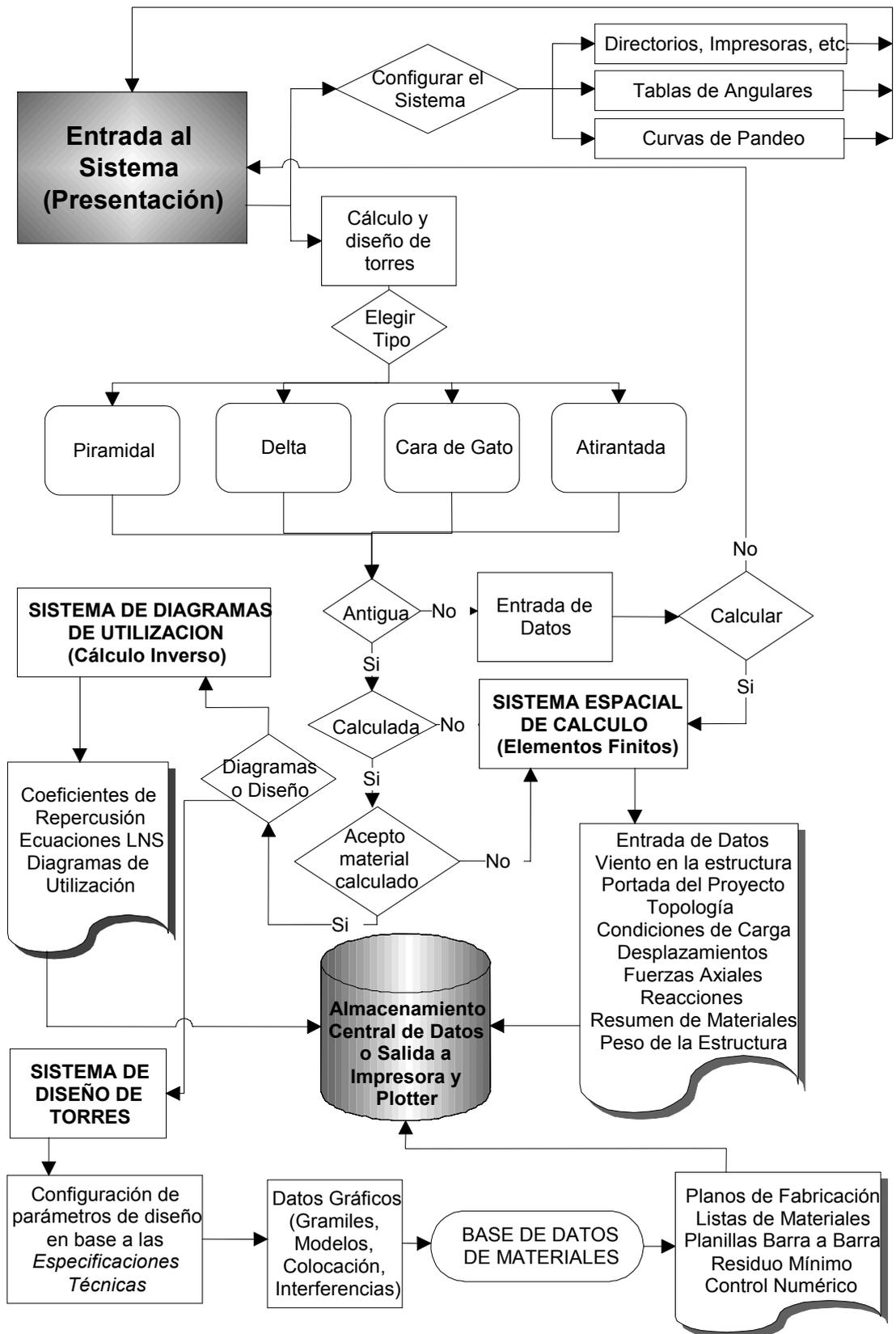


Ilustración 2-2. Diagrama Central del Sistema

2.2 Datos básicos para el sistema

Para el diseño de TTE se hace uso de los perfiles de acero o angulares en L, cuya nomenclatura es *Ala 1* × *Ala 2* × *Espesor* dada habitualmente en milímetros:

Podemos observar que el material se nombra, por ejemplo, $60 \times 60 \times 3$, $50 \times 40 \times$

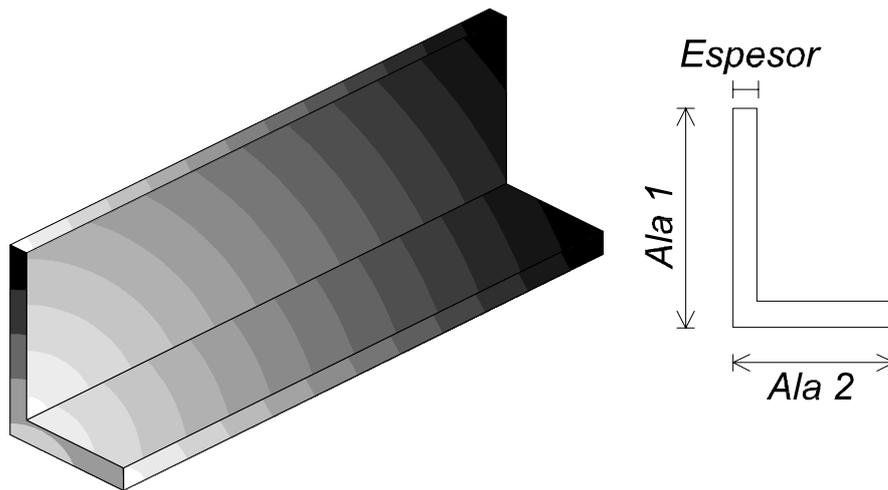


Ilustración 2-3. Nomenclatura de Angular o Perfil en L.

4, etc. En caso de que las alas 1 y 2 sean iguales, se suele escribir sólo una de ellas y tendremos barras de 70×5 , 90×6 , etc [RM02].

Para poder generar todas las listas de materiales correspondientes a cada país y a cada fabricante hemos creado una base de datos tan general como ha sido posible, que albergue los perfiles angulares más comúnmente usados en los distintos países del mundo en este campo de la ingeniería, y sobre ella se realiza la selección de los angulares disponibles para cada proyecto [Ma98]. Este procedimiento nos permite rápidamente contar con un archivo o base de datos local para cada proyecto y así poder comenzar con nuestro cálculo de estructuras. A este proceso lo llamamos “creación de tabla de angulares” y a partir de él queda un archivo del tipo :

	Notacion	Peso	Esp.	Area	R.Min	R.Med	Ala1	Ala2
"	60 * 60 * 4 "	3.70,	0.40,	4.71,	1.83,	1.18,	6.0,	6.0
"	50 * 50 * 5 "	3.77,	0.50,	4.80,	1.51,	0.97,	5.0,	5.0
"	60 * 60 * 5 "	4.57,	0.50,	5.82,	1.82,	1.17,	6.0,	6.0
"	70 * 70 * 5 "	5.37,	0.50,	6.84,	2.14,	1.38,	7.0,	7.0
"	60 * 60 * 6 "	5.42,	0.60,	6.91,	1.82,	1.17,	6.0,	6.0
"	70 * 70 * 6 "	6.38,	0.60,	8.13,	2.13,	1.37,	7.0,	7.0
"	80 * 80 * 6 "	7.34,	0.60,	9.35,	2.44,	1.57,	8.0,	8.0
"	70 * 70 * 7 "	7.38,	0.70,	9.40,	2.12,	1.36,	7.0,	7.0
"	90 * 90 * 6 "	8.31,	0.60,	10.57,	2.75,	1.77,	9.0,	9.0
"	80 * 80 * 7 "	8.49,	0.70,	10.80,	2.44,	1.57,	8.0,	8.0
"	80 * 80 * 8 "	9.63,	0.80,	12.30,	2.43,	1.55,	8.0,	8.0
"	100 * 100 * 6.5"	9.98,	0.65,	12.73,	3.06,	1.97,	10.0,	10.0
"	90 * 90 * 8 "	10.90,	0.80,	13.90,	2.74,	1.76,	9.0,	9.0
"	100 * 100 * 8 "	12.10,	0.80,	15.50,	3.06,	1.96,	10.0,	10.0
"	90 * 90 * 9 "	12.20,	0.90,	15.50,	2.74,	1.76,	9.0,	9.0
"	110 * 110 * 8 "	13.43,	0.80,	17.11,	3.37,	2.14,	11.0,	11.0
"	120 * 120 * 8 "	14.70,	0.80,	18.70,	3.68,	2.37,	12.0,	12.0
"	100 * 100 * 10"	15.00,	1.00,	19.20,	3.04,	1.95,	10.0,	10.0
"	110 * 110 * 10"	16.60,	1.00,	21.20,	3.36,	2.16,	11.0,	11.0
"	120 * 120 * 10"	18.20,	1.00,	23.10,	3.67,	2.36,	12.0,	12.0
"	130 * 130 * 10"	19.79,	1.00,	25.21,	5.08,	2.60,	13.0,	13.0
"	150 * 150 * 10"	23.00,	1.00,	29.30,	4.62,	2.97,	15.0,	15.0

En esta base de datos observamos su ordenación creciente por pesos. Esto asegurará en el cálculo de materiales que la primera selección que se encuentre y que resista a las cargas aplicadas es la que menos pesa.

Junto con la base de datos de materiales se necesita definir la curva de pandeo (*buckling curve*, [Sc02]) que en cada proyecto va a ser usada. En la mayoría de los casos, para TTE, la curva es la recomendada por ASCE (Véase Anexo) [AS88]. Pero en caso de usar cualquier otra (*Omega*, *Dutheil*, etc.) se ha de crear mediante un

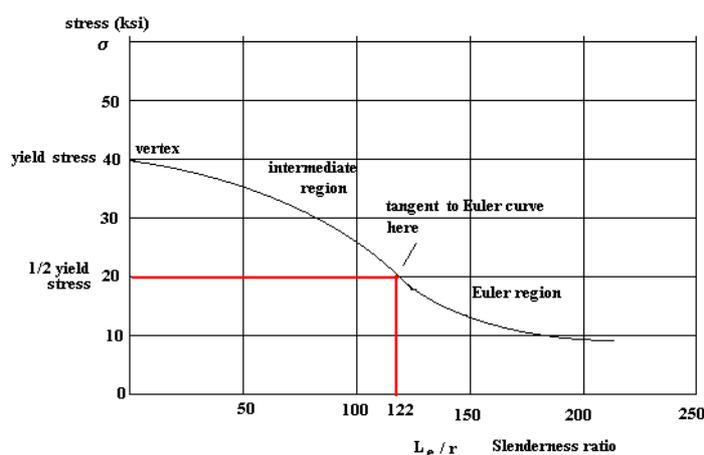


Ilustración 2-4. Curva de Pandeo (o de Euler)

gestor de curvas de pandeo que nos permita hacer la entrada de datos necesarios y observar gráficamente las propias curvas de pandeo. Éstas se usan para comprobar la

conveniencia o no de la elección de un tipo de material determinado a partir de su esbeltez y calidad de acero.

2.3 Tipos de torres y códigos geométricos

Para realizar la selección del tipo de torre [Ki02] que se desea construir, primeramente conviene destacar cuatro modelos diferentes de torres ¹:

- 1. Torres Piramidales (gráfico de torre completa)
- 2. Torres de Ventana o Delta (vemos sólo la cabeza, el cuerpo es similar a la piramidal)
- 3. Torres Cara de Gato (como en la delta, sólo hemos presentado la cabeza)
- 4. Torres Atirantadas (gráfico de torre completa)

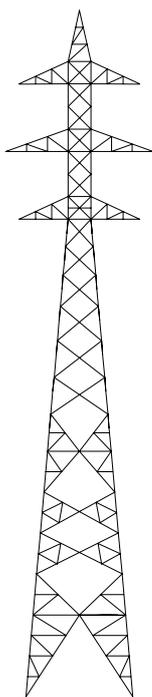


Ilustración 2-5. Torre Piramidal

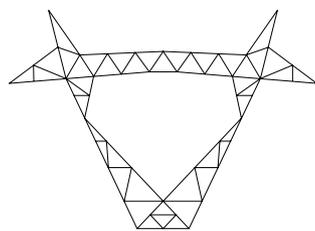


Ilustración 2-8. Cabeza de torre de

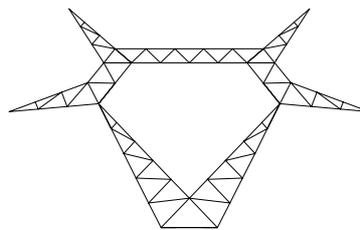


Ilustración 2-7. Cabeza de torre Cara de Gato

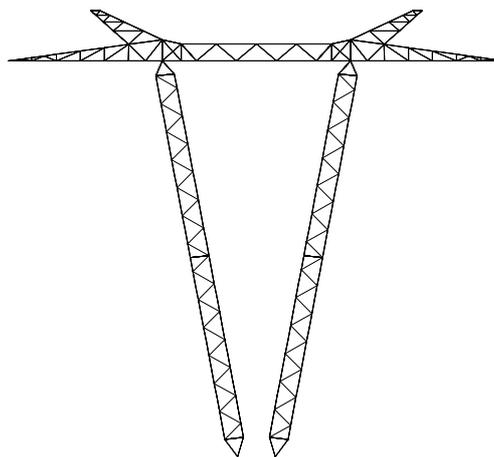


Ilustración 2-6. Torre Atirantada

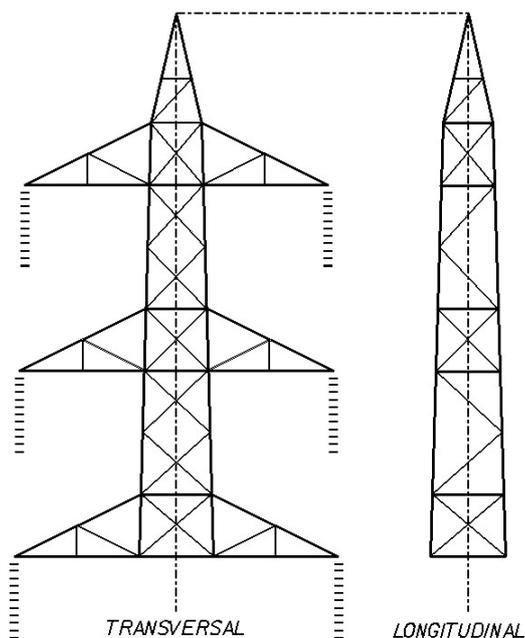
¹ Diseños de I.S.I. [IS97]

Para definir los datos mínimos necesarios para la representación de cada tipo de torre, vamos a distinguir aquellos que pudieran ser comunes a todas ellas (COM), al conjunto (CJT) formado por las tres primeras o a una en particular (PAR).

Comenzamos con la modelización de la torre más habitual en este campo, que es la torre piramidal. Estas estructuras piramidales son las más ligeras conocidas en este campo, aunque dependerá de las condiciones del entorno (cobertura contra rayos, anchura disponible de terreno, etc.) el que también sean las más adecuadas. Se trata de aquellas torres que están compuestas por una cabeza o superestructura, formada por un tronco piramidal central al que se unen unas estructuras colaterales llamadas crucetas (de estas crucetas se engancharan tanto los aisladores como cualquier tipo de herrajes y aparatos que soporten los cables de la línea de transmisión eléctrica).

En este modelo se consideran dos caras (transversal y longitudinal) y se subdivide la torre en celdas o, técnicamente, paneles. En la *Ilustración 2-10. Torre Piramidal Básica* se muestra cómo se definen las diferentes partes de este tipo de torres.

2.3.1 Codificación de caras en una TTE (COM)



A la hora de codificar las estructuras hemos de considerar que las torres están formadas por cuatro caras iguales dos a dos, por lo que realmente sólo deberemos solicitar la información de dos caras distintas. A estas dos caras las llamaremos *transversal* (es perpendicular a la propia línea de transporte de electricidad) y *longitudinal* (la paralela a esta línea).

Ilustración 2-9. Nomenclatura de las caras

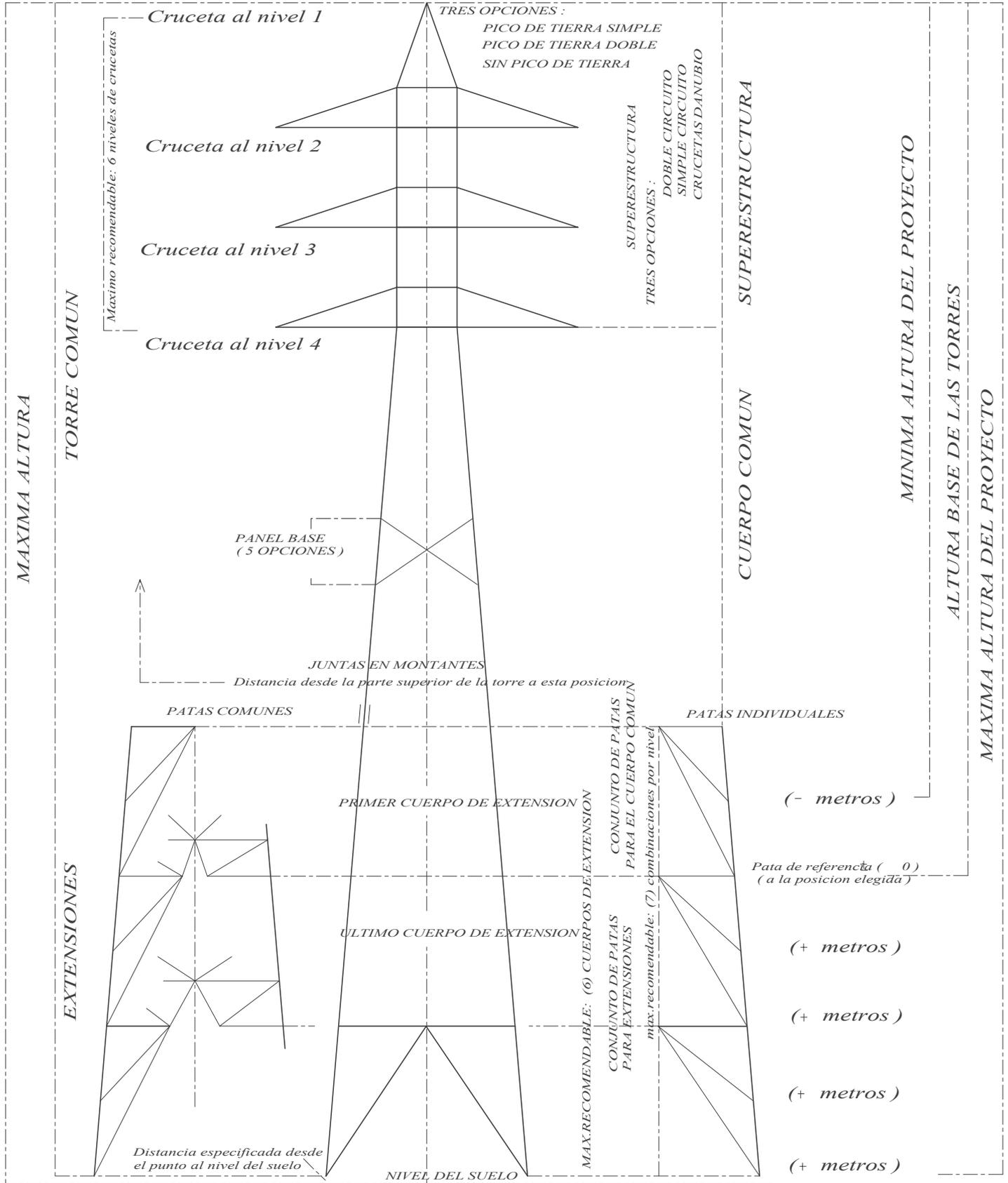


Ilustración 2-10. Torre Piramidal Básica

2.3.2 Tipos de Paneles (COM)

Los paneles podemos considerarlos básicamente, y vistos de frente, como un trapecio vacío que deseamos triangularizar. De esta forma nos centramos en aquellos modelos o tipos más usados actualmente [MS97] y que son el objeto de nuestro estudio.

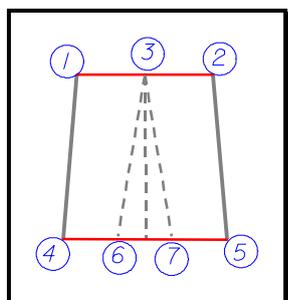


Ilustración 2-11
Numeración de nudos

En primer lugar se numeran los vértices y puntos principales del 1 al 7 de la siguiente forma: como podemos observar, los vértices 1,2,4 y 5 conforman los cuatro vértices del trapecio. El punto 3 es el centro de la base superior y los puntos 6 y 7 pertenecen a la base inferior y distan cero si el siguiente panel a aplicar es pata individual (Ver ilustración anterior de *Torre Piramidal Básica*) y la distancia necesaria en caso de patas comunes. Se consideran los siguientes tipos de paneles:

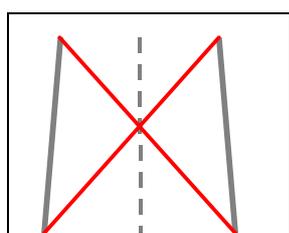


Ilustración 2-12. **Panel Tipo 1**
Nº de barras = 2
Barra 1: Entre 1 y 5
Barra 2: Entre 2 y 4

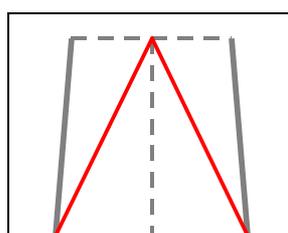


Ilustración 2-13. **Panel Tipo 2**
Nº de barras = 2
Barra 1: Entre 3 y 5
Barra 2: Entre 3 y 4

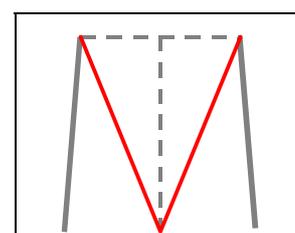


Ilustración 2-14. **Panel Tipo 3**
Nº de barras = 2
Barra 1: Entre 1 y 6
Barra 2: Entre 2 y 6

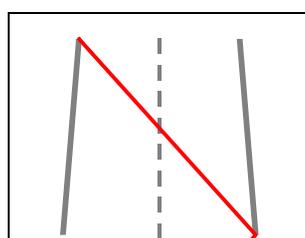


Ilustración 2-15. **Panel Tipo 4**
Nº de barras = 2
Barra 1: Entre 1 y 5
(paneles pares)
Barra 2: Entre 2 y 4
(paneles impares)

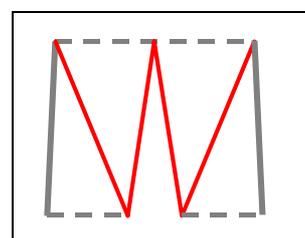


Ilustración 2-16. **Panel Tipo 5**
Nº de barras = 4
Barra 1: Entre 1 y 6
Barra 2: Entre 6 y 3
Barra 3: Entre 3 y 7
Barra 4: Entre 7 y 2

2.3.3 Tipos de Rellenos (COM)

Para disminuir las longitudes de pandeo de los elementos, conservando la resistencia de las torres, se suele aplicar diferentes configuraciones de barras de relleno dentro de cada panel. A continuación presentamos los códigos más usuales [IS97]. Están formados por dos cifras: la primera indica el número de divisiones que se producen sobre las barras exteriores y la segunda, sobre las interiores:

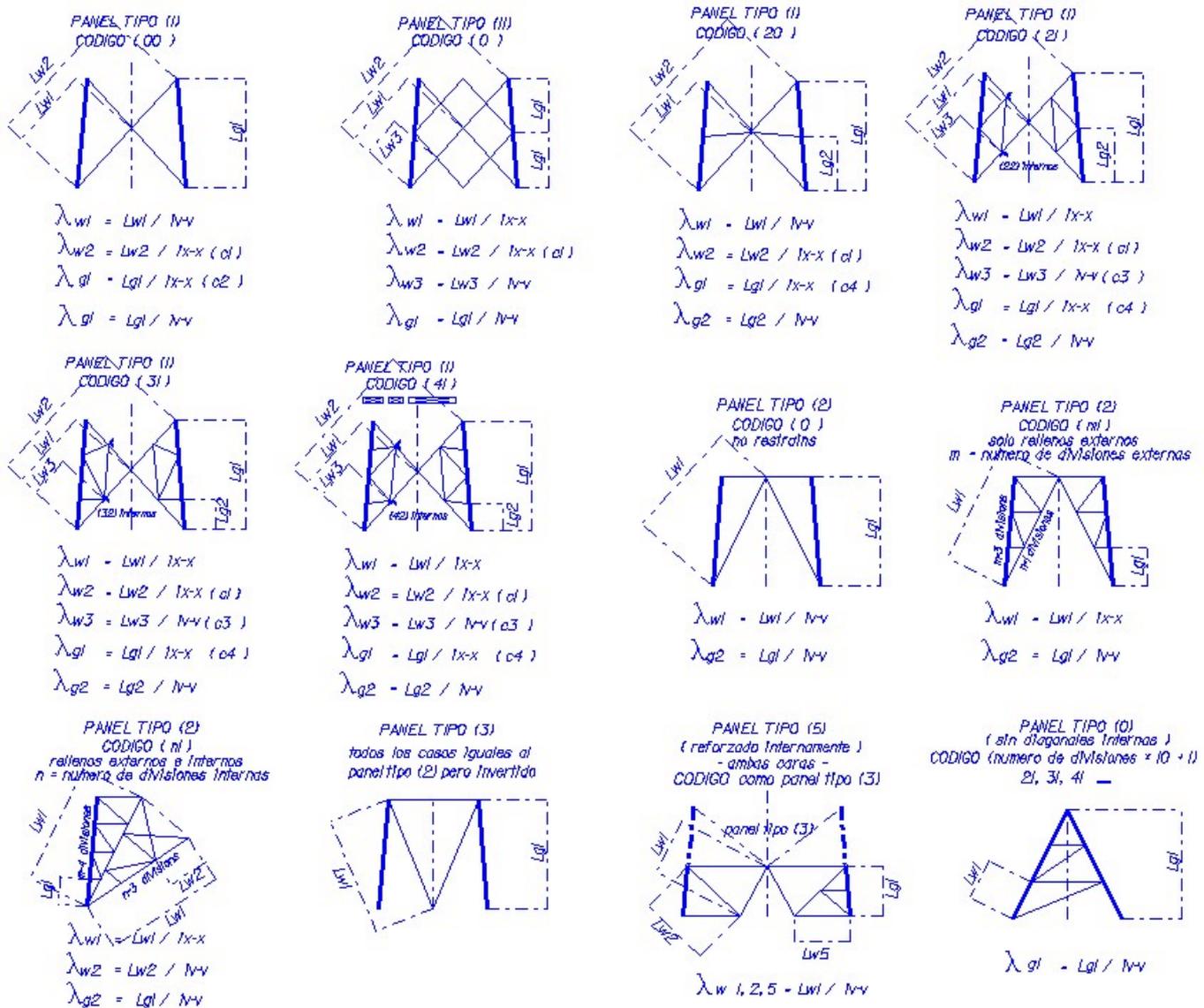
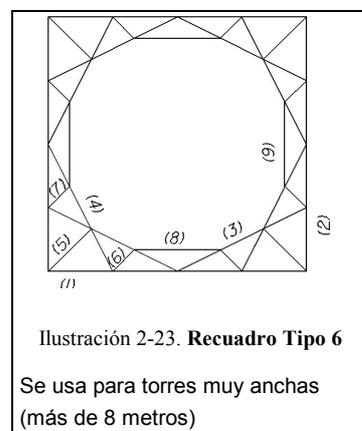
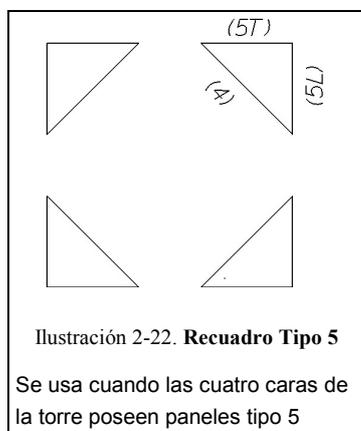
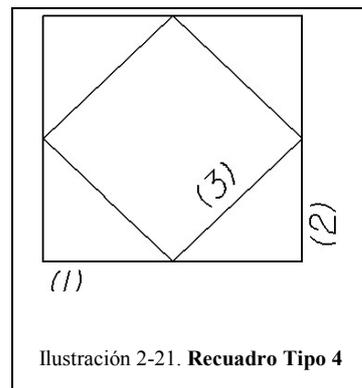
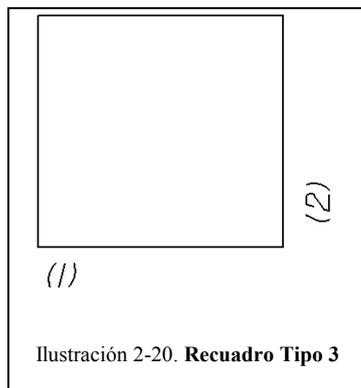
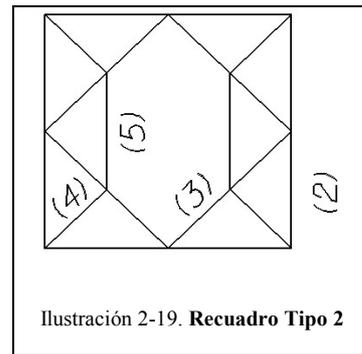
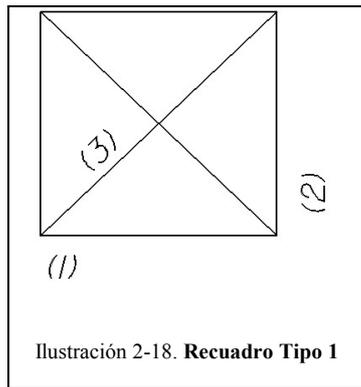


Ilustración 2-17. Tipos de Secundarios

Las fórmulas que aparecen bajo cada tipo de relleno indican la *esbeltez* (λ) de cada barra que expresa el cociente de la longitud de pandeo y el radio de inercia (mínimo v-v o medio x-x).

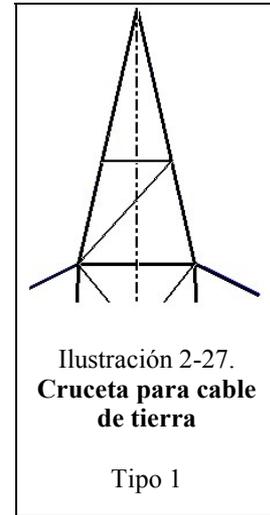
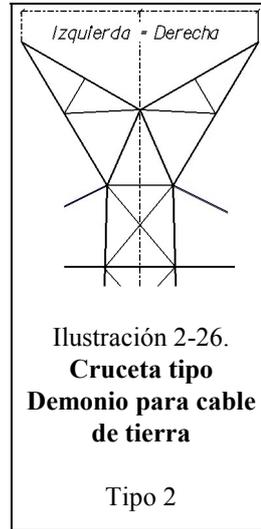
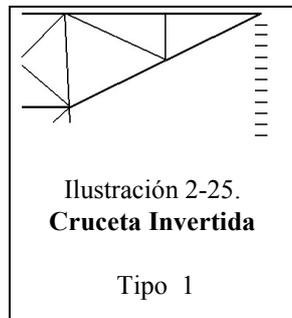
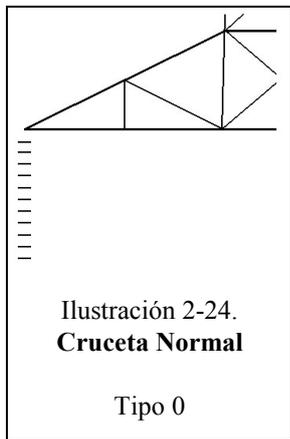
2.3.4 Tipos de Recuadros (CJT)

Junto con estas configuraciones, hemos de completar el tronco central de la torre con los diversos modelos que pueden ser observados desde una visión en planta (paralelos a la horizontal) y a los que llamaremos recuadros. En este caso, estos objetos unen interiormente las cuatro caras de la torre y también para ellos podemos desarrollar una lista de aquellos más usados [IS01, MS97]:



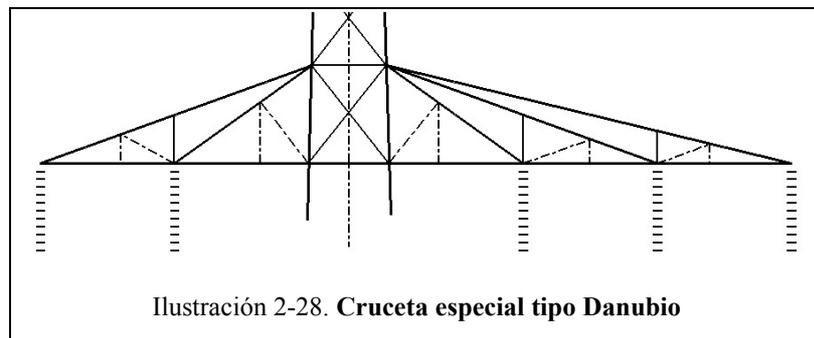
2.3.5 Tipos de Crucetas (PAR)

Para completar las configuraciones necesarias en el diseño espacial de estas TTE mostramos también los modelos de crucetas usados en este campo. Son los siguientes [IS97]:



Cruceta especial Danubio

Tipo 3. Se utiliza para enganchar muchas fases en una misma cruceta y recibe su nombre del hecho de que la primera vez que se usaron fue junto al río Danubio



2.4 Entrada de datos de las torres piramidales.

2.4.1 Datos básicos de un proyecto (COM)

Para comenzar se solicita del usuario los datos básicos del proyecto de líneas de transmisión eléctrica (que son suministrados por el cliente). Aparecen en la siguiente tabla:

Cliente	Empresa que encarga el proyecto
Proyecto	Normalmente es el nombre de las ciudades o puntos que une la línea y el voltaje eléctrico que va a transmitir
Nombre de la Torre	Identificación de la torre para su fabricación
Tipo de Torre	Suspensión, Amarre, Angulo o Terminal
Unidades de Fuerza	Kilogramos o DecaNewtons
Unidades de Dimensión	Centímetros o Pies
Tabla de Angulares	La que use el posterior fabricante de la estructura y que previamente se ha introducido en la configuración
Curva de Pandeo	ASCE por defecto u otra que se introduzca en la configuración

Antes de continuar hemos de aclarar qué entendemos por *Tipo de Torre*. Dentro de una línea eléctrica encontramos cuatro tipos fundamentales de torres:

- **AMARRE** : Las torres que soportan el tirón en caso de caerse alguna de las colindantes.
- **ANGULO** : Aquellas que se utilizan para cambiar la dirección de la línea.
- **TERMINAL** : Las que sólo les llega el cable por uno de los lados y posteriormente lo pasan a línea subterránea o a una subestación.
- **SUSPENSION** : El resto de las torres de la línea (las más habituales y las más importantes por su número a la hora de optimizar el cálculo)

2.4.2 Limitaciones numéricas de la estructura (CJT)

Comencemos entonces con los datos numéricos de la estructura. En primer lugar el sistema solicita los límites siguientes:

Nº de Paneles	Número de paneles que conforman la silueta de la torre (el mismo en ambas caras)
Nº de Paneles con Relleno	Cuántos de ellos poseen rellenos para disminuir la longitud de pandeo de los montantes o de las diagonales
Nº de Paneles con Crucetas	Cuántos de ellos (mirando la cara transversal) llevaran a su izquierda o derecha (o a ambos lados) una cruceta para coger los conductores
Nº de Extensiones	Cantidad de prolongaciones que se van a aplicar a la torre

	para elevar su altura sin modificar el cuerpo común y la superestructura
Nº de Combinaciones de Patas	Cantidad de patas de diferentes alturas que se aplican a cada extensión para superar los accidentes del terreno
Nº de Condiciones de Carga	Cada una de las distintas situaciones que deberá superar la torre cuando se coloquen sobre ella las fuerzas exigidas por las especificaciones técnicas
Nº de Juntas	Al fabricar una torre se divide en diferentes tramos o trozos que posteriormente son ensamblados, pudiendo tener cada uno de ellos un panel o más. Aquí indicaremos esa cantidad de divisiones constructivas a tener en cuenta.

2.4.3 Geometría de las Caras (CJT)

A partir de estos datos se necesita completar la información acerca de la configuración geométrica, las limitaciones de diseño y las condiciones de carga de la estructura.

Panel	Cara Transversal			Cara Longitudinal			Crucetas (SI/NO)	Recuadros Tipo
	Nº	Ancho	Alto	Tipo	Ancho	Alto		
Contador	Medida	Medida	Código	Medida	Medida	Código	Booleano	Código
1								
2								
....								

Para facilitar la entrada de datos sólo se requiere el ancho de cualesquiera de las caras en el último panel y en aquellos intermedios donde haya un cambio de pendiente de la silueta de la estructura.

Al finalizar de introducir estos datos se comprueba que:

- No se produce ningún estrechamiento irregular de la estructura.
- Coinciden la suma de las alturas de los paneles de la cara transversal y longitudinal.
- Los códigos de los tipos de paneles y de recuadros son correctos (*Véanse las respectivas codificaciones anteriormente descritas*)
- El número de niveles de crucetas elegidos coincide con las limitaciones numéricas de la tabla anterior.

2.4.4 Rellenos de paneles (CJT)

Ahora, una vez introducido el número de paneles que poseen rellenos y usando los códigos que aparecen en la nomenclatura de *tipos de rellenos* se ha de completar la siguiente entrada de datos

Panel	Cara Transversal	Cara Longitudinal
<i>Nº</i>	<i>Tipo</i>	<i>Tipo</i>
Dato	Código	Código

Al finalizar de introducir estos datos se realiza la comprobación de que:

- No se produce ningún código erróneo en función del tipo de panel que aparece a cada nivel.

2.4.5 Niveles de extensión (CJT)

Como antes hemos comentado, a la hora de diseñar una torre que va a ser construida en serie para formar parte de una línea de tensión eléctrica, ésta habrá de adaptarse al terreno por el que pasa. Evidentemente, en el caso de que pase por una llanura todas las torres poseerán la misma altura, pero en caso de tener que adaptarse al perfil de una vaguada o a las diferentes pendientes de una zona montañosa hemos de tener la posibilidad de usar la misma torre con diferentes alturas y con patas desiguales. Por eso, el sistema también solicita los niveles por los que podemos cortar la torre y sustituirlo por patas y cuántas alturas diferentes de patas se han de diseñar.

Panel	Tipo	Altura	Altura	Altura	Altura
<i>Nº</i>		<i>Extens.1</i>	<i>Extens.2</i>	<i>Extens.3</i>	<i>Extens.n</i>
Dato	Código o medida	Medida	Medida	Medida	Medida

Estas medidas de alturas de extensiones se irán introduciendo en orden decreciente para facilitar posteriormente el cálculo. Respecto al *Tipo* a introducir tenemos:

⇒ **(Código)** Tipo 2 cuando poseen anchos distintos en función de la altura a la que se vaya a adaptar, o bien

⇒ **(Medida)** Altura del panel tipo 5 que se va a adaptar en el nivel anterior para usar patas comunes² en toda la estructura.

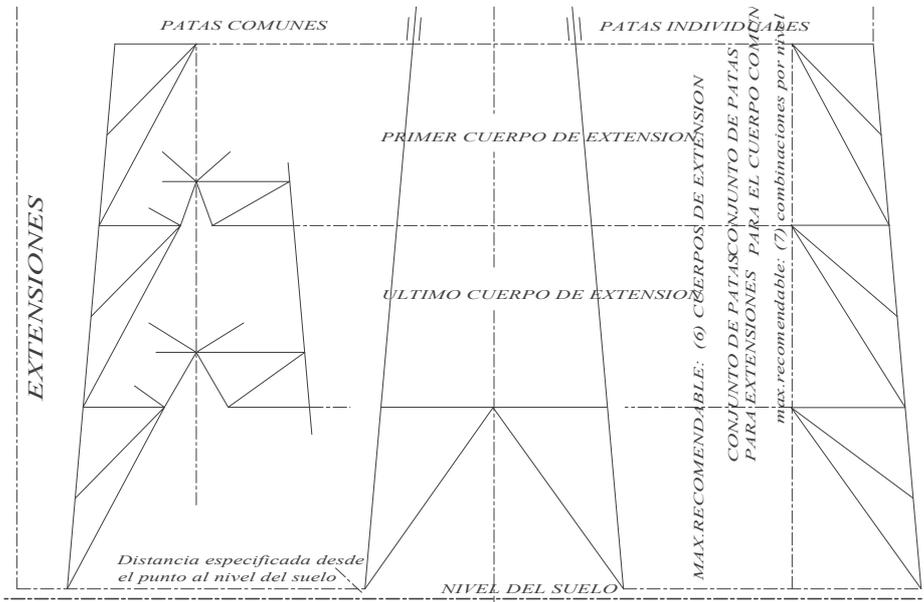


Ilustración 2-29. Tipos de Patas y Extensiones de Torre

Al finalizar de introducir estos datos se comprueba que:

- Las patas han sido introducidas en orden decreciente de altura.
- Los números de los paneles son correctos (ni mayores que el número de paneles de la estructura ni menores que los que posee la cabeza o superestructura de la torre, ya que ahí no puede haber patas).
- Los códigos o medidas de los tipos de paneles son correctos.

2.4.6 Rellenos de extensión (CJT)

Al igual que en los paneles, se han de codificar el tipo de relleno que poseen las extensiones:

² A veces interesa que una misma pata se pueda adaptar a cualquier altura de la torre. Como el ancho de la base de cada panel va aumentando al ir bajando por la torre, necesitaremos un panel especial de transición (panel tipo 5) o también llamado “pantalón” que permita su adaptación, como observamos en este gráfico.

Panel	Relleno	Relleno	Relleno	Relleno
Nº	<i>Extens.1</i>	<i>Extens.2</i>	<i>Extens.3</i>	<i>Extens.n</i>
Dato	Código	Código	Código	Código

Sólo se necesita comprobar la coherencia de los códigos introducidos respecto al tipo de panel de pata anteriormente expuesto.

2.4.7 Limitaciones de diseño (CJT)

Finalmente, los datos que pertenecen a la estructura y que no tenían cabida anteriormente, se solicitan para completar la configuración:

Ancho Transversal	Ancho superior de la cara transversal del primer panel de la estructura (recordemos que los anchos introducidos se corresponden con la parte baja de cada panel)
Ancho Longitudinal	Ancho superior de la cara longitudinal del primer panel de la estructura
Peso preliminar de la estructura	Primera aproximación al peso total para repartirlo como carga vertical sobre toda la estructura
Tipo de recuadro superior	Recuadro superior en la parte alta del primer panel de la estructura

Posteriormente se introducen las limitaciones previas del material para el diseño [BM97]:

Esbeltez máxima en montantes (<i>Valor</i>)	Límite superior en la curva de pandeo para el cálculo de los angulares necesarios en los montantes
Esbeltez máxima en diagonales (<i>Valor</i>)	Límite superior en la curva de pandeo para el cálculo de los angulares necesarios en los diagonales
Esbeltez máxima en el resto (<i>Valor</i>)	Límite superior en la curva de pandeo para el cálculo de los angulares necesarios en el resto de las barras de la estructura
Mínimo espesor en	Límite inferior para el espesor del angular necesario en los

montantes (<i>Medida</i>)	montantes
Mínimo espesor en diagonales (<i>Medida</i>)	Límite inferior para el espesor del angular necesario en los diagonales
Mínimo espesor en el resto (<i>Medida</i>)	Límite inferior para el espesor del angular necesario en el resto de las barras de la estructura
Mínimo tamaño de ala en montantes (<i>Medida</i>)	Límite inferior para el ala del angular necesario en los montantes
Cantidad de métricas de tornillos diferentes	Es el número de espesores diferentes de tornillos que vamos a utilizar (p.ej. 16 mm., 20 mm., 24 mm. , etc.)

Finalmente se introducen todas y cada una de las métricas de tornillos elegidas, tantas como indique la última cantidad introducida en el cuadro anterior.

2.4.8 Escalada y Juntas (CJT)

Para finalizar la entrada de datos de una torre piramidal hemos de introducir dos tipos de datos fundamentales en el diseño y posterior fabricación:

1.- Limitaciones de escalada: estos valores nos permitirán evitar el fallo de cualquier barra (normalmente un relleno) al apoyarse sobre ella cualquier persona que esté subiendo para realizar el montaje, mantenimiento o inspección de la torre:

- a) Angulo : Valor del ángulo máximo respecto al suelo por debajo del cual es probable que una persona se apoye para realizar la escalada.
- b) Peso estimado de esa persona junto con las herramientas y aparejos
- c) Coeficiente de seguridad que se toma en esta carga.

2.- Posiciones de juntas: tengamos en cuenta que la fabricación de una torre se

realiza por tramos o secciones y que éstas estarán compuestas por uno o varios paneles.

Evidentemente, el montante ha de ser común a todo un tramo de fabricación por lo que será importante mantener estas juntas a una distancia suficiente

que permita que los montantes no sean tan largos que después no se puedan introducir en la cuba de galvanizado (normalmente estas cubas tienen una longitud máxima de 8 m.) :

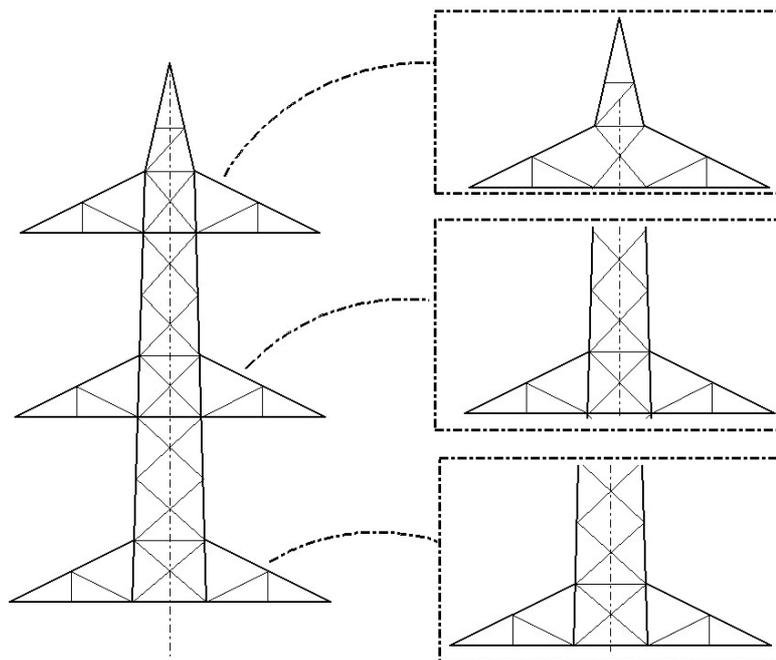


Ilustración 2-30. División de la torre en secciones para su fabricación

Junta	Panel	Distancia
<i>Nº</i>	<i>Nº</i>	<i>Respecto al tope de la torre</i>
Contador	Valor	Medida

2.4.9 Geometría de Crucetas (PAR)

Respecto a la codificación y tipos de crucetas se introduce cada una de ellas diferenciando entre lado izquierdo y lado derecho. Si queremos que posea crucetas por uno solo de los lados deberemos dejar a cero los datos del otro. El ancho de la cruceta lo medimos desde el centro exacto de la torre y no desde el borde donde se sujeta la cruceta.

Panel	Lado Izquierdo			Lado Derecho		
Nº	Ancho	Apertura	Tipo	Ancho	Apertura	Tipo
Contador	Medida	Medida	Código	Medida	Medida	Código
1						
2						
....						

Al finalizar de introducir estos datos comprobamos que:

- No se produce ningún código erróneo en el tipo de cruceta.
- La apertura de la punta de la cruceta no supera al ancho del panel de la cara longitudinal a ese nivel.

2.4.10 Rellenos de crucetas (PAR)

El código de relleno de las crucetas se compone de tres caracteres:

$$\text{Código de relleno de cruceta} = \# \# \#$$

- El primero (con valores entre 0 y 9) especifica el número de divisiones que va a tener la cruceta.
- El segundo nos dice si las barras del plano horizontal y de la cara vertical coinciden en el mismo punto (V) o en el punto medio de ambas (X).
- El tercero indica si posee barras de cierre en el plano secundario (V) o no (X).



A partir de aquí vamos introduciendo cada uno de los códigos de relleno:

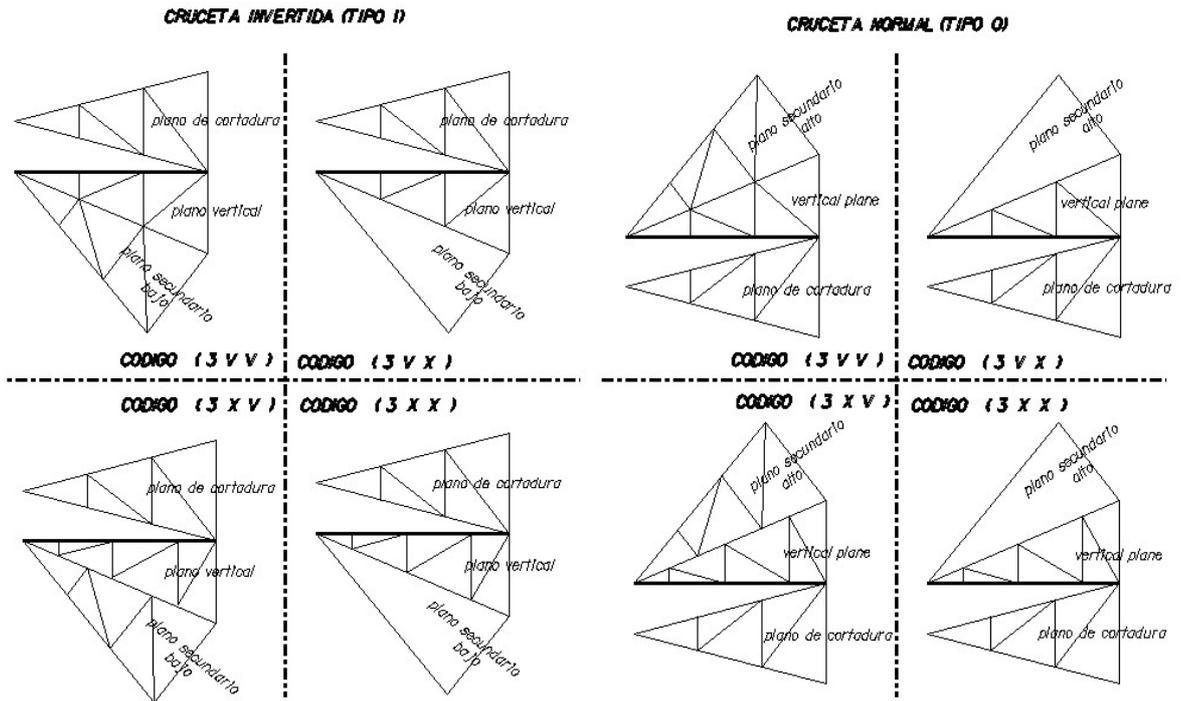


Ilustración 2-31. Tipos de Rellenos de Crucetas

Nivel	Lado Izquierdo	Lado Derecho
Nº	Tipo	Tipo
Contador	Código	Código

Al finalizar de introducir estos datos comprobamos que:

No se produce ningún código erróneo en función de la nomenclatura expuesta.

2.4.11 Condiciones de Carga (PAR)

Las cargas que debe soportar la estructura, y que habitualmente van a ser las que se aplicarán sobre el prototipo en la estación de ensayo, están compuestas por cuatro entradas de datos para cada una de las condiciones señaladas en los datos iniciales:

1.- Cargas en crucetas: Fuerza aplicada en cada una de las puntas de crucetas respecto a los tres ejes de coordenadas cartesianos XYZ, aunque en la nomenclatura de estructuras se identifican como vertical (-Z), longitudinal o en el sentido de la línea eléctrica (+X) y transversal o perpendicular al sentido de la línea (+Y).

Cruceta	Lado Izquierdo			Lado Derecho		
	Vertical	Longitudinal	Transversal	Vertical	Longitudinal	Transversal
Contador	Fuerza	Fuerza	Fuerza	Fuerza	Fuerza	Fuerza
1						
2						
...						

2.- Presión de viento sobre la estructura: Presión debida al viento que va a actuar sobre las caras de la estructura. Se introduce su valor en kg/m² en las direcciones transversal (+Y) y longitudinal (+X).

3.- Viento sobre la estructura: Fuerzas ya aplicadas a la parte inferior de los paneles que se deseen. Realmente el viento actúa sobre toda la estructura pero a la hora de realizar el ensayo se eligen unos cuantos niveles y se simula que todo el viento de la estructura estuviese concentrado en esos puntos. Por ello el sistema tratará de parecerse en lo posible al ensayo, que es a fin de cuenta el momento más crítico que ha de soportar la estructura.

Nivel	Transversal	Longitudinal
Nº	Viento	Viento
Valor	Fuerza	Fuerza

4.- Coeficientes de seguridad: Cuando las cargas de la estructura proyectada vienen mayoradas (con los coeficientes de seguridad aplicados) se introduce un “1” en estos valores. De lo contrario insertaremos lo que en cada caso corresponda. Todas las fuerzas que actúan sobre la estructura serán multiplicadas por su coeficiente de seguridad. Se solicitan tres valores. El coeficiente de seguridad *Vertical* (cargas



verticales que no sean el peso propio de la estructura), el *Horizontal* (cargas transversales y longitudinales) y el de *Peso* (preliminar) sobre la estructura.

2.5 Entrada de datos de las torres de ventana o delta.

Además de los datos anteriores de conjunto (CJT), las torres delta poseen los siguientes:

2.5.1 Datos de Geometría de la Cabeza (PAR)

En la cabeza podemos definir 2 caras, transversal y longitudinal, cuyas entradas son:

Cara Transversal: Estas distancias solicitadas siguen el esquema indicado

Nº	Valor de A
Contador	Medida

Nº	Valor de H
Contador	Medida

Nº	Valor de C
Contador	Medida

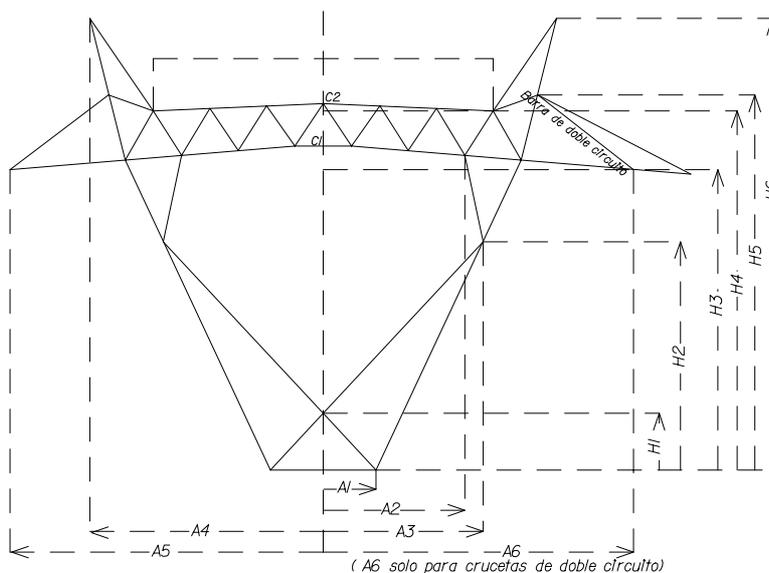
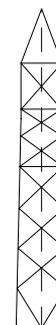


Ilustración 2-32. Medidas de la cabeza de la torre Delta

Cara Longitudinal: Se fijan los sectores, anchos en los cambios de tangente, tipos de paneles y rellenos

Panel	Sector	Ancho	Tipo	Relleno
Contador	Valor	Medida	Código	Código



2.5.2 Condiciones de Carga (PAR)

Introduciremos las cargas en las direcciones *Vertical*, *Longitudinal* y *Transversal* de la misma forma que en la piramidal pero en función del nudo sobre el que estemos trabajando, y cuya numeración aparece en el gráfico. Además del viento sobre la estructura añadiremos el viento sobre la cabeza en los nudos indicados con una flecha.

Nudo	Vertical	Longitudinal	Transversal
Contador	Valor	Valor	Valor

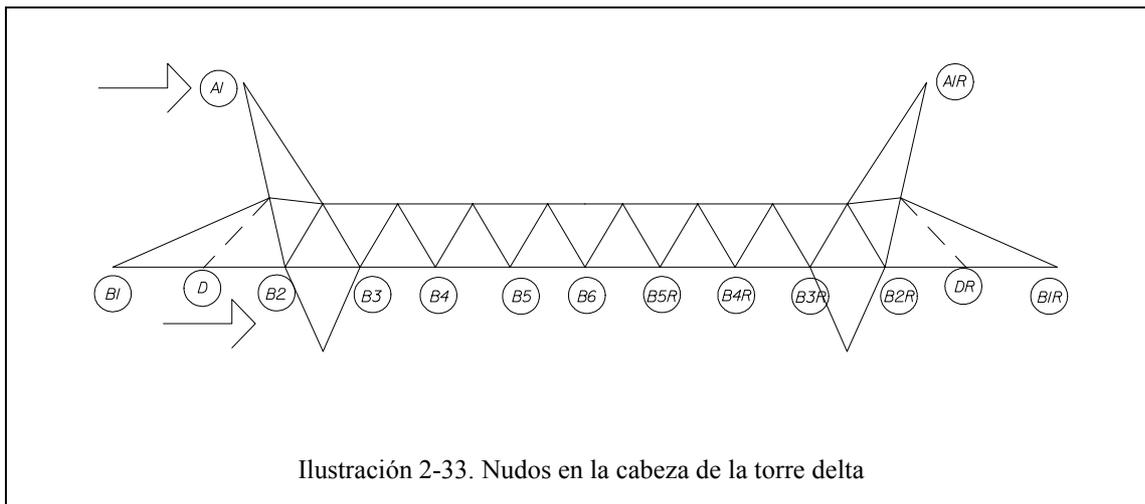


Ilustración 2-33. Nudos en la cabeza de la torre delta

2.6 Entrada de datos de las torres Cara de Gato.

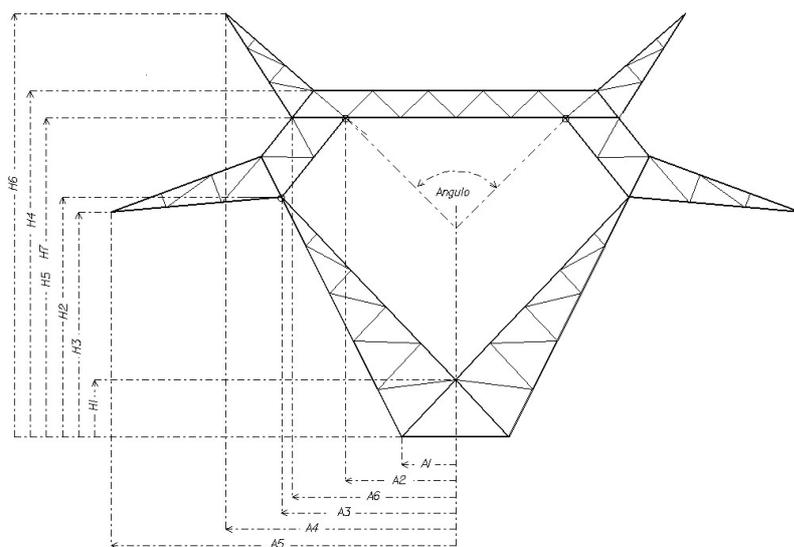
Por analogía con las torres delta, los únicos cambios son:

2.6.1 Datos de Geometría de la Cabeza (PAR)

Cara Transversal : Estas distancias solicitadas siguen el esquema indicado

Nº	Valor de A
Contador	Medida

Nº	Valor de H
Contador	Medida



Además se pedirá el **ángulo de inclinación de las cadenas** (conjunto de aisladores de cerámica o vidrio que sujetan y separan el cable eléctrico de la estructura metálica).

Cara Longitudinal: Marcamos los sectores, anchos en los cambios de tangente, tipos de paneles y rellenos

Panel	Sector	Ancho	Tipo	Relleno
Contador	Valor	Medida	Código	Código



2.6.2 Condiciones de Carga (PAR)

Introduciremos las cargas en las direcciones *Vertical*, *Longitudinal* y *Transversal* sobre los tres conductores que poseen estas estructuras:

Conductor	Vertical	Longitudinal	Transversal
Contador	Valor	Valor	Valor

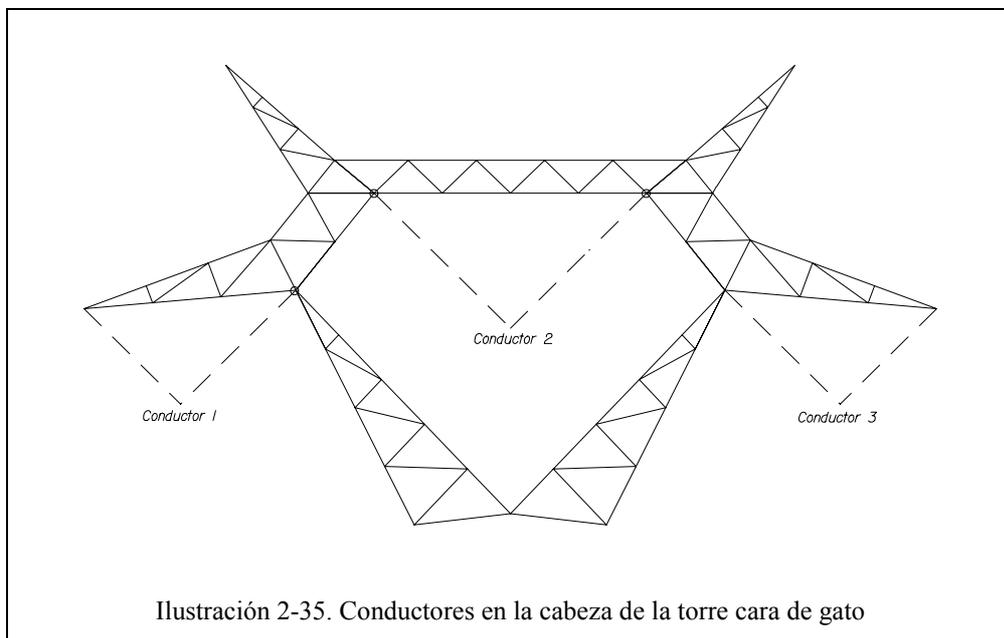


Ilustración 2-35. Conductores en la cabeza de la torre cara de gato

2.7 Entrada de datos de las torres atirantadas.

A continuación vemos, gráficamente, las diferentes opciones y datos solicitados para el diseño de las torres atirantadas.

Estas torres son usadas de forma excepcional, y se utilizan cuando los terrenos a

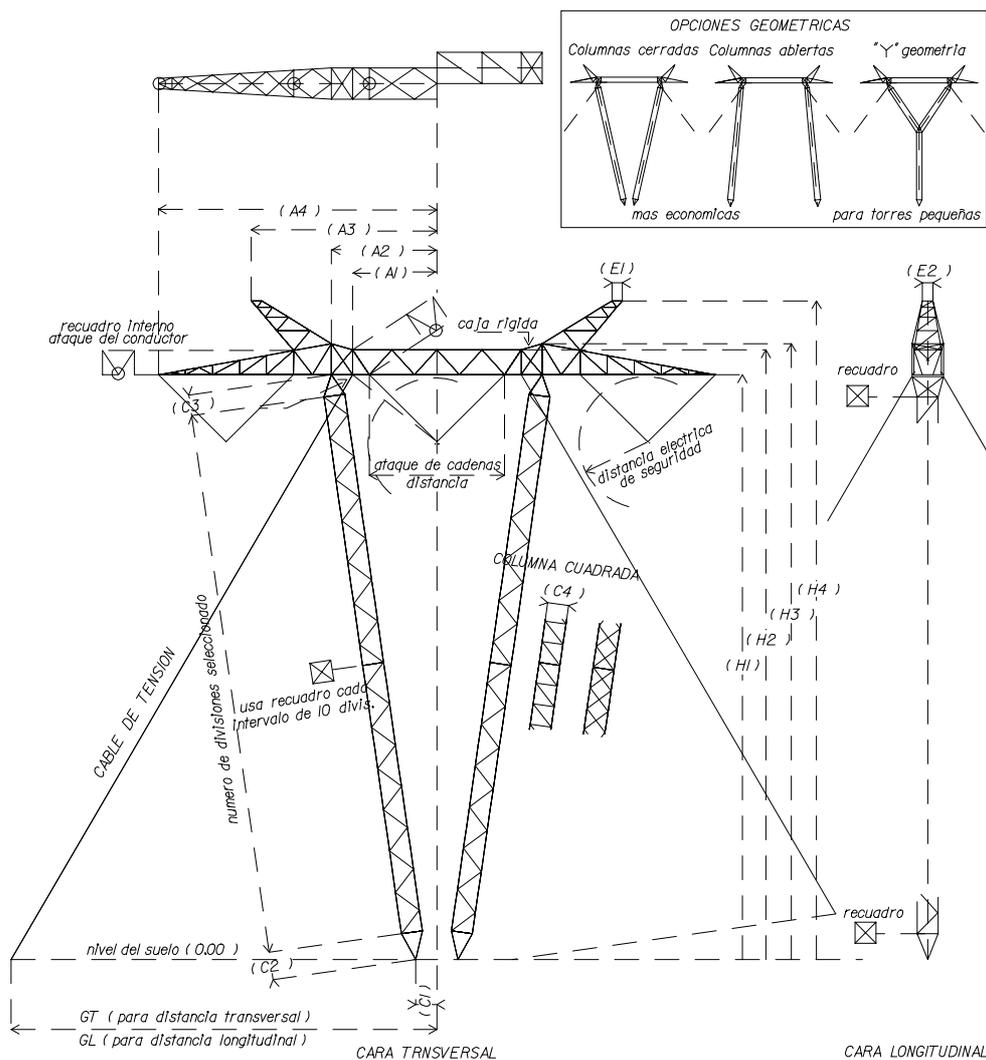


Ilustración 2-36. Medidas de la torre Atirantada

expropiar para el paso de la línea deben ser mínimos (paso por parajes naturales, parques nacionales, etc.). Su construcción es más costosa que en los casos normales de las anteriores y también se encarece el mantenimiento al tener que estar

constantemente midiendo la tensión de los cables que atirantan la estructura y poniéndolos en su valor exacto.

2.8 Aportaciones Realizadas

- Desarrollo del sistema central de procesos para la creación de un sistema experto en el diseño, cálculo y fabricación de torres
- Clasificación de parámetros comunes y particulares para los diferentes modelos de torres
- Establecimiento de la numeración nodal para la unidad mínima de fabricación (panel)

2.9 Referencias Bibliográficas

[AA99] R. Argüelles Álvarez, R. Argüelles Bustillo, F. Arriaga Martitegui, J. R. Atienza Reales. "Estructuras de Acero: Cálculo, Norma Básica y Eurocódigo". Ed. Bellisco Ediciones Técnicas y Científicas. Madrid. España. 1999

[AH88] Allen, M. B., Herrera, I. y Pinder, G. F. "Numerical Modeling in Science and Engineering", John Wiley & Sons, 1988.

[AS88] American Society of Civil Engineers. "Guide for design of steel transmission towers". Edit. ASCE. New York. EE.UU. 1988.

[BM97] R. L. Brockenrough y F. S. Merritt . "Manual de Diseño de Estructuras de Acero". Ed. McGraw-Hill Santa Fe de Bogotá. Colombia. 1997

[Ce94] Cevallos, F. J.. "Enciclopedia de Microsoft visual Basic". Edit. Ra-Ma.. Madrid. España 1994.

[CO98] Coz, J.J.; Ordieres, J. "Fundamentos para la modelización de la estructura". Manual de utilización del programa Solve. Cap. Escal3D. La Rioja.España. 1998

- [Ga98] Garcia Castellanos, D. “Modelos numéricos 3D”. Incluido en PhD Thesis *Desarrollo de modelos numéricos de flexión litosférica*. CSIC. Barcelona. España. 1998
- [IS97] ISI (Ingeniería de Sistemas Interactivos Ltda.) . “Computer integrated transmission towers system”. Manual. Málaga. España. 1997.
- [Jo87] Joyanes Aguilar, L. “Metodología de la programación” Edit. Mc Graw Hill. Madrid. España. 1987.
- [Ki02] Kipp, V.W. “Tower identification”. Site Management & Technology. Wisconsin. USA. 2002
- [Ma98] Marco García, J. "Fundamentos para el cálculo y diseño de estructuras metálicas de acero laminado". McGraw-Hill, España. 1998
- [Ma00] Marco García, J. "Curso básico de cálculo y diseño de estructuras metálicas en ordenador". McGraw-Hill, España. 2000
- [RM02] R. Ramírez Borlado, C. Martínez Lasheras y R. Martínez Laceras. *Prontuario de Estructuras Metálicas*. Centro de Estudios y Experimentación de Obras Públicas (CEDEX).
- [Sc02] Schafer, B.W. “What is a buckling curve?. Elastic buckling by the finite strip method”. Tutorials. Johns Hopkins Univ. Baltimore. USA. 2002.
- [Wi86] Wirth, N. “Algoritmos + estructuras de datos = programas” Edit. Del Castillo. Madrid. España. 1986.

3. Sistema Espacial de Cálculo

Durante muchos años, especialmente antes de la entrada de las computadoras en el entorno de la ingeniería, se utilizaron métodos gráficos, aritméticos, etc. que trataban de resolver las estructuras en un plano. Esto significaba una limitación importante debido a que el comportamiento espacial podía provocarnos desagradables sorpresas no consideradas por los métodos existentes.

A partir del desarrollo de los ordenadores (y sobre todo de su abaratamiento) las industrias fijaron sus miradas en aquellos métodos espaciales que, a pesar de que requerían un mayor aparato matemático y un gran volumen de datos, reflejaba el comportamiento espacial de las estructuras con mayor precisión, calculaba sus desplazamientos, definía las fuerzas que actuaban sobre cada barra, es decir, se aproximaba bastante más al modelo ideal que el diseñador tenía en su pensamiento.

3.1 Repaso histórico

El periodo entre 1850 y 1875 supone un punto inicial apropiado para un pequeño repaso [Ga75], aunque fuera precedido por los grandes logros de la escuela francesa de ingenieros trabajando con la elasticidad con nombres como Navier y St. Venant. Durante este período emergieron los conceptos del análisis de estructuras, gracias a los esfuerzos de Maxwell, Castigliano y Mohr, entre otros. Estos conceptos fueron el germen de la metodología del análisis matricial de estructuras, que no tomaría forma hasta casi 80 años más tarde. Precisamente este análisis forma la base del estudio de los elementos finitos [Ru95].

Entre 1875 y 1920, el progreso en el desarrollo de la teoría y de las técnicas analíticas auxiliares a los elementos finitos fue bastante lento. Eso se debía, en gran medida, a las limitaciones prácticas que permitían resolver nada más que unas ecuaciones con pocas variables. Es de señalar que entonces, para estudiar los entramados y las estructuras de barras, se empleaba de forma casi universal un análisis basado en unas suposiciones respecto a la distribución de tensión donde los parámetros de fuerza eran desconocidos.

Fue aproximadamente en 1920 gracias a los esfuerzos de Maney en EE.UU. y Ostenfeld en Dinamarca, cuando tomaron forma las ideas básicas acerca de un análisis de entramados y estructuras de barras que ya se basarían en los parámetros de desplazamiento como variables. Estas ideas fueron las precursoras de los conceptos modernos del análisis matricial de estructuras que utilizamos hoy en día. Sin embargo, siguieron existiendo hasta 1932 unas limitaciones severas respecto al tamaño de los problemas que se podían manejar, mediante las variables de fuerzas o las variables de desplazamiento. El método de Cross, introducido entonces, permitió la solución de problemas más complejos, y sería el principal método practicado durante los siguientes 25 años [Fo02].

Las computadoras digitales aparecieron por primera vez a principios de los años 50, pero su importancia real, tanto para la teoría como práctica, no se dio a conocer inmediatamente. Sin embargo, algunos preveían el impacto y codificaron el análisis de estructuras, ya bien establecido por aquellas fechas, en esa forma de matriz tan apta para el ordenador. De tal manera contribuían, por ejemplo, Argyris y Patton. Otros desarrollos importantes constituían los trabajos de Argyris y Kelsey por un lado, y los de Turner, Clough, Martin y Topp por otro. En éstos se relaciona el concepto del análisis de estructuras con el del análisis continuo, y se captan los procedimientos en formato de matriz. Esas publicaciones representan la fuente más importante para el desarrollo del método de elementos finitos durante los siguientes años. Sin embargo, varias características claves del método habían surgido ya antes de 1950 en las obras de Courant, McHenry y Hrenikoff. Un significado especial tiene el trabajo de Courant, ya que se ocupa de problemas regidos por ecuaciones que se pueden aplicar también a problemas fuera de la mecánica [Wi97].

La técnica del análisis de los elementos finitos ha avanzado bastante desde los años 50, transcurriendo por caminos diferentes. Motivados por la formulación específica de elementos para la tensión plana, los investigadores establecían unas relaciones de elementos para sólidos, placas curvadas, láminas y otras formas estructurales. Una vez establecidas éstas con el propósito del análisis lineal, estático y elástico, la atención se volvía hacia unos fenómenos especiales, como eran la reacción dinámica, el pandeo o el no cumplir las condiciones de linealidad por la geometría o el material. Era necesario ampliar, no sólo las formulaciones de los elementos, sino también el

marco general del análisis. Siguió un período de desarrollo intensivo de unos programas de ordenador de *ámbito general*, hechos con la intención de poner en manos del usuario las posibilidades del método.

En la práctica, estos programas que implementan el Método de Elementos Finitos (MEF) están muy extendidos. La posibilidad de adquirirlos a un coste moderado ha contribuido a la abundancia de aplicaciones prácticas del MEF. Por otro lado, los investigadores seguían estando ocupados con la construcción de nuevos elementos, con el avanzado desarrollo de formulaciones y algoritmos mejorados para fenómenos especiales, y con la construcción de programas nuevos. De gran interés era también el establecer el MEF para fenómenos no estructurales e interdisciplinares. Un conocido ejemplo lo constituyen los termoestructurales, donde se integran perfectamente el cálculo de la tensión térmica y la distribución transitoria de temperatura. Otro ejemplo es la interacción entre fluidos y su reacción estructural en los casos de la hidroelasticidad [RC02].

Si hace unos años el MEF era una técnica minoritaria en España, utilizada únicamente para estudiar problemas muy especiales por empresas de alta tecnología, actualmente constituye una herramienta sumamente extendida. Se emplea tanto en la investigación y desarrollo en la mayor parte de los ámbitos científicos y tecnológicos, como en bastantes sectores productivos preocupados por mejorar la calidad de sus productos y procesos. Así que, hoy en día, nos podemos encontrar aplicaciones del MEF en áreas tan lejanas entre sí como el diseño estructural, campo tradicional del MEF, y la meteorología, en donde se resuelven hoy los problemas de simulación numérica de quizás mayor tamaño. Por ahora la mayor parte de las aplicaciones donde se usa el MEF la constituye los cálculos sencillos, lineales y estáticos. En los fenómenos más específicos, como son los problemas de dinámica, de técnicas de solución no-lineales y también los problemas lineales de placas y láminas con dificultades especiales, todavía se debaten las técnicas óptimas [Ru95].

3.2 Formalización del método de los elementos finitos

En esta sección damos una breve introducción sobre el método de los elementos finitos para el análisis estático lineal¹. Para ello usaremos la notación expresada en los textos de Zienkiewicz y Taylor [ZT94] y Oñate [OÑ92] basados en la teoría de la elasticidad de Timoshenko [TY79].

3.2.1 Notación clásica de Timoshenko

Las ecuaciones matriciales de una estructura de barras se obtienen a partir del estudio del equilibrio de las diferentes barras que la componen.

Sea una barra i de longitud l^i sometida únicamente a fuerzas axiales, entonces se deduce a partir de la teoría de resistencia de materiales que la deformación ξ en cualquier punto de la barra es igual al alargamiento propio de la misma, es decir:

$$\xi = \frac{\nabla l^i}{l^i} = \frac{u^i_2 - u^i_1}{l^i}$$

donde u^i_1 y u^i_2 son los desplazamientos de los dos extremos de la barra.

Por otra parte, la tensión axial σ está relacionada con la deformación ξ por la **Ley de Hooke** y obtenemos:

$$\sigma = E^i \xi = E^i \frac{u^i_2 - u^i_1}{l^i}$$

donde E^i es el módulo de elasticidad del material de la barra. Suponiendo que el material es homogéneo e integrando las tensiones sobre la sección transversal de la barra de área A^i se obtiene el esfuerzo axial N :

¹ Los análisis lineales son precisos en un modelo cuando la relación entre esfuerzos y desplazamientos es lineal por naturaleza. En un análisis no lineal suponemos que las fuerzas no muestran una relación lineal entre desplazamientos o esfuerzos. Aunque las estructuras tienen de una forma u otra, un comportamiento no lineal, en la mayoría de los casos se puede realizar un análisis lineal que represente el comportamiento de forma suficientemente exacta. Los problemas no lineales se deben principalmente a:

- **No linealidad geométrica (exceso de deformación)**
- **No linealidad debido al material (los esfuerzos modifican su estado)**
- **No linealidad debido al análisis de contacto. (p.ej. los engranajes)**

$$N = A^i \sigma = (EA)^i \frac{u_2^i - u_1^i}{l^i}$$

Finalmente, estableciendo el equilibrio de las fuerzas axiales R_1^i y R_2^i actuantes en los extremos de la barra se obtiene que:

$$R_1^i = -R_2^i = N = A^i \sigma = (EA)^i \frac{u_2^i - u_1^i}{l^i} = k^i (u_2^i - u_1^i)$$

siendo $k^i = \left(\frac{EA}{l}\right)^i$. Recordemos que el superíndice de todas las expresiones anteriores se refiere a la barra particular. Expresando la ecuación anterior en forma matricial obtenemos:

$$q^i = \begin{Bmatrix} R_1^i \\ R_2^i \end{Bmatrix} = k^i \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{Bmatrix} u_1^i \\ u_2^i \end{Bmatrix} = K^i a^i$$

donde K^i se denomina la **matriz de rigidez** de la barra y es función únicamente de la geometría de la misma (l^i, A^i) y de sus propiedades mecánicas (E^i), y a^i y q^i son los vectores de desplazamientos y de fuerzas en los nudos de la barra, respectivamente. Esta ecuación es la expresión matricial de equilibrio de la barra aislada.

Si queremos generalizar a una estructura compuesta de barras como la considerada sólo tendremos que utilizar que la suma de las fuerzas en un nudo, debidas a las diferentes barras que concurren en el mismo, es igual a la fuerza exterior que actúa en dicho nudo. Formalmente, sea el nudo j y sean n_j el número de barras que concurren en él:

$$\sum_{e=1}^{n_j} R_e^i = R_j$$

donde e toma los valores 1 ó 2 en función del extremo de la barra que actúe sobre el nudo. Sustituyendo cada uno de estos R_e^i por sus respectivas expresiones matriciales obtenemos *la ecuación matricial de equilibrio global de la estructura* como:

$$\begin{pmatrix} K_{11} & K_{12} & \cdots & K_{1n} \\ K_{21} & K_{22} & \cdots & K_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ K_{n1} & K_{n2} & \cdots & K_{nn} \end{pmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{Bmatrix} = \begin{Bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{Bmatrix}$$

que se suele denotar como

$$Ka = f$$

donde K es la matriz de rigidez de la estructura y a y f son, respectivamente, las matrices de desplazamientos y de fuerzas exteriores de todos los nudos de la estructura. El proceso de obtención de las ecuaciones recibe el nombre de *ensamblaje*. La resolución de este sistema, donde la matriz K es simétrica y la matriz f tendrá tantas columnas como condiciones de carga distintas deba soportar la estructura, proporciona los valores de los desplazamientos (a) en todos los nudos de la estructura, a partir de los cuales se pueden medir los esfuerzos internos o fuerzas axiales que actúan sobre todas y cada una de las barras.

En el proceso de ensamblaje hemos de considerar que ciertos nudos no deben desplazarse (de lo contrario cualquier estructura estaría moviéndose sin control en cualquier dirección del espacio). Por ejemplo, en las torres piramidales, delta o cara de gato, no deben desplazarse los cuatro nudos de cogida al suelo. Para ello se aporta una limitación a los nudos, llamada coacción, que impide su desplazamiento. Esto no es más que colocar en la correspondiente posición de la matriz de rigidez donde aparecen estos nudos un valor infinito (en nuestro caso, al trabajar con ordenadores incluiremos un valor extremadamente alto, p. ej., $1e25$)

3.2.2 Notación diferencial de Castigliano

La notación del método de los elementos finitos también se establece mediante cálculo diferencial en derivadas parciales de primer orden a partir de los teoremas de Castigliano [Ar86]:

Imaginemos un cuerpo elástico sometido a un conjunto de fuerzas q_i aplicadas de manera lineal (arrancan desde un valor cero hasta llegar al módulo de q_i linealmente) del que sólo estudiamos sus deformaciones y desplazamientos locales ignorando las traslaciones que pudiera sufrir el cuerpo en su totalidad por el espacio. Si denominamos Δ_i al desplazamiento final del punto i en la dirección de q_i , el trabajo W (o energía de deformación) realizado por las fuerzas durante la deformación del cuerpo será:

$$W = \frac{1}{2} \sum_{i=1}^n q_i \Delta_i$$

Si terminado el proceso se introduce en el punto 1 un desplazamiento pequeño Δ_1 , la variación de la energía de deformación será:

$$\frac{\partial W}{\partial \Delta_1} = \frac{1}{2} \left[q_1 + \sum_{i=1}^n \frac{\partial q_i}{\partial \Delta_1} \Delta_i \right]$$

De acuerdo con el primer teorema de Castigliano, $\frac{\partial W}{\partial \Delta_1} = q_1$, luego, sustituyendo

obtenemos que:

$$q_1 = \sum_{i=1}^n \frac{\partial q_i}{\partial \Delta_1} \Delta_i = \frac{\partial q_1}{\partial \Delta_1} \Delta_1 + \frac{\partial q_2}{\partial \Delta_1} \Delta_2 + \dots + \frac{\partial q_n}{\partial \Delta_1} \Delta_n$$

o en general, para cada $j \in \{1, 2, \dots, n\}$:

$$q_j = \frac{\partial q_1}{\partial \Delta_j} \Delta_1 + \frac{\partial q_2}{\partial \Delta_j} \Delta_2 + \dots + \frac{\partial q_n}{\partial \Delta_j} \Delta_n$$

Esta ecuación se escribe en forma matricial de la manera siguiente:

$$\begin{pmatrix} \frac{\partial q_1}{\partial \Delta_1} & \frac{\partial q_2}{\partial \Delta_1} & \dots & \frac{\partial q_n}{\partial \Delta_1} \\ \frac{\partial q_1}{\partial \Delta_2} & \frac{\partial q_2}{\partial \Delta_2} & \dots & \frac{\partial q_n}{\partial \Delta_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial q_1}{\partial \Delta_n} & \frac{\partial q_2}{\partial \Delta_n} & \dots & \frac{\partial q_n}{\partial \Delta_n} \end{pmatrix} \begin{pmatrix} \Delta_1 \\ \Delta_2 \\ \vdots \\ \Delta_n \end{pmatrix} = \begin{pmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{pmatrix}$$

Aquí podemos observar el hecho de que la matriz cuadrada es simétrica ya que, a partir del resultado anterior:

$$\begin{cases} \frac{\partial q_i}{\partial \Delta_j} = \frac{\partial(\partial W / \partial \Delta_i)}{\partial \Delta_j} = \frac{\partial^2 W}{\partial \Delta_j \partial \Delta_i} \\ \frac{\partial q_j}{\partial \Delta_i} = \frac{\partial(\partial W / \partial \Delta_j)}{\partial \Delta_i} = \frac{\partial^2 W}{\partial \Delta_i \partial \Delta_j} \end{cases}$$

y por el teorema de Schwartz: $\frac{\partial q_i}{\partial \Delta_j} = \frac{\partial^2 W}{\partial \Delta_j \partial \Delta_i} = \frac{\partial^2 W}{\partial \Delta_i \partial \Delta_j} = \frac{\partial q_j}{\partial \Delta_i}$

Para relacionar las notaciones de Timoshenko y Castigliano nos basta con sustituir:

$$\frac{\partial q_i}{\partial \Delta_j} = K_{ij}$$

obteniendo por tanto:

$$\begin{pmatrix} K_{11} & K_{12} & \dots & K_{1n} \\ K_{21} & K_{22} & \dots & K_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ K_{n1} & K_{n2} & \dots & K_{nn} \end{pmatrix} \begin{pmatrix} \Delta_1 \\ \Delta_2 \\ \vdots \\ \Delta_n \end{pmatrix} = \begin{pmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{pmatrix}$$

3.3 Uso del MEF en el diseño de torres

El primer problema con el que nos encontramos cuando estamos calculando una TTE es el establecimiento de la matriz de rigidez, para lo cual es necesaria, como se muestra a continuación, la elección adecuada del modelo de numeración de nudos en el espacio.

3.3.1 Numeración de Nudos

Siempre que relacionamos dos nudos i y j significa que estamos rellenando los datos de las submatrices de dimensiones 3×3 M_{ij} y M_{ji} (desplazamientos en las 3 direcciones del espacio). Esto hace que obtengamos una matriz simétrica distribuida en banda (valores distintos de cero alrededor de la diagonal principal y cero en el resto). El MEF muestra que el mayor o menor ancho de esa banda depende directamente de la diferencia $i-j$, por lo tanto la idea consiste en elegir un algoritmo de numeración de los nudos que asigne números próximos a cualesquiera dos nudos relacionados. Existen algoritmos como el de *Cuthill-McKee Inverso* [CO98, Ma96] que desaceleran el proceso de cálculo y no aseguran a priori que puedan encontrar la mejor ordenación, por lo que optamos por una regla nemotécnica que ofrezca una solución próxima a la ordenación óptima (véase anexo sobre minimización de ancho de banda y capítulo sobre algoritmos genéticos).

A partir de esta limitación hemos ensayado distintos modelos de numeración hasta elegir el siguiente orden de numeración:

- De derecha a izquierda
- De atrás a delante
- De arriba abajo

Mediante este método conseguimos obtener un ancho de banda de la matriz aceptablemente pequeño.

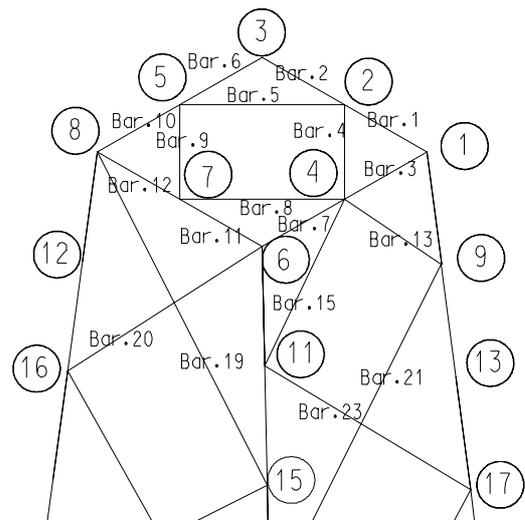


Ilustración 3-1. Numeración de nudos y barras

Otro problema que se presenta es el del almacenamiento de la matriz de rigidez en el ordenador, téngase en cuenta que las matrices de rigidez son cuadradas y tienen una dimensión igual al número de nudos de una estructura multiplicado por las tres dimensiones del espacio de trabajo, lo cual obliga a trabajar con matrices de tamaño 2000×2000 aproximadamente. Partiendo nuevamente del hecho de que la matriz es simétrica y en banda, se almacena un máximo de *Ancho de Banda* de cada fila (en las últimas será menor), para así optimizar el espacio ocupado [Oñ92].

El siguiente paso consiste en coaccionar los nudos señalando los grados de libertad que se permiten a cada uno, es decir, en qué direcciones del espacio le permitimos que tenga desplazamiento. Para ello identificamos con 0 cada dirección del nudo en la que lo dejamos libre y con 1 aquella que fijamos (o encastramos). Por eso todos los nudos de la estructura, salvo los cuatro que se fijan en el suelo y que son 111, están marcados por 000. En caso de haber cualquier otro tipo de limitación en una dirección determinada, p.ej. si se sujeta la torre por un lateral contra una montaña (es el caso de las torres de los teleféricos), colocaríamos también su correspondiente coacción.

Si n el número de nudos en la estructura y c el número de condiciones de carga, creamos un sistema de ecuaciones de la forma:

$$K_{3n \times 3n} X_{3n \times c} = C_{3n \times c}$$

donde K es la matriz de rigidez, X la de desplazamientos (a resolver) y C la de condiciones de carga aplicadas sobre los nudos de la estructura. Este método de construcción del sistema denominado “regla de ensamblaje” tiene su analogía en el caso de aplicación al equilibrio de circuitos eléctricos y es conocido como ley de Kirchoff.

3.3.2 Construcción de la Matriz

Para describir el procedimiento de ensamblaje vamos a establecer previamente la formulación teórica de cada caja de la matriz [Ar86], es decir, cada submatriz, y posteriormente el código de programación aplicado en el bloque de cálculo del sistema [IS00].

Descripción teórica

Cada grupo de cajas de la matriz simétrica de rigidez se distribuye de la forma siguiente:

$$\begin{array}{c} ij \\ \begin{bmatrix} P_x \\ P_y \\ P_z \\ M_x \\ M_y \\ M_z \\ \hline P_x \\ P_y \\ P_z \\ M_x \\ M_y \\ M_z \end{bmatrix} \end{array} = \begin{array}{c} \begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} & k_{15} & k_{16} & -k_{11} & -k_{12} & -k_{13} & k_{14} & k_{15} & k_{16} \\ & k_{22} & k_{23} & k_{24} & k_{25} & k_{26} & -k_{12} & -k_{22} & -k_{23} & k_{24} & k_{25} & k_{26} \\ & & k_{33} & k_{34} & k_{35} & k_{36} & -k_{13} & -k_{23} & -k_{33} & k_{34} & k_{35} & k_{36} \\ & & & k_{44} & k_{45} & k_{46} & -k_{14} & -k_{24} & -k_{34} & k_{47} & k_{48} & k_{49} \\ & & & & k_{55} & k_{56} & -k_{15} & -k_{25} & -k_{35} & k_{57} & k_{58} & k_{59} \\ & & & & & k_{66} & -k_{16} & -k_{26} & -k_{36} & k_{67} & k_{68} & k_{69} \end{bmatrix} \\ \hline \begin{bmatrix} k_{11} & k_{12} & k_{13} & -k_{14} & -k_{15} & -k_{16} \\ & k_{22} & k_{23} & -k_{24} & -k_{25} & -k_{26} \\ & & k_{33} & -k_{34} & -k_{35} & -k_{36} \\ & & & k_{44} & k_{45} & k_{46} \\ & & & & k_{55} & k_{56} \\ & & & & & k_{66} \end{bmatrix} \end{array} = \begin{array}{c} \begin{bmatrix} \Delta_x \\ \Delta_y \\ \Delta_z \\ \Theta_x \\ \Theta_y \\ \Theta_z \\ \hline \Delta_x \\ \Delta_y \\ \Delta_z \\ \Theta_x \\ \Theta_y \\ \Theta_z \end{bmatrix} \end{array} \begin{array}{c} i \\ \\ \\ \\ \\ \\ \\ j \end{array}$$

donde, para formar estos elementos k_{ab} partimos de las siguientes variables:

β es el ángulo de rotación del eje local y respecto al principal y_p

$$a = -\left(\frac{m}{D} \cos \beta + \frac{l \cdot n}{D} \operatorname{sen} \beta\right); \quad b = \frac{m}{D} \operatorname{sen} \beta - \frac{l \cdot n}{D} \cos \beta;$$

$$c = \frac{l}{D} \cos \beta - \frac{m \cdot n}{D} \operatorname{sen} \beta; \quad d = -\left(\frac{l}{D} \operatorname{sen} \beta + \frac{m \cdot n}{D} \cos \beta\right);$$

$$e = D \cdot \operatorname{sen} \beta; \quad f = D \cdot \cos \beta$$

En estas fórmulas l, m, n, D, a, b, c, d dependerán de la posición de la estructura respecto a los ejes de coordenadas². A partir de aquí tenemos que:

$$k_{11} = l^2 \frac{E \cdot F}{l_{ij}} + a^2 \cdot 12 \frac{E \cdot I_z}{l_{ij}^3} + b^2 \cdot 12 \frac{E \cdot I_y}{l_{ij}^3};$$

$$k_{12} = m \cdot l \frac{E \cdot F}{l_{ij}} + a \cdot c \cdot 12 \frac{E \cdot I_z}{l_{ij}^3} + b \cdot d \cdot 12 \frac{E \cdot I_y}{l_{ij}^3}; \quad k_{13} = n \cdot l \frac{E \cdot F}{l_{ij}} + a \cdot e \cdot 12 \frac{E \cdot I_z}{l_{ij}^3} + b \cdot f \cdot 12 \frac{E \cdot I_y}{l_{ij}^3};$$

$$k_{22} = m^2 \frac{E \cdot F}{l_{ij}} + c^2 \cdot 12 \frac{E \cdot I_z}{l_{ij}^3} + d^2 \cdot 12 \frac{E \cdot I_y}{l_{ij}^3}; \quad k_{33} = n \cdot m \frac{E \cdot F}{l_{ij}} + c \cdot e \cdot 12 \frac{E \cdot I_z}{l_{ij}^3} + d \cdot f \cdot 12 \frac{E \cdot I_y}{l_{ij}^3};$$

$$k_{14} = \frac{6 \cdot a \cdot b \cdot E}{l_{ij}^2} (I_y - I_z); \quad k_{24} = \frac{6 \cdot E}{l_{ij}^2} (a \cdot d \cdot I_y - b \cdot c \cdot I_z); \quad k_{15} = \frac{6 \cdot E}{l_{ij}^2} (b \cdot c \cdot I_y - a \cdot d \cdot I_z);$$

² Si el eje local x del nudo i coincide con el eje global Z del sistema de coordenadas, se tiene que:

$$l = 0; m = 0; n = 1; D = 0; a = -\operatorname{sen} \beta; b = -\cos \beta; c = \cos \beta; d = -\operatorname{sen} \beta$$

sin embargo, si el eje local x del nudo i coincide con el eje global negativo Z del sistema de coordenadas, entonces:

$$l = 0; m = 0; n = -1; D = 0; a = -\operatorname{sen} \beta; b = -\cos \beta; c = -\cos \beta; d = \operatorname{sen} \beta$$

$$k_{25} = \frac{6 \cdot c \cdot d \cdot E}{l_{ij}^2} (I_y - I_z); k_{16} = \frac{6 \cdot E}{l_{ij}^2} (b \cdot e \cdot I_y - a \cdot f \cdot I_z); k_{26} = \frac{6 \cdot E}{l_{ij}^2} (d \cdot e \cdot I_y - c \cdot f \cdot I_z);$$

$$k_{33} = n^2 \frac{E \cdot F}{l_{ij}} + e^2 \cdot 12 \frac{E \cdot I_z}{l_{ij}^3} + f^2 \cdot 12 \frac{E \cdot I_y}{l_{ij}^3};$$

$$k_{34} = \frac{6 \cdot E}{l_{ij}^2} (a \cdot f \cdot I_y - b \cdot e \cdot I_z); k_{35} = \frac{6 \cdot E}{l_{ij}^2} (c \cdot f \cdot I_y - d \cdot e \cdot I_z); k_{36} = \frac{6 \cdot e \cdot f \cdot E}{l_{ij}^2} (I_y - I_z)$$

$$k_{44} = l^2 \frac{G \cdot I_T}{l_{ij}} + a^2 \cdot 4 \frac{E \cdot I_y}{l_{ij}} + d^2 \cdot 4 \frac{E \cdot I_z}{l_{ij}};$$

$$k_{45} = m \cdot l \frac{G \cdot I_T}{l_{ij}} + a \cdot c \cdot 4 \frac{E \cdot I_y}{l_{ij}} + b \cdot d \cdot 4 \frac{E \cdot I_z}{l_{ij}}; k_{46} = n \cdot l \frac{G \cdot I_T}{l_{ij}} + a \cdot e \cdot 4 \frac{E \cdot I_y}{l_{ij}} + b \cdot f \cdot 4 \frac{E \cdot I_z}{l_{ij}};$$

$$k_{47} = -l^2 \frac{G \cdot I_T}{l_{ij}} + a^2 \cdot 2 \frac{E \cdot I_y}{l_{ij}} + b^2 \cdot 2 \frac{E \cdot I_z}{l_{ij}}; k_{48} = k_{57} = -m \cdot l \frac{G \cdot I_T}{l_{ij}} + a \cdot c \cdot 2 \frac{E \cdot I_y}{l_{ij}} + b \cdot d \cdot 2 \frac{E \cdot I_z}{l_{ij}}$$

$$k_{49} = k_{67} = -n \cdot l \frac{G \cdot I_T}{l_{ij}} + a \cdot e \cdot 2 \frac{E \cdot I_y}{l_{ij}} + b \cdot f \cdot 2 \frac{E \cdot I_z}{l_{ij}};$$

$$k_{55} = m^2 \frac{G \cdot I_T}{l_{ij}} + c^2 \cdot 4 \frac{E \cdot I_y}{l_{ij}} + d^2 \cdot 4 \frac{E \cdot I_z}{l_{ij}}; k_{56} = n \cdot m \frac{G \cdot I_T}{l_{ij}} + c \cdot e \cdot 4 \frac{E \cdot I_y}{l_{ij}} + d \cdot f \cdot 4 \frac{E \cdot I_z}{l_{ij}};$$

$$k_{58} = -m^2 \frac{G \cdot I_T}{l_{ij}} + c^2 \cdot 2 \frac{E \cdot I_y}{l_{ij}} + d^2 \cdot 2 \frac{E \cdot I_z}{l_{ij}};$$

$$k_{59} = k_{68} = -n \cdot m \frac{G \cdot I_T}{l_{ij}} + c \cdot e \cdot 2 \frac{E \cdot I_y}{l_{ij}} + d \cdot f \cdot 2 \frac{E \cdot I_z}{l_{ij}};$$

$$k_{69} = -n^2 \frac{G \cdot I_T}{l_{ij}} + e^2 \cdot 2 \frac{E \cdot I_y}{l_{ij}} + f^2 \cdot 2 \frac{E \cdot I_z}{l_{ij}};$$

siendo:

E el módulo de elasticidad (en el caso del acero 2100000)

G módulo de elasticidad transversal

I_y momento de inercia respecto del eje principal y

I_z momento de inercia respecto del eje principal z

I_T módulo de torsión

l_{ij} longitud de la barra entre los nudos i y j

y además $l = \frac{X_i - X_j}{l_{ij}}; m = \frac{Y_i - Y_j}{l_{ij}}; n = \frac{Z_i - Z_j}{l_{ij}}$

Código de programación

Cada conjunto de submatrices es colocado en un registro único de la matriz K , y así evitamos el hecho de almacenar los datos simétricos [CO98].

```
' Sean las variables:
' I1 el nudo inicial e I2 el nudo final de cada barra
' N0 el número de barras en la estructura
' CN grados de libertad de cada nudo (3 ó 6)
' U0 número de bloques (establecido a partir de la máxima memoria de CPU disponible)
' U2 número de nudos que incluye cada bloque
' U7 bloque actual
' Bar(A,1) la distancia entre los nudos inicial y final de cada barra
' Bar(A,2), Bar(A,3) y Bar(A,4) los cosenos directores
' Bar(A,5) Módulo de elasticidad multiplicado por el área de la barra
' Bar(A,6) Módulo de elasticidad multiplicado por el momento de inercia "Y"
' Bar(A,7) Módulo de elasticidad multiplicado por el momento de inercia "Z"
' Bar(A,8) Módulo de elasticidad multiplicado por el módulo de torsión
' Bar(A,9) Ángulo de torsión en radianes del eje local "y" respecto al eje global "Y"

For A = 0 To N0 - 1
  L1 = Bar(A, 1)
  L = Bar(A, 2): M = Bar(A, 3): N = Bar(A, 4)
  S = Bar(A, 5): Y7 = Int(Bar(A, 9)): D = Sqr(L ^ 2 + M ^ 2)

  If L=0 And M=0 Then
    N = Sgn(N)
    G1 = -Sin(Y7): G2 = -Cos(Y7): G3 = N * Cos(Y7)
    G4 = -N * Sin(Y7): G5 = 0: G6 = 0
  Else
    G1=- (M*Cos(Y7)+L*N*Sin(Y7))/D : G2=(M*Sin(Y7)-L*N*Cos(Y7))/D
    G3=(L*Cos(Y7)-M*N*Sin(Y7))/D : G4=- (L*Sin(Y7)+M*N*Cos(Y7))/D
    G5=D*Sin(Y7): G6=D*Cos(Y7)
  End If
  U = 12 * Bar(A, 6): V = 12 * Bar(A, 7): T = Bar(A, 8): L2 = L1 ^ 2: L3 = L1 ^ 3

  If I1 <= U2 * (U7 - 1) + U3 - 1 Then
    If I1>=U2 * (U7 - 1) Then
      P = CN * (I1 - U2 * (U7 - 1)): R = CN * (I2 - U2 * (U7 - 1))
      P1 = banda * (P + 0): P2 = banda * (P + 1): P3 = banda * (P + 2)
      P4 = banda * (P + 3): P5 = banda * (P + 4): P6 = banda * (P + 5)
      '-----
      K(P1 + 0) = K(P1 + 0) + L ^ 2 * S / L1 + (G1 ^ 2 * V + G2 ^ 2 * U) / L3
      K(P1 + 1) = K(P1 + 1) + M * L * S / L1 + (G1 * G3 * V + G2 * G4 * U) / L3
      K(P1 + 2) = K(P1 + 2) + N * L * S / L1 + (G1 * G5 * V + G2 * G6 * U) / L3
      K(P1 + 3) = K(P1 + 3) + G1 * G2 * (U - V) / (L2 * 2)
      K(P1 + 4) = K(P1 + 4) + (G2 * G3 * U - G1 * G4 * V) / (L2 * 2)
      K(P1 + 5) = K(P1 + 5) + (G2 * G5 * U - G1 * G6 * V) / (L2 * 2)
      '-----
      K(P1 + 0 + R - P) = -(L ^ 2 * S / L1 + (G1 ^ 2 * V + G2 ^ 2 * U) / L3)
      K(P1 + 1 + R - P) = -(M * L * S / L1 + (G1 * G3 * V + G2 * G4 * U) / L3)
      K(P1 + 2 + R - P) = -(N * L * S / L1 + (G1 * G5 * V + G2 * G6 * U) / L3)
      K(P1 + 3 + R - P) = G1 * G2 * (U - V) / (L2 * 2)
      K(P1 + 4 + R - P) = (G2 * G3 * U - G1 * G4 * V) / (L2 * 2)
      K(P1 + 5 + R - P) = (G2 * G5 * U - G1 * G6 * V) / (L2 * 2)
      '-----
      K(P2 + 0) = K(P2 + 0) + M ^ 2 * S / L1 + (G3 ^ 2 * V + G4 ^ 2 * U) / L3
      K(P2 + 1) = K(P2 + 1) + M * N * S / L1 + (G3 * G5 * V + G4 * G6 * U) / L3
      K(P2 + 2) = K(P2 + 2) + (G1 * G4 * U - G2 * G3 * V) / (L2 * 2)
      K(P2 + 3) = K(P2 + 3) + G3 * G4 * (U - V) / (L2 * 2)
      K(P2 + 4) = K(P2 + 4) + (G4 * G5 * U - G3 * G6 * V) / (L2 * 2)
```



```

-----
K(P2 - 1 + R - P) = K(P1 + 1 + R - P)
K(P2 + 0 + R - P) = -(M ^ 2 * S / L1 + (G3 ^ 2 * V + G4 ^ 2 * U) / L3)
K(P2 + 1 + R - P) = -(M * N * S / L1 + (G3 * G5 * V + G4 * G6 * U) / L3)
K(P2 + 2 + R - P) = (G1 * G4 * U - G2 * G3 * V) / (L2 * 2)
K(P2 + 3 + R - P) = G3 * G4 * (U - V) / (L2 * 2)
K(P2 + 4 + R - P) = (G4 * G5 * U - G3 * G6 * V) / (L2 * 2)
-----
K(P3 + 0) = K(P3 + 0) + N ^ 2 * S / L1 + (G5 ^ 2 * V + G6 ^ 2 * U) / L3
K(P3 + 1) = K(P3 + 1) + (G1 * G6 * U - G2 * G5 * V) / (L2 * 2)
K(P3 + 2) = K(P3 + 2) + (G3 * G6 * U - G4 * G5 * V) / (L2 * 2)
K(P3 + 3) = K(P3 + 3) + G5 * G6 * (U - V) / (L2 * 2)
-----
K(P3 - 2 + R - P) = K(P1 + 2 + R - P)
K(P3 - 1 + R - P) = K(P2 + 1 + R - P)
K(P3 + 0 + R - P) = -(N ^ 2 * S / L1 + (G5 ^ 2 * V + G6 ^ 2 * U) / L3)
K(P3 + 1 + R - P) = (G1 * G6 * U - G2 * G5 * V) / (L2 * 2)
K(P3 + 2 + R - P) = (G3 * G6 * U - G4 * G5 * V) / (L2 * 2)
K(P3 + 3 + R - P) = G5 * G6 * (U - V) / (L2 * 2)
-----
K(P4 + 0) = K(P4 + 0) + L ^ 2 * T / L1 + (G1 ^ 2 * U + G2 ^ 2 * V) / (3 * L1)
K(P4 + 1) = K(P4 + 1) + M * L * T / L1 + (G1 * G3 * U + G2 * G4 * V) / (3 * L1)
K(P4 + 2) = K(P4 + 2) + N * L * T / L1 + (G1 * G5 * U + G2 * G6 * V) / (3 * L1)
-----
K(P4 - 3 + R - P) = -K(P1 + 3 + R - P)
K(P4 - 2 + R - P) = -K(P2 + 2 + R - P)
K(P4 - 1 + R - P) = -K(P3 + 1 + R - P)
K(P4 + 0 + R - P) = -L ^ 2 * T / L1 + (G1 ^ 2 * U + G2 ^ 2 * V) / (6 * L1)
K(P4 + 1 + R - P) = -M * L * T / L1 + (G1 * G3 * U + G2 * G4 * V) / (6 * L1)
K(P4 + 2 + R - P) = -N * L * T / L1 + (G1 * G5 * U + G2 * G6 * V) / (6 * L1)
-----
K(P5 + 0) = K(P5 + 0) + M ^ 2 * T / L1 + (G3 ^ 2 * U + G4 ^ 2 * V) / (3 * L1)
K(P5 + 1) = K(P5 + 1) + M * N * T / L1 + (G3 * G5 * U + G4 * G6 * V) / (3 * L1)
-----
K(P5 - 4 + R - P) = -K(P1 + 4 + R - P)
K(P5 - 3 + R - P) = -K(P2 + 3 + R - P)
K(P5 - 2 + R - P) = -K(P3 + 2 + R - P)
K(P5 - 1 + R - P) = K(P4 + 1 + R - P)
K(P5 + 0 + R - P) = -M ^ 2 * T / L1 + (G3 ^ 2 * U + G4 ^ 2 * V) / (6 * L1)
K(P5 + 1 + R - P) = -M * N * T / L1 + (G3 * G5 * U + G4 * G6 * V) / (6 * L1)
-----
K(P6 + 0) = K(P6 + 0) + N ^ 2 * T / L1 + (G5 ^ 2 * U + G6 ^ 2 * V) / (3 * L1)
-----
K(P6 - 5 + R - P) = -K(P1 + 5 + R - P)
K(P6 - 4 + R - P) = -K(P2 + 4 + R - P)
K(P6 - 3 + R - P) = -K(P3 + 3 + R - P)
K(P6 - 2 + R - P) = K(P4 + 2 + R - P)
K(P6 - 1 + R - P) = K(P5 + 1 + R - P)
K(P6 + 0 + R - P) = -N ^ 2 * T / L1 + (G5 ^ 2 * U + G6 ^ 2 * V) / (6 * L1)
-----
End If
If I2 >= U2 * (U7 - 1) And I2 > U2 * (U7 - 1) + U3 - 1 Then
  R = CN * (I2 - U2 * (U7 - 1))
  R1 = banda * (R + 0): R2 = banda * (R + 1): R3 = banda * (R + 2)
  R4 = banda * (R + 3): R5 = banda * (R + 4): R6 = banda * (R + 5)
-----
K(R1 + 0) = K(R1 + 0) + L ^ 2 * S / L1 + (G1 ^ 2 * V + G2 ^ 2 * U) / L3
K(R1 + 1) = K(R1 + 1) + M * L * S / L1 + (G1 * G3 * V + G2 * G4 * U) / L3
K(R1 + 2) = K(R1 + 2) + N * L * S / L1 + (G1 * G5 * V + G2 * G6 * U) / L3
K(R1 + 3) = K(R1 + 3) - G1 * G2 * (U - V) / (L2 * 2)
K(R1 + 4) = K(R1 + 4) - (G2 * G3 * U - G1 * G4 * V) / (L2 * 2)
K(R1 + 5) = K(R1 + 5) - (G2 * G5 * U - G1 * G6 * V) / (L2 * 2)
-----

```

```

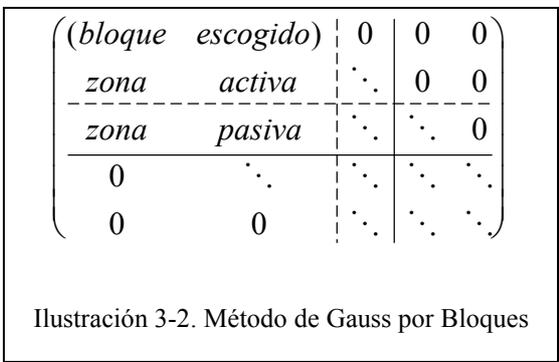
K(R2 + 0) = K(R2 + 0) + M ^ 2 * S / L1 + (G3 ^ 2 * V + G4 ^ 2 * U) / L3
K(R2 + 1) = K(R2 + 1) + M * N * S / L1 + (G3 * G5 * V + G4 * G6 * U) / L3
K(R2 + 2) = K(R2 + 2) - (G1 * G4 * U - G2 * G3 * V) / (L2 * 2)
K(R2 + 3) = K(R2 + 3) - G3 * G4 * (U - V) / (L2 * 2)
K(R2 + 4) = K(R2 + 4) - (G4 * G5 * U - G3 * G6 * V) / (L2 * 2)
'-----
K(R3 + 0) = K(R3 + 0) + N ^ 2 * S / L1 + (G5 ^ 2 * V + G6 ^ 2 * U) / L3
K(R3 + 1) = K(R3 + 1) - (G1 * G6 * U - G2 * G5 * V) / (L2 * 2)
K(R3 + 2) = K(R3 + 2) - (G3 * G6 * U - G4 * G5 * V) / (L2 * 2)
K(R3 + 3) = K(R3 + 3) - G5 * G6 * (U - V) / (L2 * 2)
'-----
K(R4 + 0) = K(R4 + 0) + L ^ 2 * T / L1 + (G1 ^ 2 * U + G2 ^ 2 * V) / (3 * L1)
K(R4 + 1) = K(R4 + 1) + M * L * T / L1 + (G1 * G3 * U + G2 * G4 * V) / (3 * L1)
K(R4 + 2) = K(R4 + 2) + N * L * T / L1 + (G1 * G5 * U + G2 * G6 * V) / (3 * L1)
'-----
K(R5 + 0) = K(R5 + 0) + M ^ 2 * T / L1 + (G3 ^ 2 * U + G4 ^ 2 * V) / (3 * L1)
K(R5 + 1) = K(R5 + 1) + M * N * T / L1 + (G3 * G5 * U + G4 * G6 * V) / (3 * L1)
'-----
K(R6 + 0) = K(R6 + 0) + N ^ 2 * T / L1 + (G5 ^ 2 * U + G6 ^ 2 * V) / (3 * L1)
'-----
End If
End If
Next A

```

3.3.3 Resolución del Sistema de Ecuaciones

Para resolver este sistema podemos invertir la matriz de rigidez (evidentemente esto es impensable para matrices de tamaño tan considerable como aquí tenemos), utilizar algún método directo (Gauss, Cholesky) o algún método iterativo (Jacobi, Gauss-Seidel, Super-Relajación Sucesiva) [MF00, Mo86]. En nuestro caso hemos elegido un método directo para no tener que ir estimando los errores de aproximación ni realizar muchas lecturas de disco. Como en cualquier sistema utilizado las dimensiones de las matrices están limitadas, hemos implementado una modificación sobre el método de Gauss, llamado *Gauss por bloques*, de tal manera que en cada paso (bloque) [BF98]:

1. Se recoge un número de ecuaciones (zona activa) para hacer ceros bajo la diagonal principal.
2. También se traen las ecuaciones que, por debajo de las anteriores, se ven afectadas al hacerle ceros (zona pasiva).
3. Entre ambos conjuntos se ocupa la memoria permitida por el ordenador.



4. Se hacen cero los términos por debajo de la diagonal a partir de los datos que hay por encima (que los tenemos almacenados y son simétricos) y los almacenamos, pasando al siguiente bloque de datos.

Este método permite resolver el sistema de una sola pasada (es decir, con acceso mínimo a disco) sin necesidad de mantener un controlador de errores, puesto que todas las operaciones son lineales y usamos valores en milímetros con números reales de doble precisión, con lo cual el error acumulado no sobrepasa las milimicras.

Una vez calculado el triángulo de ceros bajo la diagonal, el cual nos ha ido modificando paralelamente los valores de las cargas para las diferentes condiciones, se resuelve el sistema y obtenemos los valores de X , es decir, los desplazamientos de cada uno de los nudos en las diferentes direcciones del espacio.

3.3.4 Ejemplo de Gauss por Bloques

A continuación desarrollamos un ejemplo de simplificación de una matriz por el método de Gauss por bloques. En la primera columna indicamos cómo se van haciendo ceros por el método clásico de Gauss. En la segunda observamos los datos que tenemos guardados en el disco duro y en la tercera tenemos los datos que permanecen en memoria RAM con los cuales realizaremos las operaciones. Supongamos una matriz 7×7 con una banda de ancho 4 y donde el sistema trabaja con matrices de, a lo sumo, 8 datos:

Datos Originales en la matriz								Datos almacenados en disco				Datos en memoria (8+8)			
Banda 4. Zona activa=2 (Filas 1,2). Zona Pasiva=2 (Filas 3,4). Pivotará sobre fila 1															
1	2	3	5	0	0	0		1	2	3	5	1	2	3	5
2	2	1	7	0	0	0		2	1	7	0	2	1	7	0
3	1	1	4	3	2	0		1	4	3	2	1	4	3	2
5	7	4	1	2	6	1		1	2	6	1	1	2	6	1
0	0	3	2	1	5	8		1	5	8	0				
0	0	2	6	5	1	3		1	3	0	0				
0	0	0	1	8	3	1		1	0	0	0				
Banda 4. Zona activa=2 (Filas 2,3). Zona Pasiva=2 (Filas 4,5). Pivotará sobre fila 2															
1	2	3	5	0	0	0		1	2	3	5				
0	-2	-5	-3	0	0	0		-2	-5	-3	0	-2	-5	-3	0
0	-5	-8	-11	3	2	0		-8	-11	3	2	-8	-11	3	2
0	-3	-11	-24	2	6	1		-24	2	6	1	-24	2	6	1
0	0	3	2	1	5	8		1	5	8	0	1	5	8	0
0	0	2	6	5	1	3		1	3	0	0				
0	0	0	1	8	3	1		1	0	0	0				
Banda 4. Zona activa=2 (Filas 3,4). Zona Pasiva=2 (Filas 5,6). Pivotará sobre fila 3															
1	2	3	5	0	0	0		1	2	3	5				
0	-2	-5	-3	0	0	0		-2	-5	-3	0				
0	0	4,5	-3,5	3	2	0		4,5	-3,5	3	2	4,5	-3,5	3	2
0	0	-3,5	-19,5	2	6	1		-19,5	2	6	1	-19,5	2	6	1
0	0	3	2	1	5	8		1	5	8	0	1	5	8	0
0	0	2	6	5	1	3		1	3	0	0	1	3	0	0
0	0	0	1	8	3	1		1	0	0	0				
Banda 4. Zona activa=2 (Filas 4,5). Zona Pasiva=2 (Filas 6,7). Pivotará sobre fila 4															
1	2	3	5	0	0	0		1	2	3	5				
0	-2	-5	-3	0	0	0		-2	-5	-3	0				
0	0	4,5	-3,5	3	2	0		4,5	-3,5	3	2				
0	0	0	-22,222	4,3333	7,5556	1		-22,222	4,3333	7,5556	1	-22,222	4,3333	7,5556	1
0	0	0	4,3333	-1	3,6667	8		-1	3,6667	8	0	-1	3,6667	8	0
0	0	0	7,5556	3,6667	0,1111	3		0,1111	3	0	0	0,1111	3	0	0
0	0	0	1	8	3	1		1	0	0	0	1	0	0	0
Banda 4. Zona activa=2 (Filas 4,5). Zona Pasiva=2 (Filas 6,7). Pivotará sobre fila 5															
1	2	3	5	0	0	0		1	2	3	5				
0	-2	-5	-3	0	0	0		-2	-5	-3	0				
0	0	4,5	-3,5	3	2	0		4,5	-3,5	3	2				
0	0	0	-22,222	4,3333	7,5556	1		-22,222	4,3333	7,5556	1	-22,222	4,3333	7,5556	1
0	0	0	0	-0,155	5,14	8,195		-0,155	5,14	8,195	0	-0,155	5,14	8,195	0
0	0	0	0	5,14	2,68	3,34		2,68	3,34	0	0	2,68	3,34	0	0
0	0	0	0	8,195	3,34	1,045		1,045	0	0	0	1,045	0	0	0
Banda 4. Zona activa=2 (Filas 4,5). Zona Pasiva=2 (Filas 6,7). Pivotará sobre fila 6															
1	2	3	5	0	0	0		1	2	3	5				
0	-2	-5	-3	0	0	0		-2	-5	-3	0				
0	0	4,5	-3,5	3	2	0		4,5	-3,5	3	2				
0	0	0	-22,222	4,3333	7,5556	1		-22,222	4,3333	7,5556	1	-22,222	4,3333	7,5556	1
0	0	0	0	-0,155	5,14	8,195		-0,155	5,14	8,195	0	-0,155	5,14	8,195	0
0	0	0	0	0	173,13	275,1		173,13	275,1	0	0	173,13	275,1	0	0
0	0	0	0	0	275,1	434,32		434,32	0	0	0	434,32	0	0	0
Banda 4. Zona activa=2 (Filas 4,5). Zona Pasiva=2 (Filas 6,7). Finaliza															
1	2	3	5	0	0	0		1	2	3	5				
0	-2	-5	-3	0	0	0		-2	-5	-3	0				
0	0	4,5	-3,5	3	2	0		4,5	-3,5	3	2				
0	0	0	-22,222	4,3333	7,5556	1		-22,222	4,3333	7,5556	1	-22,222	4,3333	7,5556	1
0	0	0	0	-0,155	5,14	8,195		-0,155	5,14	8,195	0	-0,155	5,14	8,195	0
0	0	0	0	0	173,13	275,1		173,13	275,1	0	0	173,13	275,1	0	0
0	0	0	0	0	0	-2,7978		-2,7978	0	0	0	-2,7978	0	0	0



3.4 Modelización y cálculo de TTE

Una vez introducidos los datos correspondientes a las caras transversal y longitudinal de la torre, como paso previo a la aplicación del MEF, se necesita generar un modelo espacial a partir de los citados datos. Es decir, hemos de definir un conversor espacial que procese los datos introducidos para las caras y produzca un modelo tridimensional.

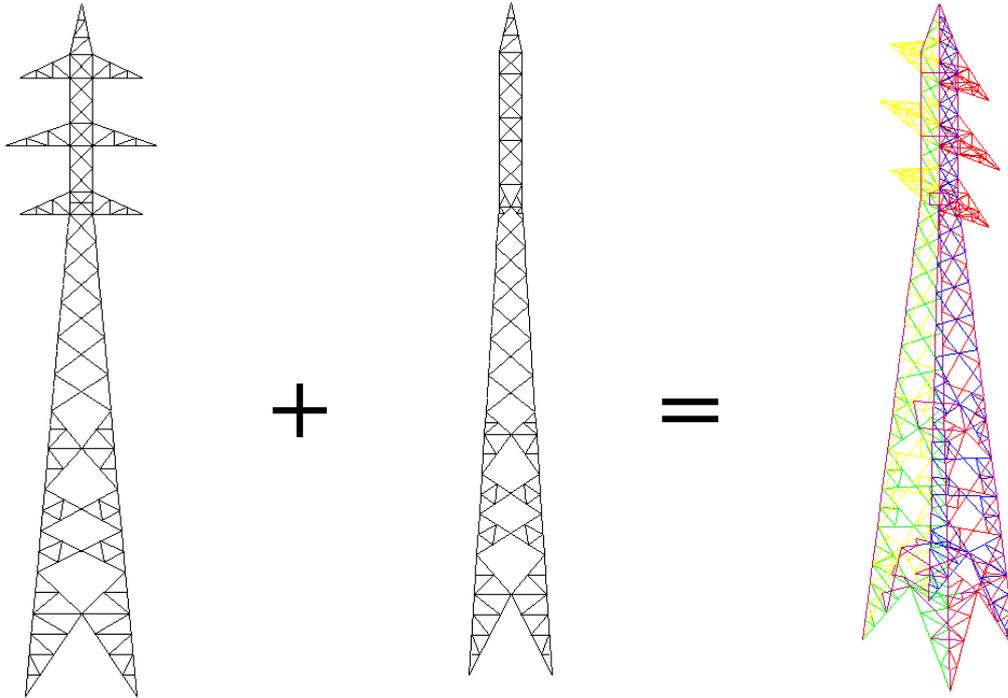


Ilustración 3-3. Conversión de entrada bidimensional en estructura tridimensional

Como primera cuestión se plantea cómo representar una estructura espacial en una superficie bidimensional (la pantalla del ordenador) para ello fue desarrollada la siguiente transformación matricial:

Sea (X_0, Y_0, Z_0) el punto original en el espacio y (X, Y) su representación en el plano. Sean $\vec{u}, \vec{v}, \vec{w}$ los vectores de la base en 3D y $\vec{u}_{2D}, \vec{v}_{2D}, \vec{w}_{2D}$ dichos vectores representados respecto a la base canónica en 2D. Entonces los valores de (X, Y) nacen de la transformación:

$$(X, Y) = (X_0, Y_0, Z_0) \cdot \begin{pmatrix} \vec{u}_{2D} \\ \vec{v}_{2D} \\ \vec{w}_{2D} \end{pmatrix}$$

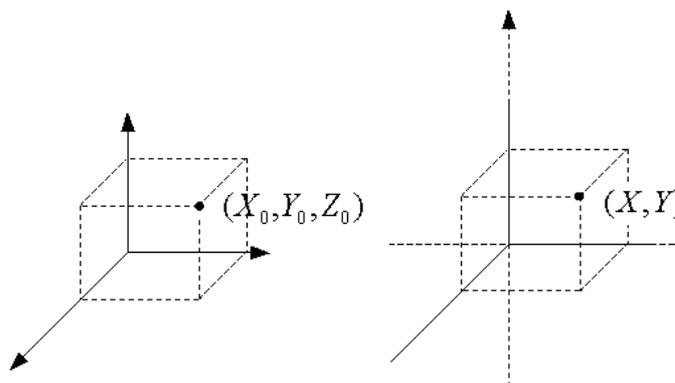


Ilustración 3-4. Visualización 3D (realidad) sobre 2D (pantalla)

En las siguientes secciones se detallan cada una de las fases por las que transcurre esta modelización espacial de la estructura, usando las herramientas de programación del lenguaje para Windows *Visual Basic* [Ce94].

3.4.1 Cálculo de Puntos Básicos de la Torre

Dado que en el diseño de una línea lo que se busca es una geometría lo más homogénea posible para todas las torres que la componen, se suele comenzar por considerar la de mayor altura y añadir o quitar aquellas subestructuras necesarias para generar todas y cada una de las alturas diferentes de la torre que permitan mantener las distancias de seguridad ante cualquier accidente del terreno (colina, valle, ladera de montaña, etc.). El número de diferentes alturas lo encontramos almacenado en la variable

Niveles de extensión

que se ha definido durante la entrada de datos. A partir del cálculo de la torre mayor estableceremos el viento global sobre la estructura para, posteriormente, calcular el resto de alturas asumiendo las fuerzas de viento correspondientes ya calculadas.

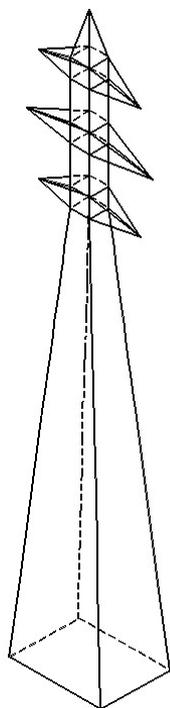


Ilustración 3-5.
Cambios de
tangente

Una vez elegida una altura determinada, comprobamos a qué niveles se encuentran los cambios de tangente de la estructura, para desarrollarla espacialmente mediante sencillos cálculos vectoriales, obteniendo los diferentes troncos piramidales superpuestos que van a conformar básicamente la estructura, como podemos observar en este

gráfico.

Cada uno de los puntos espaciales donde confluyan barras los llamaremos *nudos* de la estructura, por empezar a usar la nomenclatura del método de los elementos finitos, por lo que podemos definir ya los conceptos de:

- **Mallado de la estructura:** Abstracción de las estructuras mediante líneas imaginarias que simulan el eje de gravedad de las piezas que la conforman.
- **Barra :** Nomenclatura del cálculo estructural para esas líneas imaginarias.
- **Nudo :** Nombre que recibe cada uno de los extremos de las barras.
- **Carga :** Fuerzas representadas vectorialmente respecto a XYZ que se aplican sobre los nudos.

La declaración de la estructura Nudo es la siguiente:

```
Type  nudo
      x as double  (Coordenada x en el espacio)
      y as double  (Coordenada y en el espacio)
      z as double  (Coordenada z en el espacio)
      Código as long (Código de reconocimiento del nudo)
      Tipo as integer (Tipo de nudo para ser cargado o no)
End Type
```

Como se observa, el campo código tiene tipo *long integer* que en la práctica tomara valores de seis dígitos interpretados en tres parejas (## - ## - ##) donde:

- La primera pareja indica el panel sobre la estructura.
- La segunda, la cara a la que pertenece el nudo (1,2,3 ó 4).
- La tercera, el número dentro de esa cara que le corresponde (ver numeración en Tipos de Paneles).

El campo tipo indica si ese nudo es punta de cruceta, pico de tierra, posición de viento en la estructura (estos tres tipos poseerán cargas introducidas en el archivo de entrada), alguno de los cuatro nudos que limitan las caras de la torre (en ellos se aplicará el peso preliminar proporcional a la altura a que se encuentren), o bien se trata de un nudo sin carga.

La declaración de la estructura Barra es la siguiente:

```
Type  barra
      Panel as integer      (Panel al que pertenece)
      Códigoinicial as long (Código de nudo del extremo inicial)
      CódigoFinal as long  (Código de nudo del extremo final)
      Código as single      (Tipo de barra en la estructura)
      Sección as integer    (Tipo de material que a priori tuviese asignado
                           la barra, aunque posteriormente sea cambiado)
      Nombre as String      (Nombre habitual que recibe la barra)
End Type
```

El campo código es un número real *single* leído como parte entera y decimal (##.##) donde:

- La primera indica el tipo de barra que es dentro de la codificación de barras de cada tipo de torre (p.ej. 1 es un montante, 2 es una diagonal, 3 es un secundario o relleno de diagonales, etc.)
- La segunda indica la pieza o posición que ocupa dentro del mismo panel (p.ej. diagonal 1ª, diagonal 2ª, secundario 4º, etc.)

Gracias a esta codificación podremos posteriormente (después del cálculo) clasificar los tipos de barra que coincidan en panel, código y caras en un único grupo, al cual se le asignará el material necesario para su fabricación a partir de las condiciones más adversas que deba soportar.

La declaración de la estructura Carga es la siguiente:

```
Type  carga
      Hipótesis as integer  (Hipótesis de carga que le corresponde)
      Nudo as integer       (Nudo sobre el que aplicará la carga)
      x as double           (Coordenada x del vector de fuerza)
      y as double           (Coordenada y del vector de fuerza)
      z as double           (Coordenada z del vector de fuerza)
End Type
```

3.4.2 Desarrollo Panel a Panel

Una vez distribuida la estructura en las diferentes secciones troncopiramidales que la componen, pasamos a subdividir las en paneles, a partir de codificación introducida en el capítulo anterior. Esto permite reducir el problema a la mínima unidad de trabajo (el panel) para posteriormente unir los resultados obtenidos y resolver el problema global (método divide y vencerás).

Sea n el número de paneles diferentes en la estructura. Establecemos dentro de las secciones los siguientes límites, para i variando entre el primer panel de la estructura (1) y el último (n):

- Altura superior del panel i
- Altura inferior del panel i
- Base superior del panel i
- Base inferior del panel i

Esto lo hacemos tanto en la cara transversal como en la longitudinal, sabiendo que las caras enfrentadas son diseñadas de igual forma dos a dos. Estos límites son incorporados dentro de los campos correspondientes en la estructura panel:

```
Type Panel
  Tramo as integer
  Tipo as integer
  AlturaSuperior as double
  AnchoSuperior as double
  AlturaInferior as double
  AnchoInferior as double
  Relleno as integer
End type
```

Partiendo desde el punto superior de la torre (panel número 1), vamos generando todos los nudos y barras de las cuatro caras de los troncos piramidales, guardando los resultados en los archivos temporales

```
Torre.Nud
Torre.Bar
```

El orden en el que se sitúan los nudos, a partir de las consideraciones realizadas en la sección 3.3.1 sobre minimización de ancho de banda, será el:

- 1° de derecha a izquierda
- 2° de atrás a delante
- 3° de arriba hacia abajo

La experiencia nos ha demostrado que esta ordenación permite obtener anchos de banda aceptables de la matriz de rigidez, que posteriormente será usada en el MEF y clasificada como matriz simétrica en banda. La única excepción que se aplica a esta ordenación se produce en la torre atirantada, donde los nudos de cogida de los cables al suelo se numeran en la siguiente posición al nudo de cogida sobre la torre (de lo contrario, la distancia de índice entre el nudo de cogida a tierra y a la torre aumentaría la banda de cálculo y consecuentemente el tiempo de cálculo).

3.4.3 Barras de Cálculo y Barras Secundarias

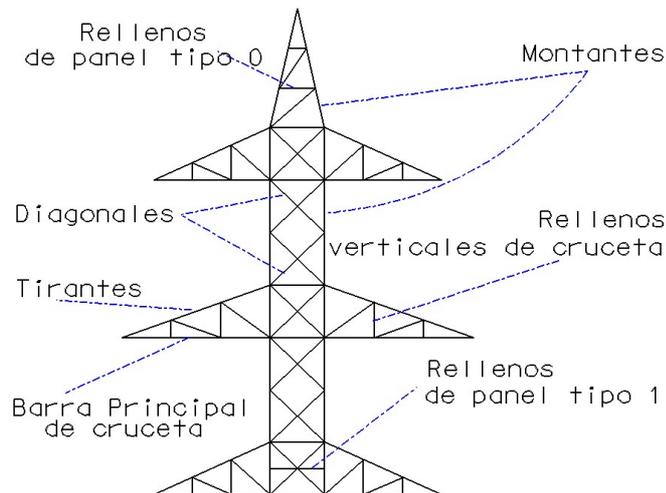


Ilustración 3-6. Nomenclatura de barras

Durante el desarrollo de los nudos y de las barras, hemos de distinguir entre barras principales, que generan nudos fundamentales para el cálculo de la estructura, y barras de relleno, que solamente se utilizan para disminuir las longitudes de pandeo de las principales (ver anexo sobre normativa ASCE para la distinción de tipos de barras).

Estas barras son las que llamamos simplemente rellenos. Estos rellenos pudieran o no ser barras de cálculo: por ejemplo, si pertenecen a un triángulo limitado por barras principales (montantes, diagonales, crucetas, etc.) podemos

prescindir de ellos como barras de cálculo, puesto que las fuerzas que soportan son nulas. No es este el caso de las que pertenecen al interior de un trapecio, como ocurre con los rellenos del plano horizontal, también llamado de cortadura, de una cruceta abierta por su punta. Por tanto tenemos esta clasificación:

- | | |
|--------------------------|-------------------------------|
| ➤ Barras Principales | ➤ Barras de Relleno |
| e) Montantes | a) Rellenos de paneles |
| f) Diagonales | b) Rellenos de crucetas |
| g) Recuadros | c) Rellenos de pico de tierra |
| h) Montante de Cruceta | d) Internos de patas |
| i) Tirante de Cruceta | |
| j) Paneles de transición | |

El considerar solamente las barras principales para cálculo es un hecho de suma importancia, ya que disminuye las dimensiones de las matrices con las vamos a trabajar y, por tanto, va a acelerar nuestros cálculos.

3.4.4 Compresión de Datos

Una vez completados los archivos de nudos y barras que componen la estructura, al haber sido hecho panel a panel por separado (lo cual disminuye la memoria usada y acelera los procesos) hemos de pasar a la compresión de datos para eliminar las repeticiones de nudos (ya que cada panel comparte los nudos superiores con el anterior y los inferiores con el posterior). Para realizar la compresión lo más rápidamente posible, sin necesidad de leer por completo los archivos de nudos y barras (pues pueden ser tantos que colapsen la memoria del ordenador), usaremos un sistema basado en el concepto de *medias móviles* de la teoría de Series Temporales, observando que:

Si para todo $i = \{1, \dots, n\}$ se cumple que:

$$\text{Panel}(i, \text{Transversal}).\text{AlturaSuperior} = \text{Panel}(i, \text{Longitudinal}).\text{AlturaSuperior}$$

y además:

$$\text{Panel}(i, \text{Transversal}).\text{AlturaInferior} = \text{Panel}(i, \text{Longitudinal}).\text{AlturaInferior}$$

bastará cargar los datos de los paneles de dos en dos, es decir, tomamos 1º y 2º, después 2º y 3º, y así sucesivamente hasta llegar a tomar $(n-1)$ -ésimo y n -ésimo. Por tanto se realizan $n-1$ paquetes de compresión.

Sin embargo, si no se da alguna de estas igualdades, y en tal caso se dice que la torre está al *tresbolillo*, aplicamos las medias móviles de orden 3, es decir, tomamos 1º, 2º y 3º, después 2º, 3º y 4º hasta llegar a $(n-2)$ -ésimo, $(n-1)$ -ésimo y n -ésimo. Total, $n-2$ paquetes de compresión.

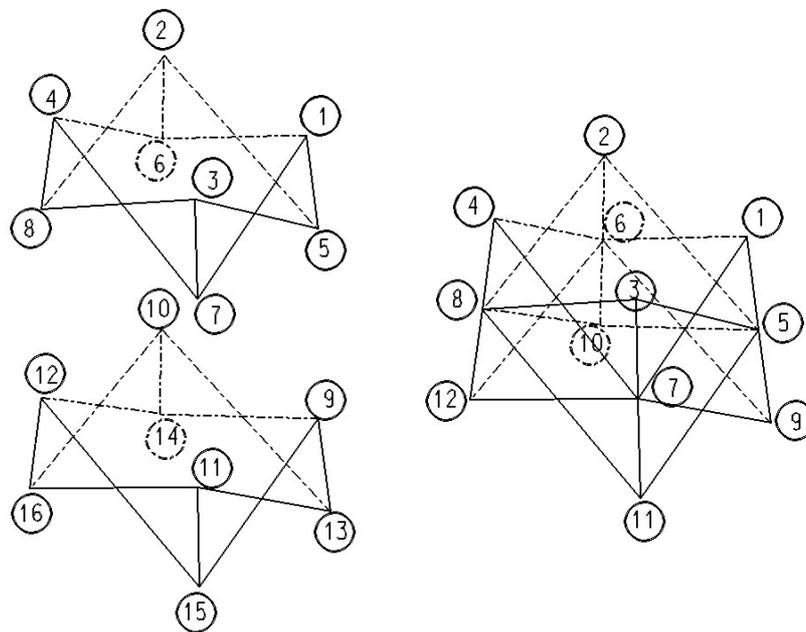


Ilustración 3-7. Compresión de nudos

En este proceso de compresión se ejecutan las fases siguientes:

1. Unificación de los casos de nudos iguales (comparando sus coordenadas).
2. Si se han igualados dos nudos, uno de los cuales no va a recibir carga y otro sí, consideramos el nuevo nudo como un nudo con posibilidad de carga.
3. Se modifica la numeración de los nudos de cada barra.
4. Se anulan las repeticiones de barras iguales (si las hubiere).

3.4.5 Nudos, Barras y Cargas

Una vez comprimidos los datos se obtiene lo que se denomina en el campo de la ingeniería “diseño de la topología de la estructura”, y está formado por:

Torre.N1 : Nudos de la estructura

Torre.B1 : Barras de cálculo de la estructura

Torre.B2 : Barras de relleno de la estructura

y con ellos ya es posible añadir las cargas sobre los nudos correspondientes para comenzar el cálculo. Con este fin iremos comparando:

- La carga que debemos aplicar (datos introducidos en la entrada)
- El tipo de nudo (con posibilidad de carga o no)
- Las coordenadas que posee el nudo (para ver si es la punta de una cruceta, un viento o un borde de la torre)

Comprobados y aceptados estos datos, se genera un archivo con todas las cargas (condiciones, nudos a cargar y fuerzas en los sentidos XYZ) denominado:

Torre.C1 : Cargas sobre los nudos de la estructura

Estos datos ya permiten aplicar el Método de los Elementos Finitos para calcular la estructura. De sus resultados es posible obtener los desplazamientos de los nudos, las fuerzas en las barras y las reacciones al nivel del suelo. El análisis de los mismos nos permitirá comprobar si la estructura estudiada es estable o no.

3.4.6 Cono de Desplazamientos

Realizado el cálculo de los desplazamientos de la estructura, nos detenemos un momento para analizar dichos desplazamientos.

Un error habitual en el diseño suele ser el no haber previsto la sujeción de algún nudo en una de las direcciones del espacio. Esto provocará una *inestabilidad local o global* dentro de la propia estructura, y en el caso de que aceptáramos los resultados obtenidos nos podría llevar a unas fuerzas enormes y, consiguientemente, a la aceptación de una estructura errónea.

Por tanto es necesario imponer unas limitaciones a los movimientos o desplazamientos de la misma.

Inestabilidad Global: en primer lugar se comprueba que el nudo o los nudos superiores de la estructura no se desplazan más del 2% de la altura total de la misma (este porcentaje es una recomendación basada en la experiencia sobre torres que poseen las estaciones de ensayo y es sólo aplicable a aquellas estructuras que vayan a ser utilizadas para transmitir electricidad). En caso contrario, se avisa al diseñador del porcentaje de movimiento que ha efectuado la estructura y se le pide si desea aceptarla o no.

Un movimiento excesivo de la parte superior nos indica la inestabilidad global de la estructura. En este caso es conveniente comprobar si se ha hecho demasiado estrecha la base inferior de la estructura, pudiendo resolverlo con un simple aumento de dicho valor.

Inestabilidad Local: Posteriormente se establece un cono invertido cuya altura es igual a la de la torre, y cuyo radio es igual al desplazamiento máximo de los nudos superiores.

Este cono nos delimita los márgenes de desplazamiento de todos los nudos desde la parte alta hasta el suelo. Si el movimiento de algún nudo no estuviese en el interior del cono en el lugar que le corresponde a su altura, la estructura en dicho punto sería inestable, y debe revisarse los datos que han sido introducidos para averiguar qué nudo no está soportado y en qué dirección del espacio ocurre esto.

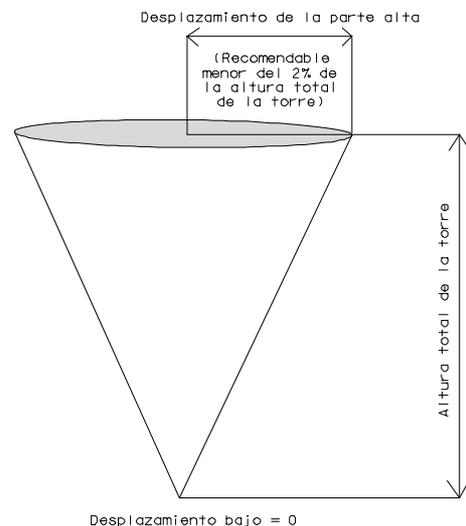


Ilustración 3-8. Cono de desplazamientos

Una vez aceptados los desplazamientos por el sistema, estos datos son recogidos en la estructura:

Type desplazamiento
 Hipótesis as integer (Hipótesis de carga que le corresponde)
 Nudo as integer (Nudo sobre el que aplicará la carga)
 x as double (Coordenada x del desplazamiento del nudo)
 y as double (Coordenada y del desplazamiento del nudo)
 z as double (Coordenada z del desplazamiento del nudo)
 End Type

y guardados en el archivo:

Torre.D1 : Desplazamiento de los nudos de la estructura

3.4.7 Fuerzas, Reacciones y Precálculo

A partir de los desplazamientos, ya es posible calcular las fuerzas axiales que soporta cada una de las barras viendo cómo se han distanciado (tensión) o acercado (compresión) sus extremos, a partir de la fórmula de la notación de Timoshenko ya expuesta:

$$N = (EA)^i \frac{u^i_2 - u^i_1}{l^i} .$$

Análogamente se calculan las reacciones que ejerce cada nudo de la estructura para evitar ser desplazado (*Principio de acción-reacción*). Van a ser especialmente interesantes las reacciones ejercidas por los nudos empotrados en el nivel inferior del último panel, ya que, a partir de ellas y del tipo de suelo donde se apoyará la torre, se podrán calcular las cimentaciones que necesita.

Tenemos las estructuras:

Type Fuerza
 Hipótesis as integer (Hipótesis de carga que le corresponde)
 Barra as integer (Barra sometida a esta fuerza)
 Valor as double (Valor en unidades de fuerza Kg ó daN)
 End Type

Type	Reacción	
	Hipótesis as integer	(Hipótesis de carga que le corresponde)
	Nudo as integer	(Nudo sobre el que aplica la reacción)
	x as double	(Coordenada x de la reacción del nudo)
	y as double	(Coordenada y de la reacción del nudo)
	z as double	(Coordenada z de la reacción del nudo)
End Type		

Estos datos son recogidos en:

Torre.AX : Fuerzas axiales que soportan las barras de la estructura

Torre.R1 : Reacciones de los nudos al nivel del suelo de la estructura

Nótese que la elección del material se obtiene mediante el procedimiento recursivo que vemos a continuación.

3.4.8 Viento sobre la Estructura

Para realizar el cálculo mediante el MEF se necesitan, a la vez que los nudos con sus relaciones (barras) y las cargas aplicadas, unas secciones de material que, aunque de forma preliminar, nos permitan establecer el módulo de elasticidad y el área de cada pieza para un primer ensamblaje de la matriz de rigidez de la estructura. Estas secciones se eligen en función de la longitud de dichas barras, sin introducirles cargas de viento (ya que aun desconocemos las cargas finales que van a soportar). Como uno de los datos de entrada solicitados es la presión que el viento ejerce sobre cada metro cuadrado de estructura, tanto en la cara transversal como longitudinal, se puede realizar un primer cálculo del empuje del viento.

Cuando la estructura es calculada, resolvemos nuevamente las cargas de viento sobre ella y así, sucesivamente, hasta que la diferencia entre un cálculo de empujes de viento y el siguiente se aproximen por encima del 95% en sus valores (tenemos errores inferiores al 5% en el viento sobre la estructura, lo que consideramos suficiente). El algoritmo tendría esta expresión:

```
TopologiaEstructura (Nudos() as nudo, Barras() as barra)
SeccionesIniciales (Barras() as barra)
VientoInicial=0 : VientoFinal = CalculoViento (Barras() as barra)
While (ABS(VientoInicial-VientoFinal)>5%)
    CalculoDesplazamientos (Nudos() as nudo, Barras() as barra, Cargas() as carga,
    Desplazamientos() as desplazamiento)
    CalculoFuerzas (Barras() as Barra, Desplazamientos() as Desplazamiento,
    Fuerzas() as fuerza)
    CalculoReacciones (Nudos() as nudo, Barras() as barra, Cargas() as carga,
    Desplazamientos() as desplazamiento, Reacciones() as reaccion)
    DiseñoMaterial (Barras() as barra, Fuerzas() as fuerza)
    VientoInicial = VientoFinal : VientoFinal = CalculoViento (Barras() as barra)
Wend
```

Estructura de la función de cálculo espacial de una torre

Para el cálculo de desplazamientos hemos hecho uso del MEF. Estos desplazamientos permiten deducir, mediante consideraciones físicas, las fuerzas axiales y las reacciones en los nudos encastrados (anclados al suelo). Esto nos lleva al diseño final de la estructura con sus angulares, calidades, tipo y cantidad de tornillos, longitudes de pandeo, áreas neta y de trabajo, etc. (es aquí donde hemos aplicado la normativa ASCE para el diseño de materiales de la estructura, véase anexo). El resultado nos aporta la estructura menos pesada (recordemos la ordenación de angulares de mínimo a máximo peso) que resiste las condiciones de carga exigidas. Estas técnicas de fijar una geometría inicial e ir modificándose progresivamente los materiales hasta conseguir el mejor modelo posible son también usadas en otras disciplinas del diseño estructural como en la optimización del comportamiento acústico de grandes recintos [CC98].

NOTA : En el caso de las torres atirantadas, realizamos un precálculo de cómo, en cada condición, se encuentran los cables (tensión o compresión). Una vez definidas estas situaciones, se agrupan las condiciones que tienen un mismo cable en tensión. Ya que no es posible que un cable responda a compresión como lo haría una barra (pues simplemente se curva), se quita este cable del conjunto de barras y se calcula

sin él. Así se hace sucesivamente con cada grupo, obteniéndose al final el cálculo de la estructura.

3.4.9 Modificación de Materiales

Aunque la torre calculada mediante el método anterior ya podría construirse e ir a ensayo, ciertamente no siempre es la más adecuada para su fabricación. Puede que en el proceso de cálculo haya tomado en consideración demasiados tipos de angulares distintos o unos materiales que pudieran ser más caros o difíciles de conseguir. Por ello se proporciona una opción al diseñador para permitirle modificar a su libre elección cualquier material de la estructura finalmente calculada.

Evidentemente esto transcurre bajo su absoluta responsabilidad, pero, por si acaso, el sistema comprueba las modificaciones efectuadas haciendo un nuevo cálculo y avisa de todas aquellas piezas que pudieran aparecer sobreesaturadas tras su modificación.

3.4.10 Cargas externas

Unido a la modificación de material se tiene asimismo la opción de introducir cargas externas al cálculo en cualquier nudo de la torre. Para ello la torre debe tener diseñada, al menos, la topología (así sabremos la numeración y posición de cada uno de los nudos). Una vez que colocamos las cargas externas que deberá soportar la torre (p.ej. la colocación de una grúa para el izado de los cables, o un cable que pudiera atirantar la estructura en cualquier nudo, etc.) el sistema calcula la estructura (no comprueba) y presenta los nuevos resultados de cálculo.

3.4.11 Salida y Presentación de Datos

Una vez finalizados los cálculos se agrupan por barras semejantes y se elige dentro de cada grupo cuál es y en qué condición se encuentra la que soporta más carga, tanto

a tensión como a compresión. A partir de aquí se genera una salida³ de archivos para presentar el proyecto. Estos datos son recogidos de la siguiente forma:

STRUCTURAL MEMBER	(3) DIMENSION	ANGLE SIZE	T/AREA	T/LENG	SLÉN	F O R C E S				B O L T S			
refff.	bar/near/far	steel grade	N/area	B/leng	r/in	kg	cond.	kg	Compress.	nu/diam	S	shear	bearing
Tower outline.	nodes	kg/cm2	cm2	cm	cm	num.	curve	mm	D	kg/cm2			
Tower Leg 7				154.20	49.0	C -25452	1	C -39663	0.64/1				
				77.10	1.55	T 19627	1	T 32760	0.60				
8	133 32 36	80 * 80 * 8	12.30	185.00	59.0	C -33894	1	C -37587	0.90/1	4/20	D	1349	5296
	131 35 39	3600	9.10	92.50	1.55	T 25780	1	T 32760	0.79				
Diagonal 7TR	123 31 34	45 * 45 * 4	3.49	203.50	137.0	C -2919	2	C -3854	0.76/4	1/16	S	1452	4560
	126 27 32	2600	1.98	120.77	0.88	T 1354	2	T 5142	0.26				
Diagonal 7LO	129 26 34	60 * 60 * 4	4.71	203.50	102.0	C -5442	1	C -7514	0.72/3	2/16	S	1353	4251
	128 27 35	2600	3.36	120.77	1.18	T 4502	1	T 8745	0.51				

donde cada variable indica:

- **Nomenclatura** : Codificación con la que es reconocido el grupo de barras dentro de una torre.
- **Número de barra (C&T)⁴**: Barra que, dentro del grupo, soporta la máxima carga.
- **Nudos inicial y final (C&T)**: Numeración de los extremos de la barra elegida.
- **Sección** : Tipo de angular que va a ser usado para que este grupo soporte las cargas, usando la nomenclatura *Ala1 x Ala2 x Espesor*.
- **Calidad** : A la hora de elegir el material podemos limitarnos a una única calidad o a varias (máximo 3 distintas e irá en función del mercado: alta, media o baja calidad)
- **Longitud** : Longitud de la barra (siempre será la misma dentro de una agrupación)
- **Longitud de Pandeo**: Es la longitud que posee la barra entre dos puntos de apoyo y puede ser mayor o menor que la Longitud.
- **Área Neta**: Área de la superficie de la barra.
- **Área de Trabajo**: El área, una vez perdida la parte de taladros que se le haga a la barra.
- **Esbeltez** : Punto de la curva de Euler (ASCE) al que ha llegado la barra mediante el material elegido y la carga que posee, y que no debe sobrepasar a las limitaciones de esbeltez introducidas en los datos de entrada.

³ Esta salida de datos fue introducida en 1990, aunque de forma parcial, por la compañía **CEGELEC (Francia)** en sus estudios de torres.

⁴ (C&T) indica que la variable tiene la misma interpretación a *Compresión* y a *Tensión*



- **Radio de Inercia:** Radio (mínimo o medio) elegido para esta barra.
- **Curva (C&T):** Número de la curva ASCE (de 1 a 4) usada. (esta numeración de 1 a 4 se corresponde con los puntos *a* hasta *d* del anexo I Normativa ASCE en el apartado de *otros miembros en compresión*)
- **Carga (C&T):** Fuerza máxima que soporta este grupo después del cálculo de fuerzas axiales. Cuando no son barras de cálculo se estima que soportan un porcentaje (habitualmente el 2.5%) de la barra principal a la que están ayudando a disminuir su longitud de pandeo.
- **Condición (C&T):** Hipótesis de carga que ha provocado los máximos.
- **Carga Máxima Permitida (C&T):** Valor máximo al que puede llegar este tipo de material.
- **Porcentaje (C&T):** Parte de la carga máxima permitida que usa el grupo en función de las máximas cargas axiales calculadas.
- **Número de Tornillos:** Cantidad de tornillos que necesitará esta barra para soportar la carga sin desgarre.
- **Diámetro de Tornillos:** Diámetro elegido dentro de aquellos que habíamos introducido en el archivo de entrada.
- **Cortadura :** Valor de cortadura de los tornillos (que no sobrepasará el límite introducido para este valor en función de la calidad de los mismos)
- **Aplastamiento :** Valor del aplastamiento que, al igual que la cortadura, deberá estar dentro de los límites expresados.

3.5 Esquema

Finalmente, después del cálculo y la salida de datos exponemos el *Esquema* de la torre donde se señalan todas sus barras, secciones, calidades, nº de tornillos y diámetros. Además se acompaña del peso (preliminar) de la estructura y de cada una de las subestructuras o extensiones requeridas. Habitualmente, estos son los datos que se precisan para participar en los concursos de torres.

3.6 Aportaciones realizadas

- Establecer el orden previo de la numeración de los nudos para la búsqueda del mínimo ancho de banda
- Implementar el algoritmo de resolución por bloques del sistema de ecuaciones desarrollado a partir del MEF
- Crear la ecuación vectorial para la representación bidimensional de estructuras espaciales con posibilidad de giros, visualización de caras por separado y cambio de perspectivas
- Desarrollar las estructuras computacionales de nudo, barra, carga, panel, fuerza y reacción
- Establecer el algoritmo de compresión de datos para la minimización del número de nudos en la estructura
- Definir los conceptos de inestabilidad global y local y las ecuaciones que las limitan. Desarrollo del cono de desplazamientos
- Crear el algoritmo recursivo para el establecimiento del viento real que afecta a la torre y el diseño estructural que, junto a las cargas, lo soporta.

3.7 Referencias bibliográficas

[Ak89] Akin, J. E. “Application and Implementation of Finite Element Methods”. Academic Press. Houston. Texas. USA. 1989.

[AR86] Argüelles Álvarez, R. “Cálculo de estructuras”. Edit. Fundación Conde del Valle de Salazar. Madrid . España. 1986.

[AS88] American Society of Civil Engineers. “Guide for design of steel transmission towers”. Edit. ASCE. New York. EE.UU. 1988.

[Ba95] Bathe, K. J.: Finite Element Procedures, Prentice-Hall, Englewood Cliffs, 1995.

- [BB66] Bergendal, G.; Brinck, I. “Lineär Algebra och Geometri”. Hakan Ohlssons. Lund. Suecia. 1972
- [BC81] Becker, E. B.; Carey, G. F. ; Oden, J. T. “Finite Elements: An Introduction”, Prentice Hall, 1981.
- [Be73] Bericua Llamas, A. “Truss, structural analysis program to design planar structures”. Artículo. Sydney. Australia. 1973.
- [BF98] Burden, R.L.; Faires, J.D. “Análisis Numérico”. Thomson International. México. 1998.
- [BR75] Bunch, J.R.; Rose, D.J. “Sparse matrix Computations”. Academic Press. San Diego. USA. 1975
- [CB99] Chandrupatla, T. R.; Belegundu, A. D. “Introducción al estudio del Elemento Finito en ingeniería”, Prentice Hall,1999
- [CC98] Corz, A.; Carillo, S. “Aplicación del método de conos para el análisis acústico de grandes espacios”. I Simposio Iberoamericano de Acústica (Lisboa). Publicación de la SEA. Málaga. España. 1998
- [Co89] Cook, R. D.: Concepts and Applications of Finite Element Analysis, J. Wiley & Sons, New York, 1989.
- [Co95] Cook, R. D.: Finite Element Modeling for Stress Analysis, J. Wiley, New York, 1995.
- [CO83] Carey, G. F. ; Oden, J. T. “Finite Elements: A Second Course”, Prentice Hall, 1983.
- [CO84] Carey, G. F. ; Oden, J. T. “Finite Elements: Computational Aspects”, Prentice Hall, 1984.

- [CO98] Coz, J.J.; Ordieres, J. “Matriz de rigidez de los sistemas de barras”. Manual de utilización del programa Solve. Cap. Escal3D. La Rioja. Málaga. España. 1998
- [Cr86] Crisfield, M. A.: Finite Elements and Solution Procedures for Structural Analysis, Vol. 1: Linear Analysis, Pineridge Press, Swansea, UK, 1986.
- [CW90] Conde, C.; Winter, G. “Métodos y algoritmos básicos del álgebra numérica”. Reverte. Madrid. 1990
- [DS96] Dym, C. L. e I. H. Shames: Energy and Finite Element Methods in Structural Mechanics, McGraw-Hill, 1996.
- [Fo02] Fonseca, J. “Ferramentas de Simulação em Mecânica: Elementos Finitos”. Apostila da Pós-Graduação em Engenharia Mecânica. Universidade Federal do Rio Grande do Sul, em Porto Alegre, Rio Grande do Sul, Brasil. 2002.
- [Ga75] Gallagher, R. H. “Finite element analysis. Fundamentals”. Edit. Prentice Hall. New Jersey. USA. 1975.
- [Ha72] Hanner, O. “Geometri och Lineär Álgebra”. Gleerups. Lund. Suecia. 1972
- [HO78] Hinton, E. ; Owen, D. R. J. “An introduction to the Finite Element Computations”, Academic Press, 1978.
- [Hu87] Hughes, T. J. R.: Finite Element Method - Linear Static and Dynamic Finite Element Analysis, Prentice-Hall, Englewood Cliffs, 1987.
- [IS97] ISI (Ingeniería de Sistemas Interactivos Ltda.) . “Computer integrated transmission towers system”. Manual. Málaga. España. 1997.
- [Ma96] Marques, O.A. “Eigensolvers and Applications in Finite Element Analyses”. Advanced Solution Procedures on Innovative Computer Architectures (M. Papadrakakis and G. Bugeda, Eds.), 66-79, CIMNE Publications, 1996.

[MF00] Mathews, J.H.; Fink, K.D. “Métodos Numéricos con Matlab”. Prentice Hall. Madrid. España 2000.

[Mo86] Mole, R.H. “Cálculo Numérico”. Anaya Multimedia. Madrid. España. 1986

[Oñ92] Oñate, E. “Cálculo de estructuras por el método de los elementos finitos” Edit. CIMNE. Barcelona. España. 1992.

[OP92] Ottosen, N. S. ; Petersson, H. “Introduction to the Finite Element Method”, Prentice Hall, 1992.

[RC02] Roberts, J.; Crandall, C.; Van Engelenhoven, J. “What is finite element analysis?”. Ciras. Iowa State University. USA. 2002.

[Re93] Reddy, J. N.: An Introduction to the Finite Element Method, McGraw-Hill Book Co., 1993.

[SB73] Szabo, B.; Babuska, I. “Finite Element Analysis”, Prentice Hall, 1973.

[Se85] Sewell, G. “Analysis of a Finite Element Method”. Springer-Verlag. New York. USA. 1985.

[TY79] Timoshenko, S.; Young, D. H. “Elementos de resistencia de materiales”. Montaner y Simón, Barcelona. España. 1979.

[Wi97] Widas, P. “Introduction to Finite Element Analysis”. Virginia Tech Material Science and Engineering. USA. 1997

[WJ84] Weaver, W.; Johnson, P.R. “Finite Element for Structural Analysis”. Prentice Hall. Stanford University. USA. 1984.

[ZT94] Zienkiewicz, O. C.; Taylor, R. L. “El método de los elementos finitos” Edit. Mc Graw Hill. Barcelona. España. 1994.

4. Algoritmos Genéticos

4.1 Introducción

Dos hechos han influido para que en los últimos años se haya presentado un desarrollo importante en el problema de la optimización:

1. La gran competencia en el campo de la producción industrial y de servicios, lleva a la necesidad de hacerlo todo, cada vez mejor y al mínimo costo; esto implica el tener que optimizar los recursos de todo tipo.
2. El avance en la tecnología de los computadores permite disponer actualmente de equipos con gran capacidad de almacenamiento y procesamiento de datos, así como una alta velocidad de operación.

De esta manera, se ha producido una especie de explosión de técnicas novedosas de optimización nacidas dentro de la inteligencia artificial [AH97], acompañada de una gran cantidad de aplicaciones en los campos más diversos.

El diseño óptimo de estructuras [Go01] se ha beneficiado del desarrollo de los métodos de optimización [Ki85, HK85, He90]. A partir de 1987 se empezó a aplicar a la optimización en ingeniería una técnica llamada *algoritmo genético*[GS87]. Esta técnica había empezado a desarrollarse a finales del decenio de 1950 con el fin de investigar la teoría y la práctica de la evolución algorítmica [Ho92]. El profesor W.M. Jenkins [Je91] presentó en 1991 el resultado de una investigación en la que aplicaba el algoritmo genético a la optimización del diseño estructural. En cierto sentido, este fue el punto de partida para lo que hoy en día es un vertiginoso desarrollo de la optimización de estructuras por medio del algoritmo genético. [GP93, Ga93, HL92, YS95, NC95, Je97, KG94, Je92]

La formulación ya tradicional del algoritmo genético (AG), consiste en ir mejorando generación tras generación, la calidad de los individuos de una especie dada. Para ello se parte de una generación básica ó inicial, normalmente obtenida en forma aleatoria; se seleccionan los mejores individuos de ella, con base en el criterio de

optimización que se persiga, y luego por un proceso de cruce y mutación se obtiene una nueva generación. Sobre esta generación vuelve a aplicarse una selección y luego cruce y mutación. De esta manera, a medida que transcurren las generaciones se van consiguiendo individuos cada vez más cercanos al óptimo buscado.

Como lo muestra *Galante* [Ga93] (para estructuras planas), el teorema fundamental de los algoritmos genéticos o teorema de los esquemas de Holland [Ho75], permite garantizar que de una manera exponencial, generación tras generación, la población va mejorando y acercándose al óptimo buscado.

Normalmente la descripción de la información o código genético de los individuos se ha escrito en código binario. Sin embargo, algunos autores [HH93, Ma93] han mostrado que no es necesario trabajar con este código de ceros(0) y unos(1) sino que puede hacerse con un código de enteros positivos.

En este trabajo los problemas planteados a resolver mediante esta técnica son los siguientes:

1. Numeración óptima de los nudos para minimizar el ancho de banda. Se usa el código de enteros positivos
2. Elección de las dimensiones óptimas de la estructura para peso mínimo. Uso de código binario.

Para la resolución de estos problemas se aplicará el algoritmo en su versión tradicional, que comprende como ya se enunció, las operaciones de selección, cruce y mutación, y posteriormente una modificación que intercambia las probabilidades de cruce y mutación en base a un exceso de padres con la misma aptitud e inferior a la óptima, lo que provoca una situación de empobrecimiento genético.

En el primero de los casos se elaboró el software necesario y se aplicó a los problemas descritos en el anexo III (Minimización de Ancho de Banda). Aquí se parte de la solución recomendada en el capítulo de cálculo de estructuras y se comprueba bajo que condiciones el número de padres y los factores de cruzamiento y mutación llevan a la solución óptima, siendo la función a minimizar el ancho de banda.

En el segundo se describe el procedimiento y se recomienda como motor de cálculo el sistema ISI de diseño de torres. Se inicia con las limitaciones de diseño y una solución preliminar, poniendo como función a minimizar el peso de la estructura.

4.2 Formulación tradicional

Las tres operaciones básicas del algoritmo genético en su formulación tradicional son selección, cruce y mutación [Go01]. Hemos incluido en este trabajo también la posibilidad de **clonación**, como el hecho de una copia exacta del padre o de la madre si el cruce es asexual.

4.2.1 Selección

La selección implica un doble proceso; primero se evalúa la bondad o aptitud de cada individuo de la generación presente con respecto al criterio que se esté tratando de optimizar [Lo97, NK02, Me02, Mer02].

Por ejemplo, si se está tratando de minimizar el peso de una estructura, tendrán mejor calificación o aptitud las estructuras livianas que las más pesadas.

Sin embargo, no basta con evaluar sólo la bondad de un individuo con respecto al criterio de optimización; también se debe evaluar si ese individuo cumple o no ciertas restricciones que se desean aplicar al individuo óptimo buscado.

Por ejemplo, si un individuo dado es una estructura muy liviana, cumplirá muy bien con el criterio del mínimo peso; pero si al mismo tiempo, esa estructura es demasiado estrecha para el paso de los montadores, entonces la estructura por muy liviana que sea, dañará su calificación de aptitud al violar las restricciones que hemos impuesto. Entonces, el primer paso en la operación de selección es asignar una aptitud a cada individuo. En la práctica si la optimización consiste en minimizar algo, por ejemplo el ancho de banda, la forma más simple de calificar a cada individuo es calcular la aptitud según la fórmula siguiente :

$$\text{APTITUD} = C - \text{Función Objetivo} \times \text{Factor de Penalización}$$

donde C es una constante arbitraria, pero lo suficientemente grande, como para que las aptitudes resulten siempre positivas.

La función objetivo es el criterio de optimización que estamos persiguiendo, por ejemplo el ancho de banda. El factor de penalización es un factor que castiga a los individuos que violan restricciones que previamente hemos impuesto.

Obsérvese que la expresión anterior funciona muy bien para calificar la aptitud de los individuos cuando el criterio de optimización es uno de minimización. Así, cuanto más pequeño sea el término afectado por el signo menos al lado derecho de la expresión, mayor será la aptitud, y eso es precisamente lo que queremos evaluar.

El factor de penalización se puede definir de una manera simple como las sumatorias de todas las violaciones a los valores permitidos, por unidad de valor permitido.

La meta a lograr es que la aptitud sea tan grande como se pueda ; ésta meta la logra el algoritmo genético generación tras generación.

Para el caso de la optimización de estructuras por mínimo peso, el algoritmo genético va produciendo en cada generación individuos más aptos; esto es, más livianos y que violen muy poco o nada las restricciones impuestas.

Lo que quiere decir esto es que los individuos con una aptitud baja están frenando el desarrollo de la búsqueda de mejores individuos. La solución obvia a esta dificultad está en privilegiar a los mejores individuos a costa de los peores. Para ello se debe aumentar la aptitud de los mejores individuos y disminuir la de los peores, para que estos tengan aun menos probabilidades de reproducirse.

Goldberg ha propuesto el llamado escalado lineal de las aptitudes [Go89]. Consiste en modificar de una manera lineal las aptitudes de los individuos de la generación. Así, para los individuos cuya aptitud bruta supere la promedia, la aptitud escalada es mayor que la bruta, pero para los individuos cuya aptitud bruta es inferior a la promedia, la aptitud escalada es inferior a la bruta.

Cuando no se usa escalado, no sólo la convergencia es lenta, sino que hay cambios bruscos en la aptitud máxima para generaciones sucesivas.

4.2.2 Cruzamiento

Una vez evaluada la aptitud de todos los individuos de una generación y luego de haber realizado el escalamiento lineal de esas aptitudes, se procede al cruzamiento de individuos para obtener la siguiente generación [Me02, NK98, Go89, Je97].

$$\left. \begin{array}{l} P1 = 001100 \quad 110011 \\ P2 = 000011 \quad 110000 \end{array} \right\} \Rightarrow \left\{ \begin{array}{l} H1 = 001100 \quad 110000 \\ H2 = 000011 \quad 110011 \end{array} \right.$$

Primero debe determinarse el número de veces que cada individuo va a participar como padre. La aptitud escalada es la medida que se emplea para asignar a cada individuo el número de veces que podrá cruzarse.

La aptitud escalada de cada individuo dividida por la sumatoria de las aptitudes escaladas de todos los individuos representa una fracción que al multiplicarla por el tamaño de la población, es decir el número total de individuos en cada generación, da el número de veces que ese individuo será padre.

Este número en general no es un número entero; entonces a cada individuo le corresponde una parte entera y un resto decimal. Aquellos individuos cuya parte entera sea nula dejarán espacios libres o vacantes que serán ocupados por los individuos que tienen parte entera mayor que 1; los demás espacios vacantes los ocuparán los individuos que tengan los mayores restos decimales

Una vez asignado el número definitivo de veces que cada individuo será padre y ya ocupando cada uno de ellos uno o varios espacios en la población total, se efectúa una selección aleatoria entre ellos para escoger las parejas que se cruzarán.

Puede especificarse además una probabilidad de cruce P_c ; se acostumbra tomar P_c entre 0.6 y 1.0; de tal manera que los individuos que se encuentran para cruzarse tienen una probabilidad, digamos del 80% de cruzarse efectivamente.

En el caso de que no se crucen, sencillamente cada uno de ellos pasará a la siguiente generación con su código genético intacto.

4.2.3 Mutación

Una vez generados por cruzamiento los nuevos individuos de la generación siguiente, se procede a una última operación llamada mutación [Je91, Ma94, NC95, NK98]. Consiste ésta, en cambiar un cierto número de bits, de la cadena total de bits de la población de manera tal que, si el bit tiene el número 1 se cambia a 0 y si es un 0 se cambia a 1, o equivalentemente, con números enteros.

Ej. : (1 2 3 4 5 6 8 9 7) => (1 8 3 4 5 6 2 9 7)

El objetivo de la mutación es efectuar ligeros cambios en el código genético de los individuos con la esperanza de que aparezcan mejores individuos; si la mutación desfavorece al individuo afectado, éste sencillamente desaparecerá en la siguiente generación. Pero si la mutación da lugar a un cambio en el parámetro al que pertenece el bit, cambio éste que favorece la aptitud del individuo, entonces este individuo empezará a transmitir su información genética en las generaciones siguientes lo que redundará en un acercarse cada vez al óptimo buscado.

El número de mutaciones que afectará la población se especifica por medio de una probabilidad de mutación de bits; normalmente esta probabilidad se define por valores del orden de 0.005 (0.5%).

Realmente no importa mucho el valor exacto [Ju95, Me02]. Pero lo que sí es un hecho es que una probabilidad muy alta, por ejemplo 0.2, perjudica notablemente la obtención de valores cercanos al óptimo; las numerosas mutaciones llevarán a que se presenten saltos irregulares en los valores de las mejores aptitudes para las generaciones sucesivas, sin que se produzca una convergencia definitiva hacia el óptimo. Por otro lado, la ausencia de mutaciones, reduce el material genético de todos los individuos de todas las generaciones prácticamente al material genético de los individuos de la generación inicial y por lo tanto, relativamente pronto, se produce una cierta esterilidad en la población para producir mejores individuos.

4.3 Software desarrollado

Aquí podemos observar como es el software desarrollado para la minimización del ancho de banda y como se ha implementado bajo la nomenclatura tradicional de algoritmos genéticos [Ho92]:

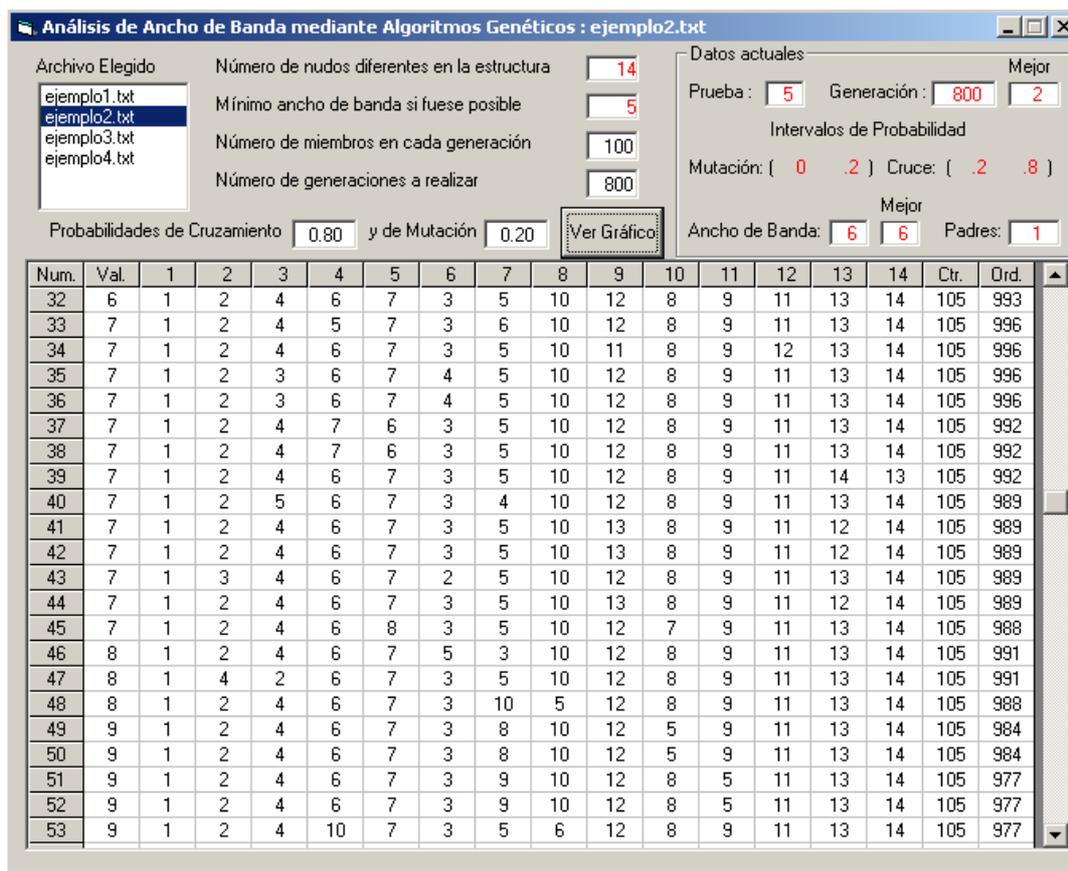
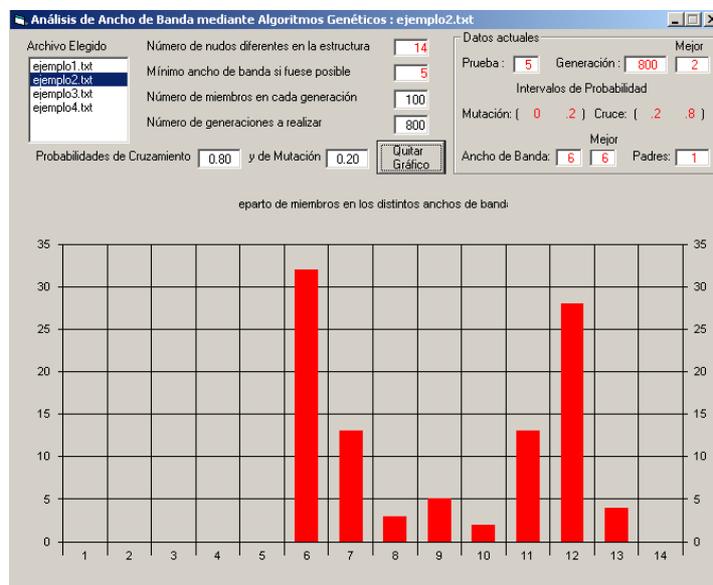


Ilustración 4-1. Minimización de Ancho de Banda mediante Algoritmos Genéticos

y como podemos ver el reparto de soluciones en base a los diferentes anchos de banda encontrados en la mejor solución:



4.4 Aplicación a la minimización de ancho de banda

Partimos del problema ya conocido de la minimización del ancho de banda en una matriz de rigidez. Deseamos resolver la ordenación óptima de manera que la diferencia entre los índices de dos nudos relacionados sea mínima. Planteamos como solución heurística la predisposición de los nudos en orden <derecha-izquierda, atrás-adelante, arriba-abajo> y vamos a usarla como solución inicial de nuestro problema para tratar de mejorarla. La descripción paso a paso del algoritmo será la siguiente:

1. Solicitamos los datos de entrada:
 - a. Número de miembros en cada generación (m)
 - b. Generaciones a desarrollar (n)
 - c. Probabilidad de cruzamiento (P_c)
 - d. Probabilidad de mutación (P_m)
2. Generamos una familia aleatoria donde el primer miembro es la solución heurística
3. Valoramos la aptitud de cada miembro (medimos el ancho de banda que generaría) y los clasificamos de mejor a peor
4. Definimos el número de ellos que actuarán de padres diferentes pero de la misma aptitud y cuales desaparecen (es adecuado entre el 50% y el 80% aunque el propio sistema va aumentando este porcentaje en función de las soluciones que va encontrando)
5. Comenzamos la reproducción creando un valor aleatorio P y aplicando lo siguiente:
 - a. Si $P > P_c$: el hijo será un clon de algún padre
 - b. Si $P_c \geq P > P_m$: elegiremos al azar un padre y una madre dentro del grupo de padres y produciremos su cruzamiento
 - c. Si $P_m \geq P$: tomaremos algún padre y produciremos una mutación de un gen aleatorio
6. Volvemos al paso 3 hasta crear n generaciones
7. El resultado final será una familia de m miembros donde los primeros poseerán las mejores aptitudes. Como sólo necesitamos una solución bastará con elegir el primero de ellos.

Apliquemos ahora este proceso a los siguientes ejemplos del Anexo III. En el observamos, además de los valores de m, n, Pc y Pm aplicados las mejores medidas, de 5 pruebas realizadas de cada tipo los mejores resultados para:

Apt : Aptitud de la primera solución

Gen : Generación en la que se obtuvo por primera vez el valor de Apt

Pad : Número de padres diferentes con la misma aptitud.

4.4.1 Caras cúbicas

Datos de entrada (el ancho de banda óptimo es 5 y el de partida 8):

(1,2)-(1,3)-(1,8)-(2,4)-(2,9)-(3,4)-(3,12)-(4,13)-(5,6)-(5,8)-(6,9)-(7,8)-(7,11)-(8,9)-
(8,12)-(9,10)-(9,13)-(10,14)-(11,12)-(12,13)-(12,15)-(13,14)-(13,16)-(15,16)

m	n	Pc	Pm	Apt	Gen	Pad
50	100	0.5	0.01	7	17	10
50	100	0.5	0.1	6	48	22
50	100	0.8	0.01	7	40	5
50	100	0.8	0.1	6	61	19
50	1000	0.8	0.2	5	123	43
100	100	0.5	0.01	7	34	20
100	100	0.5	0.1	6	8	54
100	100	0.8	0.01	7	11	21
100	100	0.8	0.1	6	41	46
100	1000	0.8	0.2	5	102	28

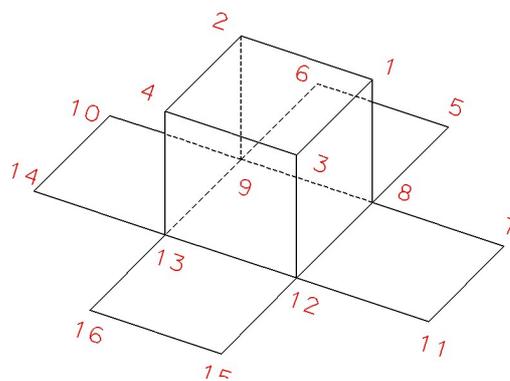


Ilustración 4-2. AG sobre caras cúbicas

Ej. de Sol.óptimas

1ª : 1 3 6 8 4 2 10 5 7 11 14 9 12 15 13 16

2ª : 2 4 5 9 1 3 7 6 8 13 12 10 11 14 15 16

En caso de efectuar reproducciones donde sólo se permite selección y mutaciones, con pocos cruzamientos y sin clonación obtenemos¹:

1ª : 2 4 6 9 1 3 7 5 8 12 11 10 13 14 15 16

2ª : 16 14 13 10 7 6 15 11 9 4 12 8 5 2 3 1

m	n	Pc	Pm	Apt	Gen	Pad
50	100	1	0.7	6	19	45
50	100	1	0.8	6	14	46
100	100	1	0.7	5	50	25
100	100	1	0.8	6	7	73
100	100	1	1	6	6	85
200	100	1	1	5	45	31
100	1000	1	1	5	113	35

¹ Para la realización de las pruebas se han aplicado reproducciones con poco cruzamiento y sin cruzamiento (formulación no sexual). Los resultados coinciden con la nueva formulación de AG expuesta por Carlos González Rodríguez (Univ. de Medellín) aplicada al diseño óptimo de estructuras en cerchas planas [CG01], aunque en ese caso su optimización se realiza sobre las ordenadas de los puntos.

4.4.2 Caras tetraédricas

Datos de entrada (el ancho de banda óptimo es 6 y el de partida 8):

(1,2)-(1,3)-(1,4)-(1,5)-(2,3)-(2,4)-(2,6)-(2,7)-(2,10)-(3,5)-(3,6)-(3,8)-(3,11)-(4,5)-(4,7)-(4,9)-
 (4,12)-(5,8)-(5,9)-(5,13)-(6,10)-(6,11)-(7,10)-(7,12)-(8,11)-(8,13)-(9,12)-(9,13)-(10,11)-(10,12)-
 (10,14)-(11,13)-(11,14)-(12,13)-(12,14)-(13,14)

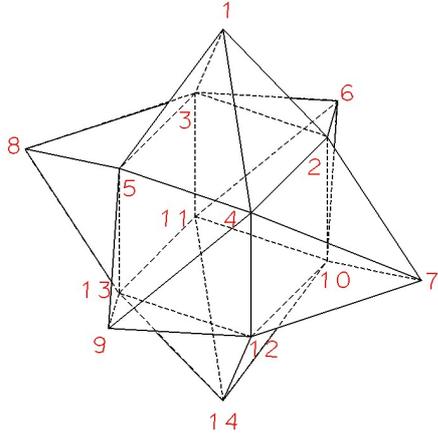


Ilustración 4-3. AG sobre caras tetraédricas

<i>m</i>	<i>n</i>	<i>Pc</i>	<i>Pm</i>	<i>Apt</i>	<i>Gen</i>	<i>Pad</i>
50	100	0.5	0.1	7	60	11
50	100	0.8	0.1	7	33	17
50	1000	0.8	0.2	7	58	50
100	100	0.5	0.1	7	19	35
100	100	0.8	0.1	7	34	32
100	1000	0.8	0.2	6	999	1
50	100	1	0.7	7	36	39
100	100	1	0.7	7	22	62
100	100	1	1	7	9	86
100	1000	1	1	6	118	1

Ej. de sol. óptimas

1ª : 1 2 4 6 7 3 5 10 12 8 9 11 13 14

2ª : 4 6 2 10 8 1 12 3 14 7 5 13 9 11

4.4.3 Sección de torre

Datos de entrada (el ancho de banda óptimo es 6 y el de partida 7):

(1,2)-(1,3)-(1,4)-(1,5)-(2,3)-(2,4)-(2,6)-(3,5)-(3,7)-(3,8)-(4,5)-(4,9)-(5,8)-(5,10)-(6,7)-(6,9)-
 (6,11)-(7,8)-(7,10)-(7,12)-(8,10)-(9, 10)-(9,13)-(10,14)-(11,12)-(11,13)-(12,14)-(13,14)-(2,7)-
 (3,6)-(3,10)-(5,7)-(4,6)-(2,9)-(5,9)-(4,10)-(11,14)-(12,13)

<i>m</i>	<i>n</i>	<i>Pc</i>	<i>Pm</i>	<i>Apt</i>	<i>Gen</i>	<i>Pad</i>
50	100	0.5	0.1	6	17	10
50	100	0.8	0.1	7	1	35
50	1000	0.8	0.2	6	16	15
100	100	0.5	0.1	6	29	10
100	100	0.8	0.1	7	1	44
100	1000	0.8	0.2	6	5	24
50	100	1	0.7	6	4	22
100	100	1	0.7	6	29	23
100	100	1	1	6	8	12
100	1000	1	1	6	7	24

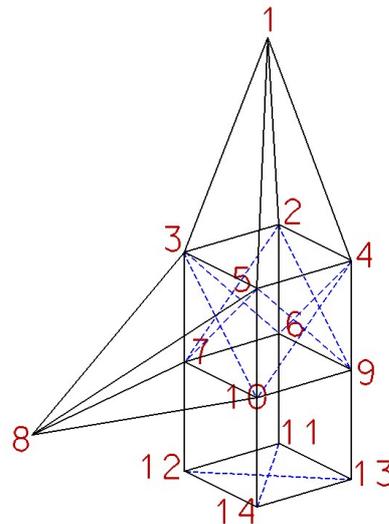


Ilustración 4-4. AG sobre sección de torre

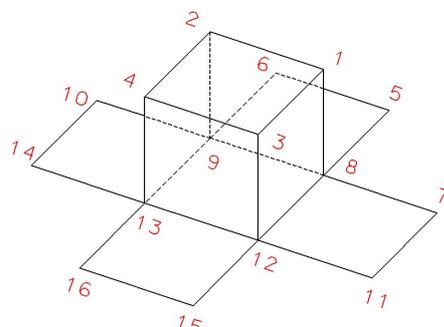
Ej. de sol. óptimas 1^a : 1 3 5 2 6 4 9 11 7 8 10 12 13 14
 2^a : 1 2 4 3 5 6 7 10 8 9 11 12 13 14

4.5 Formulación dinámica

En los casos en que el exceso de padres con el mismo valor (y superior al mínimo buscado) empobrece las soluciones buscadas se opta por intentar realizar menos cruzamientos y más mutaciones. Por ello ha sido modificado el sistema para que, además de utilizar la mejor solución de una prueba como comienzo de la siguiente (por eso en este caso el dato de la mejor generación es intrascendente y se ha quitado), en las situaciones de empobrecimiento se intercambien los valores de P_c y P_m , generando los siguientes resultados:

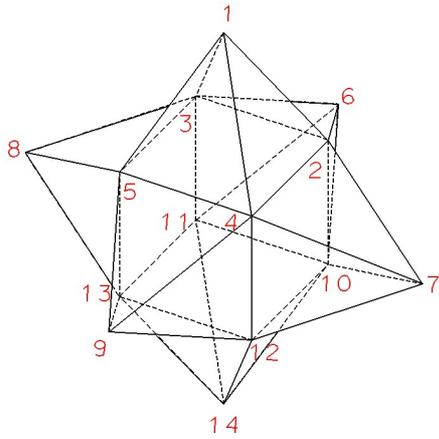
4.5.1 Caras cúbicas

m	n	P_c	P_m	A_{pt}	Pad
50	100	0.5	0.01	6	17
50	100	0.5	0.1	6	25
50	100	0.8	0.01	7	10
50	100	0.8	0.1	7	30
50	1000	0.8	0.2	5	34
100	100	0.5	0.01	7	8
100	100	0.5	0.1	6	24
100	100	0.8	0.01	7	37
100	100	0.8	0.1	6	34
100	1000	0.8	0.2	5	19



Ej. de sol. óptimas 1^a : 5 8 10 13 1 4 2 6 9 14 3 7 11 15 12 16
 2^a : 5 8 10 13 1 4 2 6 9 14 3 7 12 15 11 16

4.5.2 Caras tetraédricas



<i>m</i>	<i>n</i>	<i>Pc</i>	<i>Pm</i>	<i>Apt</i>	<i>Pad</i>
50	100	0.5	0.1	7	15
50	100	0.8	0.1	7	13
50	1000	0.8	0.2	6	1
100	100	0.5	0.1	7	18
100	100	0.8	0.1	6	1
100	1000	0.8	0.2	6	1
50	100	1	0.7	7	28
100	100	1	0.7	7	44
100	100	1	1	7	2
100	1000	1	1	6	1

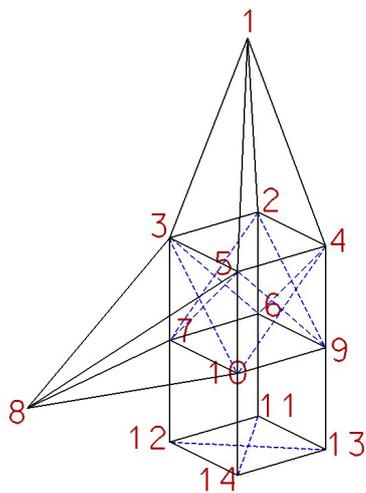
Ej. de sol. óptimas

1^a : 1 2 4 6 7 3 5 10 12 8 9 11 13 14

2^a : 1 2 5 6 7 3 4 11 12 8 9 10 13 14

3^a : 3 1 6 4 9 5 2 12 10 7 11 8 14 13

4.5.3 Sección de torre



<i>m</i>	<i>n</i>	<i>Pc</i>	<i>Pm</i>	<i>Apt</i>	<i>Pad</i>
50	100	0.5	0.1	7	23
50	100	0.8	0.1	7	22
50	1000	0.8	0.2	6	21
100	100	0.5	0.1	7	27
100	100	0.8	0.1	6	8
100	1000	0.8	0.2	6	22
50	100	1	0.7	6	11
100	100	1	0.7	6	20
100	100	1	1	6	1
100	1000	1	1	6	2

Ej. de sol. óptima

1^a : 1 2 4 3 5 8 7 10 6 9 11 13 12 14

4.6 Diseño óptimo de estructuras

Para la utilización de los algoritmos genéticos al diseño óptimo de estructuras se realiza una selección de los parámetros a aplicar en la función objetivo. En el caso del diseño de torres estos parámetros serían:

PESO : peso de la estructura

AMAX : altura total de la estructura

N : número de nudos en la estructura.

A(i) : altura a la que se encuentra el nudo *i* (con $i=1, \dots, N$)

D(i) : desplazamiento de cada nudo ($i=1, \dots, N$)

DMAX : desplazamiento máximo recomendado a la estructura en su punto superior (véase cono de desplazamientos), equivalente al 2% de la altura total en el caso de torres de transmisión eléctrica y del 6% en las torres de telecomunicaciones. Por tanto $D(1) \leq DMAX$

Dm(i) : desplazamiento máximo permitido al nudo *i* . Su cálculo depende del desplazamiento del nudo más alto (*D(1)*) y de la altura a la que se encuentra el nudo:

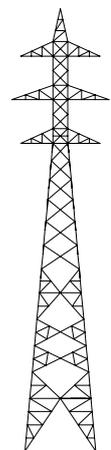
$$Dm(i) = \frac{A(i) \cdot D(1)}{AMAX} \text{ con } i = 1, \dots, N$$

con estos datos la función objetivo a minimizar es:

$$APT = C - PESO \cdot \left[1 + \sum_{i=1}^N \left(\frac{D(i) - Dm(i)}{Dm(i)} \right) \right]$$

Con estos datos se plantea una selección de individuos mediante una recta de fuerte pendiente incrementable generación tras generación, provocando un fuerte escalado lineal en oposición al escalado normal o suave de *Goldberg* [GS87, Go01]

A partir de aquí debe hacerse un diseño preliminar de la estructura y generar una familia inicial de torres donde se admita libertad de movimiento de los nudos en el sentido de los ejes X e Y. Las alturas (sentido del eje Z) suelen venir fijadas en las especificaciones técnicas, al igual que las distancias de las puntas de las crucetas al cuerpo de la estructura, por lo que estos parámetros no serán modificables.



El objetivo de reprogramar el sistema de cálculo de estructuras para la aplicación de los algoritmos genéticos lo planteamos como un trabajo futuro al actual, y entraría en el campo de las *técnicas de reánalisis* [Ki93, DX94, GV85, Sa91].

4.7 *Aportaciones realizadas*

- Introducir el concepto de clonación en la formulación de los algoritmos genéticos
- Implementar un sistema de algoritmos genéticos para la minimización del ancho de banda de una matriz
- Analizar las diferentes solución óptimas en modelos concretos mediante la formulación tradicional y mediante la formulación dinámica
- Definir el procedimiento a seguir para el diseño óptimo de torres

4.8 *Referencias Bibliográficas*

[AD97] Alexandrov, N., Dennis, J. E., Lewis, R. M., and Torczon, V., "A Trust Region Framework for Managing the Use of Approximation Models in Optimization," ICASE Report No. 97-50, 1997.

[AH97] Ansari, N., and Hou, E. "Computational Intelligence for Optimization", Kuwer Academic Publishers Boston, USA. 1997

[Ba00] Bastian, A. "Identifying Fuzzy Models Utilizing Genetic Programming", Fuzzy Sets and Systems, 113, 333-350, 2000.

[BN98] Banzhaf, W.; Nordin, P.; Keller, R.E. & Francone, F.D. "Genetic Programming: An Introduction: On the Automatic Evolution of Computer Programs and Its Applications", Morgan Kaufmann Publishers, ISBN: 155860510X, 1998.

[BS93] Bazaraa, M.; Sherali, H.; Shety, C. "Nonlinear Programming", Second Edition. Wiley, 1993.

[Bu94] Buckley, J.J. & Hayashi, Y. "Fuzzy Genetic Algorithm and Applications", *Fuzzy Sets and Systems*, 61, 129-136, 1994.

[BW93] Blanton, J.L., and Wainwright, R.L."Multiple Vehicle Routing with Time and Capacity Constraints using Genetic Algorithms", *Proc. 5th International Conference on Genetic Algorithms*, Morgan Kauffmann Pub., San Mateo, California, pp. 452-459. 1993

[Ca95] Cammesa "Procedimientos para la programación de la operación, el despacho de cargas y el cálculo de precios". Tomos I y II. Buenos Aires, Argentina. 1995

[CC95] Chen, P.; Chang, H. "Large-scale economic dispatch by genetic algorithm". 1995 IEEE/PES Winter Meeting , New York .

[CH93] Chang, K. J., Haftka, R. T., Giles, G. L., and Kao, P. -J., "Sensitivity-Based Scaling for Approximating Structural Response," *Journal of Aircraft*, 30, 1993, pp. 283-288.

[CM96] Campos, V. y Mota, E."Metaheurísticos para el CVRP", *Actas del II Symposium de Ingeniería de los Transportes*, Dep. de Transportes de la Univ. Politécnica de Madrid, (II):11-12. 1996

[CP02] Carballido, J.; Ponzoni, I. y Brignole, D. "Un algoritmo genético basado en números difusos triangulares". Artículo. Bahía Blanca, Argentina. 2002

[CR90] Chowdhury, B. H. - Rahman, S. (1990) "A review of recent advances in economic dispatch". *IEEE Trans. on Power Systems*, Vol. 5, No. 4, pp. 1248-1259.

[Da96] Davis, L. "Handbook of Genetic Algorithms". International Thompson Computer Press. Boston (USA). 1996

[De90] Desrochers, M. "A Clasification Scheme for Vehicle Routing and Scheduling Problems". *European Journal of Operational Research*, 46 (1990), pp. 322-332. 1990

[DG96] Diaz, A., Glover, F., Ghaziri H.M., González, J.L., Laguna, M., Moscato, P. y Tseng, F.T."Optimización Heurística y Redes Neuronales en Dirección de Operaciones e Ingeniería". Editorial Paraninfo S.A., Madrid. 1996

[DX94] G. Dejuan, Z. Xiangzhou, "Modified Designs and Modified Computations". Computers and Structures, Vol 49, No 6. 1994.

[EA97] Eby, D., Averill, R. C., Gelfand, B., Punch, W. F., Mathews, O., and Goodman, E. D., "An Injection Island GA for Flywheel Design Optimization" Invited Paper, Proceedings of the 5th European Congress on Intelligent Techniques and Soft Computing, Sept., 97.

[FH96] Furuya, H., and Haftka, R. T., "Combining Genetic and Deterministic Algorithms for Locating Actuators on Space Structures," Journal of Spacecraft and Rockets, Vol. 33, No. 3, 1996, pp. 422-427.

[Fo94] Fogel, D. B. "An introduction to simulated evolutionary optimization" IEEE Trans. on Neural Networks, Vol 5, No. 1, 1994, pp 3-14.

[Ga93] M. Galante, "Un algoritmo genético simple para la optimización de estructuras planas articuladas". Revista Internacional de Métodos Numéricos, para Cálculo y Diseño en Ingeniería, Vol 9,2,1993.

[GC00] Gen, M. & Cheng, R. "Genetic Algorithms and Engineering Optimization", Wiley Series in Engineering Design and Automation, John Wiley & Sons, ISBN: 0471315311, 2000.

[Gh96] Ghaziri, H.M. "Redes Neuronales en Optimización Heurística y Redes Neuronales en Dirección de Operaciones e Ingeniería". Editorial Paraninfo S.A., Madrid, pp. 163-207. 1996

[Go01] González Rodríguez, C. "Una nueva formulación del algoritmo genético aplicado al diseño óptimo de estructuras" Artículo. Medellín. Colombia. 2001.

[GP93] D.E. Grierson, W.H. Pak, "Optimal Sizing, Geometrical and Topological Design Using Genetic Algorithm". *Structural Optimization*, 6, 1993.

[Go89] Goldberg, D.E. "Genetic Algorithms in Search, Optimization and Machine Learning", Addison-Wesley Pub Co. 1989.

[GS87] D.E. Goldberg, M.P. Samtani, "Engineering Optimization Via Genetic Algorithm", *Proceedings of the Ninth Conference on Electronic Computation*, Asce, 1987.

[GT96] Goonatilake, S., and Treleaven, P. "Intelligent Systems for Finance and Business", John Wiley, 335p. 1996

[GV85] J. Garcia de Jalon, F. Viadero, "A new Direct Method for the Simple and Efficient Reanalysis of Structures". *Computers and Structures*, Vol 21, No 5, 1985.

[HB96] Hernández, J.; Brignone, S.; Ruetsch, L.; Moitre, D.; Aromataris, L. "Planificación de la operación de sistemas de energía vía optimización evolutiva". *Anales del 7º Congreso Latinoamericano de Control Automático*. Bs As, Argentina. 1996. Vol I, pp. 266 - 271.

[He90] S.I. Hernandez, *Métodos de diseño óptimo de estructuras*, Colegio de Ingenieros de Caminos, Canales y Puertos, Aragón, 1990.

[HH93] F. Hiroshi, R. Haftka, "Locating Actuators for Vibration Supression on Space Trusses by Genetic Algorithms". *Proceedings of the 1993 ASM Winter Annual Meeting*, 1993.

[HK85] R.T. Haftka, M.P. Kamat, *Elements of Stuctural Optimization*, Martinus Nijhoff Publishers, 1985.

[HL92] P. Hajela,, C.Y. Lin, "Genetic Search Strategies in Multicriterion Optimal Design". *Structural Optimization*, 4, 1992.

[HL97] Herrera, F.; Lozano, M. & Verdegay, J.L. "Fuzzy Connectives Based Crossover Operators to Model Genetic Algorithms Population Diversity", 92, 1997.

[Ho75] Holland, J. H. *Adaption in Natural and Artificial Systems*. Cambridge, MIT Press. (1975)

[Ho92] Holland, J.H. "Genetic Algorithms", *Scientific American*, September, 1992.

[HW00] Huang, Y.-P. & Wang, S.-F. "Designing a Fuzzy Model by Adaptive Macroevaluation Genetic Algorithms", *Fuzzy Sets and Systems*, 113, 367-379, 2000

[HY96] Hajela, P., and Yoo, J., "Constraint Handling in Genetic Search Using Expression Strategies," *AIAA Journal*, Vo. 34, No. 11, 1996, pp. 2414-2420.

[IE96] Special Section on Genetic Algorithms for Industry. *IEEE Trans. on Industrial Electr.*, Vol 43, No 5. , (1996).

[Je91] W.M. Jenkins, "Towards Structural Optimization Via the Genetic Algorithm", *Computers and Structures*, 40,pp, 1321-1327, 1991.

[Je92] W.M. Jenkins, "Plane Frame Optimum Design Environment Based on Genetic Algorithm". *J.of Structural Engineering*, November,1992.

[Je97] W.M. Jenkins, "On the Application of Natural Algorithms to Structural Design Optimization". *Engineering Structures*, V.19, No.4, 1997.

[JV85] J. Garcia de Jalon, F. Viadero, "A new Direct Method for the Simple and Efficient Reanalysis of Structures". *Computers and Structures*, Vol 21, No 5,1985.

[Ju95] Julstrom, B.A."What Have You Done for Me Lately? Adapting Operator Probabilities in a Steady-State Genetic Algorithm". *Proc. 6th International Conference on Genetic Algorithms*, Morgan Kauffmann Pub., San Mateo, California, pp. 81-87. 1995

- [KG94] V.K. Koumosis, P.G. Georgiou, "Genetic Algorithms in Discrete Optimization of Steel Truss Roofs". *J. of Computing in Civil Engineering*, July, 1994.
- [Ki81] U. Kirsch, *Optimum Structural Design*, Mc.Graw-Hill, 1981.
- [Ki93] U. Kirsch, "Efficient Reanalysis for Topological Optimization". *Structural Optimization*, 6, 1993.
- [Ki95] Kirsch, U., "Improved Stiffness-Based First-Order Approximations for Structural Optimization", *AIAA Journal*, Vol. 33, No. 1, 1995, pp.143-150.
- [KN96] Kodiyalam, S., Nagendra, S., and DeStefano, J., "Composite Sandwich Structural Optimization with Applications to Satellite Components," *AIAA Journal*, Vol. 34, No. 3, 1996.
- [KY95] Klir, G. & Yuan, B. "Fuzzy Sets and Fuzzy Logic, Theory and Applications", Prentice Hall, 1995.
- [LH96] Lee, J., and Hajela, P., "Parallel Genetic Algorithm Implementation for Multidisciplinary Rotor Blade Design," *Journal of Aircraft*, Vol. 33, No. 5, 1996,
- [LL92] Laporte, G., Louveaux, F., and Mercure, H. "The Vehicle Routing Problem with Stochastic Travel Times", *Transportation Science*, 26(3):162-170. 1992
- [Lo97] Louis, S.J. "Genetic Algorithms and Truss Design". Article. USA. 1997.
- [Ma93] T. Masters, *Practical Neural Network Recipes in C++*. Chapter 8, Academic Press, 1993
- [Ma94] Man, K. F.- Tang, K.- Kwong, S. "Genetic Algorithms: Concepts and Applications" *IEEE Trans. on Industrial Electronics*, Vol 43, No. 5, 1996,
- [MA96] Moitre, D.; Aromataris, L.; Hernández, J. "Evolución de los Mercados de Energía Eléctrica Argentino y Chileno". *Sesiones Técnicas del Congreso:*

Transiciones Energéticas en México, Centro y Sudamérica. México, D.F., México, 1996.

[MA97] Moitre, D.; Aromataris, L. y otros. "Aplicación de Algoritmos Genéticos a la Coordinación Hidrotérmica de Corto Plazo". Artículo. México, D.F., México, 1997.

[MB96] Moitre, D.; Brignone, S.; Ruetsch, L.; Maldonado, F.; Hernández, J.; Aromataris, L. "Técnicas emergentes en la optimización de sistemas de suministro de energía eléctrica". Ponencias del II Congreso Latinoamericano de Distribución de Energía Eléctrica, Viña del Mar, Chile. 1996.

[Me02] J.R. Medina. "Algoritmos genéticos para la optimización de redes de distribución". Artículo. Valencia. España 2002.

[Mer02] J.J. Merelo Guervós. "Informática Evolutiva: Algoritmos Genéticos". Artículo. Universidad de Granada. España 2002.

[MG95] Miller, B. L., and Goldberg, D. E., "Genetic Algorithms, Tournament Selection and the Effects of Noise," IlliGAL Report No. 95006, University of Illinois at Urbana-Champaign, 1995.

[MH96] Moitre, D.; Hernández, J.; Ruetsch, L.; Maldonado, F.; Aromataris, L. "Optimización del Despacho de un Sistema de Suministro de Energía Eléctrica combinando Técnicas Determinísticas y Estocásticas". Sesiones Técnicas del Congreso: Transiciones Energéticas en México, Centro y Sudamérica, México, D.F., México. 1996.

[MH97] Moitre, D.- Hernández, J.- Aromataris, L.- Rodríguez, G. "Programación de la Operación Óptima Económica de Sistemas Eléctricos de Potencia sujeta a restricciones energéticas: una aplicación de Algoritmos Genéticos". RVP'97 IEEE Sección Mexico. Acapulco, Mexico. 1997.

[Mi96] Michalewicz, Z. "Genetic Algorithms + Data Structures = Evolution Programs", 3ra edición revisada, Springer Verlag, ISBN: 3540606769, 1996.

- [Mi97] Miller, B. L., "Noise, Sampling and Efficient Genetic Algorithms," PhD Dissertation, University of Illinois at Urbana-Champaign, 1997.
- [NC95] K. Nukulchai, Ch.Ch. Chung, L.Y. Chage, "Efficient Genetic Algorithms for Space Truss Optimization". Proceedings of the Fifth East Asia Pacific Conference on Structural Engineering and Construction, July, 1995.
- [NH96] Nagendra, S., Haftka, R. T., Gurdal, Z., and Watson, L. T., " Improved Genetic Algorithms for the Design of Stiffened Composite Panels, " Computers and Structures, Vol. 58, No. 3, 1996, pp. 543-555.
- [NK98] P.B. Nair, A.J. Keane. "Combining approximation concepts with genetic algorithm-based structural optimisation procedures". Proceedings of the 39th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Long Beach, CA, April 1998, AIAA 98-1912
- [NK02] Nair, P. B., Keane, A. J., and Langley, R. S., "Improved First-Order Approximation of Eigenvalues and Eigenvectors," AIAA Journal, to appear.
- [Pe85] Pereira, M. "Optimal Scheduling of Hydrothermal Systems : An overview". IFAC Electric Energy Systems. Rio de Janeiro, Brazil. 1985.
- [Re95] AG Reinelt. "Discrete Optimization. Available Software and Data". 1995
- [RH97] Riubrugent, J.; Hernández, J.; Moitre, D.; Aromataris, L. "Control de Sistemas de Potencia: Aplicación de Algoritmos Genéticos a la Programación de la Operación". SIE'97. Santa Clara, Cuba., 1997.
- [Sa91] M.P. Saka, "Finite Element Applications of the Theorems of Structural Variation". Computers and Structures, Vol 41, No 3, 1991.
- [SB94] Sheble, G. B.; Brittig, K. "Refined Genetic Algorithm - economic dispatch example". IEEE Trans. on Power Systems, Vol 10, No 1. , 1994. pp. 117-124.

[SH96] Sobieszczanski-Sobieski, J., and Haftka, R. T., "Multidisciplinary Aerospace Design Optimization : Survey of Recent Developments," 34th AIAA Aerospace Sciences Meeting and Exhibit, AIAA Paper No. 96-0711, Reno, Nevada, 1996.

[Th95] Thangia, S.R. "An Adaptative Clustering Method using Geometric Shape for Vehicle Routing Problems with Time Windows", Proc. 6th International Conference on Genetic Algorithms, Morgan Kauffmann Pub., San Mateo, California, pp. 536-543.1995

[WS93] Walters, D. C. - Sheble, G. B. (1993) "Genetic algorithm solution of economic dispatch with valve point loading". IEEE Trans. on Power Systems, Vol 8, No 3. ,pp. 1331-1352.

[WS96] Whitley, D., Starkweather, T., and Shaner, D. "The Travelling Salesman and Sequence Scheduling: Quality Solutions using Genetic Edge Recombination". Handbook of Genetic Algorithms (L. Davis Ed.), International Thompson Computing Press, pp. 350-372. 1996

[WW84] Wood, A. & Wollenberg, B. Power Generation, Operation, and Control. Wiley. 1984.

[WY00] Wu, B. & Yu, X. "Fuzzy Modelling and Identification with Genetic Algorithm Based Learning", Fuzzy Sets and Systems, 113, 351-365, 2000.

[Ya96] Yang, H.; Yang. P; Huang, Ch. "Evolutionary programming based economic dispatch for units with non-smooth fuel cost functions". 1995 IEEE/PES Winter Meeting , New York .

[YS95] K.E Young, L.H. Soo, "A Study on Evolution Design Model Based on Genetic Algorithms". J.of the Architectural Institute of Korea, V.15, No.4, 1995.

[YY95] Yang, H.- Yang. P- Huang, Ch. (1995) "Evolutionary programming based economic dispatch for units with non-smooth fuel cost functions". 1995 IEEE/PES Winter Meeting , New York .

[Za65] Zadeh, L.A. "Fuzzy Sets", Inf. Control 8, 1965.

5. Minimización de datos en fábrica

Finalizado el cálculo de la TTE ya se han obtenido todos los parámetros necesarios para poder realizar los planos de ejecución de la estructura. Llegado este punto, un buen conocimiento del Álgebra y del Análisis Vectorial será útil para resolver los problemas geométricos con los que nos vamos a encontrar, tales como el análisis de las interferencias espaciales o el cálculo de áreas a partir del conocimiento de su perímetro.

Hasta ahora hemos hablado de un conjunto espacial de nudos y barras que nos acercaban a un esquema virtual de lo que deseábamos construir. A partir de este capítulo dejaremos los conceptos abstractos de nudos y barras para pasar a los de juntas, tornillos, cortes, chapas, etc., es decir, conceptos concretos que permitirán fabricar la estructura. Gráficamente podríamos definir esta conversión mediante la ilustración que sigue:

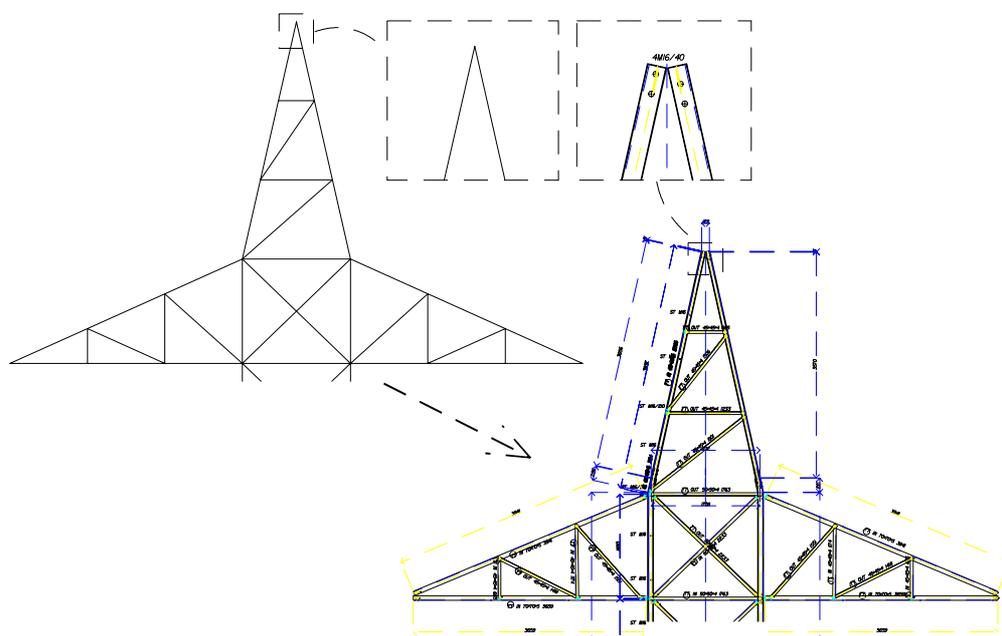


Ilustración 5-1. Transformación de Outline a Plano de Fábrica

5.1 Nomenclatura en la fabricación de TTE

Para comenzar el desarrollo de los planos de fabricación es necesario definir los conceptos básicos de este campo.

- **Barra** : Las componentes principales de una estructura, que deberá ser fabricada a partir de los angulares del mercado.
- **Chapa** : Placa de acero taladrada que reforzará la estructura allí donde se considere necesario.
- **Junta** : Lugar de unión de las barras (en el cálculo eran los nudos). Se realizarán estas uniones mediante tornillos, chapas o soldaduras.
- **Taladro** : agujero realizado sobre la barra para el paso de un tornillo.
- **Gramil** : Línea imaginaria en la barra sobre la que se realizan los taladros.
- **Progresiva** : Distancia desde el origen de la barra hasta la proyección sobre el borde de la misma del lugar taladrado.
- **Caras** : Cada una de las dos componentes o alas de un angular. Las notaremos por Izquierda y Derecha.

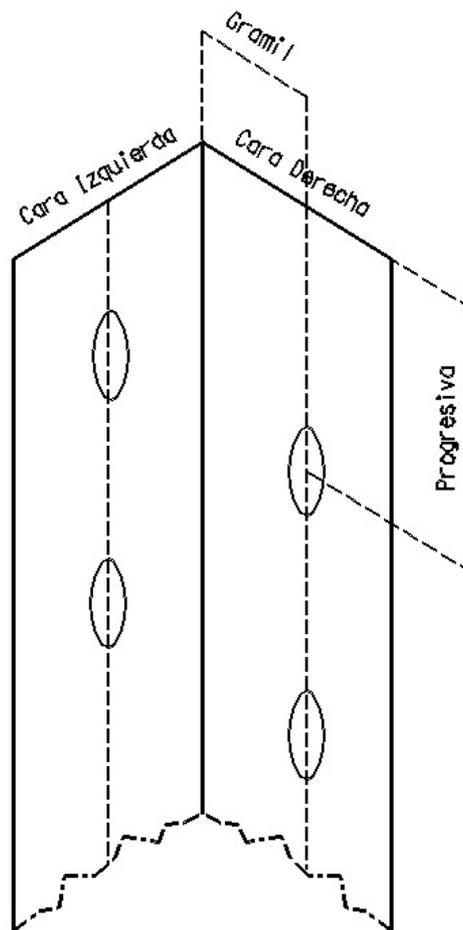


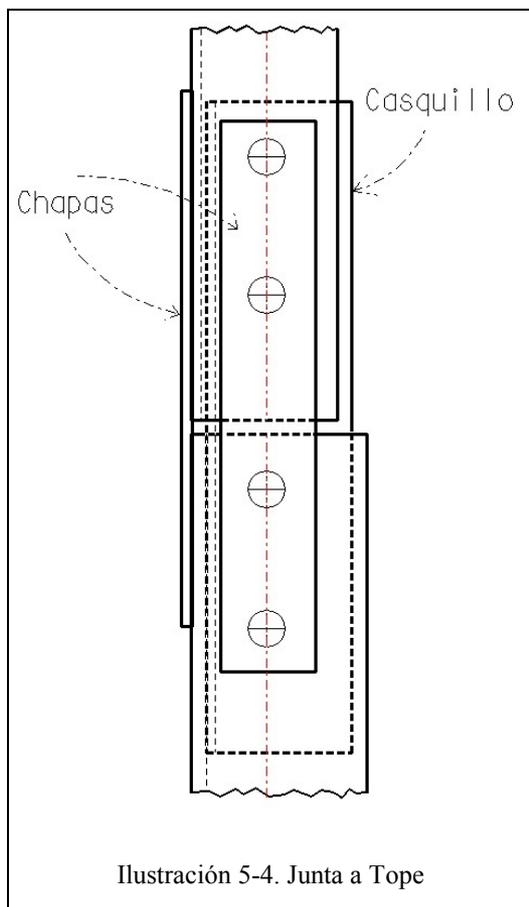
Ilustración 5-2. Distancias básicas en una barra

5.2 Tipos de juntas y desplazamiento sobre el límite

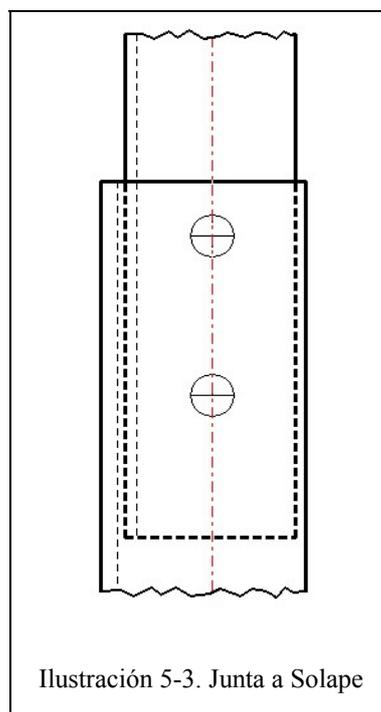
En primer lugar definimos, al igual que en el cálculo, la estructura tronco-piramidal que limita los cuatro bordes de una torre. Anteriormente teníamos unas medidas en la geometría de entrada que nos señalaba la posición de los nudos y de las barras. Ahora estas barras tienen dos caras y están fabricadas a partir de un angular en L. Por lo tanto, se estudiarán las posiciones de los materiales, de forma tal que se ajuste lo mejor posible al modelo ideal calculado.

Distinguimos dos tipos de uniones entre los montantes que son:

1. **Junta tipo Tope:** Un montante y el siguiente chocan en sus extremos, con lo cual, para ser cogidos poseen una barra interior o casquillo y dos chapas exteriores.



2. **Junta tipo Solape:** Un montante entra en el interior del siguiente y comparten los taladros con que son unidos.



Una vez definidos los dos tipos de juntas, volvemos a la especificación de la estructura de la torre.

El siguiente paso consiste en elegir qué tipo de junta se desea usar en cada empalme existente en la torre. A partir de aquí el sistema calcula a qué distancia se encontrará el borde de los montantes, de tal manera que la línea virtual de cálculo que anteriormente los identificaba pase lo más cerca posible del centro de las alas de todos ellos (debería hacerse respecto al gramil principal, pero aún no ha sido calculado).

El siguiente cálculo permite saber cuánto se desplazan todos los montantes en las caras transversal y longitudinal de su límite de origen, donde el tipo de junta alta es una variable booleana, con valor **1** si es a solape y con valor **0** en otro caso.

$$\text{Desplazamiento} = \frac{\sum_{i=1}^{\text{paneles}} \text{AlaMontante}(i) / 2 - \text{TipoJuntaAlta}(i) \cdot \text{EspesorMontante}(i)}{\text{paneles}}$$

Con esta fórmula ajustamos el conjunto de segmentos que representan el eje central de los montantes a la línea imaginaria de cálculo aplicada en los elementos finitos. Podemos observar que el uso excesivo de las juntas a solape provoca una menor aproximación entre las coordenadas de cálculo y las de la estructura real fabricada, llevándonos habitualmente a resultados impredecibles (este es uno de los principales causantes de roturas antes de llegar a la aplicación del 100% de las cargas en las estaciones de ensayo, por lo que no es una buena opción elegir la junta a solape pensando exclusivamente en el ahorro de material).

5.3 Distancias de seguridad

Cada especificación técnica nos define unas distancias mínimas de seguridad, que tendremos que tener en cuenta a la hora de colocar los taladros, y ciertas características sobre el tornillo usado en cada uno. Son las siguientes:

1. **Al borde laminado:** Distancia desde el centro del taladro al límite del ala del angular.
2. **Al corte:** Distancia desde el centro del taladro al origen o al final de la barra.
3. **Entre taladros:** Distancia entre cualesquiera dos centros de taladros en una misma barra.
4. **Tolerancia:** Diferencia u holgura entre el diámetro del taladro y el del tornillo que se usa en él.
5. **Espesor de Arandela:** Espesor de la arandela correspondiente al diámetro de cada tornillo.

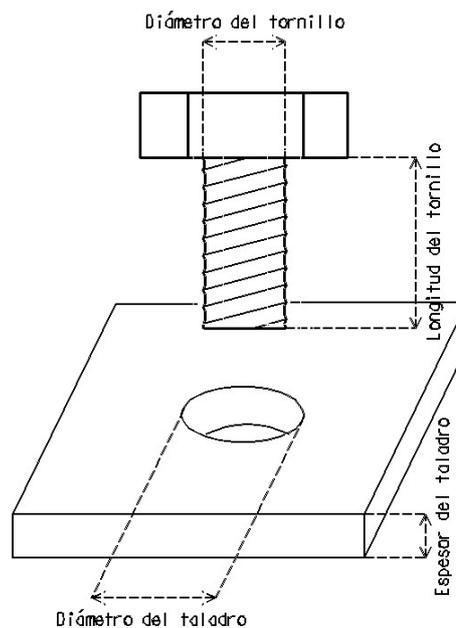


Ilustración 5-5. Dimensiones de un taladro

Estas distancias pueden venir definidas como una proporción sobre el diámetro usado, o fija para cada tipo de tornillo. El caso más habitual en los proyectos internacionales es:

Sea \varnothing el símbolo que define al diámetro de un tornillo.

1. **Al borde laminado:** $1.6 \varnothing$
2. **Al corte:** $1.5 \varnothing$
3. **Entre taladros:** $2.5 \varnothing$
4. **Tolerancia :** 1.5 mm. superior a \varnothing
5. **Espesor de Arandela:** 8 mm. para todo \varnothing

Si en el proyecto se va a usar más de un diámetro diferente de tornillos, tendremos para cada uno sus medidas de seguridad. Por ello se define la estructura:

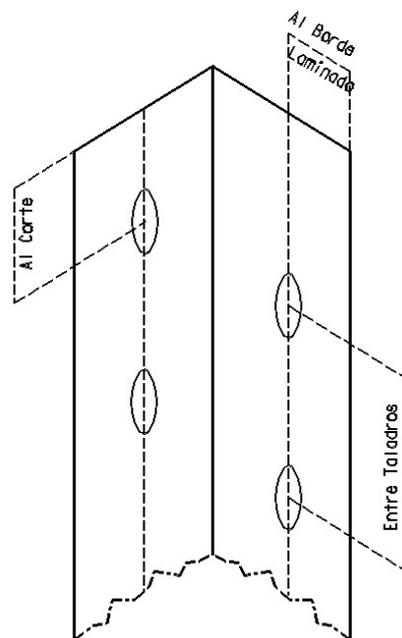


Ilustración 5-6. Distancias de seguridad

Type Seguridades

AlBorde as Single

AlCorte as Single

EntreTaladros as Single

Tolerancia as Single

EspesorArandela as Single

End Type

y por tanto, si la variable *Taladros* indica el número de taladros distintos que puede ser usado, se trabaja con la siguiente matriz de seguridad:

Seguridad (Taladros) as Seguridades

Para definir un grupo de tornillos correspondientes a los taladros calculados se usa la nomenclatura:

Cantidad M Diámetro / Longitud

Con la cual, si tenemos, por ejemplo, 3 tornillos de 16 mm. de diámetro y 45 mm. de longitud y 2 tornillos de 20 mm. de diámetro y 60 mm. de longitud diremos

3M16 / 45 + 2M20 / 60

5.4 Cálculo del desarrollo planar de la estructura

Debemos recordar que la estructura está formada por figuras tronco-piramidales superpuestas, cuyas caras planas tendrán diferentes pendientes respecto al plano del suelo y, para pasar esta estructura a un plano, se debe ir desarrollando el estudio desde arriba hasta abajo de la torre como si anduviésemos por sus fachadas exteriores (por ello la altura total definida en el plano será igual o superior a la que posea la estructura montada en el campo).

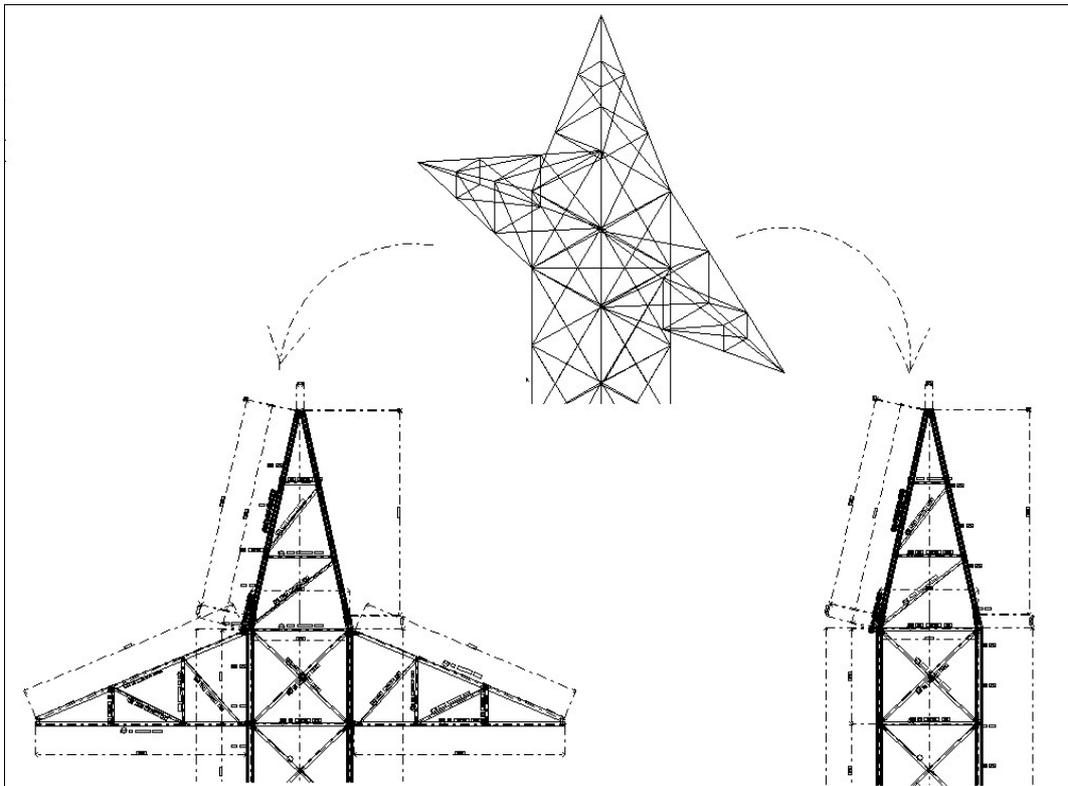


Ilustración 5-7. Transformación del cálculo 3D a los planos de fabricación

Para ello desarrollamos sobre el plano los montantes que limitan el borde exterior de la estructura. Una vez construidos éstos, se deben completar los diversos paneles que conforman las caras transversal y longitudinal de la TTE. La primera cuestión, y de especial relevancia, será el cálculo de los gramiles donde deben ir unidos los diagonales a los montantes.

5.5 Cálculo de los gramiles en los montantes y diagonales

Técnica de Búsqueda de Gramiles

Cuando observamos una junta o unión de diagonales y montantes, como la que en este gráfico tenemos, podemos notar que hay unos cuantos datos que serán necesarios para saber si se puede introducir o no los tornillos dentro del angular. Para ello es costumbre establecer las siguientes limitaciones:

1. El montante puede poseer un máximo de tres gramiles distintos por junta (esto se debe a las limitaciones habituales de las máquinas automáticas de punzonado y taladrado, llamadas normalmente *Máquinas de Control Numérico* o simplemente *Máquinas NC*)
2. El montante posee en su interior, además del espesor de la otra ala, una curva sobre la que el diagonal interno (en nuestra figura es la barra que viene por arriba) no podrá apoyarse.
3. El diagonal puede poseer un máximo de tres gramiles distintos: el mínimo (para que la tuerca no tropiece con su curva interior), el medio (a la mitad del ala) y el máximo (nunca más cerca del borde de lo que nos indican las distancias de seguridad antes señaladas).

Con el objeto de disponer de un criterio que nos permita seleccionar entre las distintas soluciones posibles, se ha diseñado un sistema de penalizaciones de acuerdo con el coste de fabricación de cada solución. Se aplica esta ponderación (penalización) a cada situación, de modo que la solución ideal sea la que tenga un menor valor ponderado.

Penalizaciones aplicadas:

- Si un tornillo sale fuera del montante, se necesitará una chapa, con el consiguiente coste, y esto se pondera con 10 unidades.
- Si un tornillo está tan cerca de la curva del montante o del otro diagonal que provoca la necesidad de un corte sobre algún diagonal, entonces se

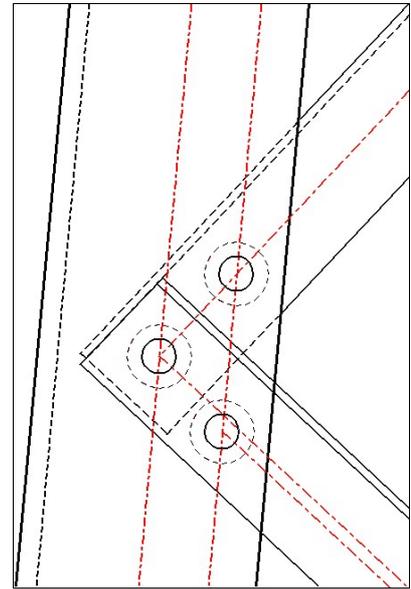


Ilustración 5-8. Diseño de Gramiles

pondera con 5 unidades (puesto que 2 cortes equivale en coste aproximadamente a una chapa).

- Si un tornillo se pone fuera del gramil medio del diagonal (en el mínimo o en el máximo), se pondera con 1 unidad (para potenciar soluciones en el gramil central frente a los laterales).

Estas limitaciones nos generan el siguiente árbol de posibilidades, en función del número de tornillos que el diseño nos indique:

Tornillo 1		Tornillo 2 Alto		Tornillo 3 Alto
Gramil 1 mínimo Provoca que el diagonal interno vaya cortado para no solapar la curva del montante		Gramil diagonal mínimo		Gramil diagonal mínimo
		Gramil diagonal medio		Gramil diagonal medio
		Gramil diagonal máximo		Gramil diagonal máximo
		Tornillo 2 Bajo		En el 2º gramil del montante hay sitio para 2 tornillos
Gramil 1 normal El diagonal interno se aleja tanto como fuere necesario para no necesitar corte		Gramil diagonal mínimo (con corte del otro diagonal o sin él)		Tornillo 3 Bajo
		Gramil diagonal medio (con corte del otro diagonal o sin él)		Gramil diagonal mínimo
		Gramil diagonal máximo (con corte del otro diagonal o sin él)		Gramil diagonal medio
				Gramil diagonal máximo
				En el 2º gramil del montante hay sitio para 2 tornillos

Si siguen quedando tornillos por colocar:

Tornillo 4 Alto		Tornillo 5 Alto
En el 2º gramil del montante hay sitio para 2 tornillos y entra sólo en el 3º		En el 2º gramil del montante hay sitio para 2 tornillos y entra con el tornillo 4 en el tercer gramil
En el 2º gramil del montante no hay sitio para 2 tornillos y entra con el tornillo 3 en el tercer gramil		Tornillo 5 Bajo
Tornillo 4 Bajo		En el 2º gramil del montante hay sitio para 2 tornillos y entra con el tornillo 4 en el tercer gramil
En el 2º gramil del montante hay sitio para 2 tornillos y entra sólo en el 3º		
En el 2º gramil del montante no hay sitio para 2 tornillos y entra con el tornillo 3 en el tercer gramil		

Al ser algunas de estas condiciones excluyentes entre sí (p.ej. si no cabe el segundo tornillo alto en el gramil máximo, no miraremos si el tercero lo hace) tendremos una cantidad P de posibilidades distintas (variable en base al número de tornillos indicado por el diseño). De ellas se escoge el conjunto de las que más tornillos introducen dentro del montante y, de este conjunto, la menos penalizada. La razón de escoger soluciones que tengan el mayor número de tornillos sobre el montante se debe a que así se disminuye el número de chapas en la estructura.

De aquí se deduce que el estudio implicará en el peor caso

$$P \times \text{Paneles} \times \text{Caras} \text{ (Transversal y Longitudinal)}$$

cálculos.

Técnica de Unificación de Gramiles

Una vez obtenidos automáticamente los tipos de juntas más “baratas y seguras”, se recomienda que el sistema escoja gramiles unificados dentro de un mismo tramo de torre o montante (con todas las juntas que éste contenga) en lugar de dejar cada unión con su configuración local calculada. Así evitamos el riesgo de que no coincidan dentro de un mismo montante los primeros gramiles (recuérdese que el cálculo del MEF asume que todos los diagonales coinciden en la misma línea imaginaria del eje central del montante).

La opción de unificar gramiles aplica el siguiente razonamiento, analizando todas y cada una de las juntas del tramo:

1. Elige el máximo de los gramiles 1. Definimos este valor como Gr_1
2. Sea *Distancia12* la distancia máxima entre el gramil 1 y el gramil 2. Entonces definimos $Gr_2 = Gr_1 + Distancia_{12}$
3. Sea *Distancia23* la distancia máxima entre el gramil 2 y el gramil 3. Entonces definimos $Gr_3 = Gr_2 + Distancia_{23}$.

De esta forma se tiene asegurado que Gr_1 , Gr_2 y Gr_3 sirven para todas las uniones. El único problema es que, mediante este cálculo, Gr_2 o Gr_3 pueden infringir las distancias de seguridad establecidas con el borde del montante, en cuyo caso se desplazan fuera del mismo. Ante esto el diseñador puede decidir entre modificar los resultados aquí establecidos o bien aceptarlos y colocar chapas que unan los tornillos internos y externos al montante.

Esta técnica de búsqueda de gramiles se puede usar de ahora en adelante, no sólo para diagonales sobre montantes, sino también para cualesquiera dos barras que deban ser unidas.

Una vez establecidos los gramiles, el resto del diseño se simplifica bastante aunque, evidentemente, se presentan nuevas dificultades que analizaremos en las secciones siguientes.

5.6 Estructuras de datos

Desde el comienzo hay que tomar la opción de cómo se deben almacenar en disco las piezas que se vayan diseñando. Cabría pensar que la solución sería almacenar cada tramo por separado como un registro, para así poder dibujarlos y construir sus planos

de fabricación; pero, por otra parte, se desea tener cada pieza aislada para poder diseñar posteriormente la orden de fábrica de dicha pieza. Entre estas dos opciones tomamos la segunda, añadiéndole a su propio esquema los parámetros necesarios para relacionarlas con el resto de la torre. Presentamos a continuación las estructuras optimizadas para cada una de las piezas:

5.6.1 Estructura de un Taladro

Type Taladros

Tipo as Integer

Cara as Integer

Gramil as Single

Progresiva as Single

Diametro as Single

Espesor as Single

Pack as Single

End Type

Todos los registros de este tipo de datos ya han sido introducidos, salvo *Tipo* y *Pack*. Con la variable *Tipo* distinguimos cuatro clases:

- Tipo 1: Taladro normal.
- Tipo 2: Taladro realizado para una junta a tope o a solape.
- Tipo 3: Taladro realizado para colocar un peldaño o escala de subida a la torre.
- Tipo 4: Taladro realizado para colocar un peldaño dentro de la junta tope o solape (es junta y escala a la vez).

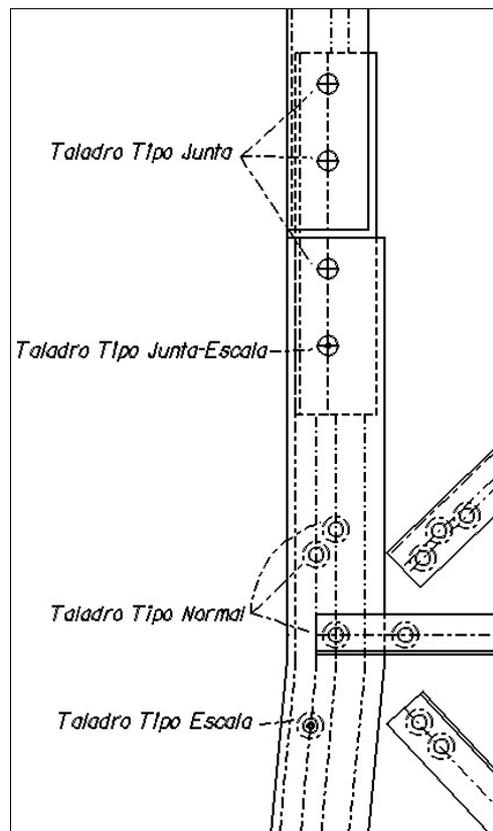


Ilustración 5-9. Tipos de Taladros

El *Pack* es una pieza añadida al tornillo (normalmente se usa una arandela o un resto taladrado de chapa) para cubrir aquellas zonas vacías existentes entre dos angulares que, aunque están unidos por el mismo tornillo, les separa alguna distancia.

5.6.2 Estructura de un Corte

Type Cortes

Tipo as Integer

Cara as Integer

Posición as Integer

X as Single

Y as Single

End Type

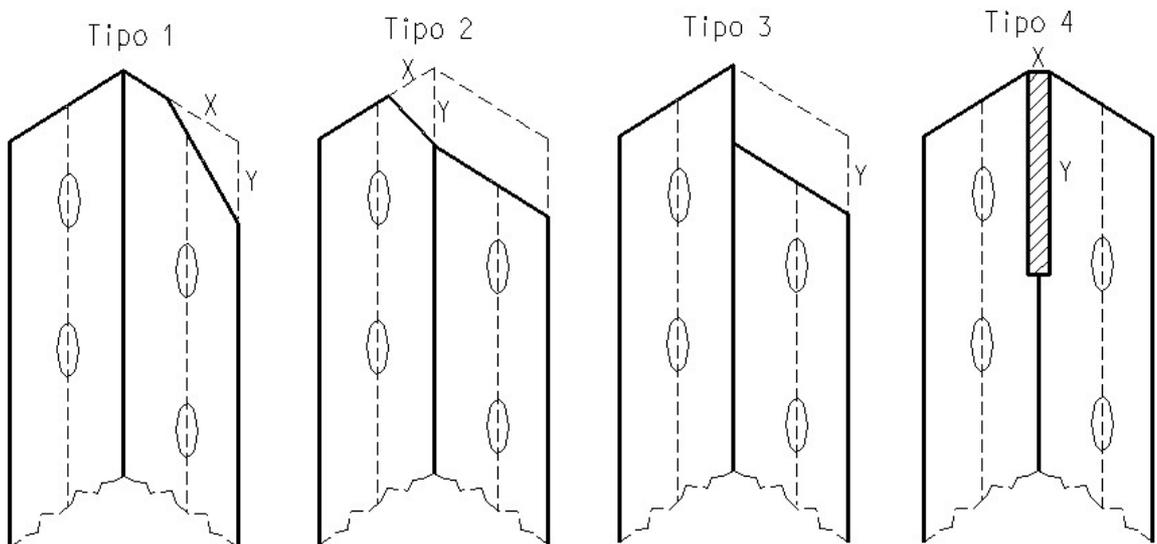


Ilustración 5-10. Tipos de Cortes

Por *Tipo* se entiende:

- Tipo 1: Inglete (corte) sobre una esquina del angular.
- Tipo 2: Corte que afecta al centro del angular de una cara y a toda la otra cara .
- Tipo 3: Corte sobre todo el ala de un lado.
- Tipo 4: Rebaje sobre el eje del angular.

La *Posición* (1,2 ó 3) nos indica Arriba, en el Centro o Abajo

5.6.3 Estructura de una Barra

Para exponer esta estructura se usan conceptos tales como: panel, diferencia entre panel y tramo, caras transversal y longitudinal, cortes, taladros, etc. que ya han sido definidos anteriormente.

Comenzamos numerando los puntos que representan a una barra en el plano (que no sea un montante):

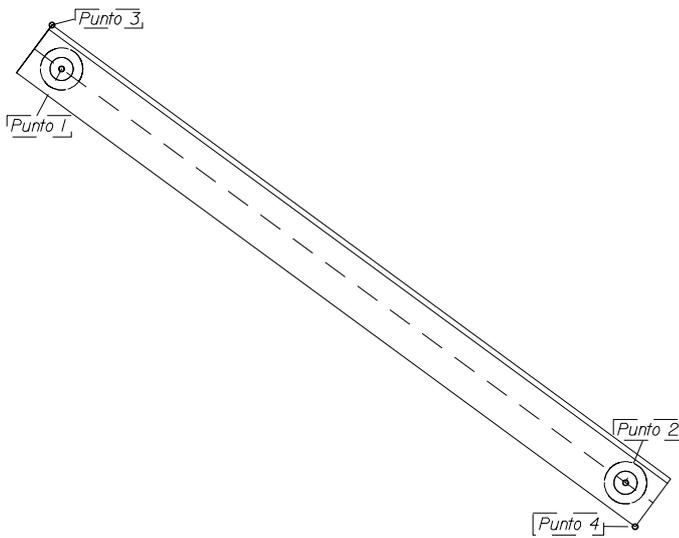


Ilustración 5-11. Puntos principales de una barra

Aquí podemos observar que el punto 3 no sólo nos define una esquina, sino que además marca el origen del angular para medir los tornillos y, de ahí, el lado donde se encuentra la otra ala proyectada del mismo [GO97]. En el caso de los montantes, se definen por tres puntos situados en el borde, siendo el central el que marca el punto de doblado de estas barras. Su

estructura será:

```
Type Puntos2D
    X as double
    Y as double
End Type
```

Incluimos, además de los cuatro puntos que aparecen en la figura, como información adicional, un máximo de 2 vectores directores de la barra. En el caso general, usaremos sólo uno y será el que señala la dirección entre los puntos 1 y 2. En el caso de los montantes, el primer vector marcará la dirección entre los puntos 1 y 2, y el segundo entre 2 y 3. Su estructura es:

```
Type Vectores2D
    X as double
    Y as double
End Type
```

Además de estas coordenadas en el plano, las definimos en el espacio (para poder comprobar si se producen interferencias de algún tipo en el montaje) a partir de las estructuras *Puntos3D* y *Vectores3D*. Con todo ello llegamos a:

```
Type Barras
    Tipo as Integer
    Tramo as Integer
    Panel as Integer
    Cara as Integer
    Grupo as Integer
   Codigo as String * 10
    Sección as integer
    Calidad as integer
    Longitud as Single
    Cantidad as Integer
    Orientación as Integer
    CantidadPuntos as Integer
    Punto(4) as Puntos2D
    CantidadVectores as Integer
    Vector(2) as Vectores2D
    CantidadPuntos3D as Integer
    Punto3D(4) as Puntos3D
    CantidadVectores3D as Integer
    Vector3D(2) as Vectores3D
    CantidadTaladros as Integer
    Taladro(200) as Taladros
    CantidadCortes as Integer
    Cortes(6) as Cortes
End Type
```

En esta estructura quedan por definir aún conceptos:

1. **Grupo** : Una vez realizado el diseño de cada tramo de torre, las barras que sean iguales a la hora de ser fabricadas (coincidan en sección, calidad, longitud, posición de taladros y cortes) deberán ser clasificadas dentro de un mismo grupo, para así ser recogidas en una única orden de fabricación o esquema de barra.
2. **Código** : El grupo aquí definido se suele notar con el siguiente código:
 Clave - Tramo - Pieza
 siendo *Clave* una tira de caracteres que identifica a cada torre en el almacén, *Tramo* el trozo de torre al que pertenece y *Pieza* la posición que ocupa dentro del tramo.
3. **Orientación**: Indica si, en el plano, el ala del angular que no se ve se dirige hacia dentro o hacia fuera (Interior-Exterior o IN-EX)

Además de la estructura de una barra tenemos la de una Chapa o Cartela que la definimos por:

Type Chapas

- Tramo as Integer
- Panel as Integer
- Cara as Integer
- Grupo as Integer
- Codigo as String * 10
- Espesor as integer
- Calidad as integer
- Cantidad as integer
- Base as Single
- Altura as Single
- CantidadVertices as Integer
- Vertice(25) as Puntos2D
- CantidadTaladros as Integer
- Taladro(100) as Taladros
- CantidadDoblados as Integer
- Doblado(6) as Doblados

End Type

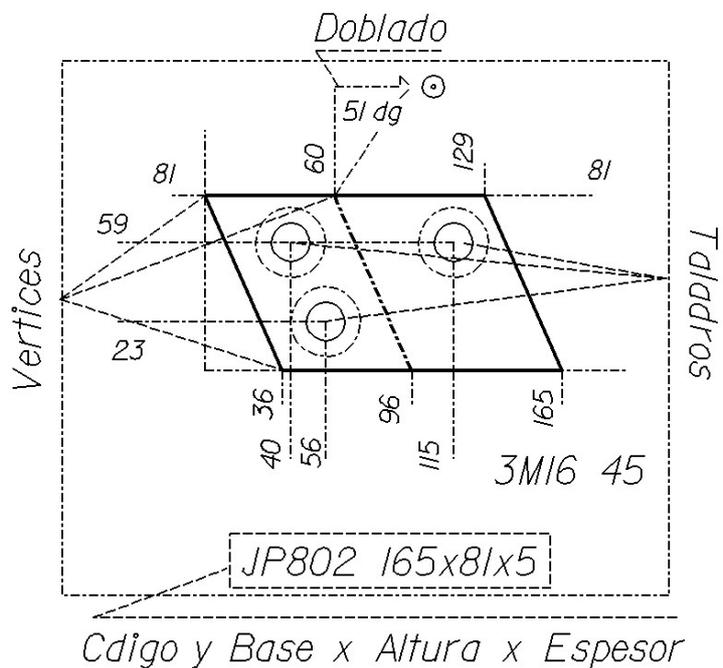


Ilustración 5-12. Distancias y datos de una chapa

Respecto a la estructura *Doblados* aquí incluida, es la que sigue:

```
Type Doblados
    VerticeOrigen as Integer
    VerticeFinal as Integer
    Angulo as Single
    Orientación as Integer
End Type
```

Con ella podemos identificar la línea imaginaria por donde debemos, en caso necesario, doblar una chapa, cuántos grados y hacia dónde. El uso de chapas dobladas es necesario en, por ejemplo, la colocación de las barras internas de las patas, en el punto de cambio de tangente en los montantes de la torre, en el interior de las crucetas, etc.

A partir de estos datos, y recorriendo panel a panel toda la estructura, se va creando cada una de las barras que la componen, tanto en su cara transversal como longitudinal, y almacenando el material en el archivo:

Torre.Mat

5.7 Modelos de recuadros y crucetas

El modelado de aquellas partes estructurales que no vienen definidas por el cálculo o que necesitan de barras complementarias han de elegirse explícitamente; este es el caso de los recuadros y de las crucetas. El sistema indica por defecto la solución más económica, pero la práctica nos muestra que la determinación del modelo final depende fundamentalmente del porcentaje de cargas al que se encuentren las barras, por lo que es recomendable tener siempre presente el resumen de cálculo.

5.7.1 Modelos de Recuadros

Se muestran a continuación los tipos de recuadros más usuales [IS97], su nomenclatura de fabricación, sus opciones y ejemplos de las mismas:

5.7.1.1 Recuadro Tipo 1

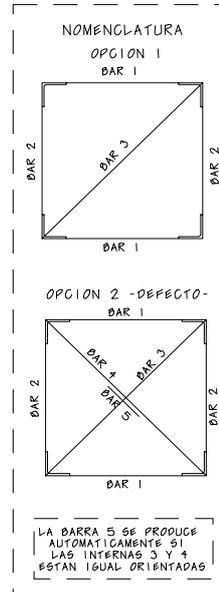
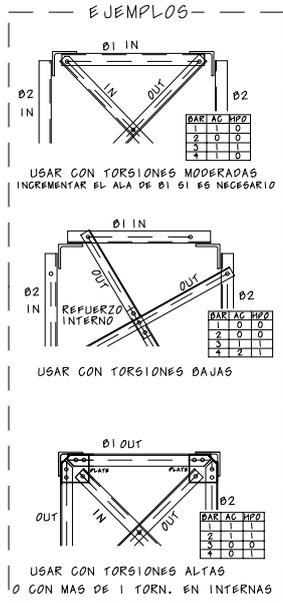


Ilustración 5-13. Ejemplos de Tipo 1

Ilustración 5-14. Nomenclatura de Tipo 1

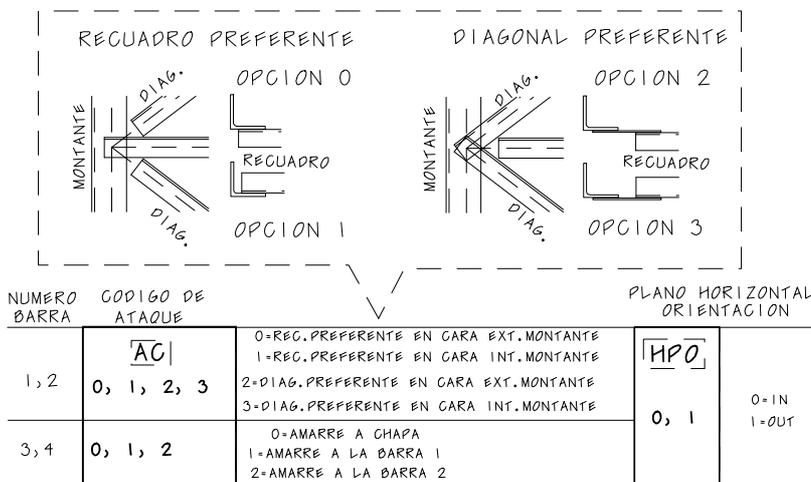


Ilustración 5-15. Opciones de posición de barras en Tipo 1



5.7.1.2 Recuadro Tipo 2

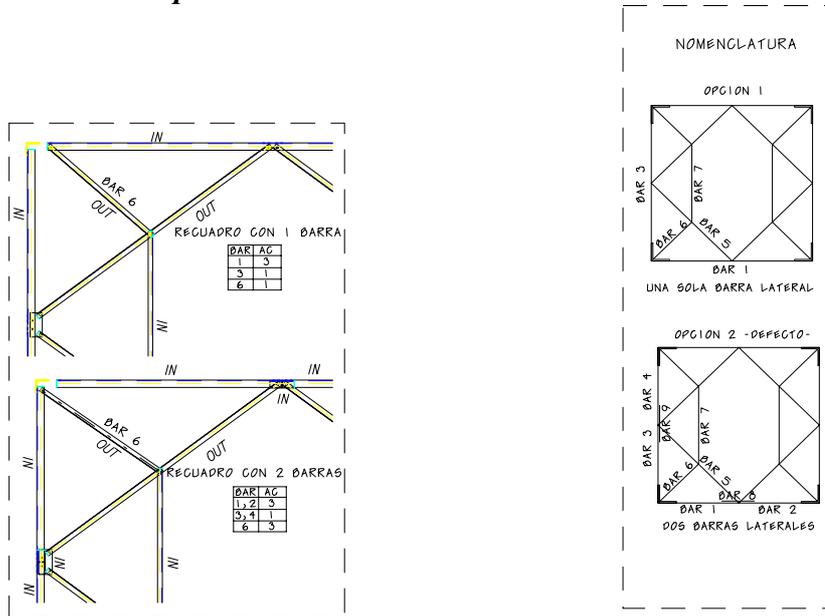


Ilustración 5-18. Ejemplos de Tipo 2

Ilustración 5-17. Nomenclatura de Tipo 2

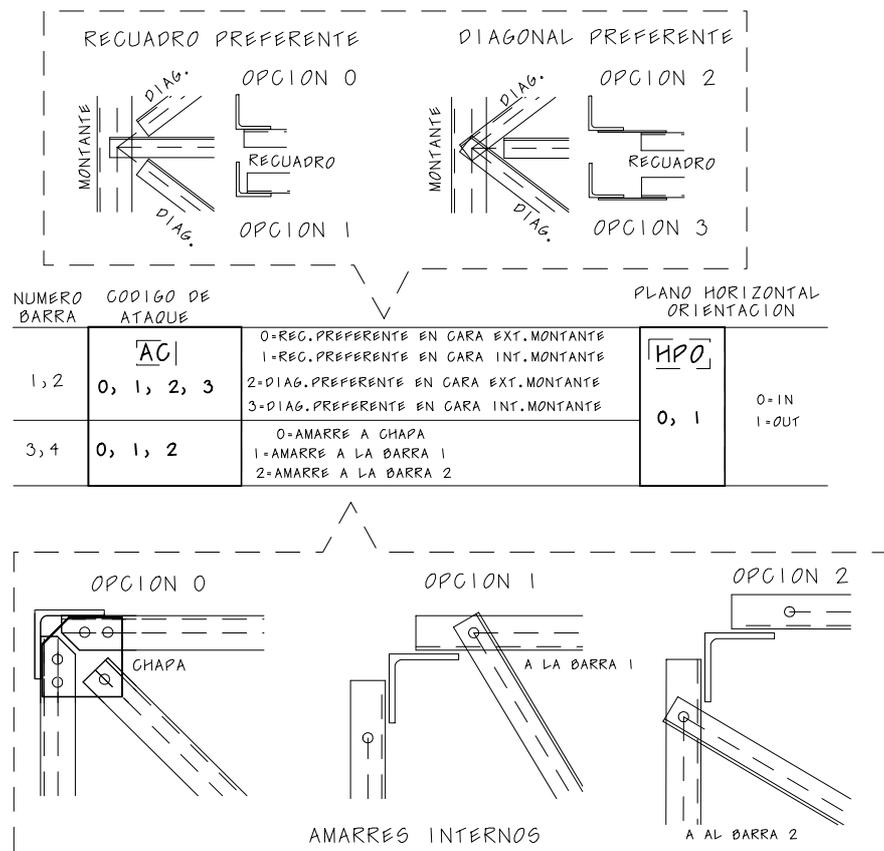


Ilustración 5-16. Opciones de posición de barra 6 en Tipo 2

5.7.1.3 Recuadro Tipo 3

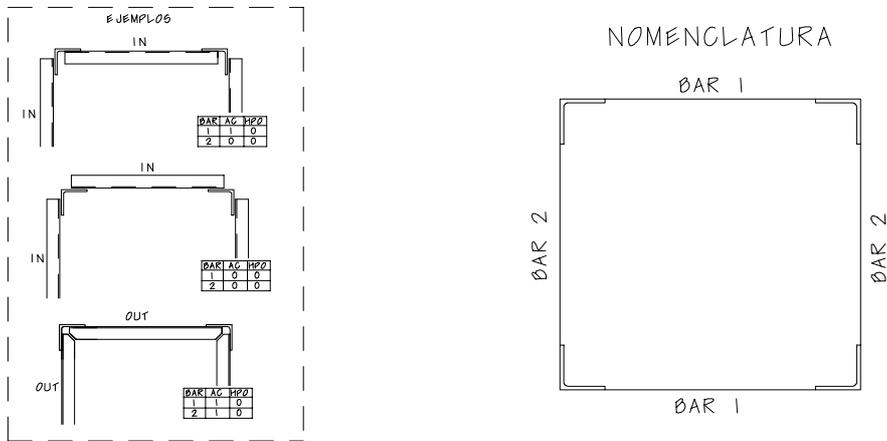


Ilustración 5-19. Ejemplos de Tipo 3

Ilustración 5-20. Nomenclatura de Tipo 3

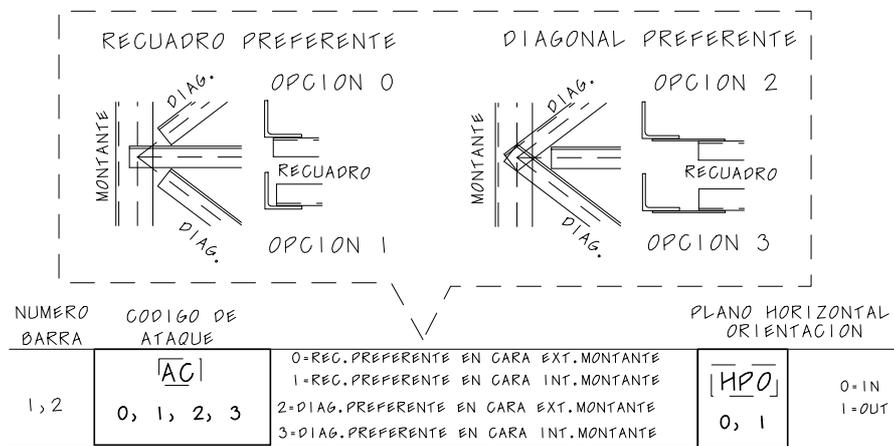


Ilustración 5-21. Opciones de barras exteriores en Tipo 3

5.7.1.4 Recuadro Tipo 4

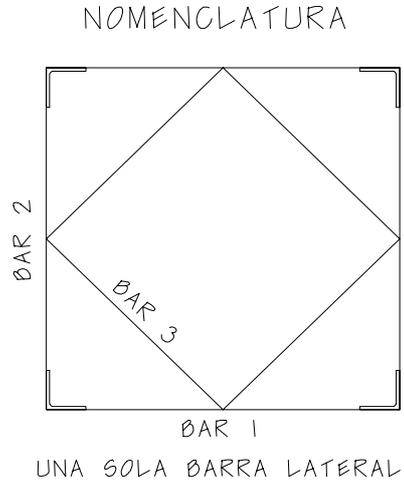
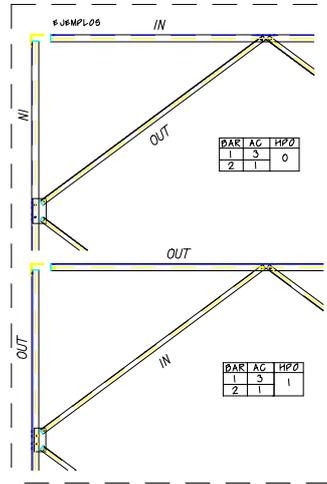


Ilustración 5-23. Ejemplos de Tipo 4

Ilustración 5-22. Nomenclatura de Tipo 4

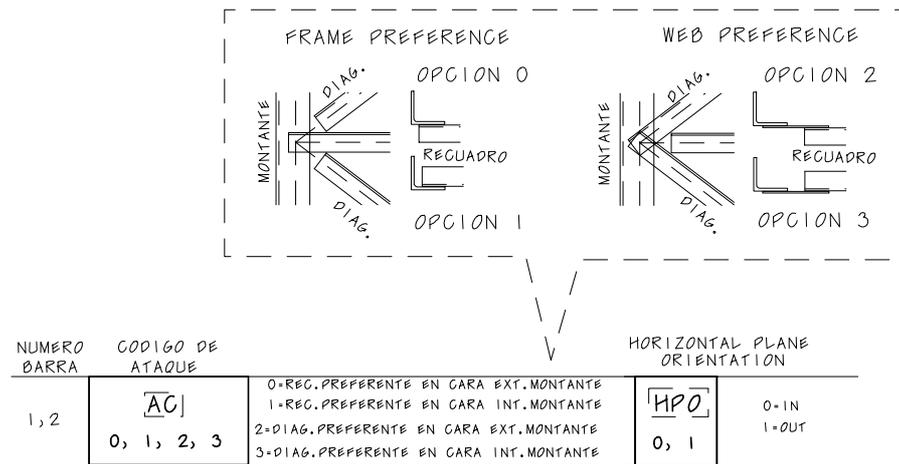


Ilustración 5-24. Opciones de barras exteriores en Tipo 4

5.7.2 Modelos de Crucetas

Elegidos los modelos de recuadros que se usarán en el diseño de la estructura y unidos a los montantes, diagonales, secundarios e internos es preciso determinar el modelo de cruceta. Los distintos modelos [IS97] numerados por C1,...,C8, se presentan a continuación:

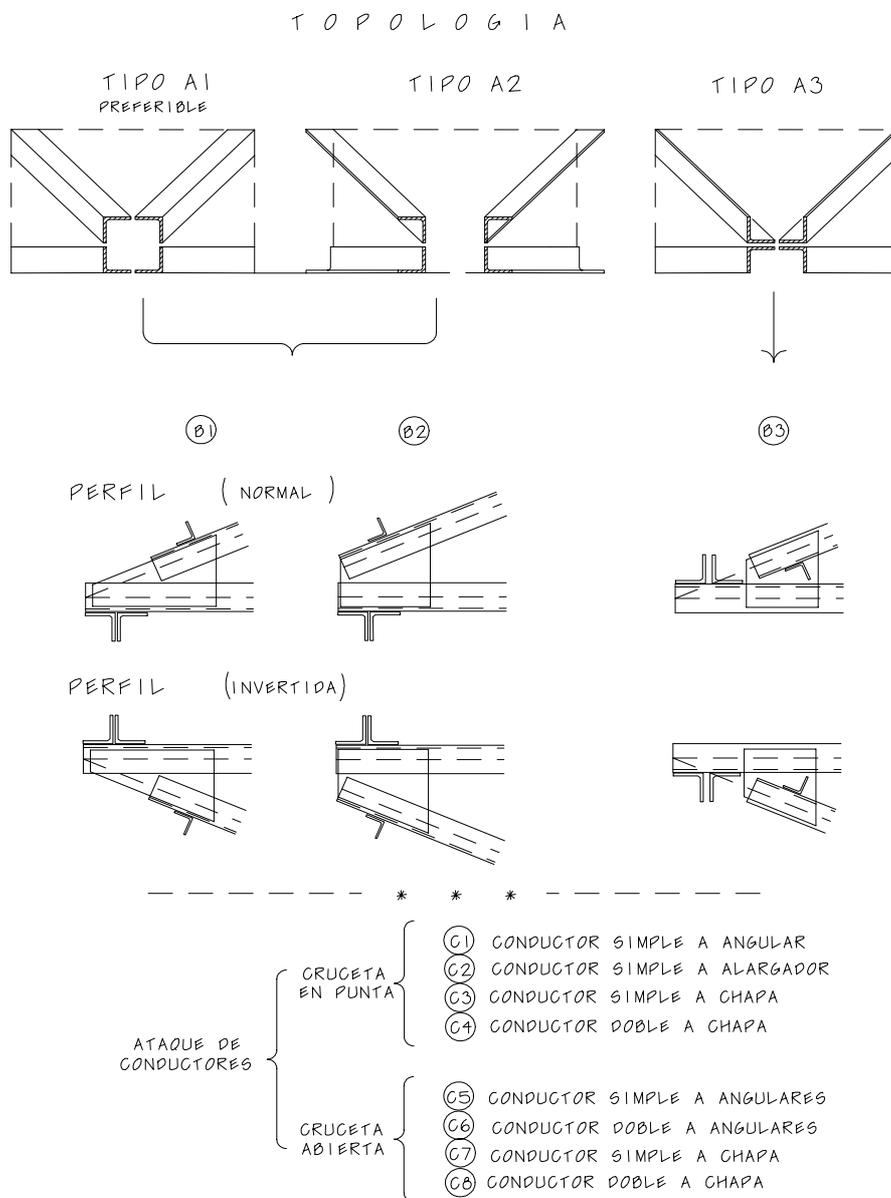


Ilustración 5-25. Disposición de piezas en crucetas



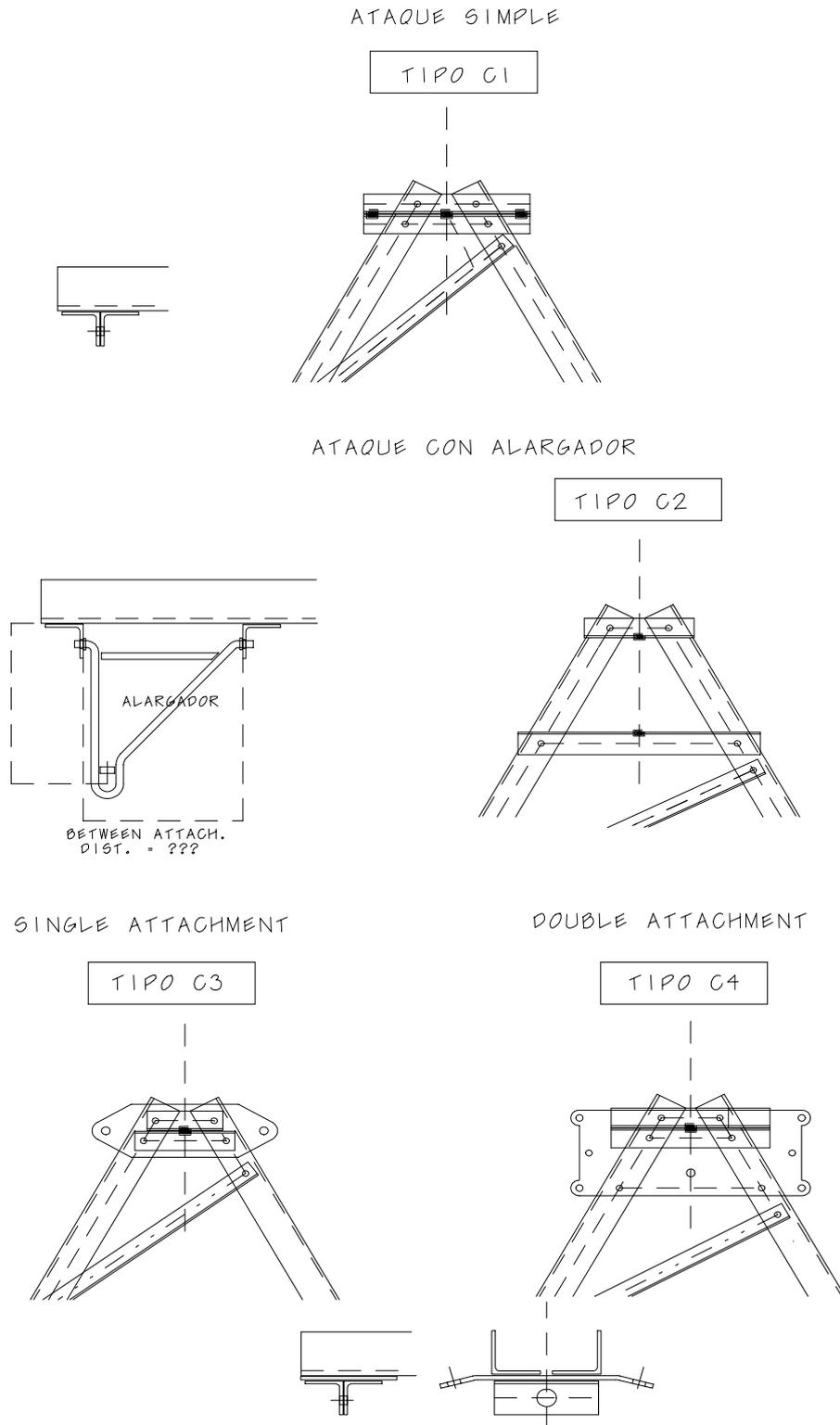


Ilustración 5-26. Cogida de conductores sobre cruceta en punta

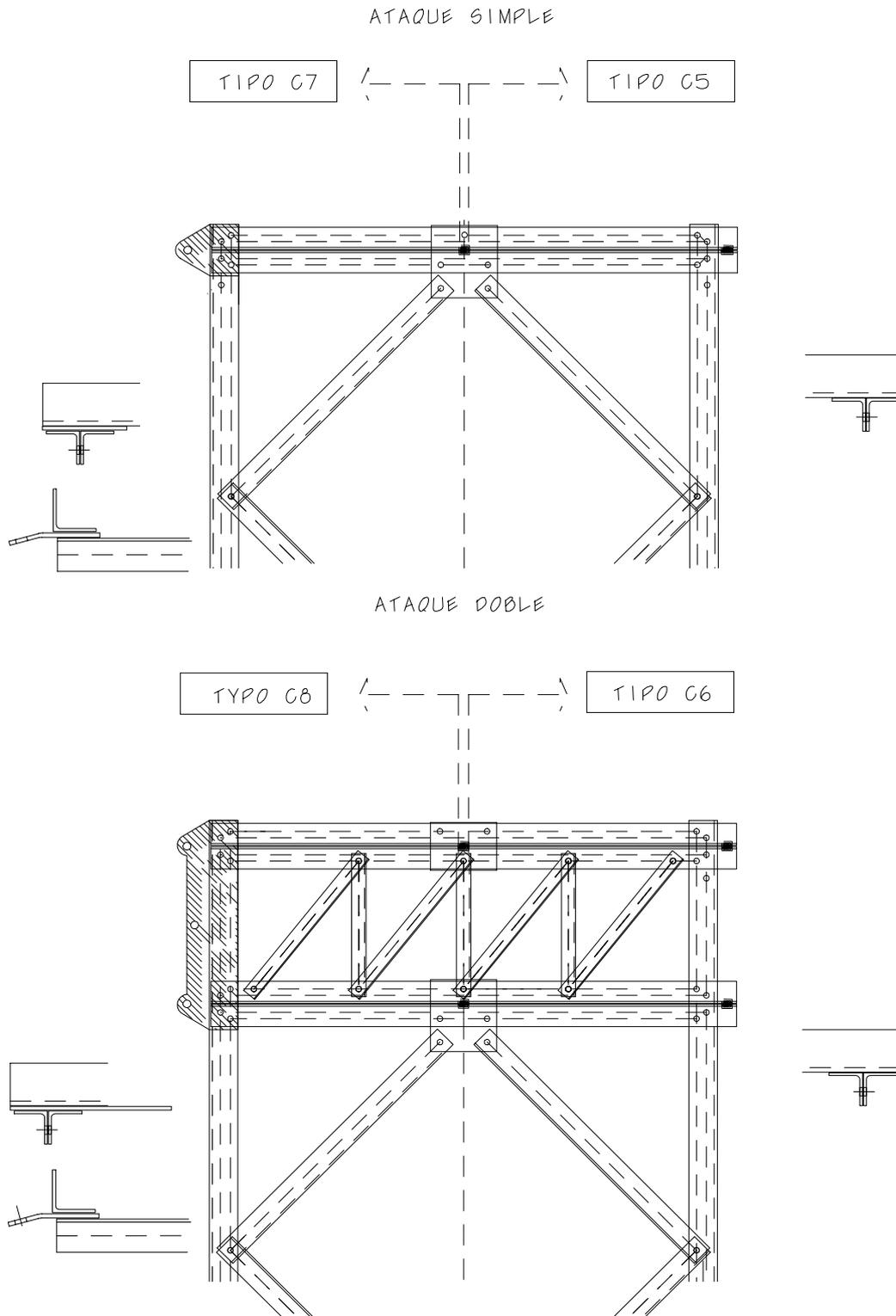


Ilustración 5-27. Cogida de conductores sobre cruceta abierta

5.8 Compresión de datos

Una vez diseñadas las barras que componen la TTE, hemos de tener en cuenta que se han podido producir repeticiones a la hora de taladrar o cortar las barras. Esto se debe a que la estructura ha sido diseñada estableciendo cada barra por separado, luego si dos diagonales, por ejemplo, están unidas en un mismo punto al montante, cada una de ellas ha provocado un taladro sobre ella misma y sobre el montante, con lo cual el sistema interpreta dos taladros sobre el montante en el mismo punto.

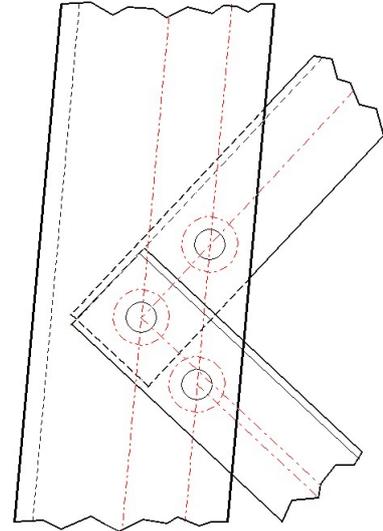


Ilustración 5-28. Compresión de datos

Para evitar estas repeticiones, se realiza una ordenación de los taladros y cortes en cada barra, desde el origen hasta el final y de la cara izquierda a la derecha, eliminando las repeticiones.

5.9 Interferencias espaciales

En el diseño de los planos no ha sido necesario utilizar todas las variables definidas en los tipos de datos correspondientes; comentamos en esta sección la utilidad de los parámetros de la estructura que nos permiten identificar cada pieza espacialmente, los cuales poseen la terminación 3D.

De forma bidimensional es posible determinar las interferencias de las barras al taladrar una sobre otra, pero siempre queda la cuestión de qué ocurrirá cuando se realice el montaje espacial de la estructura, que al fin y al cabo es el más conflictivo, ya que se realiza en el campo y las limitaciones de maquinaria para hacer una operación de corte o rebaje sobre alguna pieza son críticas.

Para analizar las posibles limitaciones en el montaje de la estructura, se realiza una simulación tridimensional y se analizan las interferencias más comunes, que son:

- Interferencia entre diagonales de caras distintas.
- Interferencia entre diagonales y recuadros.

Para realizar esta operación, se abstrae el concepto de barra y se representa como un conjunto de vértices (o puntos) y rectángulos (o caras) limitados por esos vértices, como aparece en el gráfico.

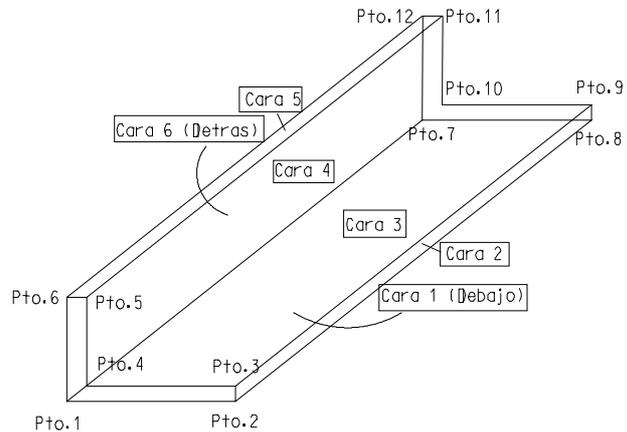


Ilustración 5-29. Modelo geométrico del Sistema Interactivo

Para el análisis de las interferencias de diagonales, en caso de no haber tresbolillo, elegimos los paneles consecutivamente de dos en dos (primero y segundo, segundo y tercero, etc.). Se recorre toda la estructura de arriba a abajo analizando las posibles interferencias entre cada dos barras y las respectivas 6 caras que cada una posee y, mediante una función vectorial de interferencia de rectángulos en el espacio determinamos su grado de interferencia [Hu02].

Mediante este método se disminuye considerablemente el número de cortes en las barras, necesarios para evitar interferencias, respecto de otros enfoques.

Cabe destacar el la representación de angulares como paralelepípedos en el programa de la Universidad de Vacon-Void (Francia) usado por Cegelec (grupo Alston), cuyo tiempo de ejecución es considerablemente más rápido (aprox. un 30%) proporciona interferencias inexistentes debido precisamente a esta representación de los angulares como paralelepípedos.

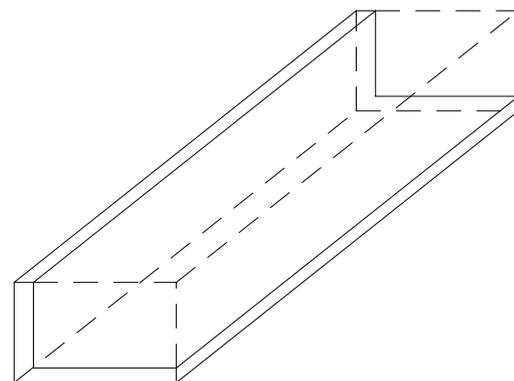


Ilustración 5-30. Modelo geométrico de la Universidad de Vacon-Void

5.9.1 Proceso algorítmico de análisis espacial de barras

Para ilustrar como se realiza el cálculo de interferencias espaciales mediante software presentamos las diferentes fases de programación.

1º Creación de estructuras para el almacenamiento de datos: sean las estructuras siguientes, donde aparecen anidadas algunas de ellas

<pre>Type Punto3D x As Double y As Double z As Double End Type</pre>	<pre>Type Vector3D x As Double y As Double z As Double End Type</pre>	<pre>Type Recta3D Punto As Punto3D Vector As Vector3D End Type</pre>
<pre>Type Plano3D Origen As Punto3D Vector As Vector3D End Type</pre>	<pre>Type Rectangulo3D Origen As Punto3D Vector1 As Vector3D Vector2 As Vector3D Ancho As Single Largo As Single End Type</pre>	<pre>Type Angular3D PlacaBase As Chapa3D Ala As Single Seccion As Integer GramilUP As Single GramilDW As Single DiametroUP As Single DiametroDW As Single AlCorteUP As Single AlCorteDW As Single End Type</pre>
<pre>Type Segmento3D Origen As Punto3D Destino As Punto3D End Type</pre>	<pre>Type Chapa3D Cara As Rectangulo3D Espesor As Single Factor As Integer End Type</pre>	

2º Modelización de las funciones geométricas básicas: a partir de las estructuras anteriores han de programarse las funciones de intersección entre ellas para llegar a la más general, que es la intersección espacial de angulares. Esta intersección está basada en el análisis de interferencias dos a dos de todos los rectángulos que forman el perímetro de cualesquiera dos angulares en los que se desee analizar sus posibles choques [Pr92, FM96]. Veamos algunos ejemplos de programación [Or92].

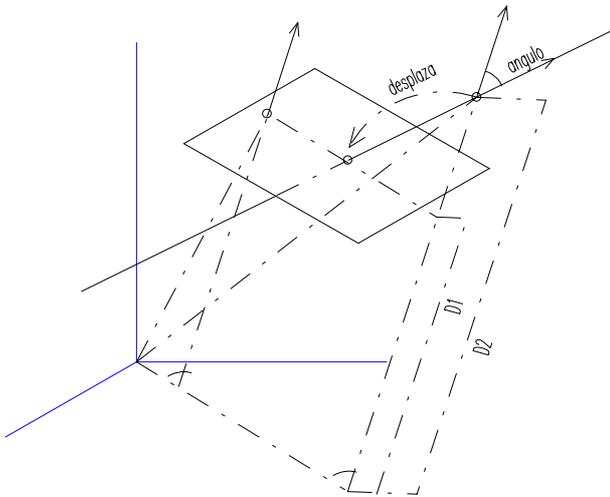


Ilustración 5-32. Intersección de plano y recta

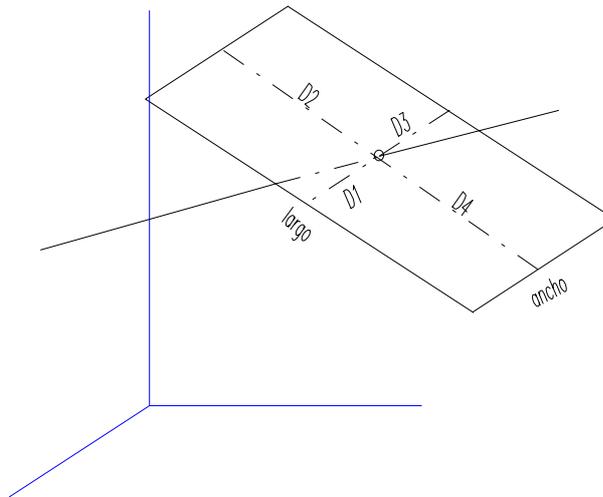


Ilustración 5-31. Intersección de recta y rectángulo

```

Sub InterseccionRectaPlano3D(Recta As Recta3D, Plano As Plano3D, Punto As Punto3D, Tipo, Desplaza)
    Coincidentes = 1: Paralelas = -1
    ' Hacemos los vectores unitarios
    Call UnificaVector3D(Plano.Vector): Call UnificaVector3D(Recta.Vector)
    Denominador = ProductoEscalar3D(Recta.Vector, Plano.Vector)
    If Abs(Denominador) = 0 Then
        Tipo = Paralelas
    Else
        D1=Plano.Origen.x*Plano.Vector.x+Plano.Origen.y*Plano.Vector.y+Plano.Origen.z*Plano.Vector.z
        D2=Plano.Vector.x*Recta.Punto.x+Plano.Vector.y*Recta.Punto.y+Plano.Vector.z*Recta.Punto.z
        Desplaza = (D1-D2) / Denominador
        Call Desplazamiento3D(Recta.Punto, Recta.Vector, Desplaza, Punto)
        Tipo = Coincidentes
    End If
End Sub

Sub InterseccionRectaRectangulo3D(Recta As Recta3D, Rectangulo As Rectangulo3D, Punto As Punto3D, Tipo, Desplaza)
    Dim Rectal As Recta3D, Centro As Punto3D, PuntoNulo As Punto3D, Plano As Plano3D
    Dim Distancial As Double, Distancia2 As Double, Distancia3 As Double, Distancia4 As Double
    Coincidentes = 1: NoCoincidentes=0: Plano.Origen = Rectangulo.Origen
    Call ProductoVectorial(Rectangulo.Vector1, Rectangulo.Vector2, Plano.Vector)
    Call InterseccionRectaPlano3D(Recta, Plano, Punto, Tipo, Desplaza)

    If Tipo = Coincidentes Then
        Rectal.Punto = Rectangulo.Origen
        Rectal.Vector = Rectangulo.Vector1: D1 = DistanPuntoRecta3D(Punto, Rectal)
        Rectal.Vector = Rectangulo.Vector2: D2 = DistanPuntoRecta3D(Punto, Rectal)
        Call Desplazamiento3D(Rectangulo.Origen, Rectangulo.Vector1, Rectangulo.Largo, Centro)
        Call Desplazamiento3D(Centro, Rectangulo.Vector2, Rectangulo.Ancho, Rectal.Punto)
        Rectal.Vector = Rectangulo.Vector1: D3 = DistanPuntoRecta3D(Punto, Rectal)
        Rectal.Vector = Rectangulo.Vector2: D4 = DistanPuntoRecta3D(Punto, Rectal)
        Difer1 = Abs(Rectangulo.Ancho - (D1 + D3))
        Difer2 = Abs(Rectangulo.Largo - (D2 + D4))
        If D1 = 0 Or D2 = 0 Or D3 = 0 Or D4 = 0 Then Difer1 = 1
        If Difer1 > 0.05 Or Difer2 > 0.05 Then Tipo = NoCoincidentes: Punto = PuntoNulo
    End If
End Sub

Sub InterseccionEntreRectangulos3D(Rectangulo1 As Rectangulo3D, Rectangulo2 As Rectangulo3D, Segmento As Segmento3D, Tipo, Distancia)
    Dim Recta As Recta3D, Punto(8) As Punto3D, PuntoNulo As Punto3D
    Coincidentes=1: NoCoincidentes=0: YES=1: NO=0: Pto=1
    '.....
    '..Intersección del borde del rectángulo 2 con el rectángulo 1..
    '.....
    Recta.Vector = Rectangulo2.Vector1: Recta.Punto = Rectangulo2.Origen
    Call InterseccionRectaRectangulo3D(Recta, Rectangulo1, Punto(Pto), Tipo, Distancia)
    If Tipo=Coincidentes And Distancia>0 And_

```

```

        DistanciaPuntos3D(Recta.Punto, Punto(Pto)) < Rectangulo2.Largo Then Pto=Pto+1

Recta.Vector = Rectangulo2.Vector2

Call InterseccionRectaRectangulo3D(Recta, Rectangulo1, Punto(Pto), Tipo, Distancia)

If Tipo=Coincidentes And Distancia>0 And_
    DistanciaPuntos3D(Recta.Punto, Punto(Pto)) < Rectangulo2.Ancho Then Pto=Pto+1

Call Desplazamiento3D(Recta.Punto, Rectangulo2.Vector1, Rectangulo2.Largo, Punto(0))
Recta.Punto = Punto(0): Recta.Vector = Rectangulo2.Vector2
Call InterseccionRectaRectangulo3D(Recta, Rectangulo1, Punto(Pto), Tipo, Distancia)
If Tipo=Coincidentes And Distancia>0 And_
    DistanciaPuntos3D(Recta.Punto, Punto(Pto)) < Rectangulo2.Ancho Then Pto=Pto+1

Call Desplazamiento3D(Recta.Punto, Rectangulo2.Vector2, Rectangulo2.Ancho, Punto(0))
Recta.Punto = Punto(0): Recta.Vector = Rectangulo2.Vector1
Call InterseccionRectaRectangulo3D(Recta, Rectangulo1, Punto(Pto), Tipo, Distancia)
If Tipo=Coincidentes And Distancia<0 And_
    DistanciaPuntos3D(Recta.Punto, Punto(Pto)) < Rectangulo2.Largo Then Pto=Pto+1

'.....
'..Intersección del borde del rectángulo 1 con el rectángulo 2..
'.....

Recta.Vector = Rectangulo1.Vector1: Recta.Punto = Rectangulo1.Origin
Call InterseccionRectaRectangulo3D(Recta, Rectangulo2, Punto(Pto), Tipo, Distancia)
If Tipo=Coincidentes And Distancia>0 And_
    DistanciaPuntos3D(Recta.Punto, Punto(Pto)) < Rectangulo1.Largo Then Pto=Pto+1

Recta.Vector = Rectangulo1.Vector2
Call InterseccionRectaRectangulo3D(Recta, Rectangulo2, Punto(Pto), Tipo, Distancia)
If Tipo=Coincidentes And Distancia>0 And_
    DistanciaPuntos3D(Recta.Punto, Punto(Pto)) < Rectangulo1.Ancho Then Pto=Pto+1

Call Desplazamiento3D(Recta.Punto, Rectangulo1.Vector1, Rectangulo1.Largo, Punto(0))
Recta.Punto = Punto(0): Recta.Vector = Rectangulo1.Vector2
Call InterseccionRectaRectangulo3D(Recta, Rectangulo2, Punto(Pto), Tipo, Distancia)
If Tipo=Coincidentes And Distancia>0 And_
    DistanciaPuntos3D(Recta.Punto, Punto(Pto)) < Rectangulo1.Ancho Then Pto=Pto+1

Call Desplazamiento3D(Recta.Punto, Rectangulo1.Vector2, Rectangulo1.Ancho, Punto(0))
Recta.Punto = Punto(0): Recta.Vector = Rectangulo1.Vector1
Call InterseccionRectaRectangulo3D(Recta, Rectangulo2, Punto(Pto), Tipo, Distancia)
If Tipo=Coincidentes And Distancia<0 And_
    DistanciaPuntos3D(Recta.Punto, Punto(Pto)) < Rectangulo1.Largo Then Pto=Pto+1

'.....
'..Carga los puntos de intersección en el segmento..
'.....

Marca = NO: Fin = Pto - 1: Tipo = NoCoincidentes
If Fin > 1 Then
    For Pto = 1 To Fin
        If (DistanciaPuntos3D(PuntoNulo, Punto(Pto)) > 0) Then
            If (Marca = YES And DistanciaPuntos3D(Segmento.Origin, Punto(Pto)) > 0) Then
                Segmento.Destino = Punto(Pto)
            Else
                Segmento.Origin = Punto(Pto): Marca = YES: Tipo = Coincidentes
            End If
        End If
    Next Pto
End If
End Sub

```

5.10 Creación de peldaños (escala)

Una vez definidos todos los taladros existentes en los diferentes montantes, puede ser necesario colocar peldaños para poder subir por la torre. Para tal fin el sistema solicita tres tipos de datos:

1. Diámetro del peldaño
2. Distancia recomendable entre un peldaño y el siguiente
3. Altura mínima sobre el nivel del suelo a partir de la cual comienza la escalada (normalmente es superior a 3 m. para que ninguna persona ajena al mantenimiento de las torres pueda subirse por ella)

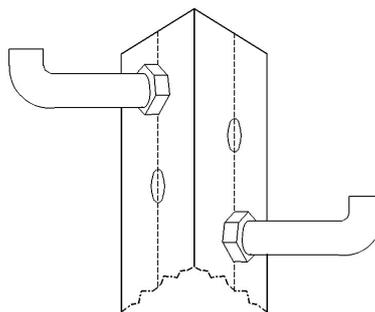


Ilustración 5-33. Peldaños

Con estos parámetros se recorre la estructura de arriba a abajo (empezando habitualmente a un metro de la cúspide) calculando el lugar donde debe colocarse cada peldaño (almacenada esta posición en la variable *Altura*) y se analiza si en el intervalo ($Altura - 100\text{ mm.}$, $Altura + 100\text{ mm.}$) aparece algún taladro ya existente (el valor de 100 mm es el desplazamiento permisible para no hacer excesivamente desiguales los pasos entre peldaños):

- Si es así, se determina cual de ellos está más cerca del valor con un diámetro igual o superior al especificado para dichos peldaños. Una vez elegido se modifica su *Tipo* y se convierte también en un taladro de escala
- Si no existen taladros, lo colocamos en el valor de *Altura* y en el primer gramil del montante.

5.11 Clasificación de material (código de fabricación)

Supóngase que se dispone de la base de datos global con todo el material necesario, en la que se han analizado los cortes y taladros de cada pieza. En este momento es conveniente clasificar todos los registros mediante el siguiente orden de prioridades:

- | | |
|---|--------------------------|
| 1. Tramos de torre (o tramos de fabricación). | 4. Calidad del material. |
| 2. Longitud de la barra. | 5. Posición de taladros. |
| 3. Sección de material. | 6. Posición de cortes. |

Cuando dos piezas coinciden en estos datos se forma un grupo, se elige una de ellas como representante y sumamos las cantidades de piezas iguales que cada una de ellas representa. Incorporamos dicha suma al representante elegido y anulamos la cantidad de la representada.

Las piezas repetidas no pueden ser borradas de la base de datos ya que, aunque son físicamente iguales (e idénticas desde el punto de vista de la fabricación), no lo son como estructura de datos porque son diferentes sus coordenadas espaciales, debiendo mantenerse para su representación posterior en los planos (son diferentes desde el punto de vista del montaje).

Una vez realizada esta clasificación se numeran los representantes de grupos por el código de fábrica al que ya nos referimos anteriormente:

Código : Clave - Tramo - Pieza

siendo *Clave* una tira de caracteres que identifica a cada torre en el almacén, *Tramo* el trozo de torre al que pertenece y *Pieza* la posición que ocupa dentro del tramo. Estos códigos de fábrica aparecen marcados sobre cada barra para facilitar su posterior montaje.

5.12 Longitudes y suma de tornillos

Cuando se clasifica el material, se realiza la representación de las piezas en los planos. Para ello se transforman los datos del archivo *Torre.Mat* en rectas, circunferencias, textos, curvas, etc. En este punto se plantea el problema de determinar la longitud necesaria de cada tornillo (no es lo mismo el tornillo que atraviesa dos barras que el que atraviesa tres o una barra y una chapa, etc.). Para ello, mientras se realiza esta representación se almacenan paralelamente los datos de cada círculo que represente a un taladro. Terminado este proceso, se clasifican estos círculos por sus coordenadas en el plano y, en caso de coincidir, se suman sus respectivos espesores y packs, con lo cual se obtiene la longitud total que debe atravesar el tornillo, y a ésta se le añaden los espesores de la arandela y de la tuerca necesarias. Este dato se aplica sobre una curva de regresión de la forma:

Longitud de mercado = F (Diámetro , Longitud atravesada)

que nos irá dando las longitudes de tornillo necesaria entre las existentes en el mercado y suficientes para salvar la longitud establecida. Colocados los textos que indican el tipo y longitud de cada tornillo, se acumulan, estableciendo cuántos tornillos y de qué tipo requiere el montaje de cada sección de torre.

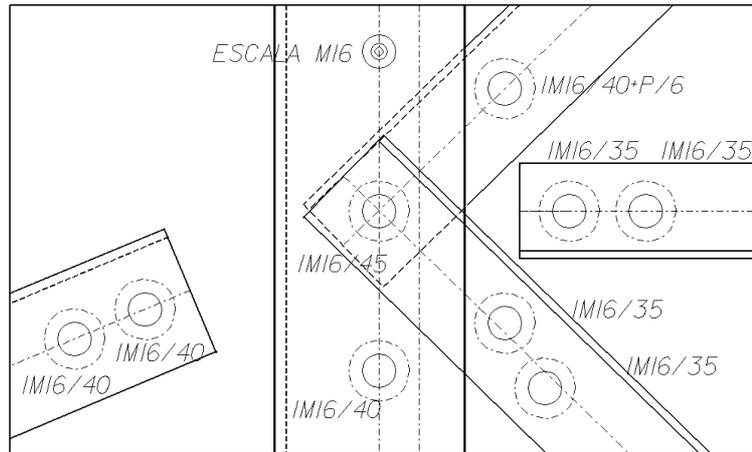


Ilustración 5-34. Acumulación de taladros

5.13 Algoritmo de Fabricación

Después de los pasos anteriores, el sistema de fabricación se resume en el siguiente algoritmo. Sea *Archivo* la variable que se direcciona a Torre.Mat:

CalculoDesplazamiento(Desplazamiento)
 EntradaFactoresSeguridad (Seguridad() as Seguridades)
 DiseñoGramiles (Barra() as Barras)
 DiseñoMontantes (Barra() as Barras, Chapa() as Chapas, Archivo)
 DiseñoRecuadros (Barra() as Barras, Chapa() as Chapas, Archivo)
 DiseñoDiagonales (Barra() as Barras, Chapa() as Chapas, Archivo)
 DiseñoSecundarios (Barra() as Barras, Chapa() as Chapas, Archivo)
 DiseñoInternos (Barra() as Barras, Chapa() as Chapas, Archivo)
 DiseñoCruceas (Barra() as Barras, Chapa() as Chapas, Archivo)
 CompresiónDatos (Barra() as Barras, Chapa() as Chapas, Archivo)
 InterferenciasEspaciales (Barra() as Barras, Archivo)
 CreaciónPeldaños (Barra() as Barras, Archivo)
 ClasificaciónMaterial (Archivo)
 DiseñoPlanos (Archivo)
 ContabilizaciónTornillos (Archivo)

5.14 Diseño de chapas (Cartelas)

Finalizada la realización automática de los planos de fabricación, es necesaria la determinación de los tipos de chapas allí donde se considere conveniente. Esta tarea no se ha implementado dentro del algoritmo general de fabricación debido a la diversidad de criterios existentes entre los profesionales del diseño de torres. Por ello se han creado unas herramientas integradas en el sistema gráfico, que permiten poder dibujar las chapas a gusto del diseñador. Como ejemplos acompañamos a continuación los dos modelos más usuales, la chapa rectangular y la periférica. El primer paso será señalar el conjunto de taladros que queremos agrupar, y proporcionar esta información al algoritmo de diseño [Kl89, Ok92]. Consideramos la estructura de un taladro para cartela de la forma:

```
Type TaladroCartela
    Radio as Single
    X as Double
    Y as Double
End Type
```

5.14.1 Cartela Rectangular

Sea n el número de taladros que se encuentran en un área determinada. Consideremos el vector $Taladro(n)$ as $TaladroCartela$. Para la simplificación de los cálculos se solicita como dato adicional que sea señalado un vector de dirección paralelo a uno de los lados del rectángulo buscado. Llamemos a este vector Eje . Calculamos el ángulo de inclinación de Eje respecto al eje de coordenadas Y . A partir de ese ángulo α se realiza una rotación de las coordenadas de cada taladro (mediante la matriz de giro), dando como solución un vector resultante de la forma:

$TaladroRotado(n)$ as $TaladroCartela$.

Esta transformación la hacemos, para todo $i=1, \dots, n$, mediante la fórmula de rotación [SU00]:

$$(Taladro(i).X, Taladro(i).Y) \cdot \begin{pmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{pmatrix} = (TaladroRotado(i).X, TaladroRotado(i).Y)$$

Situados en este nuevo sistema:

- Se calculan las coordenada límite por encima, debajo, derecha e izquierda,
- Se realiza un desplazamiento acorde con la cantidad estipulada hasta el borde en:
$$\text{Seguridad}(\text{Diámetro}) \cdot \text{AlBorde},$$
- Se determinan los cuatro vértices del rectángulo
- Se rotan estos cuatro vértices mediante el ángulo $-\alpha$, usando el mismo mecanismo anterior, y obtenemos la cartela.

Por último, guardamos la chapa rotada en el archivo Torre.MAT.

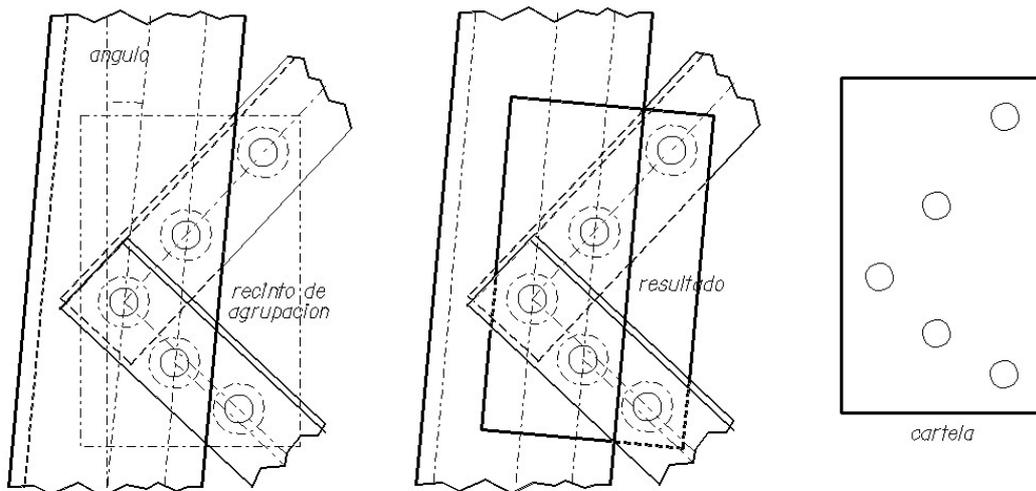


Ilustración 5-35. Creación de chapa rectangular

5.14.2 Cartela Periférica

Cuando se desea ajustar más el diseño de las chapas a los taladros, ya que se posee máquinas láser de corte que pueden seguir cualquier tipo de perímetro, se procede de la siguiente manera [Be97]:

Para una chapa periférica es necesario definir previamente un sistema de elección de los taladros del perímetro que envuelven a todo el grupo [Wi85, Wi87]. Este algoritmo se basa en:

- Elegir el taladro cuyas coordenadas tengan el máximo Y. Si hubiese más de uno tomar para el perímetro los que tengan la mínima y la máxima X.
- Definir la recta que pasa por el punto anterior con dirección $(1,0)$, o por los puntos anteriores si no fuese único.

- Rotar a derecha e izquierda desde ese punto o puntos con líneas, hasta tocar el primer taladro.
- Situarse ahora en estas posiciones encontradas y realizar el paso anterior, así continuaremos hasta cerrar el perímetro.
- Desplazarse de este perímetro la cantidad estipulada hasta el borde en:

$$\text{Seguridad}(\text{Diámetro}).\text{AlBorde},$$
- Diseñar los vértices de la chapa.
- Guardar sus datos en Torre.MAT.

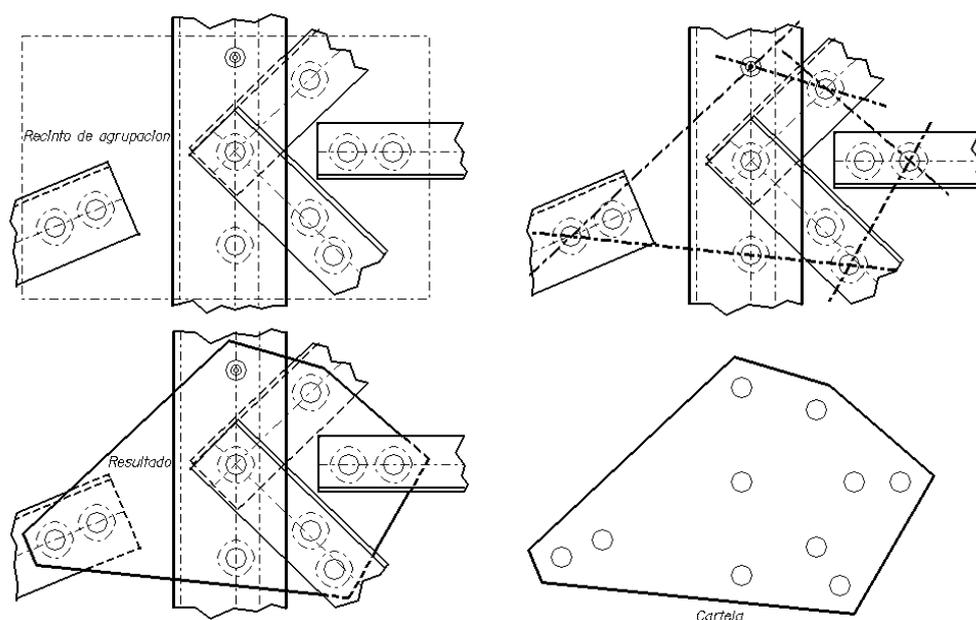


Ilustración 5-36. Creación de chapa perimétrica

Para establecer la lista de materiales de cada tramo de torre, es preceptivo añadir el peso de cada pieza. En el caso de las chapas periféricas, y dado que conocemos el perímetro de las mismas y el tipo de material usado, sólo necesitamos conocer el área de cada una. Para el cálculo del área usamos un caso particular del teorema de Green-Riemann, como sigue [Ga98]:

Sea el recinto A encerrado por el perímetro P :

Sean $P(x,y)$ y $Q(x,y)$ funciones continuas con derivadas parciales continuas en una región R y sobre su frontera C . Entonces:

$$\oint_C P(x,y)dx + Q(x,y)dy = \iint_R \left(\frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y} \right) dx dy$$

Si aplicamos los siguientes cambios sobre este resultado:

1. Sea $C=P$ (Perímetro)
2. Sea $R=A$ (Area encerrada en P)
3. Sea $P(x,y)=0$
4. Sea $Q(x,y)=x$

Obtenemos :

$$\oint_P 0dx + xdy = \iint_A \left(\frac{\partial(x)}{\partial x} - \frac{\partial(0)}{\partial y} \right) dx dy \Rightarrow \oint_P xdy = \iint_A dx dy \Rightarrow Area(A) = \oint_P xdy$$

siendo (x,y) las coordenadas paramétricas de las rectas que limitan el área calculada.

5.15 Aportaciones realizadas

- Establecer la fórmula de aproximación de la línea imaginaria de cálculo a la real del desarrollo físico de la torre
- Desarrollar el algoritmo de búsqueda automática y unificación de gramiles en el diseño de las torres
- Crear las estructuras computacionales de barra, chapa, taladro, corte y doblado
- Disminuir el número de interferencias espaciales en la estructura y sus correspondientes cortes a través de modelos matemáticos más cercanos a la ubicación tridimensional y el perímetro real de las piezas
- Establecer el algoritmo de fabricación de una torre
- Desarrollar las bases numéricas para la creación de chapas rectangulares y periféricas mínimas a partir de un conjunto de taladros en el plano

5.16 Referencias bibliográficas

[Be97] Berg. M. "Computational Geometry: Algorithms and Applications", Springer, 1997.

[FM96] Félez, J.; Martínez, M.L.; Cabanellas, J.M.; Carretero, A. "Fundamentos de Ingeniería Gráfica". Síntesis. Madrid. 1996.

[Ga98] Galán García, JL " Análisis vectorial para la Ingeniería. Teoría y problemas". Editorial Bellisco, 1998

- [GO97] Goodman, J.; O'Rourke, J. "Handbook of Discrete and Computational Geometry" CRC Press, 1997
- [Hu02] Huerta, J. "Modelado de Sólidos". Apuntes de Diseño y Fabricación asistida por Ordenador. Universidad Jaume I. Castellón. España. 2002.
- [IS97] ISI (Ingeniería de Sistemas Interactivos Ltda.) . "Computer integrated transmission towers system". Manual. Málaga. España. 1997.
- [Kl89] Klein, R. "Concrete and Abstract Voronoi Diagrams" Springer-Verlag, 1989
- [Ok92] Okabe, A. "Spatial Tessellations, Concepts and Applications of Voronoi Diagrams", John Wiley & Sons, 1992
- [Or94] O'Rourke, J. "Computational Geometry in C", Cambridge Univ. Press, 1994.
- [Pr92] Prieto, M. "Fundamentos geométricos del diseño en ingeniería". ADI. Madrid. 1992.
- [SU00] Sack, J.-R. ,Urrutia, J. "Hanbook of computational geometry" Elsevier Science Publishers B.V. North-Holland, 2000.
- [Wi85] Winter, P. "An algorithm for the Steiner problem in the Euclidean plane". Networks, vol. 15. 1985
- [Wi87] Winter, P. "Steiner problem in networks". Networks, vol. 17. 1987

6. Análisis de Residuo Mínimo

A menudo nos encontramos con problemas en la vida real que, a pesar de disponer de una modelización matemática simple, son de difícil tratamiento desde el punto de vista computacional. Ante este tipo de conflictos, la industria actual se enfrenta día a día, teniendo que compatibilizar la generalización (modelos matemáticos) [GK94, JM00] y la utilidad (algoritmos) [GK81].

6.1 Planteamiento

Desde sus inicios, un problema crucial y de difícil solución en la fabricación de torres y en otros sectores ha sido la aglomeración y aprovechamiento de los retales (o restos) que surgen tras la realización de cualquier proyecto. Este dilema llega a ser tan grave que la mayoría de las empresas, debido al enorme gasto en el almacenamiento de material usado, se ven obligadas a seguir una estricta política en la manipulación y uso del material nuevo. Todo ello, evidentemente, conlleva la necesidad de un estudio adecuado que permita combinar, satisfactoriamente, las necesidades y la minimización del residuo. Con esta motivación, basada en criterios económicos, llegamos ante al siguiente problema de optimización [Go80]:

En un proyecto de una línea de transmisión eléctrica se obtiene una ingente cantidad de material que ha de ser manufacturado para las torres. El número de barras que forma parte de ese volumen y que van a ser cortadas, galvanizadas, etcétera, será habitualmente de decenas de miles. Como esperamos, al llegar a comprar a la industria siderúrgica, las barras van a ser vendidas con una medida determinada (normalmente las medidas estándar suelen ser 12, 14 ó 16 m. aunque en casos de grandes cantidades pueden ser producidas a la longitud que se desee). Aquí necesitaremos un buen mecanismo de ordenación que nos indique cuál es el número total de barras que hemos de comprar y, sobre todo, cómo van a ser cortadas.

Es fácil observar que nos encontramos ante una decisión que puede producir beneficios o pérdidas millonarios, ya que hablamos de miles de toneladas de material.

Por ello, será preciso valorar que la solución, de no ser la óptima, sea lo suficientemente buena.

Por otra parte, en la mayoría de las ocasiones, la solución se necesita tan pronto como sea posible, ya que de ello depende el comienzo de una obra (es decir, evitar que los talleres y el personal se encuentren inactivos, lo cual implicaría también grandes pérdidas). Por tanto, será necesaria además una solución suficientemente rápida.

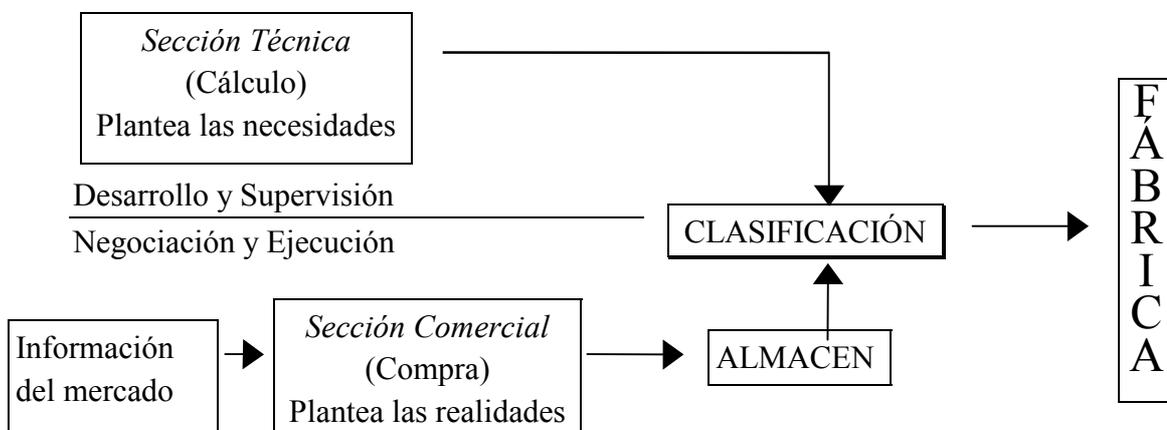
Un tercer inconveniente con que nos encontramos es la imposibilidad técnica de disponer de máquinas (computadoras) con tanta memoria como nos sea necesario (además, el hecho de necesitar gran cantidad de memoria traería consigo mayor lentitud en el proceso). Por tanto, intentaremos evitar todos aquellos procedimientos que impliquen manejos de matrices o bancos de datos dinámicos desmesurados.

6.1.1 Organización industrial

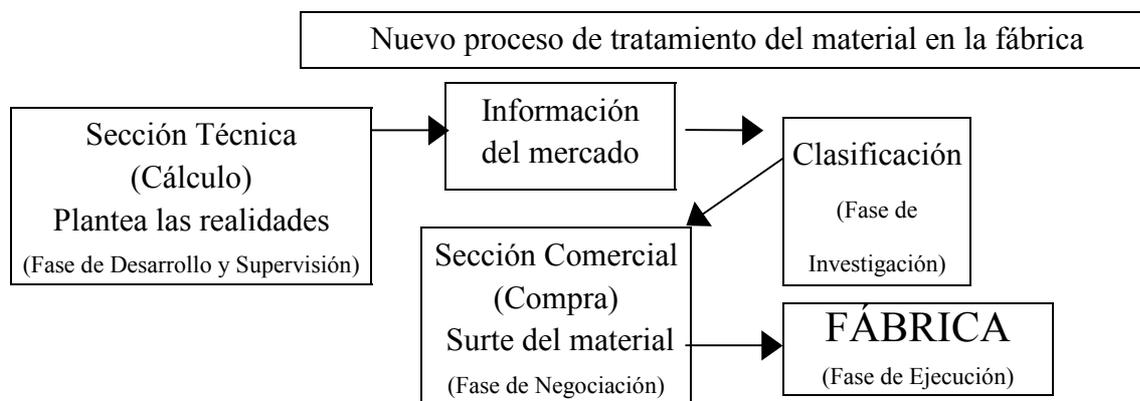
El problema que aquí nos planteamos, con todas estas restricciones, se podría denominar problema de minimización de retal o análisis de residuo mínimo, de ahí el título de este capítulo. El objetivo que nos proponemos es encontrar caminos que nos conduzcan a soluciones satisfactorias del problema de optimización de los mismos.

Actualmente, en las empresas de construcción de estructuras, los resultados y las necesidades de los departamentos de cálculo y diseño, se ven mediatizados por las realidades económicas: suele ocurrir que el stock de piezas en el almacén influye directamente en el diseño de las estructuras, de forma que se ha de resolver un problema de optimización relativo a las piezas en stock [Fe73, Pe87]. Esta práctica acarrea dificultades ya que los criterios de compra de material obedecen principalmente a motivos económicos más que de diseño [Ol70]. El organigrama podríamos expresarlo como sigue:

Proceso de tratamiento del material en la fábrica



Centrados en el objetivo de minimizar el stock de residuos en el almacén, parece razonable crear herramientas que nos calculen las dimensiones de los materiales (en este caso las barras) necesarios para obtener un aprovechamiento máximo. Es importante que esta tarea sea ejecutada rápidamente y de forma previa a la compra de material; de este modo el departamento de compras tiene la posibilidad de negociar mejores precios e incluso la fabricación de piezas *ad hoc* para el proyecto en cuestión. Estas técnicas son utilizadas actualmente por la compañía ISI como complemento al diseño final de los proyectos y marcan una nueva organización, en la que los diseñadores de las torres participan activamente en la mejor solución de clasificación para corte del material y no deben estar forzados al aprovechamiento del stock residual. La organización quedará estructurada de la siguiente forma :



Aquí podemos observar que se trata de un proceso lineal (no en paralelo, como el anterior), donde se ha suprimido el almacén y donde, tras la financiación, el proyecto

pasa directamente a la ejecución, sin limitaciones ni nuevos estudios para adaptarse a las prioridades que pudiesen existir.

6.2 Modelización

En esta sección introducimos las notaciones que usaremos para poder enunciar matemáticamente el problema de análisis de residuo mínimo.

Partimos de barras de distintas longitudes y secciones. Llamaremos χ al conjunto global de todas las barras del proyecto. Consideraremos χ clasificado por secciones y notaremos por χ_i al conjunto de las barras de sección i . Cada barra del proyecto la consideraremos una **unidad de trabajo**. Análogamente, notaremos por γ al conjunto global de barras que se compran para la manufacturación y por γ_i al conjunto de todas las que tienen sección i . Aquí, a los elementos de estos conjuntos los denominaremos **unidades de mercado**.

Para poder comenzar el estudio suponemos que todas las barras consideradas tienen la misma sección. Esto no trae consigo pérdida de generalidad ya que las conclusiones para un χ_i e γ_i son válidas para cualquier otro valor de i (simplemente nos limitamos a cada clase en χ y en γ correspondiente). Por tanto usaremos la notación:

χ es el conjunto de n unidades de trabajo: $\chi = \{x_1, x_2, \dots, x_n\}$

γ es el conjunto de m unidades de mercado: $\gamma = \{y_1, y_2, \dots, y_m\}$

En principio n lo conocemos (evidentemente, sabemos cuántas barras requiere nuestro proyecto) y m siempre podrá ser tan grande como nos haga falta ya que suponemos, en principio, ilimitada nuestra capacidad de compra (aunque, claro está, lo que deseamos es minimizarlo).

6.2.1 Enunciado y restricciones

Dados los conjuntos χ e γ donde cada elemento representa la longitud de una barra de los respectivos χ e γ y a los que notaremos por x e y subíndizadas, nuestro problema es:

Lograr que los restos sean mínimos, es decir, que quepan el máximo de unidades de trabajo en cada unidad de mercado. Por tanto hemos de:

$$\begin{aligned} \text{Minimizar :} \quad & y_1 - (x_{11} + x_{12} + \dots + x_{1n_1}) \\ & y_2 - (x_{21} + x_{22} + \dots + x_{2n_2}) \\ & \dots \\ & y_m - (x_{m1} + x_{m2} + \dots + x_{mn_m}) \end{aligned}$$

donde los x_{ij} pueden ser iguales o distintos entre ellos, bajo las hipótesis:

Primera: No excederse en la longitud de la unidad de mercado

$$\begin{aligned} y_1 &\geq x_{11} + x_{12} + \dots + x_{1n_1} \\ y_2 &\geq x_{21} + x_{22} + \dots + x_{2n_2} \\ &\dots \\ y_m &\geq x_{m1} + x_{m2} + \dots + x_{mn_m} \end{aligned}$$

Segunda: Clasificación de todas y cada una de las unidades de trabajo

$$\sum_{i=1}^m n_i = n$$

es decir que

$$\chi = \{x_{11}, \dots, x_{1n_1}, x_{21}, \dots, x_{2n_2}, \dots, x_{m1}, \dots, x_{mn_m}\}$$

Con todo esto obtenemos la base [Ac67] para adentrarnos en nuestro problema. Para empezar hemos de fijarnos en el número de posibles asignaciones de n unidades de

trabajo a m unidades de mercado. Llamemos N a dicha cantidad. Dado que cada x_i está asociado a un y_j , el máximo número de asignaciones será $VR_m^n = m^n$

Por otra parte, si sabemos que $m > 1$ (si una única unidad de mercado bastase para obtener todas las unidades de trabajo no tendríamos problema que resolver) tendremos al menos que estudiar 2^n casos. Evidentemente, de todos ellos, sólo habrá una pequeña cantidad que satisfagan las condiciones primera y segunda, pero de antemano no podemos saber cuáles de ellos serán. Por tanto, para dar solución a este problema analizaremos previamente aquellos que poseen cierta similitud con el propuesto y observaremos cómo han sido resueltos [Ma86].

6.3 Método Greedy

El método *greedy* es una técnica muy usada en la actualidad por su eficiencia, es decir, su bajo grado de complejidad. Trata de dar solución a aquellos problemas en los que tenemos n entradas y se requiere encontrar una clasificación que satisfaga ciertas restricciones. Cada asignación que cumpla dichas restricciones se denomina solución posible. Cuando encontramos una solución posible que maximice o minimice (en función del problema) a una función denominada función objetivo, decimos que dicha solución es una solución óptima. Normalmente hay un camino obvio para llegar a una solución posible, pero no necesariamente lo hay para llegar a una solución óptima.

Este método sugiere que, con un poco de imaginación, siempre se puede encontrar un algoritmo (al que llamaremos algoritmo greedy) que trabaje por etapas, considerando una entrada cada vez. En cada etapa, se toma una decisión respecto de si esa entrada mejora o no la solución, es decir, si optimiza o no el resultado hacia el que se evoluciona. Para ello se considera que las entradas se hacen en un orden determinado por algún procedimiento de selección. Si la inclusión de la siguiente entrada en la solución que vamos construyendo parcialmente resultara una solución no posible o no factible, dicha ordenación se reestructuraría. El procedimiento de selección en sí ha de estar basado en alguna medida de optimización. Esta medida puede o no ser la función objetivo. Cada medida considerada puede engendrar, sin embargo, un nuevo algoritmo que nos lleve a soluciones posibles e incluso óptimas.

Para comenzar nuestro estudio vamos a ver algunos ejemplos prácticos de aplicación del método greedy [CK76] cuyos resultados nos van a ir acercando a un caso particular de éste, el cual es el que más se acerca a nuestro problema, llamado problema de la mochila (*knapsack problem*) [MT90, CK01].

6.3.1 Almacenamiento óptimo en discos

Tenemos n programas que pueden ser almacenados en un disco, el cual decimos que tiene un tamaño B (dado en bytes). Cada programa tiene asociado un tamaño propio b_i con $i \in \{1, \dots, n\}$

Claramente, todos los programas podrán ser almacenados en el disco si y sólo si la suma de todos los tamaños es menor que B .

Nosotros asumiremos que siempre que un programa vaya a ser leído del disco, el ordenador comienza a leer en la posición inicial. Supongamos entonces que los programas son almacenados en el orden $I = \{i_1, i_2, \dots, i_n\}$ (esto es simplemente una de las $n!$ permutaciones de orden posibles entre los n programas). Entonces el tiempo t_j necesario para recuperar el programa i_j es proporcional a:

$$\sum_{k=1}^j b_k$$

Si todos los programas son recuperados con la misma frecuencia entonces el tiempo medio de recuperación (TMR) es:

$$\frac{1}{n} \sum_{j=1}^n t_k$$

En este problema nosotros necesitamos encontrar la permutación para n programas de forma que cuando estén almacenados en ese orden, el TMR sea mínimo. Esto equivale a minimizar:

$$D(I) = \sum_{j=1}^n \sum_{k=1}^j b_k$$

Ejemplo: Sea $n=3$ y sea $I = \{b_1, b_2, b_3\} = \{5, 10, 3\}$ (es decir, el primer programa tiene un tamaño 5, el segundo 10 y el tercero 3). Existen $n!=3!=6$ formas posibles de introducirlos en el disco.

Si, por ejemplo, los introducimos de la forma 5,3,10, el primer programa tardará en ser leído $t(5)$ (en función del tamaño de dicho programa). El segundo lo será en $t(5+3)$ (en función de los dos primeros) y el tercero en $t(5+3+10)$.

Todas las ordenaciones I posibles, con sus respectivos $D(I)$ serán:

Permutación	Ordenación I	$D(I)$
1,2,3	5,10,3	$5 + (5 + 10) + (5 + 10 + 3) = 38$
1,3,2	5,3,10	$5 + (5 + 3) + (5 + 3 + 10) = 31$
2,1,3	10,5,3	$10 + (10 + 5) + (10 + 5 + 3) = 43$
2,3,1	10,3,5	$10 + (10 + 3) + (10 + 3 + 5) = 41$
3,1,2	3,5,10	$3 + (3 + 5) + (3 + 5 + 10) = 29$
3,2,1	3,10,5	$3 + (3 + 10) + (3 + 10 + 5) = 34$

Por tanto la ordenación óptima será 3,5,10.

Una aproximación greedy para construir la permutación requerida debe elegir cada programa basándose en alguna medida de optimización. Una medida posible sería el valor de D de la permutación construida así. Cada programa sería almacenado en función de que minimizara el incremento de D . Si nosotros construimos la permutación i_1, i_2, \dots, i_r , al añadir el programa j dará la permutación $i_1, i_2, \dots, i_{r+1} = j$.

Esto incrementará el valor de D a $\sum_{k=1}^r b_{i_k} + b_j$. Como $\sum_{k=1}^r b_{i_k}$ es fijo e independiente de j podemos observar trivialmente que el incremento de D es mínimo si el siguiente programa elegido es aquel que tenga menor tamaño entre los que restan por almacenar.

Con este razonamiento llegamos a que, por el método greedy, los programas sólo requieren estar ordenados en orden creciente de tamaños. Esta ordenación puede ser realizada con anterioridad en un tiempo $\mathcal{O}(n \cdot \lg(n))$ usando el algoritmo de

ordenación quick-sort [BB88]. Demostraremos que el TMR es mínimo cuando los programas son almacenados en este orden.

Teorema : Si $b_1 \leq b_2 \leq \dots \leq b_n$ entonces el orden $i_j = j$ para $j = 1, 2, \dots, n$ minimiza:

$$D(I) = \sum_{j=1}^n \sum_{k=1}^j b_{i_k}$$

entre todas las posibles permutaciones de i_j

Demostración : Sea $I = i_1, i_2, \dots, i_n$ una permutación del conjunto de índices $\{1, 2, \dots, n\}$

$$D(I) = \sum_{j=1}^n \sum_{k=1}^j b_{i_k} = \sum_{j=1}^n (n-j+1) \cdot b_{i_j} \quad (\text{Véase el ejemplo anterior})$$

Si existe a y b tales que $a < b$ y $b_{i_a} > b_{i_b}$ entonces intercambiamos i_a e i_b y obtenemos la permutación I' con:

$$D(I') = \sum_{j=1; j \neq a; j \neq b}^n (n-j+1) \cdot b_{i_j} + (n-a+1) \cdot b_{i_b} + (n-b+1) \cdot b_{i_a}$$

Restando ambas expresiones:

$$D(I) - D(I') = (n-a+1) \cdot (b_{i_a} - b_{i_b}) + (n-b+1) \cdot (b_{i_b} - b_{i_a}) = (b-a) \cdot (b_{i_b} - b_{i_a}) > 0$$

se deduce que I' mejora a I .

Por tanto ninguna permutación de orden no creciente de tamaños puede minimizar D . Es fácil ver que toda permutación creciente tiene el mismo valor de D y por tanto el orden definido por $i_j = j$ para todo $j = 1, 2, \dots, n$ minimiza D

c.q.d.

Este problema de almacenamiento en discos podemos extenderlo al caso en que tengamos más de un disco. Supongamos que contamos con $m > 1$ discos iguales, D_1, D_2, \dots, D_m , entonces los programas podrán estar almacenados en cualquiera de ellos. Para cada disco existirá una permutación de almacenamiento. Si I_j es la permutación para el subconjunto de programas del disco j entonces $D(I_j)$ se define de la misma manera que antes. El tiempo total de lectura será $TD = \sum_{j=1}^m D(I_j)$. El objetivo es cargar los programas minimizando TD .

La generalización obvia de la solución para un solo disco debe ser:

Tomemos los n programas en orden creciente de b_i . El programa en curso será colocado en aquel disco que minimice el incremento de TD . Este será el que menor capacidad tenga ocupada. Si hay más de uno con esta propiedad, entonces usaremos el de menor índice (por fijar alguna norma, ya que esto no es necesario). Si los trabajos son ordenados inicialmente como $b_1 \leq b_2 \leq \dots \leq b_n$ entonces los primeros m programas serán asignados a los discos D_1, D_2, \dots, D_m , respectivamente. Los siguientes m programas serán asignados nuevamente a D_1, D_2, \dots, D_m y así sucesivamente hasta finalizar los n programas. La regla general es que el programa i será colocado en el disco $D_{i(\text{mod}(m))+1}$. Además, en cualquiera de ellos los programas están cargados en forma creciente de tamaños.

Nótese que este algoritmo no es completo puesto que no encuentra la solución óptima (de hecho no encuentra ninguna solución) al siguiente problema:

Consideremos dos discos de tamaño 5, y cinco programas, cuatro de tamaño 1 y uno de tamaño 4. Obviamente existe una solución que es poner 1+1+1 y 1+4. Sin embargo el algoritmo anterior colocaría 1+1 y 1+1 no pudiendo almacenar el programa de tamaño 4. Esta limitación desaparece considerando los discos con capacidad infinita.

Supongamos que los discos no poseen limitación de carga. No queremos decir que esta hipótesis vaya a ser necesaria en nuestros resultados finales, sólo la tenemos en cuenta para comprobar la tendencia que siguen los algoritmos óptimos y obtener así alguna pista que nos guíe en la resolución de los problemas de residuo mínimo.

El algoritmo que damos a continuación, implementado en lenguaje C, presupone que los programas están ordenados en orden creciente de tamaños, tiene un tiempo de computación $\mathcal{O}(n)$ y no necesita conocer los tamaños actuales de los programas. El siguiente teorema probará que dicho algoritmo es de resultado óptimo.

Algoritmo de almacenamiento de programas:

```

STORE (n,m)                /* n=n° de programas; m=n° de discos*/
int m,n;                   /* los definimos como enteros */
    { int i,j;              /* iniciamos los contadores i,j (enteros) */
      j=0;                  /* contador de disco */
      for(i=1;i<=n;i++)    /* contador de programas */
      {
        printf("asigna el programa %d a la permutación %d",i,j);
        j = j (mod(m))+1; /* numero de disco donde se sitúa */
      }                    /* fin de ciclo */
    }                      /* fin de algoritmo */
    
```

Teorema : Si $b_1 \leq b_2 \leq \dots \leq b_n$ el algoritmo anterior genera un modelo de almacenamiento óptimo en m discos, en caso de que los discos no tengan limitación de carga.

Demostración : Dado un modelo cualquiera de almacenamiento para m discos, sea r_i el número de programas siguientes al programa i (incluido i) en su disco correspondiente. Entonces, el tiempo total de lectura TD vendrá dado por:

$$TD = \sum_{i=1}^n r_i b_i$$

En alguno de estos modelos, para cualquier n puede haber a lo sumo m programas para los cuales $r_i = j$ (nunca más de m ya que son los discos que hay). Del teorema anterior se sigue que TD es mínimo si los m programas de mayor tamaño tienen $r_i = 1$, los siguientes m tendrán $r_i = 2$ y así sucesivamente.

Cuando los programas están ordenados por tamaños, es decir, $b_1 \leq b_2 \leq \dots \leq b_n$, entonces este criterio de minimización es satisfactorio si $r_i = \frac{n-i+1}{m}$.

Es fácil ver que el algoritmo anterior es un modelo de almacenamiento con este resultado.

c.q.d.

Esta demostración nos confirma que hay muchas posibles ordenaciones distintas que minimizan TD . Si calculamos $r_i = \frac{n-i+1}{m}$ para cada programa i , entonces, siempre que todos los programas con el mismo r_i sean almacenados en distintos discos y tengan $r_i - 1$ programas siguiéndoles, el TD es el mismo.

Si n es múltiplo de m entonces hay al menos $(m!)^{n/m}$ modelos de ordenación que minimizan TD . El algoritmo anterior tan sólo calcula uno de ellos.

6.3.2 Problema de la mochila.

Veamos ahora cómo aplicar el método greedy para resolver un problema algo más complejo, llamado *Problema de la Mochila* [MT90]. Se trata de uno de los problemas más conocidos y estudiados debido a la gran cantidad de aplicaciones en diversas disciplinas (economía, matemáticas, informática, etc.). Existen multitud de enfoques del problema que estudian tanto su especificación como estrategias de implementación eficiente [VV00] [CK01].

Sean n objetos y una única mochila. El objeto i tiene un peso w_i y un precio p_i , y la mochila tiene una capacidad de carga W . Si una fracción $x_i, 0 \leq x_i \leq 1$, del objeto i es colocada en la mochila, entonces se consigue una ganancia $p_i x_i$.

El objetivo es obtener un llenado de la mochila que maximice la ganancia. Dado que la capacidad de la mochila es W , necesitaremos que el peso total de todos los objetos no supere W .

Formalmente el problema se define como:

$$\begin{aligned} &\text{maximizar } \sum_{i=1}^n p_i x_i \\ &\text{sabiendo que } \sum_{i=1}^n w_i x_i \leq W \end{aligned}$$

$$\text{con } 0 \leq x_i \leq 1, p_i > 0, w_i > 0, 1 \leq i \leq n$$

Una solución posible (o llenado) es una n -upla (x_1, x_2, \dots, x_n) que satisfaga las condiciones segunda y tercera anteriores. Una solución óptima será aquella solución posible que satisfaga la primera condición.

Si todos los pesos suman menos de W entonces la solución trivial y óptima será la $x_i = 1$ para todo i . Por tanto asumimos que los pesos sumados exceden de W . Bajo esta hipótesis, no todos los x_i pueden valer 1. Una segunda observación es que toda solución óptima llenará la mochila totalmente. Esto es cierto porque siempre se pueden incrementar algunas fracciones para que la contribución de esos elementos

complete el peso W . Son sugeribles muchas estrategias greedy para una solución posible cuya suma complete el peso W :

- Podemos tratar de llenar la mochila con el objeto de mayor valor o premio. Si el último objeto, bajo esta consideración, no se adapta entonces ponemos una fracción de él para llenar la mochila. De esta manera, cada vez que incluimos un objeto (siempre que no lleguemos al final) en la mochila, obtenemos el mayor incremento posible en el valor de la ganancia. Pero si sólo incluimos una fracción del último objeto, sería posible aumentar dicho incremento cogiendo otra fracción de algún otro (p.ej $p_i = 4$, $w_i = 4$ y $p_{i+1} = 3$, $w_{i+1} = 2$. Evidentemente conviene $i+1$). Por tanto esta estrategia nos da soluciones posibles pero *subóptimas* ya que la mayor ganancia puede engendrar también mayor peso.
- Si ahora lo hacemos en orden decreciente de pesos tampoco llegaremos normalmente a soluciones óptimas.
- Por tanto necesitamos un algoritmo que considere paralelamente el incremento de ganancias y el decremento de pesos.

En cada paso incluiremos por tanto el objeto que tenga máxima ganancia por unidad de peso usada. Esto significa que los elementos se ordenan en orden decreciente de razón $\frac{p_i}{w_i}$. Este algoritmo llamado *GREEDY-KNAPSACK* obtiene una solución correspondiente para esta estrategia (en el siguiente teorema se muestra cómo dicha solución es óptima).

Las tres estrategias (despreciando el tiempo de ordenación) usan un tiempo $\mathcal{O}(n)$.

Teorema : Si $\frac{p_1}{w_1} > \frac{p_2}{w_2} > \dots > \frac{p_n}{w_n}$ entonces el algoritmo *GREEDY-KNAPSACK*

genera una solución óptima del problema de la mochila de la forma:

$$\left\{ \begin{array}{l} x_i = 1 \text{ si } 1 < i < j \\ x_i = 0 \text{ si } j < i < n \\ x_j = \hat{w} / w_j \text{ donde } \hat{w} = w - \sum_{k=1}^{j-1} w_k \end{array} \right.$$

Demostración : Sea $X = (x_1, x_2, \dots, x_n)$ la solución generada por el algoritmo *GREEDY-KNAPSACK*. Si todos los x_i son iguales a 1 entonces la solución es, evidentemente, óptima.

Para ser X óptima debe ser completa, en el sentido que $\sum_{i=1}^n x_i w_i = W$.

Sea $Y = (y_1, y_2, \dots, y_n)$ la solución óptima del problema. Supongamos, por reducción al absurdo, que $y_k < 1$ para algún $k < j$. Entonces debe existir algún $y_q > x_q$ para al menos un $q \geq j$.

Dado un $\varepsilon > 0$ suficientemente pequeño, podemos incrementar el valor de y_k en ε y reducir el de y_q en $\varepsilon \frac{w_k}{w_q}$, lo cual aumenta el precio de esta nueva solución en

$\varepsilon \left(p_k - p_q \frac{w_k}{w_q} \right)$ (que es mayor que cero ya que $\frac{p_k}{w_k} > \frac{p_q}{w_q}$) lo cual es una contradicción

con el hecho de ser óptima. De la misma forma se puede comprobar que no es posible encontrar un $y_k > 0$ para $k > j$. De aquí que $x_j = \hat{w} / w_j$ para alcanzar la completitud y que $Y = X$.

6.4 Algoritmos para el Problema del Residuo Mínimo

Observando las aproximaciones anteriores, podemos plantearnos: ¿Es factible un procedimiento que nos dé la solución óptima a nuestro problema de ordenación de las unidades de trabajo sobre las unidades de mercado? Evidentemente, no podemos estudiar todas las soluciones posibles y extraer la óptima (o más de una si las hubiese), ya que el coste en memoria y tiempo requeridos nos hace desechar tal idea. Por tanto requerimos un procedimiento que nos aproxime a una solución en la que los residuos se minimicen lo más posible, sin generar todas las ordenaciones exhaustivamente. Antes de pasar a exponer los algoritmos vamos a definir los siguientes conceptos que serán de utilidad.

6.4.1 Definiciones previas

Definición : Decimos que una unidad o parte de una unidad de mercado tiene *posibilidad de llenado* cuando la diferencia entre su longitud y la suma de las longitudes de las unidades de mercado que tiene asignadas hasta ese momento, es mayor que la menor de las unidades de trabajo (es decir, esa barra es aún aprovechable).

Cuando una unidad o parte de una unidad de mercado deja de cumplir tal propiedad, decimos que dicho elemento es *manipulable*. (Es decir, esa barra puede ya pasar a la siguiente sección de la fábrica para ser punzonada, taladrada, cortada, etcétera).

Definición : Decimos que un algoritmo para el problema de mínimo residuo es de *combinación múltiple* cuando mantiene en memoria todas las unidades de mercado que tienen posibilidad de llenado.

Definición : Decimos que un algoritmo para el problema de mínimo residuo es de *llenado progresivo* cuando va completando cada unidad de mercado, hasta hacerla manipulable, para pasar a la siguiente, sin necesidad de mantenerla en memoria ni de almacenar las unidades de trabajo.

6.4.2 Algoritmos para la resolución del problema de mínimo residuo

Exponemos en esta sección, como aportación de esta tesis, diversos algoritmos de clasificación para la resolución del problema de mínimo retal [DG89], junto con su implementación [Br85], la representación gráfica de su comportamiento y el diagrama de ocupación de memoria en cualquier tipo de computadora [Me91]. Usaremos las variables M como memoria y n como número de unidades de trabajo clasificadas.

Consideraciones:

- 1) En todos los casos el conjunto γ (de unidades de mercado) se encuentra ordenado de forma creciente, $y_1 \leq y_2 \leq \dots \leq y_m$, ya que en toda fábrica

siempre interesa, en caso de almacenar, que los trozos sean nuevos o de la mayor longitud posible [KI75].

- 2) Al hablar de porcentajes de residuos nos referiremos a ensayos realizados con diversos proyectos en los que el número de unidades de trabajo a clasificar variaba entre 1.000 y 10.000, y en los que se calculó el desperdicio medio entre todos los ensayos (evidentemente, dentro de cada sección siempre se eligió la unidad de mercado que provocaba la mejor clasificación).

6.4.2.1 Algoritmo de ordenación decreciente

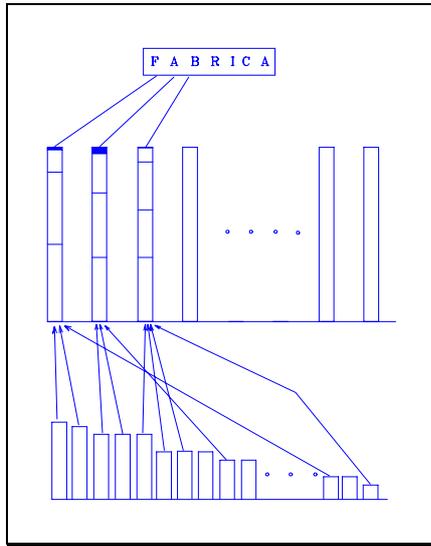


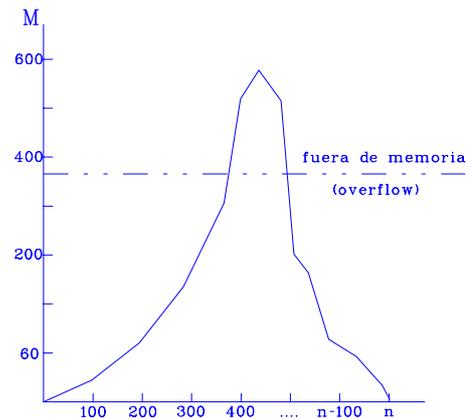
Ilustración 6-1. Algoritmo de Ordenación Decreciente

Ordenamos X de modo decreciente, y escribamos $x_1 \geq x_2 \geq \dots \geq x_n$. Comenzamos introduciendo los x_i , con $i = \{1, 2, \dots, n\}$ de la siguiente forma:

Tomamos x_1 . Comprobamos desde y_1 a y_m y escogemos el primer y_j tal que $x_1 \leq y_j$. Sustituimos y_j por $y_j - x_1$ y reordenamos nuevamente los elementos de Y . Evidentemente, en el caso que un y_j llegue a ser menor que x_n (el menor de los elementos de X a ordenar)

dejamos de considerarlo ya que es un residuo que no se puede aprovechar más, es decir, y_j es manipulable. Tomamos x_2 y realizamos la misma operación, y así sucesivamente.

Con ello conseguimos una clasificación suficientemente buena del material ya que se han obtenido residuos menores del 5%.



Algoritmo_ordenacion_decreciente(X,n,Y,m)

```

int X[],n; /* Unidades de trabajo y cantidad */
int Y[],m; /* Unidades de mercado y cantidad */
{ int i,j; /* Subíndices a usar */
  reordena(X,n,decreciente); /* Función que reordena*/
  reordena(Y,m,creciente);
  for(i=1;i<=n;i++) /* Bucle de unid. de trabajo */
    for(j=1;j<=m;j++) /* Bucle de unid. de mercado */
      if(Y[j]>=X[i]) /* Si la unid. de trabajo cabe entonces la
                    /* introduce, almacenando la unid. de mercado si
                    /* se llena */
        { incluye(X[i],Y[j]);
          reordena(Y,m,creciente);
          if(Y[j]<X[n]) { almacena(Y,j); break; }
        } /* Fin del if(Y[j]>=X[i]) */
} /* Fin de la función */

```

6.4.2.2 Algoritmo de ordenación creciente

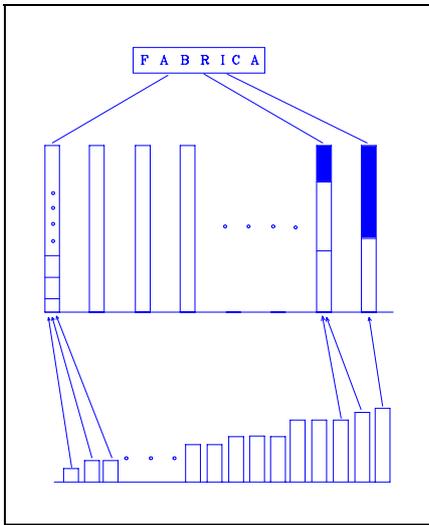
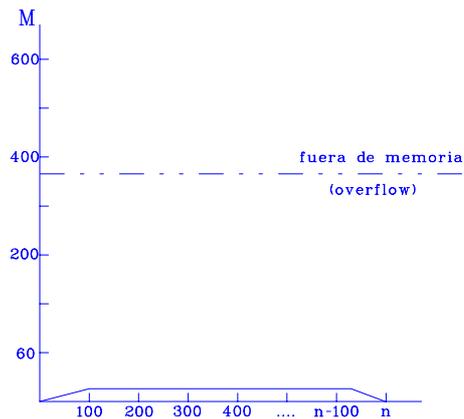


Ilustración 6-2. Algoritmo de Ordenación Creciente

Ordenamos X de forma creciente y tal que $x_1 \leq x_2 \leq \dots \leq x_n$. Comenzamos introduciendo secuencialmente x_1, x_2, \dots, x_n . Como podemos imaginar, este método apura bastante bien las primeras unidades de mercado, pero cada vez puede aprovechar menos las sucesivas unidades de mercado ya que los x_i van siendo mayores.

Además, este método realmente no combina absolutamente nada ya que van entrando progresivamente y clasificándose. El residuo provocado es excesivamente grande (superior al 20%), por tanto no es muy conveniente.



Algoritmo_ordenacion_creciente(X,n,Y,m)

```

int X[],n; /* Unidades de trabajo y cantidad */
int Y[],m; /* Unidades de mercado y cantidad */
{ int i,unidad; /* Subíndices a usar */
reordena(X,n,creciente); /* Función que reordena*/
reordena(Y,m,creciente);
unidad=1;
for(i=1;i<=n;i++) /* Bucle de unid. de trabajo */
{
incluye(X[i],Y[unidad]); /* Incluye la unidad de trabajo */
if(Y[unidad]<X[i+1]) /* Si la siguiente no va a entrar */
{ almacena(Y,unidad); unidad++; } /*Guarda y coge otra */
}
} /* Fin del FOR */
} /* Fin de la función */
    
```

6.4.2.3 Algoritmo de ordenación alternada

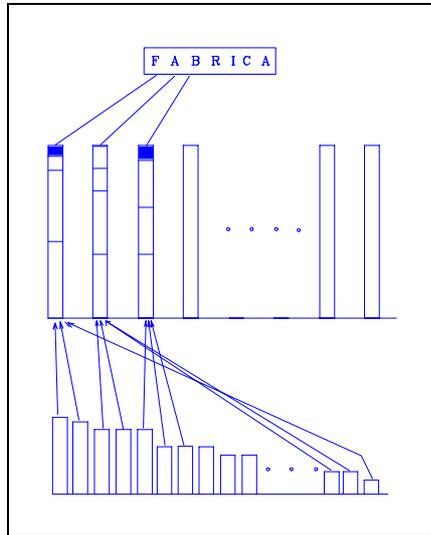


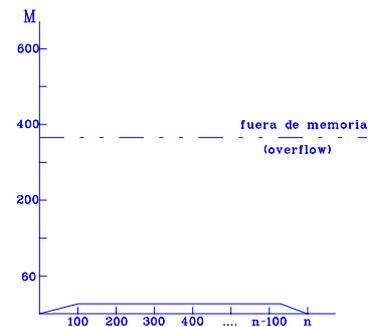
Ilustración 6-3. Algoritmo de Ordenación Alternada

En este caso ordenamos de forma decreciente el conjunto X .

Vamos introduciendo secuencialmente x_1, x_2, \dots hasta que ya no entren en y_1 . En este momento comenzamos a incluir en y_1 los x_i pero desde el final (los menores), x_n, x_{n-1}, \dots mientras sea posible. Con ello completamos y_1 y no hemos de conservarla en memoria ya que pasa directamente al almacén.

Este método mejora bastante

al anterior ya que aprovecha un mayor número de unidades de mercado y en él realmente se produce una combinación de unidades de trabajo *largas* y *cortas*. El residuo provocado por dicho algoritmo es menor del 15%, y empezamos a considerarlo como bueno (en algunos casos iguala al algoritmo de ordenación decreciente siendo, evidentemente, mucho más rápido).



Algoritmo_ordenacion_alternada(X,n,Y,m)

```

int X[],n; /* Unidades de trabajo y cantidad */
int Y[],m; /* Unidades de mercado y cantidad */
{ int i,j,unidad,ultima; /* Subíndices a usar */
reordena(X,n,decreciente); /* Función que reordena*/
reordena(Y,m,creciente);
unidad=1; ultima=n+1;
for(i=1;i<=n;i++) /* Bucle de unid. de trabajo */
{
if(Y[unidad]<X[i]) /* Si no entran las mayores */
for(j=ultima;j<=1;j--) /* Empiezo por las menores */
if(Y[unidad]>=X[j]) /* Si la menor cabe, la pongo */
{ ultima--; incluye(X[ultima],Y[unidad]); }
else /* Si no, ya puedo almacenar */
{ almacena(Y,unidad); unidad++; break; }
else /* Pero si entra la mayor */
{ incluye(X[i],Y[unidad]); }
} /* Fin del FOR */
} /* Fin de la función */

```

6.4.2.4 Algoritmo de ordenación mayorada

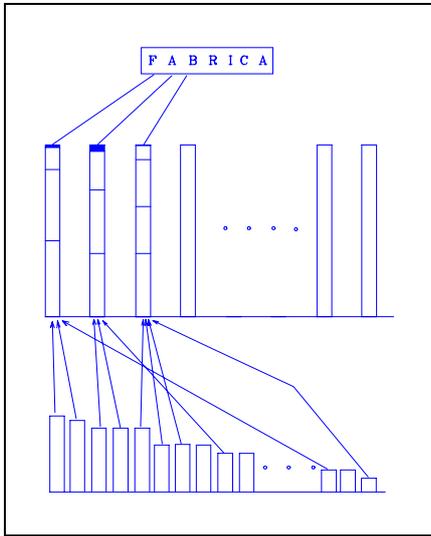
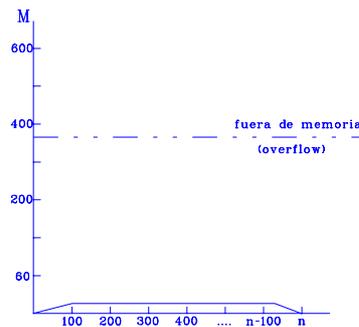


Ilustración 6-4. Algoritmo de Ordenación Mayorada

Nuevamente ordenamos de forma decreciente el conjunto X .

Volvemos a introducir secuencialmente x_1, x_2, \dots mientras que entren en y_1 . Cuando ya no sea posible, buscamos en X el mayor x_i que pueda entrar en y_1 , y así sucesivamente hasta que hagamos que y_1 sea manipulable. Entonces tomaremos y_2 y haremos el mismo proceso. El algoritmo acaba cuando hayamos clasificado el conjunto X , evidentemente.

Como podemos apreciar, es un algoritmo que no usa la memoria para almacenar datos, lo cual lo convierte en un algoritmo bastante rápido. Por otra parte su resultado es suficientemente bueno (menor del 15%), por tanto es un buen candidato para la resolución de nuestro problema.



Algoritmo_ordenacion_mayorada(X,n,Y,m)

```

int X[],n; /* Unidades de trabajo y cantidad */
int Y[],m; /* Unidades de mercado y cantidad */
{ int i,unidad,quedan; /* Subíndices a usar */
  reordena(X,n,decreciente); /* Función que reordena*/
  reordena(Y,m,creciente);
  unidad=1; quedan=n;
  while(quedan > 0) /* Mientras haya unid. de trabajo */
    for(i=1;i<=n;i++) /* Bucle de unid. de trabajo */
      {
        if(Y[unidad]<X[n]) /* Si no entran la menor */
          { almacena(Y,unidad); unidad++; }
        if(Y[unidad]>=X[i]) /* Pero si entra la actual */
          { incluye(X[i],Y[unidad]); quedan--; }
      }
    /* Fin del FOR */
  }
  /* Fin de la función */

```

6.4.2.5 Proceso de llenado progresivo óptimo

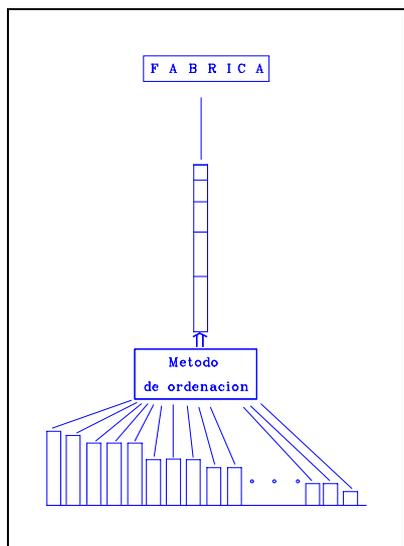


Ilustración 6-5. Proceso de Llenado Progresivo Óptimo

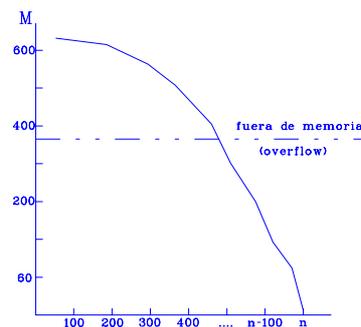
Dados x_1, x_2, \dots, x_n tenemos la tabla:

	x_1	x_2	x_3	x_n
y_1					

Consideramos todas las soluciones posibles de llenado de y_1 y busco la óptima, o consideramos a y_1 el destino y a x_1, x_2, \dots, x_n los orígenes, aplicando seguidamente el método del transporte para conseguir el reparto óptimo.

De esta manera obtenemos el llenado óptimo de y_1 , lo enviamos al almacén y análogamente vamos haciendo lo mismo con y_2, \dots, y_m mientras sigan quedando unidades de trabajo (mientras $X \neq \emptyset$)

Como era de esperar, este es el método que consigue la mejor solución, pero es de muy lenta ejecución (es de orden exponencial) y necesita máquinas con una capacidad de memoria muy grande.



Proceso_de_llenado_progresivo_optimo(X,n,Y,m)

```

int X[],n; /* Unidades de trabajo y cantidad */
int Y[],m; /* Unidades de mercado y cantidad */
{ int unidad,unidades_de_trabajo; /* Subindices a usar */
  reordena(X,n,decreciente); /* Función que reordena*/
  reordena(Y,m,creciente);
  unidad=1; unidades_de_trabajo=n;
  while(unidades_de_trabajo>0) /* Mientras haya unidades */
  { /* de trabajo, clasifica */
    llena_optimamente(Y[unidad],X); /* Proceso de clasificación */
    almacena(Y,unidad); /* Guarda la unidad llena */
    unidad++;
  } /* Fin del FOR */
} /* Fin de la función */

```

6.4.3 Clasificación de los algoritmos

Según los conceptos previos que hemos definido podemos clasificar estos algoritmos de la siguiente forma :

Tipo I . Algoritmos de combinación múltiple.

1. Algoritmo de ordenación decreciente.
2. Proceso de llenado progresivo óptimo.

Estos algoritmos tienen la propiedad de efectuar ordenaciones de residuo muy bajo, acercándose e incluso alcanzando la solución óptima. El problema es que utilizan una gran capacidad de memoria provocando gran lentitud en los procesos de clasificación, hasta el punto de no hacerlos rentables para la industria.

No obstante, el primero es un algoritmo más sencillo de programar que no necesita en memoria las unidades de trabajo (pero sí todas las unidades de mercado con posibilidad de llenado).

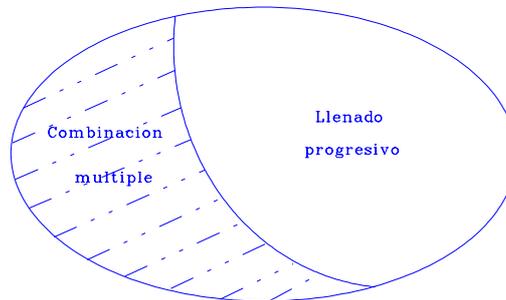
El segundo es de más lenta ejecución y necesita en memoria sólo las unidades de trabajo.

Tipo II . Algoritmos de llenado progresivo.

3. Algoritmo de ordenación creciente.
4. Algoritmo de ordenación alternada.
5. Algoritmo de ordenación mayorada.

Estos algoritmos son muy rápidos y no tienen necesidad de usar memoria. Su problema radica en el exceso de residuo que provocan, lo que los hace menos fiables. Todos son de fácil implementación, aunque el segundo y el tercero mejoran bastante la solución ofrecida por el primero de los métodos.

Por tanto, y ante este panorama, observamos que tenemos dividido el espacio de los algoritmos para la resolución de nuestro problema en dos:



¿Serán dichos espacios de intersección vacía?

Esta pregunta viene provocada porque la solución podría estar en encontrar un algoritmo que dando una solución suficientemente buena (menor del 10% en grandes lotes), es decir, del tipo I, no usara mucha memoria y fuera suficientemente rápido, o sea, del tipo II. Con esta intención comenzamos a investigar y llegamos a un nuevo concepto:

6.4.4 Mixtura algorítmica y aplicaciones

Definición : Llamamos *mixtura algorítmica* al hecho de que ante un conjunto de diversos algoritmos para la resolución de un mismo problema, utilizemos uno u otro en función de la situación encontrada.

Bajo este concepto definimos una nueva clase de algoritmos:

Definición : Decimos que un algoritmo es *mixto* cuando utiliza técnicas de combinación múltiple y de llenado progresivo a la vez.

Consideremos los siguientes límites:

M : Número máximo de unidades de mercado que el ordenador es capaz de mantener en memoria.

M : Número de unidades mercado en memoria, a partir del cual la velocidad de ejecución decrece notablemente (a elección del usuario). Evidentemente $\mathbf{M} \geq M$

N : Número de unidades mercado en memoria, por debajo del cual el tiempo de ejecución es prácticamente nulo (acceso casi instantáneo, dependiendo de la computadora).

6.4.4.1 Algoritmo mixto decreciente - alternado o algoritmo en banda

Utilizaremos simultáneamente los algoritmos decreciente y alternado. Partimos de una ordenación de forma decreciente de las unidades de trabajo y comenzamos el proceso utilizando el algoritmo de ordenación decreciente. Cuando el número de unidades de mercado en memoria sea igual a M , aplicamos entonces el algoritmo de ordenación alternada, que introducirá las unidades de trabajo por el final x_n, x_{n-1}, \dots , con lo que irá ocupando los huecos existentes y transformando las unidades de mercado en unidades manipulables. Este segundo algoritmo se aplica hasta que el número de unidades de mercado en memoria es igual a N , donde nuevamente se aplica el algoritmo primero, y así sucesivamente.

```

Algoritmo_en_banda(X,n,Y,m)
int X[],n;                               /* Unidades de trabajo y cantidad */
int Y[],m;                               /* Unidades de mercado y cantidad */
{ int unidad,unidades_de_trabajo;       /* Subíndices a usar */
  reordena(X,n,decreciente);             /* Función que reordena*/
  reordena(Y,m,creciente);
  unidad=1; unidades_de_mercado=0;
  for(i=1;l<=n;i++)                       /* Bucle de unidades de trabajo */
  {
    if(unidad_de_mercado < M)             /* Mientras no llegue el limite */
      algoritmo_de_ordenacion_decreciente(X,n,Y,m)
    else                                   /* si llega, comienza a almacenar */
      while(unidades_de_mercado>N)
        algoritmo_de_ordenacion_alternada(X,n,Y,m)
  }                                       /* Fin del FOR */
}                                       /* Fin de la función */

```

6.4.4.2 Algoritmo mixto decreciente - mayorado o algoritmo en cascada

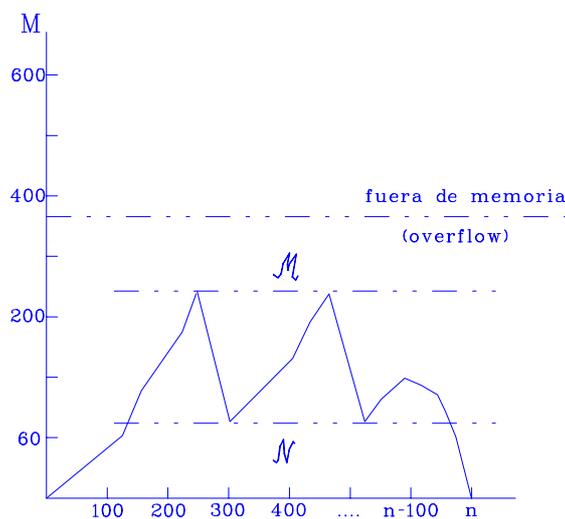
Al igual que antes comenzamos el proceso ordenando las unidades de trabajo de forma decreciente y aplicamos entonces el algoritmo de ordenación decreciente. Cuando el número de unidades de mercado sea igual a M , tomamos las $M-N$ primeras sobre las que aplicamos el algoritmo de ordenación mayorada, que se encargará de transformarlas en manipulables. Entonces se vuelve a aplicar el primer algoritmo y así sucesivamente hasta agotar las unidades de trabajo existentes.

```

Algoritmo_en_cascada(X,n,Y,m)
int X[],n;                               /* Unidades de trabajo y cantidad */
int Y[],m;                               /* Unidades de mercado y cantidad */
{ int unidad,unidades_de_trabajo;       /* Subíndices a usar */
  reordena(X,n,decreciente);             /* Función que reordena*/
  reordena(Y,m,creciente);
  unidad=1; unidades_de_mercado=0;
  for(i=1;1<=n;i++)                      /* Bucle de unidades de trabajo */
  {
    if(unidad_de_mercado < M)            /* Mientras no llegue el limite */
      algoritmo_de_ordenacion_decreciente(X,n,Y,m)
    else                                  /* si llega, comienza a almacenar */
      while(unidades_de_mercado>N)
        algoritmo_de_ordenacion_mayorada(X,n,Y,m)
  }                                       /* Fin del FOR */
}                                       /* Fin de la función */

```

El diagrama de uso de memoria de estos dos últimos algoritmos sería el siguiente:



6.5 Análisis de los algoritmos propuestos

6.5.1 Tiempos de ejecución

Consideremos las unidades x_1, x_2, \dots, x_n . Evidentemente, muchas longitudes se repiten, por tanto consideramos x_1, x_2, \dots, x_n como pares de la siguiente forma $x_i = (\text{longitud}, \text{cantidad})$. De esta manera, a la hora de ordenarlas no hemos de hacerlo con todas las unidades de trabajo individualmente sino por grupos o parejas.

Pasemos entonces a analizar los tiempos de ejecución de cada uno de los algoritmos para la solución de nuestro problema.

A la hora de reordenar las unidades de trabajo o las unidades de mercado elegimos un algoritmo determinado previamente. Es recomendable usar el algoritmo *Quick-Sort*, actualmente el más rápido de los usados en las bibliotecas de programación y cuyo tiempo de ejecución de n elementos pertenece a $\mathcal{O}(n \cdot \log(n))$. El único problema es que se trata de un método recursivo, por lo que utilizamos lenguajes de programación que permitan definir funciones recursivamente. El algoritmo *Quick-Sort* se define como sigue:

```
Quick_sort(T[i,..,j]) /* Ordena la matriz T en orden creciente */
{
  if(j < i)
    insert(T[i,..,j]);
  else
  {
    pivot(T[i,..,j],l)
    Quick_sort(T[i,..,l-1])
    Quick_sort(T[l+1,..,j])
  }
}/* Fin de la función */
```

Donde, tanto *insert* (función que ordena una pequeña cantidad de elementos) como *pivot* (función que pasa los elementos menores que $T[l]$ a la izquierda y los mayores a la derecha), pueden encontrarse en la bibliografía que existe al respecto (Véase *Gilles Brassard - 1988*). Este método fue creado por C.A.R. Hoare en 1962 basado en la idea de *divide y vencerás* (divide and conquer), proceso que descompone un problema en pequeños subproblemas, les da solución a cada uno de ellos e intenta combinarlas para dar una solución general al problema inicial.

Pasemos entonces a analizar los tiempos de ejecución de cada uno de los algoritmos para la solución de nuestro problema.

Como en casi todos ellos se produce un ordenamiento inicial de las unidades de trabajo y de mercado, consideramos este paso común a todos los algoritmos. Para analizar la complejidad de este paso común, sea $c(n,m) = n \cdot \log(n) + m \cdot \log(m)$.

Dicha ordenación tiene complejidad $\mathcal{O}(c(n, m))$. Como siempre se cumple que $n \geq m$ entonces

$$\mathcal{O}(c(n, m)) = \mathcal{O}(n \cdot \log(n) + m \cdot \log(m))$$

Por ser la función logaritmo estrictamente creciente se cumple que

$$n \cdot \log(n) \geq m \cdot \log(m)$$

entonces

$$\mathcal{O}(n \cdot \log(n) + m \cdot \log(m)) = \mathcal{O}(\max\{n \cdot \log(n), m \cdot \log(m)\}) = \mathcal{O}(n \cdot \log(n))$$

Esto es coherente, ya que la lentitud en el proceso depende de la magnitud del proyecto y no de las disponibilidades del mercado. Entonces tenemos $c(n, m) = c(n) = n \cdot \log(n)$

6.5.1.1 Tiempo de ejecución para el algoritmo de ordenación decreciente

Bucle de unidades de trabajo = n pasos

Bucle de unidades de mercado = m pasos en cada unidad de trabajo

Ordenaciones de unidades de mercado en cada inclusión o almacenamiento = 1 (a lo sumo) $\Rightarrow m \cdot \log(m)$

Total de pasos sin contar ordenaciones iniciales: $n \cdot m \cdot (m \cdot \log(m))$

Conclusión: el algoritmo de ordenación decreciente, observando las ordenaciones iniciales y el número de pasos clasificatorios pertenece a :

$$\mathcal{O}(c(n) + n \cdot m \cdot (m \cdot \log(m))) = \mathcal{O}(c(n) + n \cdot m^2 \cdot \log(m))$$

6.5.1.2 Tiempo de ejecución para el algoritmo de ordenación creciente

Bucle de unidades de trabajo = n pasos

Total de pasos sin contar ordenaciones iniciales : n

Conclusión : el algoritmo de ordenación creciente, observando las ordenaciones iniciales y el número de pasos clasificatorios pertenece a :

$$\mathcal{O}(c(n) + n)$$

6.5.1.3 Tiempo de ejecución para el algoritmo de ordenación alternada

Análogo al algoritmo anterior, el algoritmo de ordenación alternada pertenece a:

$$\mathfrak{O}(c(n) + n)$$

6.5.1.4 Tiempo de ejecución para el algoritmo de ordenación mayorada

En este caso, para clasificar la primera barra, recorrerá a lo sumo n barras (peor caso). Para la segunda n-1 y así sucesivamente. Por tanto la clasificación ejecuta un número de pasos:

$$\sum_{i=1}^n i = \frac{(n+1) \cdot n}{2}$$

Conclusión : el algoritmo de ordenación mayorada, observando las ordenaciones iniciales y el número de pasos clasificatorios pertenece a:

$$\mathfrak{O}\left(c(n) + \frac{(n+1) \cdot n}{2}\right)$$

6.5.1.5 Tiempo de ejecución para el proceso de llenado progresivo óptimo

El tiempo de ejecución es de orden exponencial al realizar todas las permutaciones posibles.

6.5.2 Tiempos de ejecución para algoritmos mixtos

Dados los algoritmos A_1, A_2, \dots, A_p , consideramos una mixtura algorítmica dentro de dicho conjunto. Sean n_1, n_2, \dots, n_p los elementos manipulados por cada uno de ellos, y n el total de elementos a clasificar ($n = \sum_{i=1}^p n_i$). Entonces, el tiempo de ejecución de la mixtura es de orden

$$\mathfrak{O}(t(n)) = \mathfrak{O}\left(\sum_{i=1}^p \lambda_i t_i(n)\right)$$

donde cada algoritmo A_i es de orden $\mathfrak{O}(t_i(n))$ y $\lambda_i = \frac{n_i}{\sum_{i=1}^p n_i} = \frac{n_i}{n}$, de donde se

cumple que $\sum_{i=1}^p \lambda_i = 1$. Es por tanto una combinación lineal convexa de algoritmos.

6.5.2.1 Tiempo de ejecución para el algoritmo en banda

Sea α el número de unidades de trabajo clasificadas por el algoritmo de ordenación creciente, y β el de las ordenadas por el algoritmo de ordenación alternada ($\alpha + \beta = n$).

Entonces, por la definición anterior obtenemos que el algoritmo de ordenación en banda pertenece a

$$\mathfrak{O}(c(n) + \lambda_1 \cdot n \cdot m^2 \cdot \log(m) + \lambda_2 \cdot n)$$

donde $\lambda_1 = \frac{\alpha}{\alpha + \beta}$ y $\lambda_2 = \frac{\beta}{\alpha + \beta}$

6.5.2.2 Tiempo de ejecución para el algoritmo en cascada

Ya que el algoritmo de ordenación mayorada pertenece a $\mathfrak{O}\left(c(n) + \frac{(n+1) \cdot n}{2}\right)$ y haciendo uso del mismo resultado anterior obtenemos que este algoritmo pertenece a

$$\mathfrak{O}\left(c(n) + \lambda_1 \cdot n \cdot m^2 \cdot \log(m) + \lambda_2 \cdot \frac{(n+1) \cdot n}{2}\right)$$

donde $\lambda_1 = \frac{\alpha}{\alpha + \beta}$ y $\lambda_2 = \frac{\beta}{\alpha + \beta}$

6.5.3 Comparación y elección de algoritmos

Hemos visto que el algoritmo de ordenación decreciente sería suficientemente bueno si no hiciera uso de tanta memoria. Parece, sin embargo, que el algoritmo de ordenación mayorada da un resultado similar y su tiempo para llegar a la solución es mejor al del anterior. Podríamos incluso plantearnos si habrá casos en los que

mejoraría la solución del primero. Para ver que no es así tenemos el siguiente resultado :

Teorema : Toda solución dada por el algoritmo de ordenación decreciente mejora o iguala a la dada por el algoritmo de ordenación mayorada.

Demostración: El algoritmo de ordenación mayorada se basa en cómo buscar los mejores candidatos para completar la única unidad de mercado sobre la que estamos trabajando. Diremos entonces que es un algoritmo de orden uno y lo notaremos por clase O_1 .

Sea $Res(O_1)$ el residuo generado por un algoritmo de orden uno. Podemos comprobar que $Res(O_2) \leq Res(O_1)$, es decir, el orden dos iguala o mejora al orden uno, ya que el número de candidatos en memoria para clasificar es mayor.

Este argumento podemos utilizarlo de forma recursiva y establecer que $Res(O_k) \leq Res(O_{k-1})$ para todo $k \in \{1, 2, \dots, m\}$, o de forma más general, $Res(O_k) \leq Res(O_h)$ para todo $k \geq h$ con $k, h \in \{1, 2, \dots, m\}$.

El algoritmo de ordenación decreciente para m unidades de mercado es de clase O_m (en caso de no tener limitaciones es en algoritmo de clase infinito, O_∞) y el algoritmo de ordenación mayorada es O_1 , aplicando que $Res(O_m) \leq Res(O_1)$, tenemos que *toda solución dada por el algoritmo de ordenación decreciente mejora o iguala a la dada por el algoritmo de ordenación mayorada.* c.q.d. ■

Ejemplo: Sea el caso de las unidades de mercado $\{\boxed{12}, \boxed{12}, \boxed{12}\}$ en la que hay que clasificar las unidades de mercado $\{\boxed{9}, \boxed{6}, \boxed{4.5}, \boxed{1.5}, \boxed{1}, \boxed{1}, \boxed{1}\}$. Tenemos los siguientes pasos (léase “☞” como finalizada y ya manipulable, y “↑” como sube en el orden).

Paso	Alg. de ord. decreciente (en este caso O_3)	Alg. de ord. Mayorada (O_1)
1	$\boxed{12} \mapsto \{\boxed{9}\}$	Queda $\{\boxed{3}, \boxed{12}, \boxed{12}\}$
2	$\boxed{12} \mapsto \{\boxed{9}\}$ $\boxed{12} \mapsto \{\boxed{6}\}$	Queda $\{\boxed{3}, \boxed{6}, \boxed{12}\}$
3	$\boxed{12} \mapsto \{\boxed{9}\}$ $\boxed{12} \mapsto \{\boxed{6} + \boxed{4.5}\}$ ↑	Queda $\{\boxed{1.5}, \boxed{3}, \boxed{12}\}$
4	$\boxed{12} \mapsto \{\boxed{6} + \boxed{4.5} + \boxed{1.5}\}$ ☞ $\boxed{12} \mapsto \{\boxed{9}\}$	Queda $\{\boxed{0}, \boxed{3}, \boxed{12}\}$
5	$\boxed{12} \mapsto \{\boxed{9} + \boxed{1}\}$	Queda $\{\boxed{0}, \boxed{2}, \boxed{12}\}$
6	$\boxed{12} \mapsto \{\boxed{9} + \boxed{1} + \boxed{1}\}$	Queda $\{\boxed{0}, \boxed{1}, \boxed{12}\}$
7	$\boxed{12} \mapsto \{\boxed{9} + \boxed{1} + \boxed{1} + \boxed{1}\}$ ☞	Queda $\{\boxed{0}, \boxed{0}, \boxed{12}\}$



Observamos que el residuo en el caso del algoritmo de ordenación decreciente es cero (hay una unidad de mercado que no llega a usar) y en el de ordenación mayorada es once (sólo puede colocar “uno” en la última unidad sobrando el resto).

Conclusión :

Después de realizar este estudio y en base a los resultados encontrados, podemos observar que son los algoritmos mixtos los que nos llevan a la solución más aproximada a la óptima sin hacer uso de una cantidad de memoria excesiva (es más, podemos controlarla ya que son de clase O_M con M por debajo del límite de memoria disponible), y, sobre todo, en un tiempo suficientemente rápido.

Por tanto, hemos cumplido los objetivos que nos planteábamos, dentro de las limitaciones indicadas en el planteamiento inicial del problema.

Nos encontramos ante la duda de que algoritmo elegir entonces (en banda o en cascada). Para tomar una decisión nos basamos en el tiempo de ejecución de ambos.

Suponiendo que λ_1 y λ_2 son iguales para ambos algoritmos, en orden a compararlos, tenemos que:

Sea $k(n) = c(n) + \lambda_1 \cdot n \cdot m^2 \cdot \log(m)$

El algoritmo en banda pertenece a $\mathcal{O}(k(n) + \lambda_2 \cdot n)$

El algoritmo en cascada pertenece a $\mathcal{O}(k(n) + \lambda_2 \cdot \frac{(n+1) \cdot n}{2})$

Si despreciamos $k(n)$ (por ser idéntico en ambos casos), obtenemos que:

$$\lim_{n \rightarrow \infty} \frac{\lambda_2 \cdot n}{\lambda_2 \cdot \frac{(n+1) \cdot n}{2}} = \lim_{n \rightarrow \infty} \frac{2}{n+1} = 0$$

Por tanto el tiempo de ejecución del primero es menor que el del segundo \Rightarrow la mejor decisión es elegir el algoritmo en banda.

6.6 *Introducción al Análisis Espacial*

Una vez resuelto el problema de cómo cortar las barras para cualquier volumen de producción, vamos a tratar de resolver el mismo problema pero en cualquier dimensión, es decir, cómo, por ejemplo, cortar chapas a partir de una plancha comprada en el mercado [AG90, Mar00] o cómo almacenar el máximo de cajas en un camión desaprovechando el mínimo espacio [Me95, Me01].

De entrada, parece que nuestros resultados anteriores pudieran tener poco que ver con lo que ahora nos planteamos, pero iremos viendo cómo aparecen similitudes y comportamiento análogos. Vamos a resolver el caso tridimensional, cuya solución es aplicable a 2 dimensiones y a más de 3.

6.7 *Limitaciones*

- No se trata de un problema lineal o modelizable mediante programación lineal ya que la solución que debemos encontrar son las coordenadas espaciales de posición de, al menos, dos vértices opuestos de una figura rectangular con caras paralelas a los planos coordenados.
- Encontrar todas las posibles ordenaciones y depurar la óptima es un trabajo inabordable.
- Las soluciones han de ser propuestas rápidamente pues, en muchos casos, dependerá de ello la eficacia práctica de una actividad determinada.
- No siempre poseeremos máquinas (computadoras) tan rápidas como deseamos, por lo que los métodos de resolución han de adaptarse a las disponibilidades técnicas.

Al igual que en las secciones anteriores haremos uso de la *algoritmia*. En esta línea, hemos dividido el proceso en cuatro fases:

1. Modelización matemática del problema.
2. Estrategias propuestas para la resolución de problemas concretos.
3. Análisis de los algoritmos y del tiempo de ejecución.

4. Ejemplo de aplicación.

Las soluciones aquí propuestas están basadas nuevamente en la técnica de *Divide y Venceras*. Las ventajas de este tipo de procedimientos son las de evitar el desaprovechamiento de espacio que en multitud de ocasiones se produce en almacenes o en medios de transporte, dando un método sencillo y útil para ser usado en casos concretos. Pensemos que, en la práctica, para transportes a corta distancia quizás no sean graves las pérdidas producidas por una incorrecta colocación de la carga, pero a larga distancia (barcos, aviones, convoyes de alimentos, etc.) o en caso de almacenamiento fijo (almacenes de distribución, hangares, polvorines, cámaras frigoríficas, etc.) los costes sí pueden ser elevados.

6.8 Modelización matemática

Deseamos generar un sistema que permita la ordenación de prismas rectangulares de 3 dimensiones dentro de una unidad de almacenamiento. Sea \mathbf{R} el conjunto de todos los rectángulos en \mathbb{R}^3 . Por similitud con los apartados anteriores vamos a usar la notación que sigue:

Sean X_1, X_2, \dots, X_k las unidades de trabajo (p. ej. las cajas que van a ser ordenadas)

Sean Y_1, Y_2, \dots, Y_m las unidades de almacenamiento (p. ej. los camiones o cámaras donde vamos a almacenar dichas cajas)

Partimos de que $X_i, Y_j \in \mathbf{R}$ para todo i, j .

6.8.1 Definición de rectángulo

Llamamos *rectángulo* en \mathbb{R}^3 a:

$R = [a_1, b_1] \times [a_2, b_2] \times [a_3, b_3]$ donde cada $[a_i, b_i]$ es un intervalo cerrado en \mathbb{R} .

Por la naturaleza física del problema que tratamos de resolver, suponemos que $a_1, b_1, a_2, b_2, a_3, b_3 \in \mathbb{R}^+$.

Cada unidad de trabajo queda definida por sus dimensiones:

$$X = [x_1, x_2, x_3]$$

o, como subconjunto de \mathbb{R}^3 , por $X = [0, x_1] \times [0, x_2] \times [0, x_3]$. Usaremos indistintamente ambas notaciones.

6.8.2 Propiedades de los rectángulos

Consideremos las siguientes propiedades de los rectángulos en \mathbb{R}^n :

Unión de rectángulos de dimensión n

Dados dos rectángulos X e Y , si tienen $n-1$ componentes iguales (sean, sin pérdida de generalidad, $x_1 = y_1; x_2 = y_2; \dots; x_{i-1} = y_{i-1}; x_i \neq y_i; x_{i+1} = y_{i+1}; \dots; x_n = y_n$

llamamos rectángulo unión (o suma) R y lo denotamos por $R=X+Y$ al rectángulo :

$$R = [x_1, x_2, \dots, x_{i-1}, x_i + y_i, x_{i+1}, \dots, x_n]$$

Dilatación

Dado un vector en \mathbb{R}^n , $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_n)$, la dilatación por un factor λ de un rectángulo X se denota por λX y se define como $\lambda X = [\lambda_1 x_1, \lambda_2 x_2, \dots, \lambda_n x_n]$

Rotación

Para considerar las posiciones de un rectángulo en \mathbb{R}^n , $X = [x_1, x_2, \dots, x_n]$ que conserven sus caras paralelas a los planos cartesianos, hemos de contemplar el conjunto de todas las permutaciones posibles de sus coordenadas. Por tanto dado el rectángulo X disponemos de $n!$ posiciones distintas.

En consecuencia consideramos que dos rectángulos R y R' son equivalentes si son iguales salvo permutaciones, y lo notaremos por: $R \overset{\sigma}{\approx} R'$ (es fácilmente comprobable que se trata de una relación de equivalencia).

6.8.3 Medidas de rectángulos

Una cuestión que va a tener especial importancia, al abordar cualquier método de optimización para el problema que nos ocupa, es el orden inicial en el que consideremos los rectángulos. La consideración de esta cuestión requiere como paso previo decidir la forma de medir un rectángulo, para lo cual partiremos del concepto matemático de *norma* y otros tipos de medidas. Entre las más conocidas podemos encontrar:

- 1) **Norma Euclídea** : $\|X\| = \sqrt{x_1^2 + x_2^2 + \dots + x_n^2}$

1. Se colocan las unidades de trabajo X_1, X_2, \dots, X_n en orden de medida volumétrica (o volumen) decreciente. Se colocan las unidades de almacenamiento Y_1, Y_2, \dots, Y_m en el orden que deseemos poniendo en primer lugar, como es lógico, las unidades que antes queramos llenar. Sea Y la primera unidad de almacenamiento.
2. Se busca desde X_1 hasta X_k el primer X_i tal que X_i cabe en Y .
3. Una vez introducido se definen tres nuevas unidades de almacenamiento Y_1, Y_2, Y_3 como indica la figura anterior.
4. Se procede nuevamente por el paso 2 del algoritmo en el orden Y_1, Y_2, Y_3 fijo.

Para ilustrar las funciones necesarias para su programación implementamos las siguientes instrucciones, donde el origen y el final de cada recuadro se especifican como aparece en el gráfico:

1. Estructuras

```
typedef {Double x,y,z;} Punto3D;
typedef {
    Punto3D Origen;
    Punto3D Final;
    Int cantidad;
} Unidad;
```

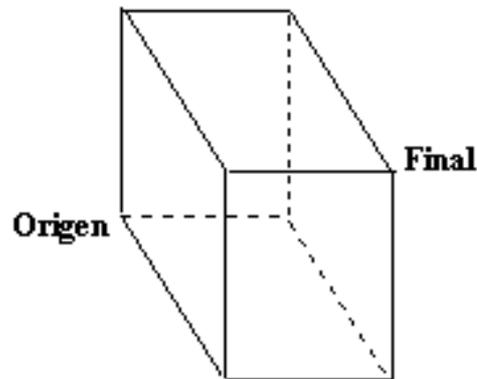


Ilustración 6-7. Puntos básicos de un rectángulo

2. Función Caber

```
Caber(X,Y)
Unidad X,Y;
{
    Const Short SI=1;
    Punto3D LadoX,LadoY;
    LadoX.x=X.Final.x-X.Origen.x; LadoY.x=Y.Final.x-Y.Origen.x;
    LadoX.y=X.Final.y-X.Origen.y; LadoY.y=Y.Final.y-Y.Origen.y;
    LadoX.z=X.Final.z-X.Origen.z; LadoY.z=Y.Final.z-Y.Origen.z;
    if(LadoX.x<LadoY.x & LadoX.y<LadoY.y & LadoX.z<LadoY.z) Caber=SI;
    if(LadoX.x<LadoY.x & LadoX.z<LadoY.y & LadoX.y<LadoY.z) Caber=SI;
    if(LadoX.y<LadoY.x & LadoX.x<LadoY.y & LadoX.z<LadoY.z) Caber=SI;
    if(LadoX.y<LadoY.x & LadoX.z<LadoY.y & LadoX.x<LadoY.z) Caber=SI;
    if(LadoX.z<LadoY.x & LadoX.x<LadoY.y & LadoX.y<LadoY.z) Caber=SI;
    if(LadoX.z<LadoY.x & LadoX.y<LadoY.y & LadoX.x<LadoY.z) Caber=SI;
}
```

3. Función Colocar

```

Colocar(X,Y)
Unidad X,Y;
{
  Const Short SI=1;
  if (Caber(X,Y)==SI)
  {
    X.Final.x +=X.Origen.x;
    X.Final.y +=X.Origen.y;
    X.Final.z +=X.Origen.z;
    X.Origen=Y.Origen;
  }
}

```

En nuestro estudio consideramos todas las posiciones posibles, algunas de las cuales observamos en la siguiente figura, que siempre pueden ser limitadas en función de las circunstancias de cada unidad de trabajo (sabemos que hay ocasiones en las que una unidad no puede, por ejemplo, volcarse), por lo que es necesario incluir restricciones en los movimientos de esas unidades:

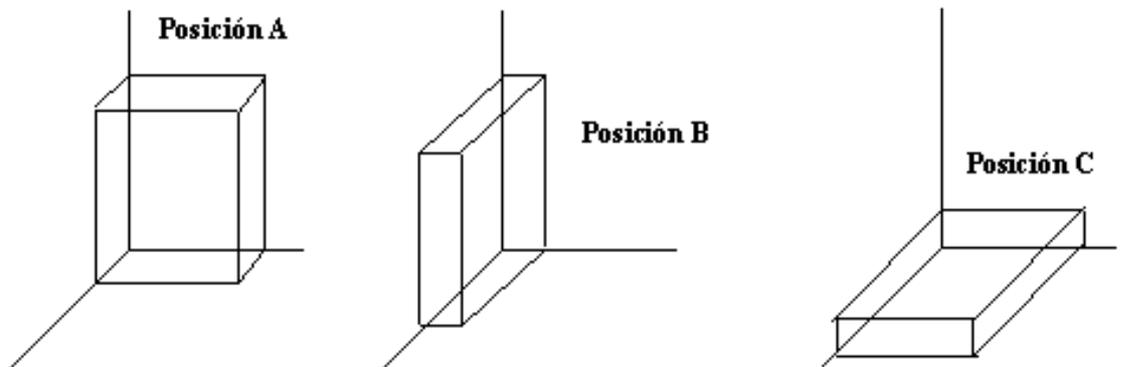


Ilustración 6-8. Giros de una unidad de trabajo

6.9.1 Tiempo de ejecución del algoritmo espacial

Para calcular el tiempo de ejecución, consideramos que:

1. En el primer paso podemos, en el peor de los casos, realizar un máximo n de comparaciones para ver cual de los X_i es el primero que cabe.

2. En los siguientes pasos tenemos n comparaciones menos las unidades ya clasificadas, además de que las unidades de almacenaje se multiplican por 3.

Por tanto la cota superior del tiempo de ejecución de este algoritmo será:

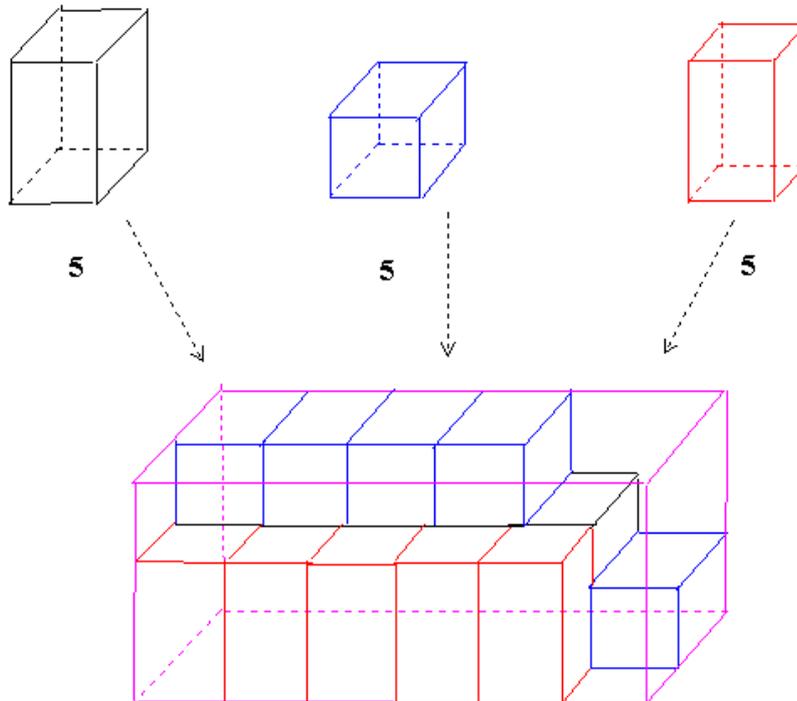
$$n + 3 \cdot (n-1) + 3^2 \cdot (n-2) + \dots + 3^{n-2} \cdot 2 + 3^{n-1} = n + \sum_{i=1}^{n-1} 3^i (n-i)$$

es decir, la velocidad de crecimiento está limitada superiormente mediante una suma aritmético-geométrica¹

$$T(n) < c \left(n + \sum_{i=1}^{n-1} 3^i (n-i) \right) \Rightarrow T(n) \in \mathcal{O} \left(n + \sum_{i=1}^{n-1} 3^i (n-i) \right)$$

6.10 Ejemplo de aplicación

Aquí podemos observar un ejemplo simple de la aplicación de este algoritmo a la colocación de un conjunto formado por 3 rectángulos distintos, cada uno de ellos repetido 5 veces, introducidos dentro de una unidad de almacenamiento:



¹ Esta suma finita se puede reducir: $n + \sum_{i=1}^{n-1} 3^i (n-i) = n + \frac{3^{n+1} - 3^2}{4} - \frac{3}{2}(n-1)$

6.11 Aportaciones Realizadas

- Plantear mediante programación lineal el problema y las restricciones para la minimización de residuos unidimensionales en la industria
- Crear, analizar, implementar y clasificar los algoritmos:
 - Ordenación Creciente
 - Ordenación Decreciente
 - Ordenación Alternada
 - Llenado Progresivo Óptimo
 - En Banda
 - En Cascada
- Modelar el problema de residuos en cualquier dimensión
- Generalizar los algoritmos anteriores a los casos bidimensional y espacial

6.12 Referencias Bibliográficas

[Ac67] Acher, J. “Algebre lineaire et programmation lineaire”. Dunod. Paris. 1967

[AG90] Arochena,H.E., Guerra,F. “Sistema de Optimización de Corte de Piezas Rectangulares”. Revista Investigación Operacional, Vol.XI,No.3, Cuba. 1990

[BB88] Brassard, G.; Bratley P. “Algorithmics. Theory and practice”. Prentice Hall. Montreal. Canadá. 1988.

[Br85] Brassard, G. “Crusade for a better notation”. Sigact News. ACM, vol. 17. Montreal. Canadá. 1985.

[Ce94] Ceballos, F. J.. “Enciclopedia de Microsoft Visual Basic”. Edit. Ra-Ma.. Madrid. España 1994.

[CK76] Chang, L.; Korsh, J. “Canonical coin changing and greedy solutions”.
Journal ACM, vol. 23. USA. 1976

[CK01] Caccetta, L.; Kulanoot, A. “Algorithms for some hard knapsack problems”
Kluwer Acad. Publ. 2001. Dordrecht. Holanda

[DG89] Díaz, P., Guerra, F. Sistema de Optimización de Corte de Piezas
Unidimensionales. Revista Investigación Operacional, Vol. X, No. 2. Cuba 1989.

[Fe73] Feller, W. “Introducción a la teoría de probabilidades y sus aplicaciones”.
Wiley. Princeton. Madrid. España. 1973

[GK81] Greene, D. H.; Knuth, D. E. “Mathematics for the analysis of algorithms”.
Birkhauser. Boston. USA. 1981

[GK94] Graham, R. L.; Knuth, D. E.; Patashnik, O. “Concrete Mathematics”.
Addison-Wesley. Stanford University. USA. 1994

[Go80] Gordon, G. “Simulación de Sistemas”. Diana. México 1980

[HL91] Hillier, F.S.; Lieberman, G. J. “Introduction to operations research” Edit. Mc
Graw Hill. Standford. EE.UU. 1991.

[IS97] ISI (Ingeniería de Sistemas Interactivos Ltda.) . “Computer integrated
transmission towers system”. Manual. Málaga. España. 1997.

[JM00] Johnson, D. B.; Mowry, T. A. “Matemáticas Finitas”. Thomson International.
México. 2000.

[K175] Kleinrock, L. “Queueing Systems”. Wiley Interscience. Los Angeles. USA.
1975

[Ma86] Mason, J.C. “Basic Matrix Methods”. Butterworth & Co. Londres. 1986

[Mar00] Marcelo, B. “Corte Certo 2000. Sistema de optimización de cortes para chapas”. Silicon Valley. USA. 2000

[Me91] Merino Córdoba, S. “Análisis de residuo mínimo” Tesis de Grado. Facultad de Ciencias Matemáticas. Universidad de Málaga. Dirección : Javier Erice. Málaga. España. 1991.

[Me95] Merino Córdoba, S. “Técnicas algorítmicas para almacenamiento” Artículo para la Secretaría General Técnica del Ministerio de Defensa. Málaga. España. 1994.

[Me01] Medina, I. “Algoritmos para la simulación de ordenación espacial”. Proyecto fin de carrera dirigido por Salvador Merino. Málaga. España. 2001

[MT90] Martello, S. y Toth, P. “Knapsack Problems” Edit. Wiley. Chichester. Inglaterra. 1990.

[OI70] Olesten, N.O. “Numerical control”. Wiley interscience. La Jolla. California. USA. 1970

[Pe87] Peña Sánchez de Rivera, D. “Estadística, modelos y métodos”. Alianza Universidad. Madrid. España. 1987

7. Cálculo Inverso. Diagramas de Utilización

En este capítulo trataremos el aprovechamiento óptimo de las torres ya fabricadas y almacenadas o en uso.

En el campo de las TTE ocurre frecuentemente que, transcurrido el tiempo, es necesario actualizar el tipo de conductor usado (puede haberse quedado anticuado el que tenía o no fabricarse ya ese modelo) o modificar levemente el trazado de la línea (por crecimiento de las poblaciones) [Ch88, FC96].

Evidentemente se puede resolver este problema realizando un nuevo proyecto y calculando y fabricando otras torres. Pero habitualmente estos cambios en la línea pueden venir acompañados de :

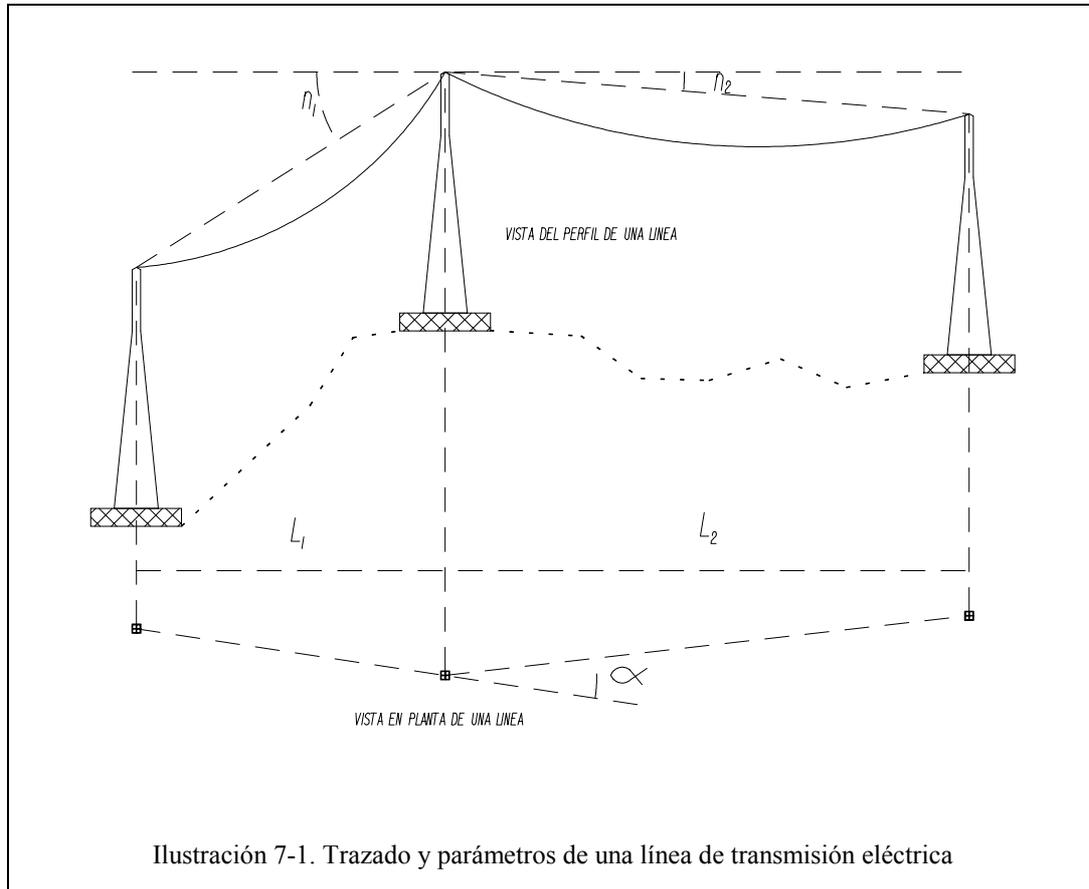
- Brevedad de tiempo para realizar el cambio del tendido
- Existencia de torres iguales en el almacén
- Mayor ahorro económico si utilizamos la torre que ya hay

Por tanto se nos plantea una situación nueva, y es ¿cómo sabemos qué tipos de modificaciones de entorno pueden resistir las torres sin cambiar su diseño y cuáles serían sus límites?. Estos interrogantes no siguen las pautas del sistema de cálculo anterior, ya que no es simplemente la comprobación de una estructura donde se cambian las cargas, sino entre qué límites pueden estar estos cambios [RA92].

Actualmente estas modificaciones están siendo fundamentales en España y en muchos otros países al incorporarse a las líneas de alta tensión el uso de la transmisión de información. Está siendo habitual el cambio del cable de tierra clásico por otro en cuyo interior (también llamado *alma del conductor*) contiene fibra óptica para realizar la transmisión de internet, radio, telefonía y televisión por cable [Mo88, IM90].

7.1 Parámetros de entorno

Para comenzar se analizan los parámetros que intervienen en este tipo de modificaciones[Me03]. Como podemos observar en el siguiente gráfico las características en el trazado de una línea son diversas:



A partir de estos datos se definen las siguientes variables [IS97]:

- $L = \frac{L_1 + L_2}{2}$, llamado *Vano Viento* y es la media aritmética de los vanos a los lados del apoyo.
- $N = \text{tg}(n_1) + \text{tg}(n_2)$, es la suma de las tangentes de la desviación vertical de la línea, debida a las irregularidades de nivel del terreno
- $S = 2 \text{sen}\left(\frac{\alpha}{2}\right)$, define la desviación horizontal de la línea y nace del hecho de no poder situar las torres en línea por la existencia de algún obstáculo en el camino

A partir de ellos se puede asumir que todos los esfuerzos externos transmitidos a un apoyo, para un conductor, zona y condiciones de tendido determinadas, son funciones lineales de las magnitudes que se han designado por L, N y S, las que a su vez dependen de la situación de cada apoyo dentro de la línea.

7.2 *Diseño de ecuaciones LNS*

Dada cualquier barra de la estructura, esta tiene unos límites de cargas que puede soportar, tanto a tensión como a compresión [AS88], como se definió en el capítulo de cálculo espacial de estructuras. Estos límites los llamamos cargas de saturación y en ellos van a influir tanto las cargas fijas propias de la estructura (su propio peso y el viento sobre la torre) como todas aquellas cargas horizontales, verticales o tensiones de conductores que se aplican sobre la misma [Ra73, AA90].

A partir de estas premisas se puede establecer que la carga de saturación de una barra debe ser una función lineal de la forma:

$$F = F_H L + F_V N + F_T S + F_F$$

donde :

- F es la carga o fuerza de saturación de cada barra en cada condición
- F_H es la suma de los factores horizontales
- F_V es la suma de los factores verticales
- F_T es la fuerza debida a la tensión del conductor
- F_F son las cargas fijas sobre la barra

Por su expresión se conoce a estas ecuaciones como LNS.

Además de estas ecuaciones para las barras, definimos de forma similar otra para las cadenas de aisladores de la forma:

$$A_W L + A_V N + A_T S = A_F$$

donde :

- A_W es la fuerza del viento sobre el conductor
- A_V es la suma de los factores verticales
- A_T es la tensión del conductor
- A_F son las cargas fijas sobre la cadena

7.3 Factores de Repercusión y Entrada de Datos

En primer lugar deben calcularse los factores que nos informan de cómo cada carga con valor de una unidad dispuesta de forma vertical, longitudinal o transversal sobre cada nudo de carga (puntos donde se amarran los conductores o lugares donde se simula el viento sobre la estructura) repercute sobre las diferentes barras de la estructura. Son llamados *factores de repercusión* [Gö88, Mo88, IS97].

En el caso de que estos factores de repercusión se establezcan para cada nudo de la estructura en todas las direcciones del espacio obtendríamos en el sistema de ecuaciones a resolver el siguiente resultado [Me03]:

$$M \cdot X = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 \end{pmatrix} \Rightarrow M \cdot X = I \Rightarrow X = M^{-1}$$

Por tanto realmente nuestro resultado es la inversa de la matriz de rigidez, de ahí el nombre de *cálculo inverso*. Para disminuir el tiempo de computación y dado que los puntos de aplicación de las cargas son algunos en particular, realmente pasamos a resolver un *cálculo inverso selectivo*, ciñéndonos exclusivamente a los nudos que nos interesan.

El primer paso será realizar un cálculo de la estructura con tantas condiciones (que llamaremos *condiciones de aplicación* para distinguirlas de las de cálculo) como sean necesarias. El número de tales condiciones es la suma de:

- La cantidad de nudos que posea la estructura donde se sujeten cables, multiplicada por las tres posibles direcciones del espacio
- La cantidad de nudos de viento, multiplicada por las dos direcciones de aplicación posibles que son transversal y longitudinal (el viento nunca es vertical)

- Una carga unitaria repartida en la torre para ver cómo repercute el peso propio de la estructura en todas las barras

Por tanto: $Condiciones = 3 \times Cables + 2 \times Vientos + 1$

Una vez finalizado el cálculo de los factores de repercusión se obtiene la influencia de cualquier carga unitaria sobre cada barra de la torre. Para no interferir en los datos originales de diseño, se crean nuevos archivos de cargas y fuerzas axiales unitarias con los nombres:

Torre.C2

Torre.AX2

Conocidos los datos de repercusión, el sistema solicita los datos del nuevo entorno de la torre, es decir, tipos de cables, vano, zona climática (en España existen 3 diferentes A, B y C en función de las condiciones atmosféricas del entorno y de su altura sobre el nivel del mar) y condiciones de carga a aplicar.[RAT, Mo88]

Las estructuras creadas para almacenar los datos acerca de los cables utilizados (cable de tierra y conductor de electricidad) y de la cadena de aisladores, son las siguientes [Me03, JM99]:

Type Cables

Fase as Integer
 Tipo as String
 Nombre as String
 Composicion as String
 Diametro as Single
 area as Single
 Peso as Single
 ModuloElasticidad as Double
 ExpansionLineal as Double
 CargaRotura as Single
 PresionViento as Single
 CargaHielo as Single
 tension as Single
 Temperatura as Integer
 Viento as Boolean
 Hielo as Boolean
 MaxWinTension as Single
 MaxWinTemperatura as Integer
 MaxWinViento as Boolean
 MaxWinHielo as Boolean
 MaxTemTemperatura as Integer
 MaxTemViento as Boolean
 MaxTemHielo as Boolean

End Type

Type Cadenas
MaximoAngulo as Single
Longitud as Single
Peso as Single
Contrapeso as Single
PresionViento as Single
ReduccionPresionViento as Single
End Type

7.4 Procedimiento de Cálculo

A partir de la entrada de datos se completa la siguiente estructura de coeficientes para ecuaciones LNS (obsérvese que cada estructura está anidada en la siguiente):

Type CargaEspacial
X as Double (Longitudinal)
Y as Double (Transversal)
Z as Double (Vertical)
End Type

Type CoeficientesCable
CargaVariable(Condiciones) as CargaEspacial
CoeficienteS as Double
End Type

Type CoeficientesLNS
CoeficienteRepercusion(CondicionesAplicacion) as Double
Conductor as CoeficientesCable
Tierra as CoeficientesCable
CargaFija(Condiciones) as Double
CargaPeso as Double
End Type

Sea CoeficienteLNS una variable del tipo *CoeficientesLNS*. Se crea un archivo de almacenamiento con tantos registros como barras de cálculo aparezcan en la torre y donde cada uno de ellos poseerá la información de CoeficienteLNS para la barra. Este archivo se nota por :

Torre.COE

Describamos el procedimiento de cálculo de coeficientes:

- Se introducen los datos de Torre.AX2 en Torre.COE ocupando la posición de cada condición de aplicación en:

CoefficienteLNS. CoeficienteRepercusion(CondicionAplicacion)

- Para cada condición de aplicación que actúe sobre un nudo donde pueda aplicarse un cable (tierra o conductor) en sentido transversal (Y) se acumula su factor sobre la variable

CoefficienteLNS. Conductor.CoefficienteS ó

CoefficienteLNS. Tierra.CoefficienteS (según el tipo de cable)

- Si, dada una condición original de cálculo de la estructura, ésta tuviera carga en un nudo, se acumula sobre su posición (la condición y su dirección X,Y ó Z) todos los factores que pertenezcan a cargas puntuales en esa dirección y nudo sobre los valores de :

CoefficienteLNS. Conductor.CargaVariable(Condicion).X , Y ó Z ó

CoefficienteLNS. Tierra.CargaVariable(Condicion).X , Y ó Z

- Paralelamente, si se comprueba que es un nudo de un conductor y además se trata de una torre de suspensión (ya que las de otro tipo no poseen cadenas de aisladores), se introducen las cargas fijas, según su dirección, de la forma siguiente:

Dirección X (Longitudinal)= Tensión del conductor x Factor

Dirección Y (Transversal)= Presión de viento en la cadena x Factor

Dirección Z (Vertical)= Peso de la cadena x Factor

donde el *Factor* es el coeficiente de repercusión de la condición de aplicación actual sobre la barra en que trabajamos. Este dato lo acumulamos en los valores de :

CoefficienteLNS. CargaFija(Condicion)

- En estas posiciones de cargas fijas también se introducen las debidas al viento sobre la estructura, para lo cual, una vez comprobado en cada condición si posee viento o no, se calcula el valor de :

Dirección X (Longitudinal)= Carga de Viento Longitudinal x Factor

Dirección Y (Transversal)= Carga de Viento Transversal x Factor

siendo el *Factor* el mismo que antes, introduciéndolo nuevamente en:

CoefficienteLNS. CargaFija(Condicion)

- En la última condición de aplicación se encuentra el factor de repercusión del peso propio de la estructura y directamente es aplicado en:

CoefficienteLNS. CargaPeso = Peso de la Estructura x Factor

Con ello quedan establecidos los datos necesarios para construir las ecuaciones LNS. A partir de aquí bastará con recorrer cada barra y dentro de ella cada condición de carga y cada situación (tensión o compresión), para rellenar la estructura:

```
Type EcuacionLNS
  CargaSaturacion as Single
  CoeficienteL as Single
  CoeficienteN as Single
  CoeficienteS as Single
  CargasFijas as Single
End Type
```

de la forma siguiente (los datos de resistencia provienen del cálculo previo de la estructura):

$$F = \text{EcuacionLNS.CargasSaturación} = \text{Resistencia de la Barra (Cond., Situa.)}$$

$$F_L = \text{EcuacionLNS.CoeficienteL} = L_1 + L_2 + L_3 + L_4$$

$$F_N = \text{EcuacionLNS.CoeficienteN} = N_1 + N_2$$

$$F_S = \text{EcuacionLNS.CoeficienteS} = S_1 + S_2$$

$$F_F = \text{EcuacionLNS.CargasFijas} = \\ = \text{CoeficienteLNS.CargaFija(Condicion)} + \text{CoeficienteLNS.CargaPeso}$$

donde :

$$L_1 = \text{CoeficienteLNS.Tierra.CargaVariable(Condicion).Z} \times \text{Peso del Tierra}$$

$$L_2 = \text{CoeficienteLNS.Tierra.CargaVariable(Condicion).Y} \times \text{Presión Viento Tierra}$$

$$L_3 = \text{CoeficienteLNS.Conductor.CargaVariable(Condicion).Z} \times \text{Peso del Conductor}$$

$$L_4 = \text{CoeficienteLNS.Conductor.CargaVariable(Condicion).Y} \times \text{Presión Viento Conductor}$$

$$N_1 = \text{CoeficienteLNS.Tierra.CargaVariable(Condicion).Z} \times \text{Tensión Tierra}$$

$$N_2 = \text{CoeficienteLNS.Conductor.CargaVariable(Condicion).Z} \times \text{Tensión Conductor}$$

$$S_1 = \text{CoeficienteLNS.Tierra.CoeficienteS} \times \text{Tensión Tierra}$$

$$S_2 = \text{CoeficienteLNS.Conductor.CoeficienteS} \times \text{Tensión Conductor}$$

$$\text{obteniendo la ecuación : } F = F_L L + F_N N + F_S S + F_F$$

Si estamos en el caso de que la barra es un montante y es una torre de suspensión entonces se puede calcular la ecuación de las cadenas de la forma :

$A_L = \text{EcuacionLNS.CoeficienteL} = \text{tg}(\alpha) \times \text{Peso Conductor} - \text{Carga Viento}$

$A_N = \text{EcuacionLNS.CoeficienteN} = \text{tg}(\alpha) \times \text{Peso Conductor} \times$

$\frac{\text{TensionConductorConMaximoViento}}{\sqrt{\text{PesoConductor}^2 + (\text{PresionVientoConductor} \times \text{DiametroConductor})}}$

$A_S = \text{EcuacionLNS.CoeficienteS} = -\text{TensionConductorConMaximoViento}$

$A_F = \text{EcuacionLNS.CargasFijas} = \text{tg}(\alpha) \times (\text{PesoCadena}/2 + \text{Contrapeso}) -$

$\text{PresionVientoCadena}/2$

obteniéndose análogamente la ecuación:

$$A_L L + A_N N + A_S S = A_F$$

En caso de querer representar las ecuaciones, como tenemos 3 variables, su expresión sería la de una interferencia de planos en el espacio. Al igual que en Econometría, y para facilitar su interpretación, se fija el valor de alguna de ellas, normalmente $S=0$. Entonces se calculan las intersecciones de todas las rectas y se desarrolla la poligonal que más se acerque al origen de coordenadas, es decir, $(L=0, N=0)$.

7.5 Representación de las ecuaciones sobre el diagrama

El diagrama representado contiene por tanto sólo aquellas barras que, en su peor condición y situación, van a provocar las limitaciones de la torre dentro de su disposición geográfica en la línea de transmisión eléctrica:

En este diagrama, el eje de ordenadas indica los diferentes valores de N (recordemos que expresa la suma de las tangentes de desviación vertical de la línea) y el de abscisas las medidas de L (máximos vanos de viento permitidos para separar las torres)

En la parte superior se encuentra (en azul) las ecuaciones de aquellas barras que están más cerca de la saturación y, por tanto, provocan los límites de posibles modificaciones en el entorno de la línea. Como podemos observar se trata de dos diagonales y un montante (leg).

En la parte inferior aparece la ecuación que limita la inclinación de las cadenas y que cierra el diagrama. Dentro del área delimitada será posible variar la estructura sin llegar a la saturación de ninguna de sus barras con un ángulo de desviación horizontal S nulo.

Si modificamos S , se observa cómo aparece otro diagrama dentro del anterior que limita aún más el entorno de trabajo.

El último límite que indica el diagrama es la máxima separación entre torres para que, cuando aumente el empuje de viento, los conductores no se acerquen tanto que infrinjan las distancias de seguridad exigidas y provoquen saltos eléctricos entre sí. [DS87, WA91, MA01] Esta distancia viene indicada por la expresión internacional *Max. Recommended Wind Span for conductors separation*. Para el cálculo de esta distancia se hace uso de la ecuación cúbica del cambio de condiciones, donde se utilizan las herramientas necesarias del cálculo numérico para su resolución (en este caso se aplica la aproximación por el *método iterativo de Newton-Raphson*).

7.6 Hiperestructuras

A partir de este tipo de cálculo, que hemos dado en llamar *Cálculo Inverso* [MV99], nos introducimos en un campo del cálculo más global que vamos a definir como el de las Hiperestructuras.

Generalizando los conceptos expuestos en este trabajo ordenamos las restricciones de diseño estructural en los siguientes conjuntos:

E = Especificaciones de un proyecto (Datos del proyecto \cup Limitaciones de diseño)

F_0 = Fuerzas particulares iniciales (Cargas \cup Viento \cup Escalada para Mantenimiento)

G = Geometría (Distancias \cup Códigos \cup Juntas)

A partir de aquí decimos que la estructura básica para transmisión eléctrica (o datos iniciales de entrada de una torre) T_0 es una terna formada por:

$$T_0 = EF_0G$$

Si a estos datos le aplicamos un sistema de cálculo (como es en nuestro caso el MEF), al que llamaremos la función *cálculo* ó C , obtendremos un modelo de torre

que pertenezca a los tres conjuntos (ha sido diseñada, con sus correspondientes secciones de material existente y necesario, para resistir las fuerzas aplicadas inicialmente bajo la geometría definida), y la llamamos \mathbf{T} . Por tanto :

$$\mathbf{T} = C(T_0)$$

De la misma manera notamos al Cálculo Inverso por C^{-1} , por el cual obtenemos :

$$T = C^{-1}(\mathbf{T})$$

El problema que nos encontramos en el cálculo inverso de \mathbf{T} , cuyo resultado es T , es que no se corresponde estrictamente con T_0 . ¿Qué respuesta nos está dando entonces?. Realmente lo que ocurre es que la función *cálculo* no es biyectiva, sino sobreyectiva (por ello el nombre de cálculo *inverso* no sería estrictamente correcto al no existir biyectividad). Como dos ternas diferentes pueden dar la misma torre

$$\mathbf{T} = C(T_0) = C(T_1) \text{ con } T_0 \neq T_1$$

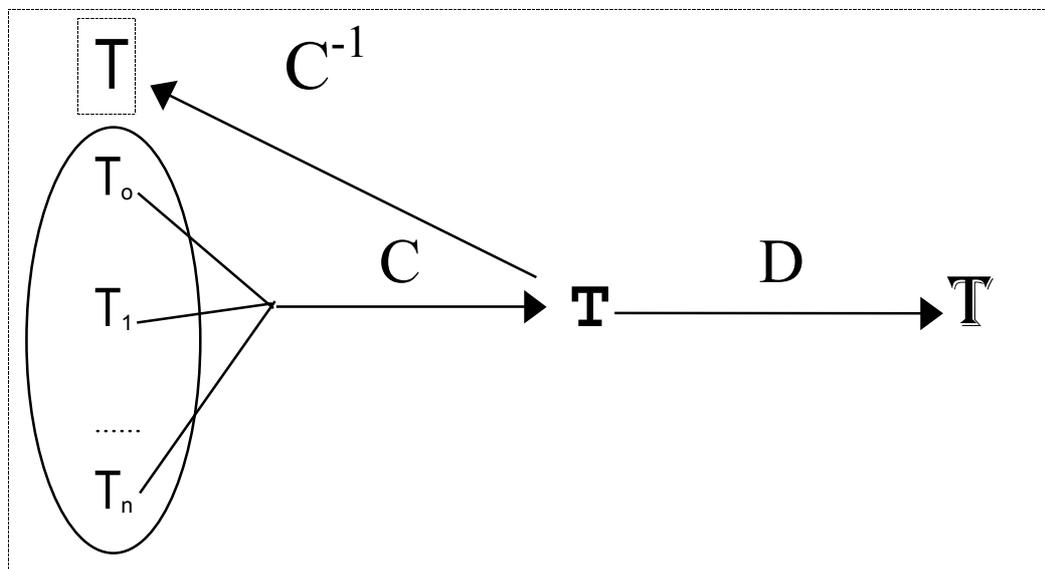
la representación de $\mathbf{T} = \text{EFG}$, mediante el cálculo inverso, no es una torre sino el poliedro (véase el gráfico del diagrama en tres dimensiones) donde están contenidas todas las condiciones de carga posibles tales que sus cálculos den la misma torre:

$$T = \{T_i = \text{EF}_i\text{G} \mid \mathbf{T} = C(T_i)\}$$

Unido a lo anterior, podemos definir también una función de diseño o dibujo D , sólo aplicable a una torre \mathbf{T} (ya que hemos de conocer el material con el que se debe diseñar) y cuyo resultado nos servirá para la fabricación en el espacio y lo notaremos por T :

$$T = D(\mathbf{T}) = D(C(T_0))$$

Con todo ello se obtiene el diagrama:



De aquí que definamos como Hiperestructura al conjunto :

$$T = \{T_i = EF_i G / T = C(T_i)\}$$

y, rigurosamente, al cálculo inverso como aquel que, a partir de los datos de una torre ya calculada T nos devuelve su hiperestructura original (todas aquellos datos de origen que producen la misma torre de cálculo). El diagrama en tres dimensiones que genera todo el conjunto T sería:

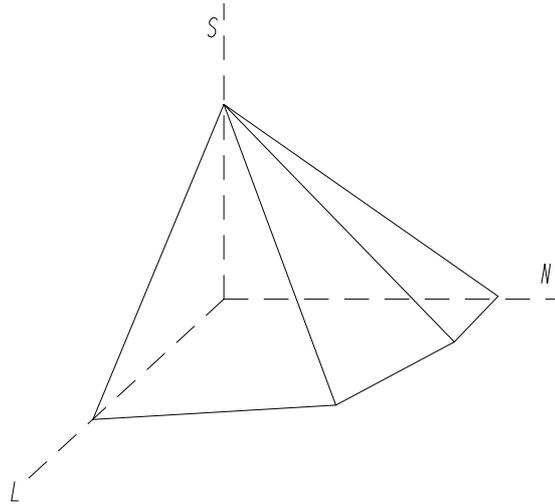


Ilustración 7-2. Poliedro de utilización

7.7 Torres de Comunicaciones

La información proporcionada por los diagramas de utilización es usada para establecer el comportamiento de una torre ante posibles cambios frecuentes y actualizaciones, proporcionándonos además un *orden parcial* [Go74, AM01] desde el punto de vista de su clasificación cara a la resistencia estructural.

Un campo sin duda de constante renovación es el de las telecomunicaciones. Para describir el procedimiento de cálculo de las torres aquí usadas hemos optado por el análisis particular de una familia de torres y del tipo de anclaje de antenas usado en ellas, ambos patentados por la compañía ISI (*Oficina Nacional de Patentes y Marcas. Registros nº 200001529/200001530*). En los cálculos observaremos como repercuten en estas torres tanto las diferentes antenas (parabólicas, crosspolares, focales, radome, etc.) como los vientos aplicados por cada normativa. Paralelamente iremos interpretando el software de cálculo desarrollado por ISI para la comprobación de sus torres de comunicaciones [IS100].

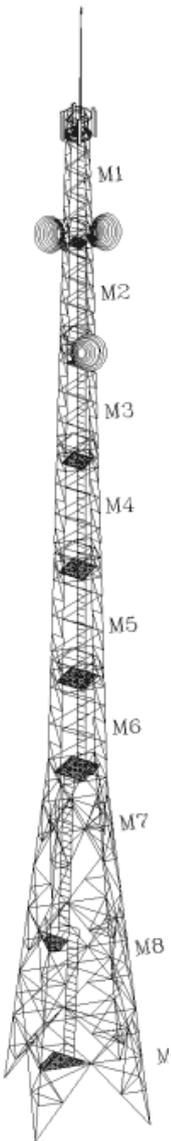


Ilustración 7-3. Torre Multiusos

7.7.1 Estructuras multiusos

Con este nombre se presenta un tipo de torres formado por nueve módulos de alturas semejantes que conforman una altura global nominal de 52 metros. Cada uno de ellos es una torre en sí, salvo los tres últimos (que se utilizan únicamente para incremento de altura), y contiene todos los elementos necesarios para poder colocar sobre ellos plataformas, antenas, soportes, cableado, etc. De ahí que reciba el nombre comercial de “torre de torres”. Por tanto a la hora de combinar sabemos que poseerán un máximo de 6 posibles niveles de antenas.

Para el cálculo de una torre de estas características ha de tenerse en cuenta que el abanico de posibles usos que una empresa de telecomunicaciones va a hacer de ella es excesivamente amplio y evolutivo (amplio porque el número de modelos de antenas disponibles en el mercado lo es, y evolutivo porque cada día surgen nuevos sistemas de telefonía como son analógica, GMS, WAP, UMTS,... que exigen un cambio global de los receptores y emisores). Pero la inversión realmente costosa para estas compañías es la estructura de soporte. En ella deben incurrir características como fácil y rápido montaje, acceso fácil, trabajo en óptimas condiciones de seguridad o posibilidad de abarcar el máximo espacio y espectro radioeléctrico.

Por ello vamos a ver cada una de las fuerzas que afectan a las torres de este tipo y como se incorporan a la estructura.

7.7.2 Fuerzas de viento sobre la estructura

Debido a la gran altura que poseen las estructuras de telecomunicaciones, o al hecho de posicionarse encima de edificios o estructuras mayores que las que hay a su alrededor, el cálculo de los vientos que les afectan es ligeramente diferente al de las torres de transmisión eléctrica¹.

En los años 60, la *US Navy* desarrolló el proyecto *OMEGA* alrededor del mundo. Se trataba de torres de comunicaciones entre 300 y 400 metros de altura que permitían

¹ La necesidad de mayor altura es claramente comprensible desde el momento que observamos que las torres eléctricas deben vencer ciertas distancias de seguridad con cualquier objeto que se encuentre bajo ellas, mientras que las torres de telecomunicaciones deben tratar de radiar el mayor espacio a su alrededor. Por ello buscaremos estructuras que no se desplacen mucho con el viento (para no perder señal de radio) y tan altas como sea posible.

establecer la posición de cualquier barco en un punto de la Tierra (esas torres han caído posteriormente en desuso al aparecer recientemente el sistema GPS por satélite). A la hora de calcular como era la fuerza del viento que afectaba a estas estructuras se lanzaron globos aerostáticos para la toma de datos, y de sus investigaciones se dedujo algo que hasta el momento no había sido tomado en cuenta (nunca se habían hecho estructuras tan altas y ligeras). El viento, a partir de los 10 metros de altura, va sufriendo un incremento que no se estabiliza hasta pasados los 200 metros de altura. La fórmula que deduce el factor a aplicar sobre la velocidad de viento actuante es:

$$V = \sqrt[4]{\frac{h}{10}}$$

donde h es la altura sobre el nivel de tierra a partir de los 10 metros. Este fenómeno se debe a que el viento, al ir incrementando la altura sobre el nivel de tierra, va encontrando un menor número de obstáculos (edificios, montañas, árboles, etc.) que vencer y por ello incrementando su velocidad

Por tanto, volviendo al sistema de cálculo, el primer paso será establecer que modelo de presiones de viento sobre la estructura se va a utilizar:

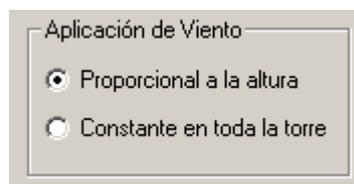


Ilustración 7-4. Decisión booleana que indica el modelo de aplicación de viento

Para poder calcular cual es la fuerza de viento a aplicar, sea cual sea la normativa y las velocidades de vientos que marquen cada zona geográfica, se necesitan los siguiente datos. Sean:

1. $k(h) = \sqrt[4]{\frac{h}{10}}$ el factor proporcional a la altura
2. $Op(h)$ el porcentaje de opacidad por cada metro de altura en cada nivel. Como dato base partimos del porcentaje de ocupación de las escaleras y cables emisarios en un metro cuadrado de superficie y le añadimos el porcentaje debido a las barras y chapas de la estructura.

Área de escaleras y emisarios (m² por m) 0.12

3. $C(h)$ el valor de una carga unitaria aplicado al tope de cada panel. Si el viento es constante en toda la torre este valor será siempre 1 pero si el viento es proporcional a la altura entonces $C(h) = k(h)$
4. La velocidad de viento expresada en metros por segundo a partir de la velocidad base (calculada para cada zona geográfica a nivel de tierra en las peores condiciones históricas registradas) es:

$$V_{ms} = V_{base} \cdot \sqrt[4]{\frac{h_n}{10}} = V_{base} \cdot K(h)$$

5. $P(h)$ la presión de viento por metro cuadrado de superficie en base a la velocidad del mismo. Se calcula mediante la fórmula:

$$P(h) = \left(\frac{V_{ms}^2}{16} \right) \cdot 1.6 \cdot 1.5$$

siendo 1.6 el factor forma del acero y 1.5 el factor debido a las dos caras de la torre.

Finalmente las cargas aplicadas por empuje del viento sobre la estructura, a cada altura, será de:

$$Carga(h) = P(h) \cdot Op(h) \cdot C(h)$$

Por tanto nuestros cálculos deben partir de 4 condiciones iniciales para establecer los factores de repercusión: carga unitaria a cada nivel y carga proporcional a cada nivel, aplicando ambos casos tanto en el sentido transversal como longitudinal².

7.7.3 Fuerzas de viento sobre las antenas

A la hora de estimar como se transmite el empuje del viento sobre las antenas a la estructura es importante considerar cual es el sistema mecánico de acoplamiento de las antenas. En el caso de las estructuras multiusos también fue patentado un sistema que permitiese gran versatilidad en el número de posiciones posibles de las antenas (dentro de los 360° que rodean a la estructura) con el máximo de seguridad para el personal de construcción y mantenimiento. La opción final que unía a estas

² Usamos la nomenclatura “transversal y longitudinal” de forma incorrecta, ya que aquí no tenemos los cables de la línea eléctrica como punto de referencia, pero son definiciones que han heredado de ellas y simplemente las conservamos. En caso contrario podríamos hablar de viento frontal y viento lateral

características la de facilidad de fabricación y montaje fue la que aparece en el siguiente gráfico:

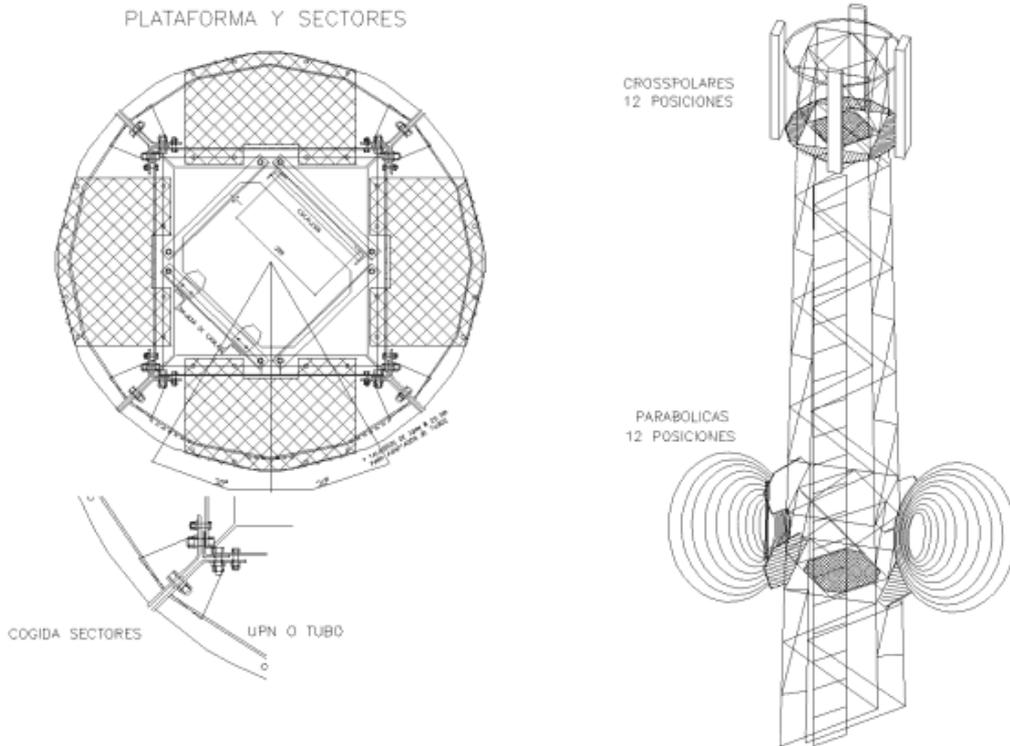


Ilustración 7-5. Anclaje de Antenas de ISI

Se trata de un arco de circunferencia tubular que cubre 90 grados de ángulo (visto en planta) y con 3 posibles posiciones de las antenas (las cuales son atornilladas a unas barras que actúan de soporte, siendo colocados estos soportes en dichas posiciones). Una vez definido este modelo, cualesquiera combinación que se desee de antenas podrá aplicarse a la estructura. Aunque, matemáticamente, el número de posibles posiciones alrededor de la estructura es infinito, hemos de tener en cuenta que a partir de 12 posiciones es probable que las antenas parabólicas estén excesivamente próximas unas a otras y comiencen a chocar entre si. Para el caso de antenas crosspolares de telefonía móvil, los modelos existentes en el mercado cubren un ángulo mínimo en planta de 90° por lo que habitualmente se utiliza una configuración de 4 antenas en círculo que cubra toda el área.

Elegida la configuración deseada, han de transmitirse las fuerzas a la estructura. Para ello se usa el sistema de análisis de empuje de viento sobre antenas desarrollado por

la compañía Andrews [AN00] y del que aparece a continuación una imagen de su funcionamiento para estos cálculos

Antenna Windloading

Default Offsets ?

Wind

Antenna Vertex

Antenna Mounting Pipe

Wind Angle: 30

Wind Speed: 125 MPH, 200 km/h, 55.6 m/s

Diameter: 1 ft (0.3m)

Forces and Moments					
Fa	Fs	M	Fat	Fst	Mt
14.4807	7.24038	1.20264	14.4807	7.24038	12.6424
lb	lb	ft-lb	lb	lb	ft-lb
Fa	Fs	M	Fat	Fst	Mt
64.4137	32.2068	1.63056	64.4137	32.2068	17.1408
N	N	N-m	N	N	N-m

- P Series (Standard Antenna)
- FP Series (Focal Plane)
- P/FP Series with Radome
- HP/UHX Series (Shielded)
- GP Series (Grid Antennas)
- KP Series (High Freq.)
- KP Series (Low Freq.)

Ilustración 7-6. Cálculo de Cargas por el método de Andrews

En este sistema se han realizado todas las pruebas de túnel de viento necesarias para poder conocer con exactitud el comportamiento del viento sobre las antenas a partir del ángulo de incidencia del mismo.

Para estimar la presión de viento sobre las antenas se distinguen dos tipos de cálculo en función de la clase de antenas:

1. Las antenas crosspolares: se definen por su área de pantalla, con coeficiente de forma =1, y por tanto la presión ejercida tiene la expresión:

$$\left(\frac{V_{ms}^2}{16}\right) \cdot 1.0$$

de acuerdo con la modalidad de viento, posición etc.

2. Las antenas parabólicas se definen por su área frontal, con factores de carga variables dependiendo de la dirección del viento respecto al eje de las mismas. Se consideran las antenas en la posición local más desfavorable con

relación a los vientos normal y diagonal a la estructura. Los factores de carga aplicados son:

$$\text{Factor carga axial : } C_a = \frac{F_a}{A * V^2}$$

$$\text{Factor carga normal: } C_s = \frac{F_s}{A * V^2}$$

donde: F_a y F_s son las cargas aplicadas al eje de la parábola
 A es el área frontal de la parábola en m^2
 V es la velocidad de viento en km/h.

A la hora de especificar los distintos tipos de antenas parabólicas que pueden colocarse en estas estructuras (ya que el cálculo de sus coeficientes de transferencia de incidencia de viento a la estructura es diferente) el sistema se ciñe a los modelos de Andrews que están internacionalmente comprobados y aceptados. Se eligen:

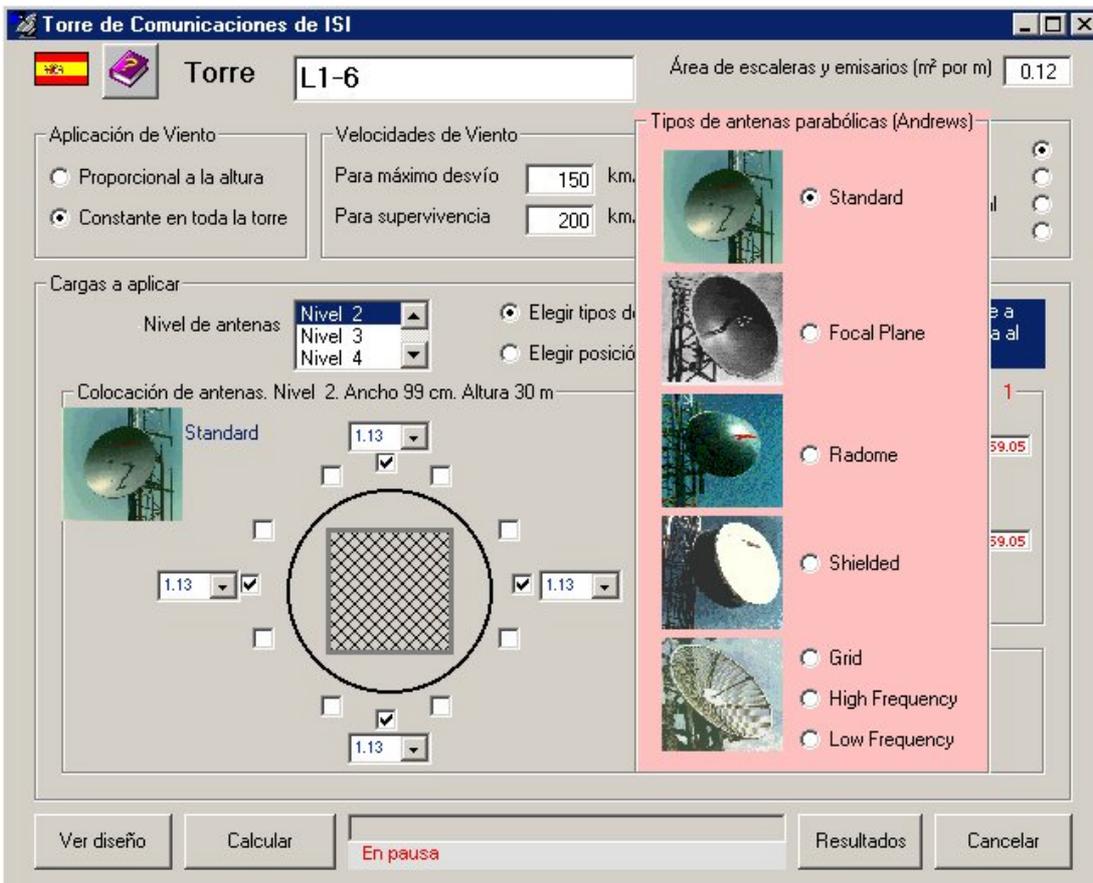
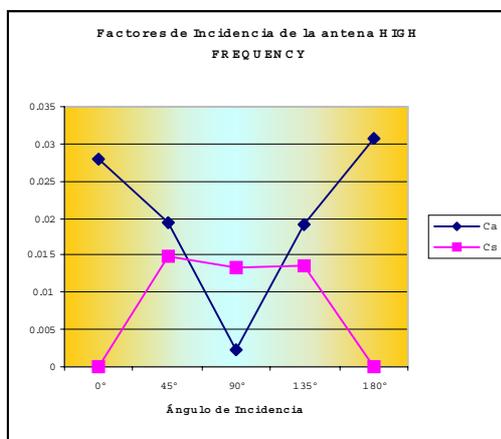
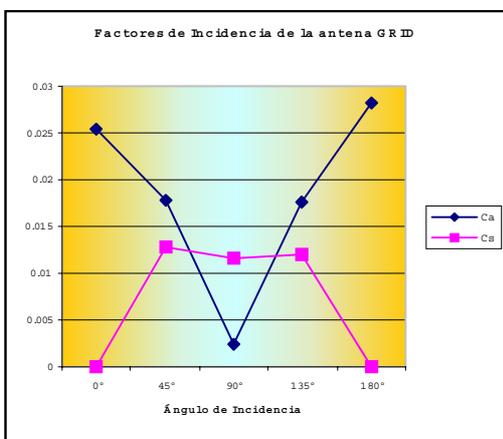
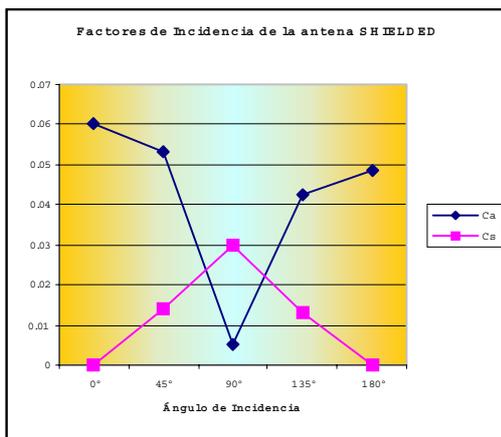
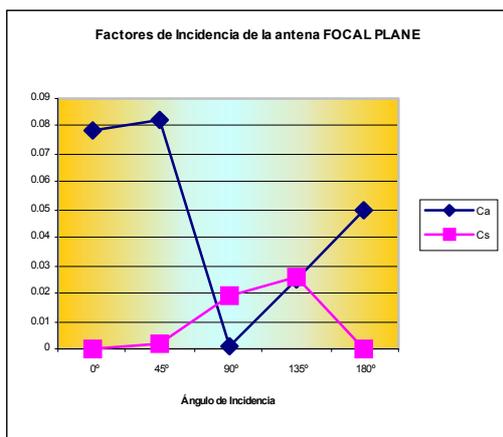
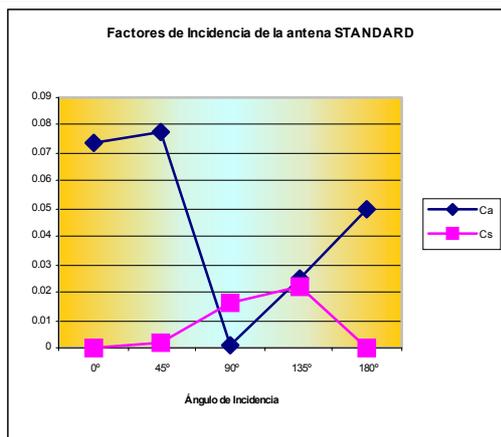
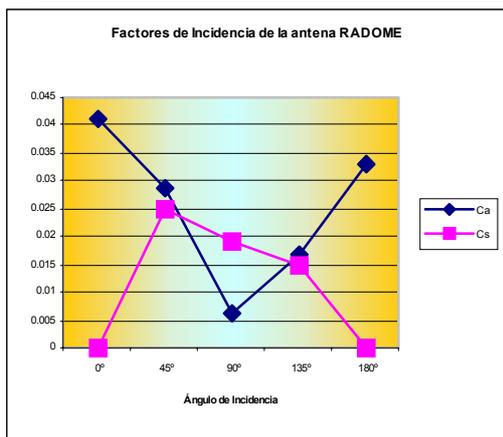


Ilustración 7-7. Elección de Antenas

A continuación se aprecia el análisis de los datos de cálculo de los factores Ca y Cs para cada tipo de antena y sus gráficos correspondientes (datos extraídos de AN00):

	Standard		Focal Plane		Radome		Shielded		Grid		High Frequency		Low Frequency	
	Ca	Cs	Ca	Cs	Ca	Cs	Ca	Cs	Ca	Cs	Ca	Cs	Ca	Cs
0°	0.0738	0	0.0786	0	0.0411	0	0.0601	0	0.0255	0	0.02786	0	0.02547	0
45°	0.07752	0.00232	0.08255	0.00232	0.0288	0.0249	0.0534	0.0139	0.0179	0.0128	0.01945	0.01475	0.01785	0.01283
90°	0.00056	0.01636	0.00056	0.01934	0.0063	0.0193	0.0052	0.0297	0.0024	0.0115	0.00226	0.01326	0.00242	0.01153
135°	0.02482	0.02175	0.02482	0.0257	0.0167	0.0149	0.0427	0.0129	0.0177	0.0119	0.01925	0.01369	0.01766	0.0119
180°	0.05019	0	0.05019	0	0.0329	0	0.0483	0	0.0283	0	0.0308	0	0.02826	0



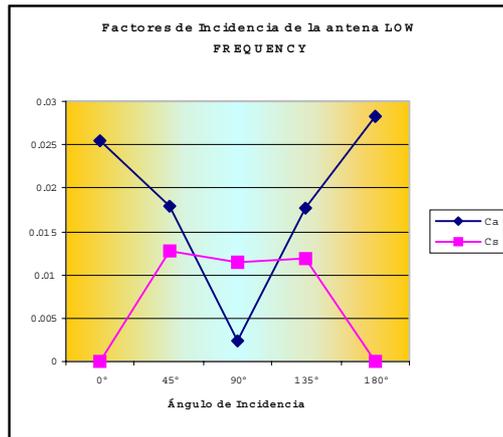


Ilustración 7-8. Gráficos para Antenas Normalizadas

7.7.4 Hipótesis de cargas a aplicar

Para el cálculo de las fuerzas han de definirse las siguientes hipótesis de carga o situaciones límite de estudio:

1. Condición de Máximo Desvío: es aquella situación donde el empuje de viento sobre la torre es el máximo permisible de manera que por debajo del mismo los desplazamientos de la torre no deben provocar que sus antenas pierdan la señal de ondas que reciben y transmiten.
2. Condición de Supervivencia: viento máximo soportado por la torre antes de fallar estructuralmente.

Ilustración 7-9. Entrada de Velocidad de Viento

En ambas condiciones hemos de estudiar la aplicación del viento en dirección lateral y diagonal, por tanto se obtiene el siguiente cuadro de condiciones:

Ilustración 7-10. Condiciones de carga

7.7.5 Cálculo de fuerzas sobre la estructura

A partir de aquí se obtiene, por ejemplo, que si aplicamos la siguiente configuración de antenas de tipo estándar sobre un nivel de la torre:

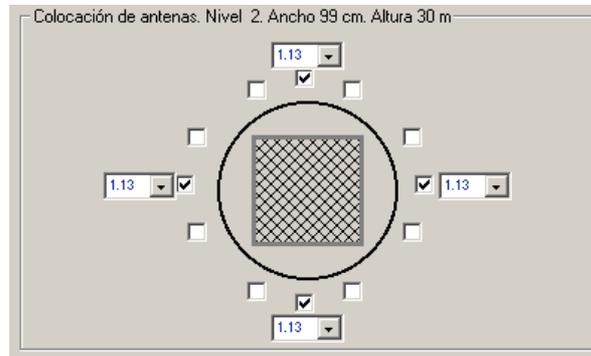


Ilustración 7-11. Colocación de antenas

los resultados de carga son (la flecha indica la incidencia del viento en la estructura):

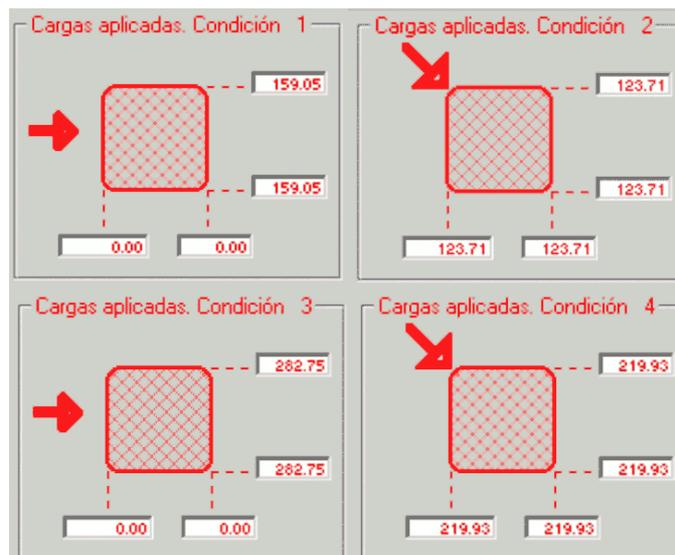


Ilustración 7-12. Cuadros de cargas

El máximo ángulo de desvío que las antenas soportan para no perder la señal es diferente en caso de antenas crosspolares (permiten mayor movimiento y por eso están situadas en la parte más alta de las torres) y de parabólicas.

Cada normativa nacional especifica cuales son los ángulos de desvío máximos permitidos respecto a la verticalidad de la torre y la perpendicularidad al suelo de las caras exteriores del nivel donde se sitúan las antenas. Sean, a modo de ejemplo:

Máximo ángulo de desvío	
Crosspolares	1 °
Parabólicas	0.6 °

Ilustración 7-13. Ángulos de desviación permitidos en antenas

Con todas estas variables de cálculo se puede interpretar el siguiente modelo de entrada de datos para el cálculo de las estructuras:

The screenshot shows the 'Torre de Comunicaciones de ISI' software interface. At the top, it displays 'Torre L1-6' and 'Área de escaleras y emisarios (m² por m) 0.12'. The interface is divided into several sections:

- Aplicación de Viento:** Radio buttons for 'Proporcional a la altura' and 'Constante en toda la torre'.
- Velocidades de Viento:** Input fields for 'Para máximo desvío' (150 km/h) and 'Para supervivencia' (200 km/h).
- Condiciones a aplicar:** Four checked checkboxes: '1. Viento máx. desvío transversal', '2. Viento máx. desvío diagonal', '3. Viento supervivencia transversal', and '4. Viento supervivencia diagonal'.
- Cargas a aplicar:** A dropdown menu for 'Nivel de antenas' (set to 'Nivel 2'), radio buttons for 'Elegir tipos de antenas' and 'Elegir posición de cargas', and a text box for 'Datos de antenas en base a superficie (en m²) expuesta al viento'.
- Colocación de antenas:** A diagram showing a central antenna grid with four '1.13' input fields and checkboxes for placement.
- Cargas aplicadas. Condición 1:** A diagram showing a red grid with a red arrow pointing to it, and numerical values: 159.05, 159.05, 0.00, and 0.00.
- Máximo ángulo de desvío:** A small version of the table from Illustration 7-13.

At the bottom, there are buttons for 'Ver diseño', 'Calcular', 'En pausa', 'Resultados', and 'Cancelar'.

Ilustración 7-14. Entrada de datos para torres de telecomunicaciones

La nomenclatura que siguen las torres, como es aquí en el ejemplo la L1-6, especifica cuantos tramos de la torre multiusos se están utilizando (desde el 1 al 6). Con este sistema se consigue realizar un estudio exhaustivo de cada torre obteniendo datos referentes a:

- Cuánto se desplaza en cada condición
- Cómo se desvía de la verticalidad
- Qué fuerzas soporta cada pieza, o bien

- Cómo son las reacciones en el suelo (este factor es de suma importancia, ya que muchas de las torres de comunicación están situadas en las azoteas de los edificios y es importante definir si el punto donde se va a colocar resistirá o no los desplazamientos de la estructura).

A la hora de transferir las cargas aplicables a los nudos de la estructura en sus posibles 6 niveles de antenas se calcula, de forma previa, las siguientes 4 situaciones posibles de carga:

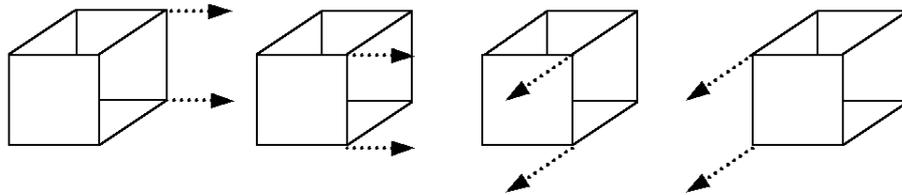


Ilustración 7-15. Aplicación de las cargas sobre las caras de la torre

En cada una de estas condiciones se aplican factores de $0.5 + 0.5 \text{ Kg.}$ y así se obtiene el factor de repercusión unitario para la combinación de cargas. Al ser 4 posibilidades por 6 niveles serán necesarios 24 nuevos factores de repercusión o situaciones de carga unitaria distintas para cada barra.

En resumen las condiciones estudiadas son:

- ❑ 1ª Presión de viento unitario y constante en toda la torre en sentido lateral
- ❑ 2ª Presión de viento unitario y constante en toda la torre en sentido diagonal
- ❑ 3ª Presión de viento unitario y proporcional a la altura en sentido lateral
- ❑ 4ª Presión de viento unitario y proporcional a la altura en sentido diagonal
- ❑ 5ª Opacidad de cada nivel con viento constante en toda la torre
- ❑ 6ª Opacidad de cada nivel con viento proporcional a la altura
- ❑ 7ª Factor peso unitario aplicado de forma proporcional al peso superior soportado, y aplicado en sentido vertical sobre cada nivel.
- ❑ 8ª a 31ª, las 4 combinaciones de cada uno de los 6 niveles

¿Cómo combinar los resultados unitarios para obtener los globales?. Nos basamos en el siguiente resultado algebraico:

Teorema: La combinación lineal finita de desplazamientos producidos por las fuerzas unitarias es equivalente al desplazamiento global de la estructura

Demostración:

Sean las variables:

- **M** matriz de rigidez de la estructura, que sólo depende de su propia geometría y del material usado.
- **C** cargas aplicadas a la estructura
- C_i cargas unitarias aplicadas a la estructura $i \in \{1, \dots, n\}$
- **X** desplazamientos de los nudos en las 3 direcciones del espacio
- X_i desplazamientos producidos por las cargas C_i

Tenemos que resolver el sistema: $\mathbf{M} \cdot \mathbf{X} = \mathbf{C}$

En nuestro caso, como hemos resuelto todos los sistemas $\mathbf{M} \cdot \mathbf{X}_i = \mathbf{C}_i$ previamente, y $\mathbf{C} = \alpha_1 \mathbf{C}_1 + \alpha_2 \mathbf{C}_2 + \dots + \alpha_n \mathbf{C}_n$ siendo los α_i reales, entonces:

$$\mathbf{M} \cdot \mathbf{X} = \mathbf{C} = \alpha_1 \mathbf{C}_1 + \alpha_2 \mathbf{C}_2 + \dots + \alpha_n \mathbf{C}_n = \alpha_1 \mathbf{M} \mathbf{X}_1 + \alpha_2 \mathbf{M} \mathbf{X}_2 + \dots + \alpha_n \mathbf{M} \mathbf{X}_n$$

Sacando factor común la matriz M obtenemos:

$$\mathbf{M} \cdot \mathbf{X} = \mathbf{M}(\alpha_1 \mathbf{X}_1 + \alpha_2 \mathbf{X}_2 + \dots + \alpha_n \mathbf{X}_n)$$

e igualando nos queda que: $\mathbf{X} = \alpha_1 \mathbf{X}_1 + \alpha_2 \mathbf{X}_2 + \dots + \alpha_n \mathbf{X}_n$

que justifica que *el desplazamiento global de la estructura es igual a la combinación lineal de desplazamientos unitarios*

c.q.d. ■

Corolario: El mismo argumento es aplicable tanto a las fuerzas axiales como a las reacciones

Demostración: Es obvio ya que ambas se obtienen como combinación lineal sobre los desplazamientos. Sean las variables:

- **F** fuerzas axiales que soportan las barras de la estructura
- F_i fuerzas axiales al aplicarle las cargas C_i con $i \in \{1, \dots, n\}$
- **R** reacciones que transmite la estructura a sus 4 puntos de apoyo en el suelo
- R_i reacciones producidas al aplicarle las cargas C_i con $i \in \{1, \dots, n\}$

obteniéndose las relaciones:

$$\mathbf{F} = \alpha_1 \mathbf{F}_1 + \alpha_2 \mathbf{F}_2 + \dots + \alpha_n \mathbf{F}_n$$

$$\mathbf{R} = \alpha_1 \mathbf{R}_1 + \alpha_2 \mathbf{R}_2 + \dots + \alpha_n \mathbf{R}_n$$

c.q.d. ■

Por ello es posible diseñar cada torre con la combinación de antenas que se desee y rápidamente resolver el cálculo de sus desplazamiento, fuerzas axiales y reacciones. Aquí tenemos un ejemplo:

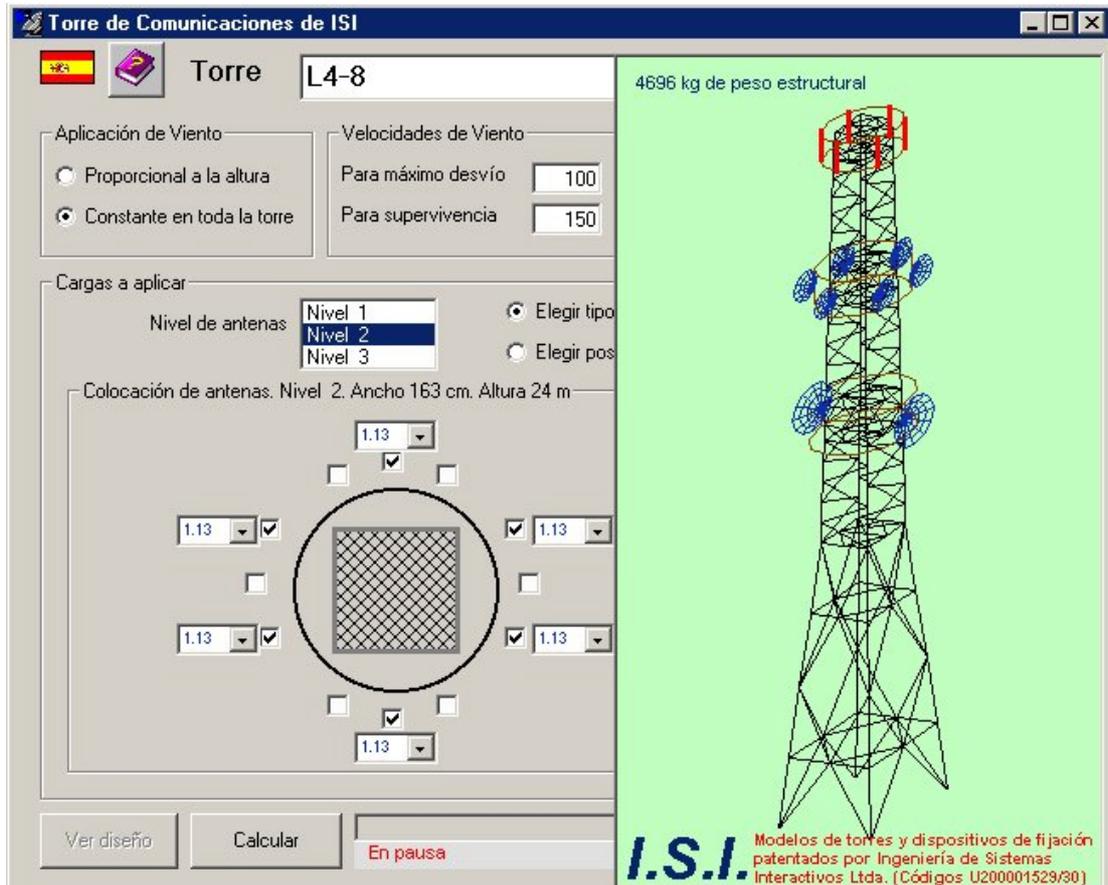


Ilustración 7-16. Presentación de antenas sobre la torre

en el que al realizar los cálculos sobre esta estructura se obtienen los resultados:

ANGULOS DE DESVÍO Y DESPLAZAMIENTOS

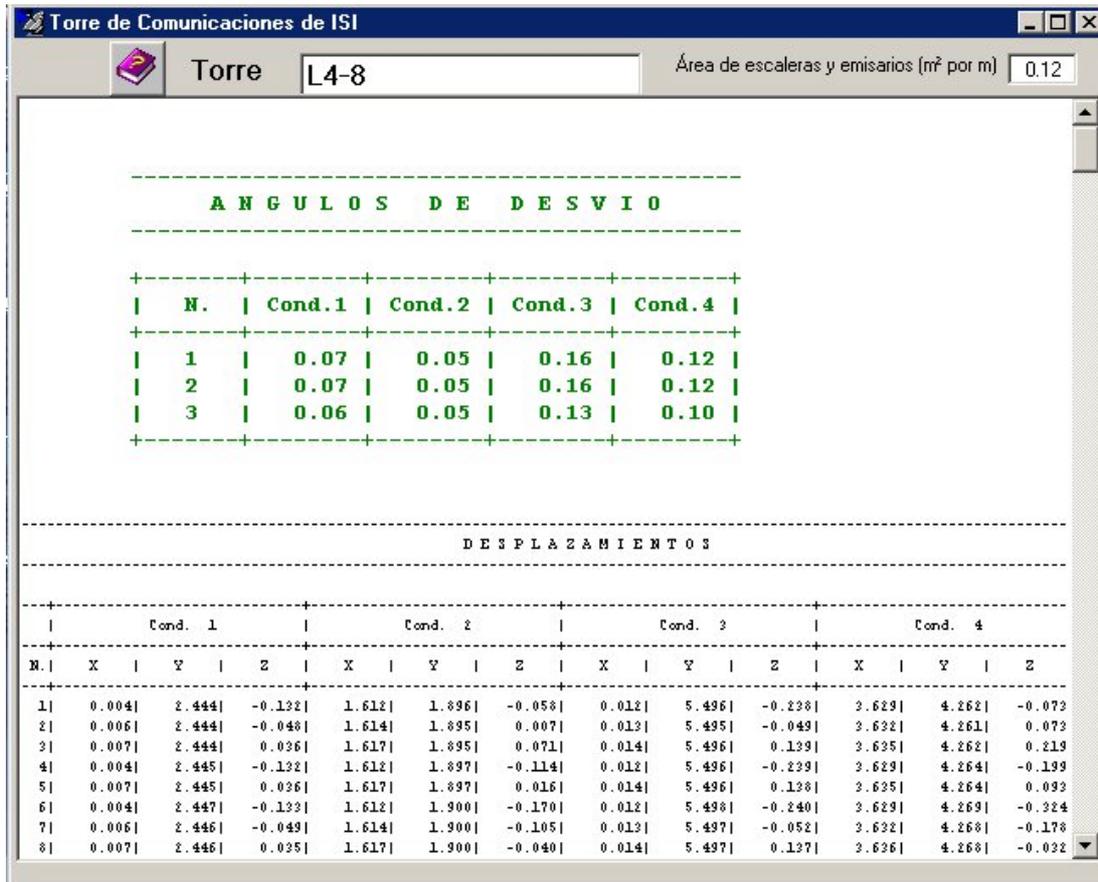


Ilustración 7-17. Ángulos de Desvío y Desplazamientos

El cálculo de los desplazamiento de cada nudo expresa las deformaciones máximas

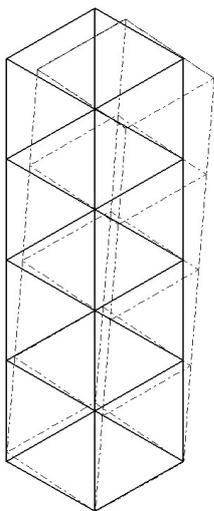


Ilustración 7-19. Desplazamiento 3D

de la estructuras (aumentadas convenientemente en los gráficos). A partir de ellos se miden las diferencias entre el máximo desplazamiento de los nudos inferiores y de los superiores de cada nivel de antenas y lo dividimos por la altura de dicho nivel, obteniendo la tangente de los ángulos buscados.

La aceptación del desplazamiento estructural supone el cumplimiento de una recepción y emisión de ondas adecuadas.

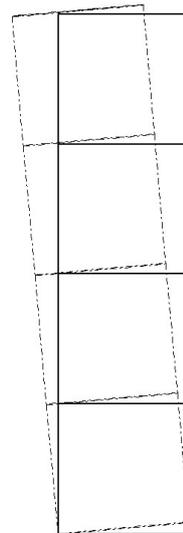


Ilustración 7-18. Desplazamiento Lateral

FUERZAS AXIALES

Bar n.	Nomenclatura I.S.I.	Condiciones de Carga				ESTADO ÚLTIMO (límite elástico)			
		de Carga 1	de Carga 2	de Carga 3	de Carga 4	Comp. admisible %	Tens. admisible %		
1	Tower Leg 1	0.01	0.00	0.01	0.00	- 46902.00	0.0%	39405.00	0.0%
2	Tower Leg 1	- 32.51	- 51.30	- 73.07	- 115.34	- 46902.00	0.2%	39405.00	0.0%
3	Tower Leg 1	- 33.00	0.34	- 74.16	0.84	- 46902.00	0.2%	39405.00	0.0%
4	Tower Leg 1	- 0.01	0.00	- 0.01	0.00	- 46902.00	0.0%	39405.00	0.0%
5	Tower Leg 1	0.01	0.01	0.01	0.02	- 46902.00	0.0%	39405.00	0.0%
6	Tower Leg 1	33.00	51.60	74.16	116.02	- 46902.00	0.0%	39405.00	0.3%
7	Tower Leg 1	32.51	- 0.64	73.08	- 1.52	- 46902.00	0.0%	39405.00	0.2%
8	Tower Leg 1	- 0.01	- 0.01	- 0.01	- 0.02	- 46902.00	0.0%	39405.00	0.0%
9	Diagonal LTR	25.15	19.58	56.52	42.99	- 6635.00	0.0%	6538.00	0.9%
10	Diagonal LTR	- 25.23	- 19.69	- 56.94	- 44.25	- 6635.00	0.9%	6538.00	0.0%
11	Diagonal LTR	24.38	18.97	54.79	42.63	- 6635.00	0.0%	6538.00	0.8%
12	Diagonal LTR	24.56	19.09	55.20	42.90	- 6635.00	0.0%	6538.00	0.8%
13	Diagonal LTR	- 24.56	- 19.09	- 55.20	- 42.89	- 6635.00	0.8%	6538.00	0.0%
14	Diagonal LTR	- 25.15	- 19.57	- 56.52	- 42.98	- 6635.00	0.9%	6538.00	0.0%
15	Diagonal LTR	- 24.38	- 18.98	- 54.79	- 42.64	- 6635.00	0.8%	6538.00	0.0%
16	Diagonal LTR	25.23	19.69	56.94	44.24	- 6635.00	0.0%	6538.00	0.9%
17	Diagonal LLO	0.09	20.06	0.20	45.14	- 6635.00	0.0%	6538.00	0.7%
18	Diagonal LLO	- 0.09	- 19.45	- 0.19	- 43.76	- 6635.00	0.7%	6538.00	0.0%
19	Diagonal LLO	0.10	- 19.95	0.22	- 44.88	- 6635.00	0.7%	6538.00	0.0%
20	Diagonal LLO	- 0.09	19.95	- 0.20	44.90	- 6635.00	0.0%	6538.00	0.7%
21	Diagonal LLO	0.09	- 19.34	0.19	- 43.52	- 6635.00	0.7%	6538.00	0.0%
22	Diagonal LLO	- 0.09	19.34	- 0.21	43.51	- 6635.00	0.0%	6538.00	0.7%
23	Diagonal LLO	0.09	19.46	0.21	- 43.78	- 6635.00	0.0%	6538.00	0.7%
24	Diagonal LLO	- 0.10	- 20.07	- 0.22	- 45.16	- 6635.00	0.7%	6538.00	0.0%
25	Frame LTR 1	35.20	27.20	80.04	62.05	- 16174.00	0.0%	14132.00	0.6%
26	Frame LTR 1	- 47.98	- 35.73	- 106.96	- 79.40	- 16174.00	0.7%	14132.00	0.0%
27	Frame LTR 1	71.98	57.56	162.70	130.26	- 16174.00	0.0%	14132.00	1.2%

Ilustración 7-20. Fuerzas Axiales

A la hora de estudiar las fuerzas axiales en cada barra, los datos son presentados bajo los siguientes conceptos:

- Número de la barra: numeración de la misma en el conjunto total de la estructura.
- Nomenclatura I.S.I: a ella se hizo referencia en el capítulo de *Cálculo Espacial* al hacer referencia a las barras principales y secundarias.
- Condición de Carga: cada una de las cuatro hipótesis estudiadas en las torres de telecomunicaciones.
- Estado Último: Máxima capacidad de las barras, tanto a tensión como a compresión, antes de quedar deformadas de forma permanente. Se tienen dos datos, el primero hace referencia a dicha capacidad y el segundo al porcentaje sobre este valor al que está sometida la condición peor de esta barra.

La aceptación de las fuerzas axiales supone la resistencia de la estructura a las peores condiciones de trabajo exigidas.

7.8 Aportaciones Realizadas

- Establecer el modelo matemático que subyace en el cálculo inverso a partir de la inversa de la matriz de rigidez
- Crear las estructuras computacionales de cable, cadena de aisladores, carga espacial, y coeficientes de las ecuaciones LNS
- Sintetizar el cálculo de las ecuaciones LNS de todas las barras de una estructura en cualquier condición posible y la búsqueda automática de las que limitan el diagrama de utilización
- Modelizar algebraicamente el cálculo inverso e introducir el concepto de *hiperestructura*
- Demostrar la equivalencia entre desplazamiento global de las estructuras y combinación lineal de los desplazamientos unitarios

7.9 Referencias Bibliográficas

[AA90] Arrillaga, J.; Arnold, C.P. “Computer Analysis of Power Systems”. John Wiley, 1990.

[AM01] Alberca, P.; Martín, D. “Métodos matemáticos: álgebra lineal y geometría”. Aljibe. Málaga. España. 2001

[AS88] American Society of Civil Engineers. “Guide for design of steel transmission towers”. Edit. ASCE. New York. EE.UU. 1988.

[Ch88] L. Checa, M. “Líneas de Transporte de Energía”, Marcombo Boixareu Editores, 1988

[DS87] Duncan, J.; Sarma, M. “Power System Analysis and Design”. PWS Publishers, 1987.

[FC96] Faulkenberry, L.M.; Coffey, W. “Electrical Power Distribution and Transmission”. Prentice-Hall, 1996

- [Go74] Godement, R. “Álgebra”. Tecnos. Madrid. España. 1974
- [Gö88] Gönen, T. “Electric Power Transmission System Engineering. Analysis and Design”. John Wiley, 1988.
- [IM90] IMEDEXSA (Industrias Mecánicas de Extremadura S.A.) . “Torres metálicas para líneas eléctricas de media tensión”. Manual. Cáceres. España. 1990.
- [IS97] ISI (Ingeniería de Sistemas Interactivos Ltda.) . “Computer integrated transmission towers system”. Manual. Málaga. España. 1997.
- [JM99] Joyanes, L.; Muñoz, A. “Microsoft, Visual Basic 6.0”. Mc Graw Hill. Universidad Pontificia de Salamanca. España. 1999.
- [MA01] Magnusson, P. C.; Alexander, G. C.; Tripathi, V. K.; Weisshaar, A. “Transmission lines and wave propagation”. Ed. CRC Press,2001
- [Me03] Merino, S. “Cálculo Inverso, Diagramas de Utilización”. Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería (para publicación en Diciembre) . 2003
- [Mo88] Moreno Clemente, J. “Cálculo de líneas eléctricas aéreas de alta tensión”. Manual. Málaga. España. 1988.
- [MV99] Mata, M.; Vinacua, A.; Brunet, P. “Inclusión de metodos de ingeniería inversa en sistemas CAS/CAD para la optimización del Proceso de diseño de productos” CEIG'99 IX Congreso Español de Informática Gráfica, Jaen. 1999
- [Ra73] Ras, E. “Teoría de líneas eléctricas”. Ed. Marcombo Boixareu, 1973
- [RA92] Rifaldi, A.; Arnera, P.; Giacopinelli, A. “Diseño de Líneas Eléctricas”. Apuntes de la Facultad de Ingeniería. Dpto. Electrotecnia Universidad Nacional de la Plata. Argentina. 1992

[RAT] Reglamento de líneas eléctricas aéreas de Alta Tensión

[RBT] Reglamento electrotécnico para baja tensión

[RCE] Reglamento sobre centrales eléctricas, subestaciones y centros de transformación

[Wa91] Wadhwa, C.L. “Electrical Power Systems”. Segunda Edición, John Wiley, 1991.

8. Aportaciones, Aplicaciones y Trabajo Futuro

Todo este trabajo ha tratado de dar “modo y orden” al diseño industrial de torres. Desde esta perspectiva concluimos que queda un largo camino por recorrer en paralelo al desarrollo de los sistemas de cálculo estructural y a las nuevas posibilidades que la matemática va abriendo en el mundo de la computación y de la inteligencia artificial. Por ello consideramos que este puede ser un buen punto de partida que vaya acogiendo esos nuevos horizontes y permitiendo la creación de industrias de mayor calidad y mejor gestión medioambiental.

8.1 Resumen final de aportaciones

Recogemos a continuación las aportaciones realizadas por el autor que se han desarrollado en este trabajo. El lector podrá comprender que algunas de ellas han sido presentadas en su correspondiente capítulo de forma resumida por encontrarse protegidas bajo patentes en el Sistema ISI de cálculo, diseño y fabricación de torres.

1) Datos de entrada

- a) Desarrollo del sistema central de procesos para la creación de un sistema experto en el diseño, cálculo y fabricación de torres
- b) Clasificación de los parámetros comunes y particulares para los diferentes modelos de torres
- c) Establecimiento de la numeración nodal para la unidad mínima de fabricación

2) Sistema espacial de cálculo

- a) Establecimiento del orden previo de la numeración de los nudos para la búsqueda del mínimo ancho de banda
- b) Implementación del algoritmo de resolución por bloques del sistema de ecuaciones desarrollado a partir del MEF
- c) Creación de la ecuación vectorial para la representación bidimensional de estructuras espaciales con posibilidad de giros, visualización de caras por separado y cambio de perspectivas
- d) Desarrollo las estructuras computacionales de nudo, barra, carga y panel

- e) Establecimiento del algoritmo de compresión de datos para la minimización del número de nudos en la estructura
- f) Definición de los conceptos de inestabilidad global y local y las ecuaciones que las limitan. Desarrollo del cono de desplazamientos
- g) Creación del algoritmo recursivo para el establecimiento del viento real que afecta a la torre y el diseño estructural que, junto a las cargas, lo soporte.

3) Algoritmos genéticos

- a) Introducción del concepto de clonación en la formulación de los algoritmos genéticos
- b) Implementación de un sistema de algoritmos genéticos para la minimización del ancho de banda de una matriz
- c) Análisis de las soluciones óptimas en modelos concretos mediante la formulación tradicional y mediante la formulación dinámica
- d) Definición del procedimiento a seguir para el diseño óptimo de torres

4) Minimización de datos en fábrica

- a) Establecimiento de la fórmula de aproximación de la línea imaginaria de cálculo a la real del desarrollo físico de la torre
- b) Desarrollo del algoritmo de búsqueda automática y unificación de gramiles en el diseño de las torres
- c) Creación de las estructuras computacionales de barra, chapa, taladro, corte y doblado
- d) Disminución del número de interferencias espaciales en la estructura y sus correspondientes cortes a través de modelos matemáticos más cercanos a la ubicación tridimensional y el perímetro real de las piezas
- e) Establecimiento del algoritmo de fabricación de una torre
- f) Desarrollo de las bases numéricas para la creación de chapas rectangulares y periféricas mínimas a partir de un conjunto de taladros en el plano

5) Análisis de Residuo Mínimo

- a) Planteamiento mediante programación lineal del problema y las restricciones para la minimización de residuos unidimensionales en la industria
- b) Creación, análisis, implementación y clasificación de los algoritmos:

- i) Ordenación Creciente
 - ii) Ordenación Decreciente
 - iii) Ordenación Alternada
 - iv) Llenado Progresivo Óptimo
 - v) En Banda
 - vi) En Cascada
- c) Modelización del problema de residuos en cualquier dimensión
 - d) Generalización de los algoritmos a los casos bidimensional y espacial

6) Cálculo Inverso y Diagramas de Utilización

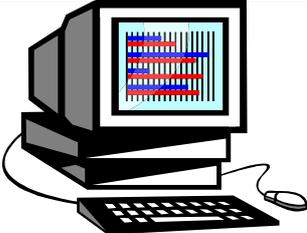
- a) Establecimiento del modelo matemático que subyace en el cálculo inverso a partir de la inversa de la matriz de rigidez
- b) Creación de las estructuras computacionales de cable, cadena de aisladores, carga espacial, y coeficientes de las ecuaciones LNS
- c) Sintetización del cálculo de las ecuaciones LNS de todas las barras de una estructura en cualquier condición posible y la búsqueda automática de las que limitan el diagrama de utilización
- d) Modelización algebraica del cálculo inverso e introducción del concepto de *hiperestructura*
- e) Demostración de la equivalencia entre desplazamiento global de las estructuras y combinación lineal de los desplazamientos unitarios

8.2 Aplicaciones de la teoría de torres

Durante el desarrollo de este trabajo, junto a la realización de decenas de ensayos a escala real en colaboración con la compañía ISI, han ido surgiendo ideas y proyectos a los cuales se ha tratado de dar solución, encontrando en ellos nuevos y apasionantes campos de aplicación de la teoría de torres. De la colaboración entre la empresa privada y la Universidad nacen los resultados aquí presentados.

8.2.1 Simulación de planta industrial

Para mejorar el control de producción en las plantas industriales, y basándose en el modelo teórico “Three Models System” desarrollado por Acacio Bericua para la compañía Allco Steel (Australia), fue estudiada la posibilidad de realizar un sistema de tres simuladores gráficos como observamos en el siguiente gráfico.

		
<p>Modelo 1. Los datos de la situación actual de carga de trabajo de las diferentes máquinas son presentados gráficamente en la pantalla. Ello permite al técnico de control tener una visión clara de la situación de la fábrica</p>	<p>Modelo 2. Una segunda computadora realiza una simulación de los cambios a realizar en la planta para acometer un nuevo proyecto. Aquí se estudian todas las necesidades de maquinaria y empleados y se prevén las saturaciones.</p>	<p>Modelo 3. La tercera estudia la posibilidad de introducir el segundo proyecto en mitad de la ejecución del primero sin colapsar la capacidad de la planta. Planteada la solución rescribe las nuevas órdenes de trabajo para modificar los trabajos que en ese instante se efectúan.</p>

Sistema de Tres Modelos (A.Bericua 1981)

Para llevar a cabo esta labor se dividió el trabajo en tres proyectos:

1º. Creación de la estructura de una base de datos que permitiese simular cualquier planta industrial (realizado).

2º. A partir de esta base de datos se creo un simulador de planta industrial que permite gráficamente construir cualquier planta industrial, marcar las relaciones existentes entre las máquinas y el proceso de carga de una a otras (realizado).

3º. Estudio de los algoritmos necesarios para optimizar la carga de la planta industrial (en proyecto)

8.2.2 Cálculo de Torre Delta

En colaboración con el Dpto. de Transporte Eléctrico de la EUP y la compañía ISI, y con motivo del Campeonato del Mundo de Esquí en Sierra Nevada (1994), fue realizado el proyecto de cálculo de una torre terminal tipo Delta para la compañía Sevillana de Electricidad (actualmente Endesa). La dificultad de este proyecto radicaba en el hecho de que, cada vez que finalizaba el deshielo, la torre, situada en un lugar de mucha pendiente, caía al suelo. [C. Cuadra y M. A. Mateo, 1993]

El problema nacía de no aplicar cargas horizontales sobre la torre cercanas al nivel del suelo que simularan las presiones producidas bajo la nieve por el arrastre de piedras, lodos e hielo de las corrientes de agua que se originan. Una vez aplicadas estas cargas se comprobó que piezas debían ser modificadas, fueron diseñadas y corregidas, no volviendo a repetirse las caídas.

8.2.3 Método de los Elementos Finitos

Acerca de este tema se han desarrollado principalmente tres proyectos, aunque su aplicación se ha llevado a cabo en todos los cálculos de torres realizados.

1º. Un texto sobre iniciación al cálculo mediante el MEF y su aplicación en el sistema SAP para su conocimiento por los alumnos de ingeniería. [J.F. Ruiz García, 1995]

2º. Un segundo texto sobre cálculo de edificios mediante el MEF y su aplicación mediante los sistemas ANSYS y CYPECAD. [J. M. Rivero, 2000]

3º. Un visualizador espacial de estructuras que, una vez efectuado el MEF, dotaba a las mismas de una coloración adecuada (visualización de esfuerzos Von Mises) para observar que zonas trabajaban a un mayor rendimiento en función de las distintas hipótesis de carga [A. Pacetti, 1999]

8.2.4 Colocación de sensores contra incendios

El problema que planteamos originalmente fue la óptima colocación de sensores térmicos en la parte alta de las TTE que, minimizando el número de ellos (debido a su alto coste) maximizara el área protegida contra incendios. Estos sensores, desarrollados por la industria Bazán, funcionan vigilando un área circular alrededor de los mismos donde detectan cualquier cambio local de temperatura superior a cinco grados respecto de su entorno próximo. [M. Fernández, 1999]

El sistema gráfico desarrollado permite la entrada de las coordenadas geográficas de cada torre y fijar el porcentaje mínimo de área a cubrir o el número máximo de sensores que pueden ponerse por limitaciones presupuestarias. A partir de aquí dispone los sensores u optimiza el área a vigilar. Matemáticamente es el problema de los conjuntos compactos, es decir, fijar mediante círculos un subrecubrimiento finito.

Es posible aplicar este sistema para la ubicación de las torres de telefonía móvil con un máximo de cobertura (actualmente en proyecto).

8.2.5 Algoritmos de minimización de residuos

A partir de las teorías desarrolladas en el capítulo de minimización de residuos fueron desarrolladas dos aplicaciones:

1º Un sistema de minimización de materiales necesarios para la colocación de sistemas de riego por goteo. A partir de la ubicación de los árboles y del punto de suministro de agua el sistema busca la mejor solución que permita regar todos y cada uno de los árboles usando el mínimo de material. Sus aplicaciones en el diseño de circuitos integrados es también importante. [T. Rivas, 1998]

2º Un simulador de optimización de carga de camiones, en colaboración con los técnicos de la compañía Mayoral, donde, a partir de las dimensiones de las cajas que se van a almacenar y de los camiones disponibles, establece la forma de ordenarnos

con mínimo desaprovechamiento, calculando a posteriori el volumen reaprovechable por parte de los operarios. [I. Medina, 2001]

8.2.6 Torres de ocio

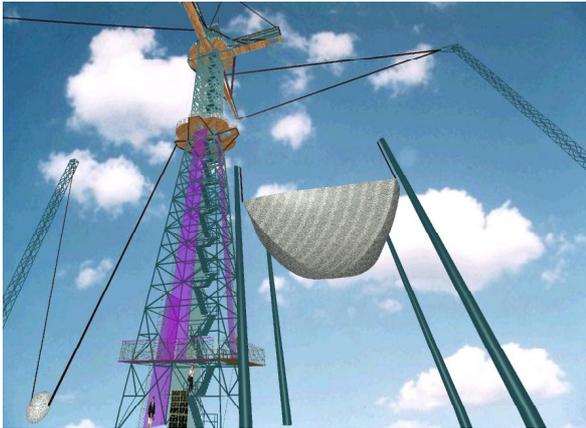


Ilustración 8-2. Torre para deportes extremos

Actualmente se desarrollan en la compañía ISI torres para deportes de riesgo. En este campo el uso de tecnologías que permitan salvaguardar la seguridad de los usuarios es de suma importancia. La innovación nace del hecho de que aquí las cargas son menores pero los factores de seguridad, que en TTE están entre 1.1 y 2 normalmente, se elevan a 10, y las

cargas pasan a ser golpes dinámicos, que exigen mayor precisión de cálculo y de detalles de fabricación.

Además de estos deportes hemos de resaltar el diseño de torres emblemáticas, como son:



Ilustración 8-1. Salto desde torre

- Dos torre de 330 metros de altura para atravesar los 2.5 kilómetros de ancho del río amarillo en China.
- La torre de televisión portuguesa de 120 metros de altura, desarrollado para la compañía Sermague.

Un diseño importante ha sido el de la *torre de Málaga*, de 230 metros de altura, para comercio, ocio y comunicaciones. Veamos alguna de sus características:

Cimentación: La torre se apoya en una plataforma rectangular de hormigón que permite el reparto uniforme de los empujes laterales y su carga. Mediante 3 anillos interconectados genera una corona desde la que se profundiza por pilotaje en busca de un suelo adecuado.

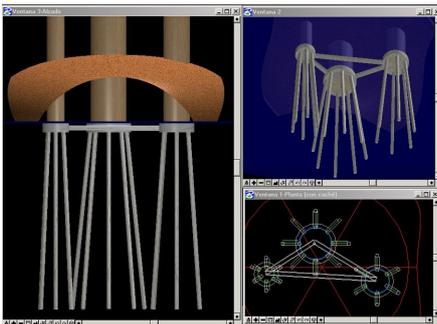


Ilustración 8-3. Diseño de Cimentación para la Torre de Málaga

La cúpula de base: Se trata de una estructura de 15 a 20 m. de altura y 60 m. de diámetro en su base que albergaría cinco plantas de servicios en su interior.

En el diseño podemos observar tres mástiles cilíndricos de alturas, diámetros y disposición variables y cuatro plataformas de servicios que, de abajo a arriba o de mayor a menor tienen las

siguientes funciones:

Primera plataforma: situada a 60 m. de altura desde la base tiene tres plantas de altura. Con una superficie total de 2.500 m² encontramos los servicios de restaurantes, tiendas, miradores y salas. Podemos considerarla como la planta lúdica destinada al uso de los ciudadanos en general.

Segunda plataforma: situada a 115 m. de altura con dos plantas y una superficie total de 1200 m². En ella encontramos los servicios de logística. Se controla el comercio, tráfico, transporte, servicios de ambulancias, radio-taxis, iluminación, etc.

Tercera plataforma: situada a 150 m. de altura y con 1700 m² de superficie utilizable. Encontramos los servicios de telefonía móvil y fija, cadenas de televisión locales, nacionales e internacionales y radio, junto con las comunicaciones intercontinentales con África.

Cuarta y última plataforma: situada a 190 m. de altura y con una superficie de 300 m² repartidos en cuatro plantas se trata de un espacio destinado a mirador de Málaga.

Otras características a resaltar serían:



Ilustración 8-4. Torre de Málaga

1. **Sistemas de generación de energía** compuestos por:
 - a. Generadores eólicos de electricidad a partir de turbinas situadas en la parte superior de cada uno de los cilindros.
 - b. Placas de energía solar, para los sistemas de calefacción y calentamiento de agua situadas en la parte superior de cada una de las plataformas, y placas fotovoltaicas, que forman toda la cúpula de base para la generación de electricidad.

2. **Sistemas de emergencia** compuestos por:
 - a. Depósitos de agua ubicados en la parte inferior de cada una de las plataformas como sistema anti-incendios y de consumo propio.
 - b. Pasillos a modo de contrafuertes para unir los mástiles entre sí y pasar de uno a otro en paralelo en caso necesario.

Este diseño requiere el uso del MEF con herramientas como NASTRAN o ANSYS.

8.3 *Patentes Desarrolladas*

Durante el desarrollo de este trabajo se han creado diversas patentes en la Oficina Española de Patentes y Marcas del Ministerio de Ciencia y Tecnología donde se aplican algunos de los resultados presentados en este trabajo:

- **Modelo de Utilidad 1046411-U200001530. Dispositivo de fijación de antenas en torres**
 - Inventores: Acacio Bericua, Salvador Merino y Robert Bericua.
 - Publicada el 16 de Diciembre de 2000

- **Modelo de Utilidad 1046512-U200001529. Torre Multiusos**
 - Inventores: Acacio Bericua, Salvador Merino y Robert Bericua.
 - Publicada el 1 de Enero de 2001

- **Patente 1053866-U200300262. Dispositivo protector de Antenas de Radiofrecuencia**
 - Inventores: Acacio Bericua, Salvador Merino y Robert Bericua.
 - Publicada el 16 de Mayo de 2003

- **Modelo Industrial I0151835. Torre (Torre de Málaga)**
 - Inventores: Acacio Bericua, Salvador Merino y Robert Bericua.
 - Publicada el 16 de Septiembre de 2002

8.4 Trabajo futuro

Exponemos a continuación nuevas líneas de investigación:

Hasta aquí hemos definido las líneas de una herramienta que responda a las necesidades actuales del trabajo en TTE. Una vez realizado el proyecto y construido el prototipo hemos de llevarlo a la estación de ensayos para la simulación de las condiciones que elija el cliente sobre la torre. Este procedimiento nos va a permitir verificar si realmente la torre responde a nuestras conclusiones o, si a pesar de nuestras suposiciones de cálculo, existen puntos críticos que hacen que falle la estructura.

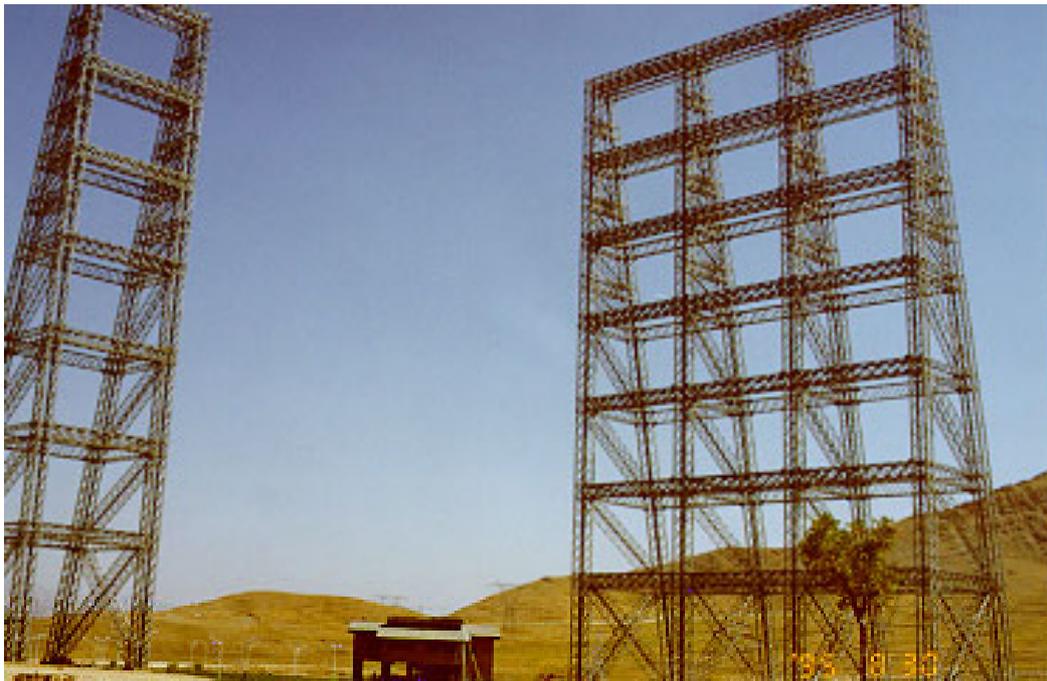


Ilustración 8-5. Estación de ensayos con su centro de control abajo

El ensayo consta de las siguientes fases :

- **Protocolo de ensayo:** Definición de las condiciones que se van a ensayar y el orden de las mismas.

- **Colocación de tiros:** Sobre el prototipo se enganchan los diferentes cables que, a partir de mecanismos hidráulicos, tirarán de la estructura a modo de conductores en el campo. Como el viento sobre la torre no se puede simular, elegimos puntos o niveles sobre los que colocamos tiros, con sistemas de cédulas de carga que nos permitan comprobar la tensión que está soportando cada uno, de forma que simulen que todo el viento sobre una porción de torre se concentrara en ellos.



Ilustración 8-6. Colocación de cables y cédulas de carga

- **Realización del ensayo:** Se comienza cada condición haciendo que los cables o tirantes que simulan pasen, de una tensión inicial cero, hasta llegar a la estipulada. Esto se ha de hacer progresivamente de manera que todos los tirantes vayan a la vez del 0% al 100% de su tensión final. En el 100% de cada condición se espera un cierto tiempo (25 ó 30 segundos) y se pasa a preparar la siguiente. Una vez realizada la última y aceptado el 100%, se continua tirando (101%, 102%,...) hasta su destrucción. Esto último se hace porque si pasamos de un cierto límite (normalmente el 115%) nuestra estructura podría estar sobredimensionada, por lo cual puede darse el caso de que el cliente no acepte el proyecto y debamos aligerar la torre.



Ilustración 8-7. Controles de una estación de ensayos

Como podemos imaginar se trata de un procedimiento muy costoso y a veces difícil de explicar cuando ocurre un fallo (un tornillo de distinta calidad puesto por error, un material no homogéneo, una fisura microscópica en el borde de un taladro, una soldadura mal hecha, etc.) y este no dependía del cálculo. Evidentemente estos

errores nunca se van a poder resolver matemáticamente, sólo estimar estadísticamente su posibilidad de existencia, de ahí la aplicación de factores de seguridad (1.1, 1.2 ó 1.5 habitualmente) sobre las condiciones estructurales.

8.4.1 Proceso de simulación virtual

Podemos plantearnos el realizar una simulación matemática del ensayo que nos permitiese detectar cualquier error de diseño no contemplado. Para ello nos bastará con hacer uso de los coeficientes de repercusión desarrollados para las ecuaciones LNS. Tomemos los coeficientes del archivo:

Torre.COE

Definimos el protocolo a ensayar. Como los coeficientes nos dan la repercusión de una carga unitaria sobre un punto en todas las barras de la torre, definimos que puntos y direcciones son las que voy a cargar y hasta que valor máximo (100%).

Comenzamos el ensayo, incrementando todas las cargas desde el 0% al 100%, observando espacialmente las modificaciones y desplazamientos sobre la estructura:

- Para las fuerzas soportadas es conveniente ir coloreando las barras del diseño espacial en base al porcentaje de su carga de saturación al que se está llegando.
- Para los desplazamientos de los nudos, como tenemos su desplazamiento final por el cálculo original, hacemos que siga la dirección desde sus coordenadas originales a las finales desplazadas proporcionalmente al porcentaje de ensayo en que nos encontremos. En este punto es conveniente multiplicar por un factor (p.ej. 10) para ser observables los desplazamientos, ya que en una estructura de 50 mts. de altura una variación de 10 cm. es inapreciable.

8.4.2 Aplicaciones en costes, tiempos y seguridad

Este procedimiento nos permitirá visualizar gráficamente los cambios y repartos de cargas en la estructura durante los ensayos, siendo muy importante desde el punto de vista pedagógico en la formación de los técnicos en este campo. Con él no tendremos

que asumir los costes que suponen los ensayos de prototipos (esto no significa que dejen de hacerse, ya que hemos señalado que existen otros errores paralelos sólo probabilísticamente estimables) en diseñadores con menor experiencia y así les permitimos que aceleren su aprendizaje.

Otra de las aplicaciones de esta simulación será la de ensayar condiciones que pueden no establecerse en el cálculo pero que pudiesen ocasionar fallos en otras situaciones, como por ejemplo, en el montaje de la estructura en el campo. El hecho de que mientras montamos las estructuras haya una cruceta donde tengamos al personal que está izando los conductores para colocarlos en su lugar mientras en su cruceta simétrica aún no existe carga podría provocar la inestabilidad o sobrecarga de alguna barra que, una vez colocados los dos conductores, habría compensado tensión y compresión y por ello en su máxima resistencia no aparecería la posibilidad de sólo tensión o sólo compresión.

A pesar de que estas condiciones de montaje deben tenerse en cuenta a la hora de colocar las condiciones iniciales, nos puede servir de ejercicio el observar y prevenir errores donde pudiesen estar involucradas vidas humanas.

También, mediante este mecanismo podemos analizar si un error de ensayo se ha producido por un fallo en la estación. En algunas ocasiones se producen bloqueos en alguno de los tiros hidráulicos, lo que descompensa la aplicación del porcentaje de cargas, o se rompe un cable en pleno ensayo, con el consiguiente golpe dinámico sobre un punto de la estructura (esto es más peligroso por la descarga de energía que se produce de forma instantánea).

9. Anexo I. Teoría Algorítmica

Para comenzar, podríamos preguntarnos ¿Qué es un algoritmo?. Esta palabra tiene su origen en el nombre del autor persa Muhammad ibn Musa al Juwarizmi (aprox. 780-850 d.C.), y se entiende como un método especial de resolución para un cierto tipo de problemas. Pero esta palabra tiene una significación especial en la teoría computacional, donde algoritmo viene a referirse a un método preciso utilizable por un ordenador para la solución de un problema.

9.1.1 Eficiencia de los algoritmos

Cuando tenemos que resolver un problema, nos interesa, obviamente, no sólo encontrar un algoritmo, sino además un algoritmo eficiente. Esto requiere disponer de métodos para determinar la eficiencia de un algoritmo. La aproximación empírica (o a posteriori) consiste en programar los diversos algoritmos disponibles para nuestro problema y hacerlos competir en una computadora, tomando medidas en diferentes instantes y comparándolas. La teórica (o a priori) consiste en determinar matemáticamente la cantidad de recursos (tiempo de ejecución, espacio de memoria, etc.) necesarios para cada algoritmo, como una función del tamaño de los casos considerados.

El tamaño de un caso x , que notamos por $|x|$, corresponde formalmente al número de bits necesarios para representar dicho caso en un ordenador usando un código predefinido y razonable.

Para clarificar nuestro análisis, sin embargo, solemos usar el vocablo *tamaño* que nos viene a indicar aquel entero que mide el número de componentes (o bien alguna medida de cada componente) en cada caso. Por ejemplo, cuando estamos realizando una ordenación de un caso formado por n componentes, generalmente consideramos que es de tamaño n , a pesar de que cada componente pueda poseer más de un bit al ser representada en una computadora. Cuando tratemos de problemas numéricos, daremos en ciertas ocasiones la eficiencia de nuestros algoritmos en términos del

valor del caso considerado más que de su tamaño, que sería el número de bits necesarios para representar este valor en código binario.

La ventaja de la aproximación teórica es que nunca depende ni del ordenador usado ni del lenguaje de programación, ni siquiera de la habilidad del programador. Ello elimina el tiempo que hubiese sido necesario para programarlos y el tiempo de máquina usado para comprobarlos. También permite estudiar la eficiencia en el caso de muestras sin restricción de tamaño. Este no es el caso de la aproximación empírica, normalmente, ya que por cuestiones prácticas nos obliga a probar nuestros algoritmos en caso de tamaños moderados. Este último punto es particularmente importante si consideramos que un nuevo algoritmo puede sólo mejorar a su predecesor cuando ambos se usan en casos de gran tamaño.

También es posible analizar un algoritmo usando una aproximación híbrida, donde la forma de la función que describe la eficiencia del algoritmo es determinada teóricamente y entonces, los parámetros numéricos requeridos son determinados empíricamente para un programa y máquina particulares, normalmente por algún mecanismo de regresión. Esta aproximación permite que sean hechas predicciones sobre lo que tardará en resolver casos de mayor tamaño que los probados. Pero si tal extrapolación se hace sobre la base de pruebas empíricas únicamente, ignorando cualquier consideración teórica, puede conducir, no sólo a imprecisiones, sino también a incorrecciones.

En este punto es natural preguntarse qué unidad debe ser usada para expresar la eficiencia teórica de un algoritmo. Si el recurso analizado es el tiempo de ejecución, no es cuestión de expresarlo en segundos ya que no tenemos un ordenador estándar al que todas las medidas se refieran. En respuesta a esta cuestión hemos de basarnos en el principio de invariancia, que nos dice que dos implementaciones de un mismo algoritmo no difieren en eficiencia salvo alguna constante multiplicativa. Es decir, si dos implementaciones tardan $t_1(n)$ y $t_2(n)$ segundos, respectivamente, para resolver un caso de tamaño n , entonces siempre existe una constante positiva c tal que $t_1(n) \leq ct_2(n)$ para un n suficientemente grande. Este principio siempre es verdad para cualquier ordenador usado, indiferentemente del lenguaje de programación empleado y de la habilidad del programador. Por ello, un cambio en la computadora

nos puede ayudar a resolver algo (por ejemplo 10 ó 100 veces más rápido), pero sólo un cambio del algoritmo nos permitirá incrementar sustancialmente el tamaño de los casos que pueden ser resueltos.

Ciertos órdenes ocurren tan frecuentemente que acaban teniendo nombre propio. Por ejemplo, si un algoritmo utiliza un tiempo de orden n , donde n es el tamaño del caso que va a ser resuelto, decimos que tiene tiempo lineal o también que es un algoritmo lineal. Análogamente, un algoritmo es cuadrático, cúbico, polinómico o exponencial si el tiempo es de orden n^2, n^3, n^k, c^n respectivamente, donde k y c son constantes apropiadas.

Otros recursos necesarios para la ejecución de los algoritmos, como el espacio de memoria usado, pueden ser determinados teóricamente de una manera similar. Es interesante estudiar la mejor relación entre tiempo de ejecución, aproximación al resultado óptimo y espacio de memoria usado, como veremos con más detalle en la comparación de algoritmos que nacieron para resolver nuestro problema en cuestión.

9.1.2 ¿Porqué son necesarios algoritmos eficientes?

Dado que hoy en día los equipos son cada vez más y más rápidos, parece una pérdida de tiempo trabajar en el diseño de algoritmos más eficientes. ¿No sería más simple esperar a una nueva generación de computadoras?. Como vamos a ver en este apartado, esto no es verdad. Supongamos, para ilustrar este argumento, que en la resolución de un problema particular contamos con un algoritmo exponencial, y una computadora es capaz de resolver un caso de tamaño n en $10^{-4} \cdot 2^n$ segundos. El programa resuelve un caso de tamaño 10 en una décima de segundo. Para un tamaño 20 tardará aproximadamente dos minutos y para un caso de tamaño 30 no será suficiente con un día de trabajo. Suponiendo que el ordenador fuera capaz de trabajar sin interrupción durante un año, sólo sería capaz de resolver un caso de tamaño 38.

Como necesitamos resolver casos mayores que éstos, decidimos comprar una nueva computadora 100 veces más rápida que la primera. Con el mismo algoritmo podemos resolver un caso de tamaño n en $10^{-6} \cdot 2^n$ segundos. Parece que hemos rentabilizado nuestra inversión, sin embargo, si hacemos trabajar la máquina durante un año, no

podemos resolver apenas casos de tamaño 45. En general, si fuimos capaces de resolver casos de tamaño n con la primera máquina, resolveremos casos de a lo sumo $n+7$ con la segunda, en el mismo tiempo.

Supongamos entonces que decidimos investigar en nuevos algoritmos y encontramos para nuestro problema un algoritmo cúbico. Supongamos que en la máquina original resuelve casos de tamaño n en $10^{-2} \cdot n^3$ segundos. En un día podemos resolver casos de tamaños mayores de 200 y en un año llegaremos a 1500.

Sin embargo, como podemos observar en el gráfico el segundo algoritmo no siempre debe ser usado. Por ejemplo, tarda 10 segundos en resolver casos de tamaño 10, mientras que el anterior lo hacía en 1 décima de segundo, por lo que en estos casos es 100 veces más lento. Por lo tanto hemos de usarlo en casos de tamaño superior a 20. Evidentemente hemos de combinar los dos algoritmos en un tercero que mida primero el tamaño y luego elija el algoritmo a usar.

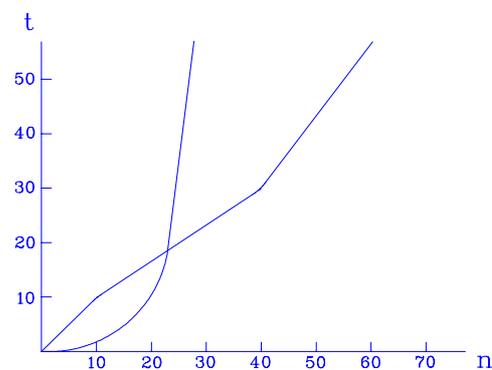


Ilustración 9-1. Tiempo de Ejecución de Algoritmos

9.1.3 Análisis de Algoritmos

¿Cómo analizar un algoritmo?. Cuando un algoritmo se ejecuta, hace uso de la Unidad Central de Procesos (CPU) para realizar las operaciones y usa las memorias residente (RAM) y auxiliar para procesar el programa y sus datos. El análisis de algoritmos hace referencia a la determinación tanto del tiempo como de la capacidad que un algoritmo requiere para funcionar.

9.1.3.1 Tiempo de ejecución

El tiempo de ejecución de un programa depende principalmente de:

1. Los datos de entrada al programa
2. La calidad del código generado por el compilador utilizado para crear el programa objeto.
3. La naturaleza y rapidez de las instrucciones de máquina empleadas en la ejecución del programa
4. La complejidad de tiempo del algoritmo base del programa.

Al depender este tiempo del tamaño de la entrada, suele notarse como $T(n)$ al tiempo de ejecución de un programa con una entrada de tamaño n (en nuestro caso n sería el número de unidades de trabajo). Las unidades de $T(n)$ se dejan sin especificar, pero se puede considerar a $T(n)$ como el número de instrucciones ejecutadas en un ordenador idealizado.

Para muchos programas, el tiempo de ejecución es en realidad una función de la entrada especificada y no sólo del tamaño de ésta. En este caso, se define $T(n)$ como el tiempo de ejecución del peor caso, es decir, el máximo valor del tiempo de ejecución para entradas de tamaño n . También se considera a veces $T_{prom}(n)$, que es el valor medio del tiempo de ejecución de todas las entradas de tamaño n .

Algunos autores denominan al estudio del tiempo de ejecución como análisis de la complejidad y lo examinan de la siguiente forma:

Sea V_n el conjunto de todas las entradas de tamaño n y sea i un elemento en particular de V . Notamos por $p(i)$ a la probabilidad de que entre el elemento i y sea $t(i)$ el número de operaciones básicas que realiza el algoritmo sobre la entrada i . La conducta esperada o complejidad media de una entrada de tamaño n se puede definir como:

$$A(n) = \sum_{i \in V_n} p(i)t(i)$$

y esto no es más que $T_{prom}(n)$. Además, como $t(i)$ es posible calcularlo aritméticamente, $p(i)$ se suele hallar en función de la experiencia o por tanteo, por lo que se suele tomar el peor de los casos, como ya hicimos anteriormente. Entonces definimos la complejidad como:

$$W(n) = \max_{i \in V_n} \{t(i)\}$$

donde $W(n)$ determina la cota superior de complejidad de un algoritmo para una entrada de tamaño n

9.1.3.2 Notación asintótica

Definición Decimos que la velocidad de crecimiento de un programa, cuyo tiempo de ejecución es $T(n)$, es $f(n)$ si existen dos enteros positivos c y n_0 tales que:

$$\text{Para todo } n \geq n_0 \text{ se verifica que } T(n) \leq c \cdot f(n)$$

donde $f(n)$ es una función definida sobre los enteros no negativos y que toma valores no negativos (no necesariamente enteros).

Notación asintótica:

Se dice que $T(n)$ es $\mathcal{O}(f(n))$ para indicar que $f(n)$ es una cota superior para la velocidad de crecimiento de $T(n)$. Para especificar una cota inferior se usa la notación $T(n)$ es $\Omega(g(n))$ y decimos que existe una constante c tal que $T(n) \geq c \cdot g(n)$ para un número infinito de valores de n .

10. Anexo II. Normativa ASCE

En este anexo ofrecemos un resumen de la normativa ASCE (American Society of Civil Engineers) [AS88] aplicado al campo de las torres (Manual nº52)

10.1 Diseño de piezas

El objetivo que aquí nos proponemos es presentar los métodos de cálculo para piezas de acero en estructuras reticulares.

10.2 Materiales usados

Los materiales que se usarán en las TTE seguirán las especificaciones estándar (UNE, DIN, ASTM, etc) en base a la última fecha de emisión de las mismas.

Respecto a las éstas, ASCE, en caso de estructuras de acero normales, recomienda la ASTM A36.

10.3 Secciones mínimas

El mínimo espesor de angular es de 1/8" (aprox. 3 mm.) para las piezas normales y de 3/16" (aprox. 5 mm.) para las conectadas con chapas (p.ej. montantes en uniones a tope). Estas condiciones pueden variar en casos de zonas de alta corrosión (p.ej. cercanas al mar o a polígonos petroquímicos)

10.4 Cálculo de la esbeltez

Usaremos la siguiente notación en las fórmulas expuestas a continuación:

- F_y = Límite elástico (habitualmente es 2750 en baja calidad y 3550 en alta)
- E = módulo de elasticidad (29000 ksi)

- L = Longitud de la pieza
- r = radio de inercia
- K = coeficiente de longitud efectiva ($K \cdot L$ es lo que llamamos longitud de pandeo)

El límite de esbeltez permitida para miembros calculados en tensión/compresión debe ser de:

$$\text{Montantes:} \quad \frac{L}{r} \leq 150$$

$$\text{Otros:} \quad K \frac{L}{r} \leq 200$$

$$\text{Rellenos:} \quad K \frac{L}{r} \leq 250$$

$$\text{Miembros sólo en tensión:} \quad K \frac{L}{r} \leq 500 \quad (\text{p.ej. tirantes de las crucetas})$$

10.5 Propiedades de los materiales

Las propiedades de un material tales como área, momento de inercia, radio de giro, módulo de la sección, etc. deben estar basados en el área total excepto cuando se especifique un área reducida o un área neta (Véase ASCE 52 4.9). El área neta es la diferencia entre el área de la sección (Suma de los productos del espesor del angular por cada una de las alas medidas desde el eje del elemento) menos el área de los taladros o cualquier otra apertura provocada en la sección de material investigada. Si hay un conjunto de taladros en diagonal o en zigzag, el área neta será determinada restando al área de la sección la de todos los taladros en línea y sumando para cada línea de taladros (gramil) la cantidad de:

$$\frac{s^2}{4g}$$

donde s es la distancia longitudinal (espaciado) y g la distancia transversal (gramil) de cada dos taladros consecutivos. El área neta crítica será obtenida en aquella cadena de taladros que dé el mínimo valor.

10.6 Compresión permisible

La compresión máxima permitida, que notaremos por F_a , en una sección de área del material o en un área reducida que se especifique, para miembros cargados con fuerzas axiales en compresión será de:

$$F_a = \begin{cases} \left[1 - \frac{1}{2} \left(\frac{K \frac{L}{r}}{C_c} \right)^2 \right] \cdot F_y & \text{si } K \frac{L}{r} \leq C_c \\ \frac{286.000}{\left(K \frac{L}{r} \right)^2} & \text{si } K \frac{L}{r} \geq C_c \end{cases} \quad (10.6 - 1)$$

$$F_a = \begin{cases} \left[1 - \frac{1}{2} \left(\frac{K \frac{L}{r}}{C_c} \right)^2 \right] \cdot F_y & \text{si } K \frac{L}{r} \leq C_c \\ \frac{286.000}{\left(K \frac{L}{r} \right)^2} & \text{si } K \frac{L}{r} \geq C_c \end{cases} \quad (10.6 - 2)$$

$$\text{donde } C_c = \Pi \cdot \sqrt{\frac{2E}{F_y}} \quad (10.6 - 3)$$

10.7 Miembros a compresión : Angulares

Los resultados expuestos aquí son aplicables solamente a angulares en L (90°). Si los angulares son en V (p.ej. 60°) deben aplicarse otro tipo de resultados (ver ASCE 52 4.9)

10.7.1 Máximo coeficiente w/t

El coeficiente $\frac{w}{t}$, donde w es el ancho del ala del material usado y t el espesor del mismo, no deberá exceder de 25

10.7.2 Esfuerzos permitidos a compresión

Los esfuerzos permitidos a compresión en una sección de área deben ser F_a de acuerdo con la sección 10.6. partiendo de que $\frac{w}{t}$ no haya excedido de los valores de 10.7.

10.7.3 Determinación de F_{cr}

Si $\frac{w}{t}$ (definido en 10.7.1.) excede de $\left(\frac{w}{t}\right)_{lim}$ dado por:

$$\left(\frac{w}{t}\right)_{lim} = \frac{80}{\sqrt{F_y}} \quad (10.7-1)$$

El esfuerzo permitido F_a debería ser el valor definido en 10.6. donde F_y (en las ecuaciones 10.6-1 y 10.6-3) es reemplazado por F_{cr} dado por:

$$F_a = \begin{cases} \left[1'677 - 0'677 \frac{w/t}{\left(\frac{w}{t}\right)_{lim}} \right] \cdot F_y & \text{si } \left(\frac{w}{t}\right)_{lim} \leq \frac{w}{t} \leq \frac{144}{\sqrt{F_y}} \\ \frac{9500}{\left(\frac{w}{t}\right)^2} & \text{si } \frac{w}{t} \geq \frac{144}{\sqrt{F_y}} \end{cases} \quad (10.7-2)$$

$$\text{si } \frac{w}{t} \geq \frac{144}{\sqrt{F_y}} \quad (10.7-3)$$

10.7.4 Longitud efectiva

Por último veamos como determinar la longitud efectiva $K \frac{L}{r}$ para cada caso. Esta longitud efectiva la denominaremos esbeltez y se notará con la variable δ , por tanto consideraremos:

$$\delta = K \frac{L}{r}$$

10.7.4.1 Montantes

Para montantes atornillados en ambas caras y extremos a otros:

$$\delta = \frac{L}{r'} \quad \text{con} \quad 0 \leq \frac{L}{r} \leq 150$$

10.7.4.2 Otros miembros en compresión

a) Para miembros con la carga concentrada en ambos extremos de su cogida al panel:

$$\delta = \frac{L}{r'} \quad \text{con} \quad 0 \leq \frac{L}{r} \leq 120$$

b) Para miembros con la carga concentrada en un extremo de su cogida al panel (ej. barras principales de crucetas):

$$\delta = 30 + 0.75 \frac{L}{r'} \quad \text{con} \quad 0 \leq \frac{L}{r} \leq 120$$

c) Para miembros con excentricidad normal en ambos extremos (ej. diagonales):

$$\delta = 60 + 0.5 \frac{L}{r'} \quad \text{con} \quad 0 \leq \frac{L}{r} \leq 120$$

d) Para miembros sin restricción a la rotación¹ en ambos extremos de su cogida (con un solo tornillo):

$$\delta = \frac{L}{r'} \quad \text{con} \quad 120 \leq \frac{L}{r} \leq 200$$

e) Para miembros con restricción parcial a la rotación en uno de sus extremos:

$$\delta = 28.6 + 0.762 \frac{L}{r'} \quad \text{con} \quad 120 \leq \frac{L}{r} \leq 225$$

f) Para miembros con restricción parcial a la rotación en ambos extremos:

$$\delta = 46.2 + 0.615 \frac{L}{r'} \quad \text{con} \quad 120 \leq \frac{L}{r} \leq 250$$

10.7.4.3 Miembros de relleno (secundarios)

$$\delta = \frac{L}{r'} \quad \text{con} \quad 0 \leq \frac{L}{r} \leq 120$$

¹ Una conexión mediante un solo tornillo en cada extremo de la pieza debe considerarse que no aporta restricciones a la rotación de la propia pieza. Cuando esta conexión se realiza mediante múltiples taladros y ha sido diseñada evitando excentricidades sí que hemos de considerar que ofrece restricciones parciales a la rotación (si la unión se realiza a otro miembro capaz de resistir la rotación de toda la junta)

a) Para miembros sin restricción a la rotación en ambos extremos de su cogida (con un solo tornillo):

$$\delta = \frac{L}{r'} \quad \text{con} \quad 120 \leq \frac{L}{r} \leq 250$$

b) Para miembros con restricción parcial a la rotación en uno de sus extremos:

$$\delta = 28.6 + 0.762 \frac{L}{r'} \quad \text{con} \quad 120 \leq \frac{L}{r} \leq 290$$

c) Para miembros con restricción parcial a la rotación en ambos extremos:

$$\delta = 46.2 + 0.615 \frac{L}{r'} \quad \text{con} \quad 120 \leq \frac{L}{r} \leq 330$$

10.8 Tensión permisible

La tensión máxima permitida, que notaremos por F_t , en una sección de área del material es el cociente de F_y por el área neta A_n .

NOTA FINAL: Todos los ensayos y/o análisis demuestran que algunos detalles específicos producen restricciones diferentes de las aquí recomendadas. En tales casos los valores de δ especificados deberán ser modificados.

10.9 Crítica a la norma ASCE

En este punto hemos de resaltar algunos apartados que consideramos que deberían

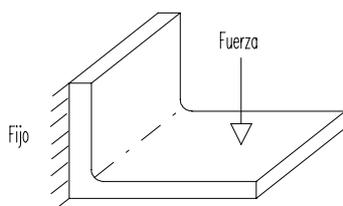


Ilustración 10-1. Fuerza aplicada

ser modificados en esta normativa internacional a raíz de la experiencia recogida en ensayos de materiales y fabricación de torres.

1.- El cociente ala y espesor w/t que se considera para la capacidad de alabeo del ala debe ser modificado, ya que, como podemos observar, esta deformación no afecta a un ala completa, sino que lo

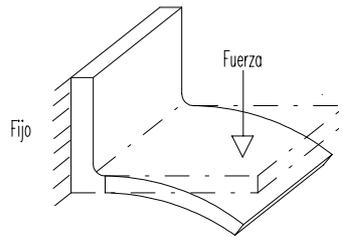


Ilustración 10-2. Doblado o Alabeo

hace al ala menos el espesor y menos el radio de la curva interna (es una parte más resistente a este alabeo).

Por tanto, en esta fórmula debería ser modificado el ala w por $w - [(w - 40) * 0.082 + 6.5 + t]$ que no es más que la curva de regresión que ajusta a cada ala y espesor su curva en la barra de acero laminado.

2.- El cálculo de área de pandeo es demasiado engorroso y podemos encontrarnos con la incongruencia siguiente:

Material con área de acero = 10 cm^2

Taladros a realizar: 5 en línea de 2 cm^2 cada uno

Area neta = $10 - 5 * 2 = 0 \text{ cm}^2 = \text{imposible}$

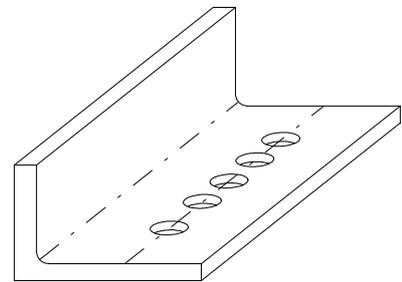


Ilustración 10-3. Taladros en línea

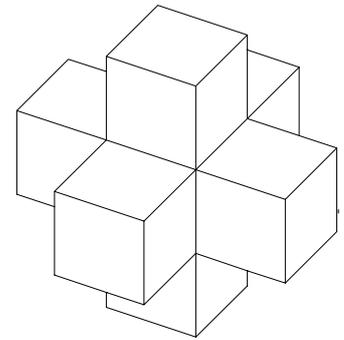
11. Anexo III. Minimización de Ancho de Banda

En este anexo buscamos una estrategia que nos permita comprobar si los algoritmos desarrollados nos llevan a soluciones óptimas. En particular, estudiamos un modelo espacial que permita la interacción en cualquier dirección del espacio.

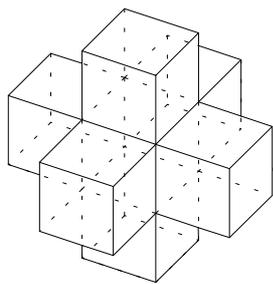
11.1 Aplicación a las estructuras espaciales

Procedemos de la siguiente manera:

1.- Planteamos un problema genérico que abarque todas las posibles numeraciones en cualquier dirección del espacio. El modelo que aquí estudiaremos será el de la imagen:



2.- Discretizamos el sólido representándolo por sus vértices y lados, quedándonos el modelo de la figura. En él observamos 32 nudos



diferentes con sus correspondientes relaciones entre ellos (barras). Para ver cuales de las posibles numeraciones satisfacen que la máxima diferencia de índice entre dos nudos cualesquiera que estén relacionados es mínima, generamos el campo de todas las soluciones. Esto nos obligaría a establecer $2.6313 \cdot 10^{35}$ ordenaciones

(permutaciones de 32 elementos), lo que exige demasiado tiempo de computación.

De ahí que necesitemos un caso más sencillo para estudiar.

3. – Por ello, y basándonos en la simetría del propio sólido, pasamos a un nuevo modelo del que trataremos de extraer las conclusiones para el anterior. Generamos la parte superior del sólido y marcamos las posibles posiciones de los nudos. En este caso estudiaremos $2 \cdot 10^{13}$ posibilidades.

11.1.1 Programación de permutaciones

Usamos el siguiente algoritmo exhaustivo para generar todo el espacio de soluciones con la menor diferencia posible. De entre las $2 \cdot 10^{13}$ combinaciones e han obtenido **46656** soluciones con diferencia 5 que, por supuesto, es la mínima al existir nudos relacionados con al menos cinco nudos más (los nudos de posición 8,9,12 y 13).

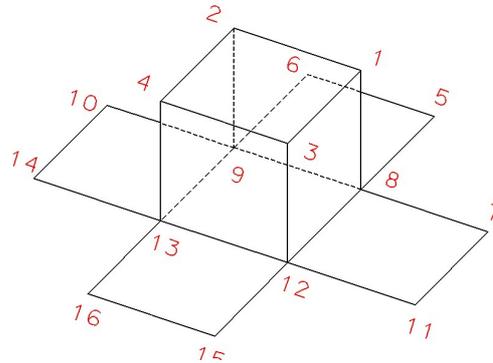


Ilustración 11-1. Numeración inicial de nudos

```

DIM a(16) AS INTEGER, b(16) AS INTEGER, maximo as INTEGER
DIM Banda AS integer, BandaLocal AS INTEGER
DIM NO AS INTEGER, YES AS INTEGER, Archivo AS STRING
DIM i16,i15,i14,i13,i12,i11,i10,i9,i8,i7,i6,i5,i4,i3,i2,i1 AS INTEGER
NO = 0: YES = 1
maximo = 16: Banda = maximo - 1
Archivo = "solucion.txt": OPEN Archivo FOR OUTPUT AS #1

\-----Permutaciones y relaciones posibles-----
FOR i1 = 1 TO maximo: a(1) = i1
FOR i2 = 1 TO maximo: a(2) = i2
  IF Vale(2, a()) = NO THEN NEXT i2
  IF compara(2, 1, a(), b(), Banda) = NO THEN NEXT i2
  FOR i3 = 1 TO maximo: a(3) = i3
    IF Vale(3, a()) = NO THEN NEXT i3
    IF compara(3, 1, a(), b(), Banda) = NO THEN NEXT i3
    FOR i4 = 1 TO maximo: a(4) = i4
      IF Vale(4, a()) = NO THEN NEXT i4
      IF compara(4, 2, a(), b(), Banda) = NO THEN NEXT i4
      IF compara(4, 3, a(), b(), Banda) = NO THEN NEXT i4
      FOR i5 = 1 TO maximo: a(5) = i5
        IF Vale(5, a()) = NO THEN NEXT i5
        FOR i6 = 1 TO maximo: a(6) = i6
          IF Vale(6, a()) = NO THEN NEXT i6
          IF compara(6, 5, a(), b(), Banda) = NO THEN NEXT i6
          FOR i7 = 1 TO maximo: a(7) = i7
            IF Vale(7, a()) = NO THEN NEXT i7
            FOR i8 = 1 TO maximo: a(8) = i8
              IF Vale(8, a()) = NO THEN NEXT i8
              IF compara(8, 1, a(), b(), Banda) = NO THEN NEXT i8
              IF compara(8, 5, a(), b(), Banda) = NO THEN NEXT i8
              IF compara(8, 7, a(), b(), Banda) = NO THEN NEXT i8
              FOR i9 = 1 TO maximo: a(9) = i9
                IF Vale(9, a()) = NO THEN NEXT i9
                IF compara(9, 2, a(), b(), Banda) = NO THEN NEXT i9
                IF compara(9, 6, a(), b(), Banda) = NO THEN NEXT i9
                IF compara(9, 8, a(), b(), Banda) = NO THEN NEXT i9
                FOR i10 = 1 TO maximo: a(10) = i10
                  IF Vale(10, a()) = NO THEN NEXT i10
                  IF compara(10, 9, a(), b(), Banda) = NO THEN NEXT i10
                  FOR i11 = 1 TO maximo: a(11) = i11
                    IF Vale(11, a()) = NO THEN NEXT i11
                    IF compara(11, 7, a(), b(), Banda) = NO THEN NEXT i11
                    FOR i12 = 1 TO maximo: a(12) = i12
                      IF Vale(12, a()) = NO THEN NEXT i12
                      IF compara(12, 3, a(), b(), Banda) = NO THEN NEXT i12
                      IF compara(12, 8, a(), b(), Banda) = NO THEN NEXT i12
                      IF compara(12, 11, a(), b(), Banda) = NO THEN NEXT i12
                      FOR i13 = 1 TO maximo: a(13) = i13

```

```

IF Vale(13, a()) = NO THEN NEXT i13
IF compara(13, 4, a(), b(), Banda) = NO THEN NEXT i13
IF compara(13, 9, a(), b(), Banda) = NO THEN NEXT i13
IF compara(13, 12, a(), b(), Banda) = NO THEN NEXT i13
FOR i14 = 1 TO maximo: a(14) = i14
  IF Vale(14, a()) = NO THEN NEXT i14
  IF compara(14, 10, a(), b(), Banda) = NO THEN NEXT i14
  IF compara(14, 13, a(), b(), Banda) = NO THEN NEXT i14
  FOR i15 = 1 TO maximo: a(15) = i15
    IF Vale(15, a()) = NO THEN NEXT i15
    IF compara(15, 12, a(), b(), Banda) = NO THEN NEXT i15
    FOR i16 = 1 TO maximo: a(16) = i16
      IF Vale(16, a()) = NO THEN NEXT i16
      IF compara(16, 13, a(), b(), Banda) = NO THEN NEXT i16
      IF compara(16, 15, a(), b(), Banda) = NO THEN NEXT i16
    \-----
    `En el caso que haya superado las condiciones anteriores establecemos `la máxima
    diferencia contenida en la permutación (<=Banda)

BandaLocal = b(2)
FOR kk = 2 TO maximo
  IF b(kk) > BandaLocal THEN BandaLocal = b(kk)
NEXT kk

\-----
` Recalculamos la banda si la solución la mejora => Inicializamos el
` archivo de soluciones
IF Banda > BandaLocal THEN
  Banda = BandaLocal
  CLOSE #1: OPEN Archivo FOR OUTPUT AS #1
END IF

\----- Guardamos la solución -----
FOR kk = 1 TO maximo: PRINT #1, a(kk); : NEXT kk: PRINT #1,

\----- Fin de los bucles -----
next i16,i15,i14,i13,i12,i11,i10,i9,i8,i7,i6,i5,i4,i3,i2,i1

\-----Cerramos el archivo-----
CLOSE #1

\----- Finalizamos el algoritmo -----
END

\+-----+
\| Función que comprueba si la permutación es correcta, es decir, el |
\| valor no ha aparecido anteriormente en la solución que se está |
\| construyendo |
\+-----+

FUNCTION Vale (indice, a()) AS INTEGER
  FOR kk = 1 TO indice - 1
    IF a(kk) = a(indice) THEN
      Vale = 0: EXIT FOR
    ELSE
      Vale = 1
    END IF
  NEXT kk
END FUNCTION

\+-----+
\| Función que compara si la diferencia supera a la banda y en caso |
\| negativo va guardando la máxima diferencia desde el índice |
\| estudiado |
\+-----+

FUNCTION compara (indice1,indice2,a() AS INTEGER,b() AS INTEGER,Banda)

  DIM Valor AS INTEGER
  IF ABS(a(indice1)-a(indice2)) > Banda THEN compara = 0:EXIT FUNCTION

  Valor = ABS(a(indice1) - a(indice2)): compara = 1
  IF Valor > b(indice1) THEN b(indice1) = Valor

END FUNCTION

```

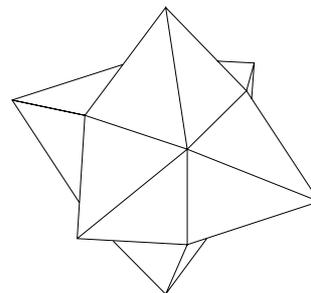
11.1.2 Análisis de soluciones

Si giramos una solución dada 90 grados respecto al eje Z obtenemos una nueva solución, pero no nos aporta información alguna. Por tanto debemos considerar realmente que tenemos $46656/4 = 11664$ numeraciones.

La extensión de cada una de estas numeraciones al caso inicial de 32 nudos proporciona soluciones inapropiadas, ya que el mínimo de banda posible que se obtiene es de 14. Si tenemos en cuenta que con cualquier ordenación que siga un patrón fijo de elección (atrás-adelante, derecha-izquierda y arriba-abajo) obtenemos al menos una banda de 12, observamos que las soluciones obtenidas no son extensibles al caso de 32 nudos.

11.1.3 Problema mediante pirámides

Partimos por tanto de otro caso en el que analizamos el cubo central completo junto con las 6 prolongaciones piramidales sobre cada una de sus caras. En este nuevo caso partimos del modelo que aparece en la figura.



1.- Nuevamente discretizamos el sólido, quedándonos la figura y su numeración de posiciones como siguen:

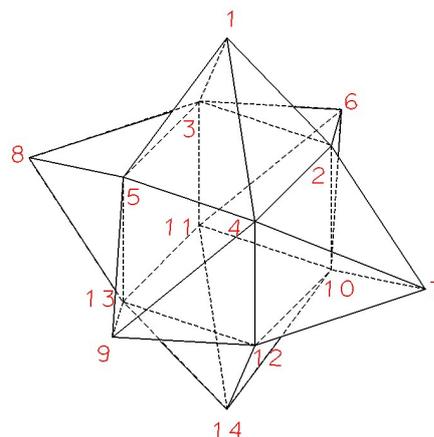
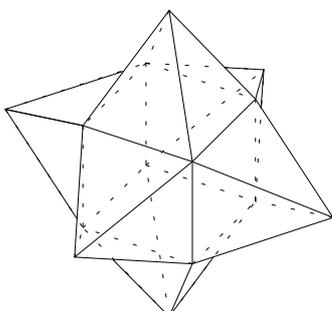


Ilustración 11-2. Numeración tetraédrica

2.- A partir de ella modificamos el algoritmo mediante el siguiente bucle de *Permutaciones y Relaciones posibles*:

```

\-----Permutaciones y relaciones posibles-----
FOR i1 = 1 TO maximo: a(1) = i1
  FOR i2 = 1 TO maximo: a(2) = i2: IF Vale(2, a()) = NO THEN NEXT i2
  IF compara(2, 1, a(), b(), Banda) = NO THEN NEXT i2
  FOR i3 = 1 TO maximo: a(3) = i3: IF Vale(3, a()) = NO THEN NEXT i3
  IF compara(3, 1, a(), b(), Banda) = NO THEN NEXT i3
  IF compara(3, 2, a(), b(), Banda) = NO THEN NEXT i3
  FOR i4 = 1 TO maximo: a(4) = i4: IF Vale(4, a()) = NO THEN NEXT i4
  IF compara(4, 1, a(), b(), Banda) = NO THEN NEXT i4
  IF compara(4, 2, a(), b(), Banda) = NO THEN NEXT i4
  FOR i5 = 1 TO maximo: a(5) = i5: IF Vale(5, a()) = NO THEN NEXT i5
  IF compara(5, 1, a(), b(), Banda) = NO THEN NEXT i5
  IF compara(5, 3, a(), b(), Banda) = NO THEN NEXT i5
  IF compara(5, 4, a(), b(), Banda) = NO THEN NEXT i5
  FOR i6 = 1 TO maximo: a(6) = i6: IF Vale(6, a()) = NO THEN NEXT i6
  IF compara(6, 2, a(), b(), Banda) = NO THEN NEXT i6
  IF compara(6, 3, a(), b(), Banda) = NO THEN NEXT i6
  FOR i7 = 1 TO maximo: a(7) = i7: IF Vale(7, a()) = NO THEN NEXT i7
  IF compara(7, 2, a(), b(), Banda) = NO THEN NEXT i7
  IF compara(7, 4, a(), b(), Banda) = NO THEN NEXT i7
  FOR i8 = 1 TO maximo: a(8) = i8: IF Vale(8, a()) = NO THEN NEXT i8
  IF compara(8, 3, a(), b(), Banda) = NO THEN NEXT i8
  IF compara(8, 5, a(), b(), Banda) = NO THEN NEXT i8
  FOR i9 = 1 TO maximo: a(9) = i9: IF Vale(9, a()) = NO THEN NEXT i9
  IF compara(9, 4, a(), b(), Banda) = NO THEN NEXT i9
  IF compara(9, 5, a(), b(), Banda) = NO THEN NEXT i9
  FOR i10 = 1 TO maximo: a(10) = i10: IF Vale(10, a()) = NO THEN NEXT i10
  IF compara(10, 2, a(), b(), Banda) = NO THEN NEXT i10
  IF compara(10, 6, a(), b(), Banda) = NO THEN NEXT i10
  IF compara(10, 7, a(), b(), Banda) = NO THEN NEXT i10
  FOR i11 = 1 TO maximo: a(11) = i11: IF Vale(11, a()) = NO THEN NEXT i11
  IF compara(11, 3, a(), b(), Banda) = NO THEN NEXT i11
  IF compara(11, 6, a(), b(), Banda) = NO THEN NEXT i11
  IF compara(11, 8, a(), b(), Banda) = NO THEN NEXT i11
  IF compara(11, 10, a(), b(), Banda) = NO THEN NEXT i11
  FOR i12 = 1 TO maximo: a(12) = i12: IF Vale(12, a()) = NO THEN NEXT i12
  IF compara(12, 4, a(), b(), Banda) = NO THEN NEXT i12
  IF compara(12, 7, a(), b(), Banda) = NO THEN NEXT i12
  IF compara(12, 9, a(), b(), Banda) = NO THEN NEXT i12
  IF compara(12, 10, a(), b(), Banda) = NO THEN NEXT i12
  FOR i13 = 1 TO maximo: a(13) = i13: IF Vale(13, a()) = NO THEN NEXT i13
  IF compara(13, 5, a(), b(), Banda) = NO THEN NEXT i13
  IF compara(13, 8, a(), b(), Banda) = NO THEN NEXT i13
  IF compara(13, 9, a(), b(), Banda) = NO THEN NEXT i13
  IF compara(13, 11, a(), b(), Banda) = NO THEN NEXT i13
  IF compara(13, 12, a(), b(), Banda) = NO THEN NEXT i13
  FOR i14 = 1 TO maximo: a(14) = i14: IF Vale(14, a()) = NO THEN NEXT i14
  IF compara(14, 10, a(), b(), Banda) = NO THEN NEXT i14
  IF compara(14, 11, a(), b(), Banda) = NO THEN NEXT i14
  IF compara(14, 12, a(), b(), Banda) = NO THEN NEXT i14
  IF compara(14, 13, a(), b(), Banda) = NO THEN NEXT i14
\-----

```

3.- En este caso el número de ordenaciones es de $8.7 \cdot 10^5$ y hemos encontrado 528 numeraciones de banda 6. Como cualquiera de ellas genera a 6 (las seis posiciones de los vértices de las pirámides que rodean al cubo central) tendremos que el número de soluciones diferentes, salvo giros, es 88. En particular, al analizar las numeraciones, observamos que todas tienen al 1 ó al 2 pero no a ambos en un vértice piramidal. Las 88 soluciones diferentes son:

Anexo III. Minimización de Ancho de Banda

Sol.	Pos.1	Pos.2	Pos.3	Pos.4	Pos.5	Pos.6	Pos.7	Pos.8	Pos.9	Pos.10	Pos.11	Pos.12	Pos.13	Pos.14
1	1	2	4	6	7	3	5	10	12	8	9	11	13	14
2	1	2	5	6	7	3	4	11	12	8	9	10	13	14
3	1	2	5	6	7	4	3	11	12	8	10	9	13	14
4	1	2	6	4	7	5	3	12	10	8	11	9	13	14
5	1	2	6	5	7	3	4	12	11	8	9	10	13	14
6	1	2	6	5	7	4	3	12	11	8	10	9	13	14
7	1	4	2	7	6	3	10	5	12	9	8	13	11	14
8	1	4	7	2	6	10	3	12	5	9	13	8	11	14
9	1	5	2	7	6	3	11	4	12	9	8	13	10	14
10	1	5	2	7	6	4	11	3	12	10	8	13	9	14
11	1	5	7	2	6	11	3	12	4	9	13	8	10	14
12	1	5	7	2	6	11	4	12	3	10	13	8	9	14
13	1	6	2	7	4	5	12	3	10	11	8	13	9	14
14	1	6	2	7	5	3	12	4	11	9	8	13	10	14
15	1	6	2	7	5	4	12	3	11	10	8	13	9	14
16	1	6	7	2	4	12	5	10	3	11	13	8	9	14
17	1	6	7	2	5	12	3	11	4	9	13	8	10	14
18	1	6	7	2	5	12	4	11	3	10	13	8	9	14
19	1	7	4	6	2	10	12	3	5	13	9	11	8	14
20	1	7	5	6	2	11	12	3	4	13	9	10	8	14
21	1	7	5	6	2	11	12	4	3	13	10	9	8	14
22	1	7	6	4	2	12	10	5	3	13	11	9	8	14
23	1	7	6	5	2	12	11	3	4	13	9	10	8	14
24	1	7	6	5	2	12	11	4	3	13	10	9	8	14
25	2	1	4	6	8	3	5	10	12	7	9	11	14	13
26	2	1	4	7	8	3	5	10	13	6	9	11	14	12
27	2	1	5	6	8	3	4	11	12	7	9	10	14	13
28	2	1	5	6	8	4	3	11	12	7	10	9	14	13
29	2	1	5	7	8	3	4	11	13	6	9	10	14	12
30	2	1	5	7	8	4	3	11	13	6	10	9	14	12
31	2	1	6	4	8	5	3	12	10	7	11	9	14	13
32	2	1	6	5	8	3	4	12	11	7	9	10	14	13
33	2	1	6	5	8	4	3	12	11	7	10	9	14	13
34	2	1	6	7	8	3	4	12	13	5	9	10	14	11
35	2	1	6	7	8	4	3	12	13	5	10	9	14	11
36	2	1	7	4	8	5	3	13	10	6	11	9	14	12
37	2	1	7	5	8	3	4	13	11	6	9	10	14	12
38	2	1	7	5	8	4	3	13	11	6	10	9	14	12
39	2	1	7	6	8	3	4	13	12	5	9	10	14	11
40	2	1	7	6	8	4	3	13	12	5	10	9	14	11
41	2	4	1	8	6	3	10	5	12	9	7	14	11	13
42	2	4	1	8	7	3	10	5	13	9	6	14	11	12
43	2	4	8	1	6	10	3	12	5	9	14	7	11	13
44	2	4	8	1	7	10	3	13	5	9	14	6	11	12
45	2	5	1	8	6	3	11	4	12	9	7	14	10	13
46	2	5	1	8	6	4	11	3	12	10	7	14	9	13
47	2	5	1	8	7	3	11	4	13	9	6	14	10	12
48	2	5	1	8	7	4	11	3	13	10	6	14	9	12
49	2	5	8	1	6	11	3	12	4	9	14	7	10	13
50	2	5	8	1	6	11	4	12	3	10	14	7	9	13
51	2	5	8	1	7	11	3	13	4	9	14	6	10	12
52	2	5	8	1	7	11	4	13	3	10	14	6	9	12
53	2	6	1	8	4	5	12	3	10	11	7	14	9	13
54	2	6	1	8	5	3	12	4	11	9	7	14	10	13
55	2	6	1	8	5	4	12	3	11	10	7	14	9	13
56	2	6	1	8	7	3	12	4	13	9	5	14	10	11
57	2	6	1	8	7	4	12	3	13	10	5	14	9	11
58	2	6	8	1	4	12	5	10	3	11	14	7	9	13
59	2	6	8	1	5	12	3	11	4	9	14	7	10	13
60	2	6	8	1	5	12	4	11	3	10	14	7	9	13
61	2	6	8	1	7	12	3	13	4	9	14	5	10	11
62	2	6	8	1	7	12	4	13	3	10	14	5	9	11
63	2	7	1	8	4	5	13	3	10	11	6	14	9	12
64	2	7	1	8	5	3	13	4	11	9	6	14	10	12
65	2	7	1	8	5	4	13	3	11	10	6	14	9	12
66	2	7	1	8	6	3	13	4	12	9	5	14	10	11
67	2	7	1	8	6	4	13	3	12	10	5	14	9	11
68	2	7	8	1	4	13	5	10	3	11	14	6	9	12
69	2	7	8	1	5	13	3	11	4	9	14	6	10	12
70	2	7	8	1	5	13	4	11	3	10	14	6	9	12
71	2	7	8	1	6	13	3	12	4	9	14	5	10	11
72	2	7	8	1	6	13	4	12	3	10	14	5	9	11



73	2	8	4	6	1	10	12	3	5	14	9	11	7	13
74	2	8	4	7	1	10	13	3	5	14	9	11	6	12
75	2	8	5	6	1	11	12	3	4	14	9	10	7	13
76	2	8	5	6	1	11	12	4	3	14	10	9	7	13
77	2	8	5	7	1	11	13	3	4	14	9	10	6	12
78	2	8	5	7	1	11	13	4	3	14	10	9	6	12
79	2	8	6	4	1	12	10	5	3	14	11	9	7	13
80	2	8	6	5	1	12	11	3	4	14	9	10	7	13
81	2	8	6	5	1	12	11	4	3	14	10	9	7	13
82	2	8	6	7	1	12	13	3	4	14	9	10	5	11
83	2	8	6	7	1	12	13	4	3	14	10	9	5	11
84	2	8	7	4	1	13	10	5	3	14	11	9	6	12
85	2	8	7	5	1	13	11	3	4	14	9	10	6	12
86	2	8	7	5	1	13	11	4	3	14	10	9	6	12
87	2	8	7	6	1	13	12	3	4	14	9	10	5	11
88	2	8	7	6	1	13	12	4	3	14	10	9	5	11

Del análisis del conjunto de soluciones podemos extraer que valores extremos en la numeración (el 1 y el 14, ó 2 y 13, en este caso) deben estar también en los extremos de la propia estructura o modelo, y además lo más alejados posible entre sí.

11.2 Aplicación a las torres

Realizamos, bajo el mismo algoritmo, dos modelos de sólidos que se asemejen más a las torres, como las que se indican en la figura:

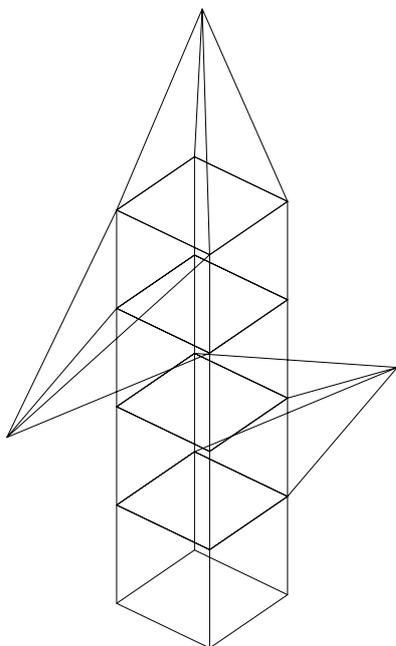


Ilustración 11-3. Sin relación diagonal

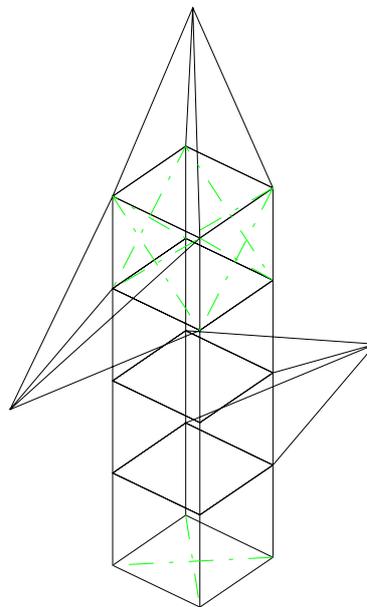


Ilustración 11-4. Con relación diagonal

En el primer caso no relacionamos los vértices entre sí salvo de forma horizontal o vertical (salvo con la punta de la cruceta), mientras que en el segundo relacionamos todos los vértices (aunque hayamos dibujado cómo lo hacemos verticalmente y horizontalmente sólo en un caso para no cargar excesivamente el gráfico).

En el primer caso obtuvimos ordenaciones con mínimo de banda 5. Pero al relacionar los vértices entre sí (caso 2), lo cual responde de una manera más cercana a la realidad de las torres, la banda mínima necesaria fue 7. En este segundo caso nuestro patrón fijo de elección (atrás-adelante, derecha-izquierda y arriba-abajo) sí nos lleva a una de las formas de ordenación escogidas. De aquí podemos deducir que si seguimos este sistema de ordenación antes de construir la matriz de rigidez podemos prescindir de un sistema de preprocesado de numeración nodal que nos aumenta de forma alarmante el tiempo de cálculo.

Aplicaremos por tanto este patrón fijo de elección de nudos en el cálculo de las torres.

12. Anexo IV. Sistema de Gestión de Materiales

12.1 Presentación

Como aplicación complementaria al Sistema de Torres, creamos el sistema de gestión de todas las piezas que conforman la estructura al que llamaremos MATERIAL. Éste provee al proyectista a través de su interfaz, de la base de datos correspondiente a las barras y cartelas que componen la estructura, así como de las herramientas para la creación de documentación y manipulación de los elementos integrantes de la misma.

Esta base de datos está generada a partir de la sección de fabricación del sistema principal. Este último construye a partir de los cálculos del proyecto una relación parametrizada del material empleado en forma de archivo o fichero de extensión *.MAT que debe estar localizado en la carpeta o directorio del PROYECTO con el mismo nombre que la denominación de la estructura analizada.

La pérdida del archivo de materiales correspondiente a un proyecto requiere la reconstrucción total de la base de datos mediante la opción del sistema antes referida.

12.2 El Programa MATERIAL

La aplicación MATERIAL dispone de una interfaz en el entorno WINDOWS, que permite la visualización de las barras y las chapas o cartelas en el formato en el que van a ser documentadas, integrando todos los datos necesarios tales como nomenclatura, proyecto, magnitudes, etc.. necesarios para la fabricación.

Las características de MATERIAL son:

- Observación de las planillas de barras y chapas con todos sus detalles constructivos y dimensionales.
- Observación de la lista de materiales.

- Impresión de planillas y listas.
- Modificación, integración y eliminación (posterior a la manipulación del diseño) de los elementos constitutivos del material de la estructura, tales como :
 - ◆ Gramil y progresiva de taladros
 - ◆ Cortes de barra
 - ◆ Doblados de barras y cartelas
 - ◆ Deformaciones
 - ◆ Puntos de soldadura
 - ◆ Etc.

El Sistema de Torres está ligado también a una aplicación de diseño gráfico en la cual se generan los planos y detalles de fabricación. A través de ésta se pueden crear cartelas y barras que son incluidas en la base de datos de MATERIAL automáticamente o manualmente. La inclusión manual de estos elementos en el diseño requiere la atenta modificación manual de la base de datos de MATERIAL, y siempre en este mismo orden de actualización.

12.2.1 Archivos Necesarios

1. *Tabla de Angulares*
2. *Torre.Cod* (Códigos de las diferentes secciones de la Torre)
3. *Torre.Mcf* (Configuración de Materiales)
4. *Torre.Mat* (Materiales de la estructura)
5. *Torre.Sel* (Limitaciones físicas a considerar en el material)
6. *Torre.Tow* (Datos básicos de la geometría de la torre)

12.3 Pantalla de Trabajo

La Pantalla de trabajo se estructura de la siguiente manera:

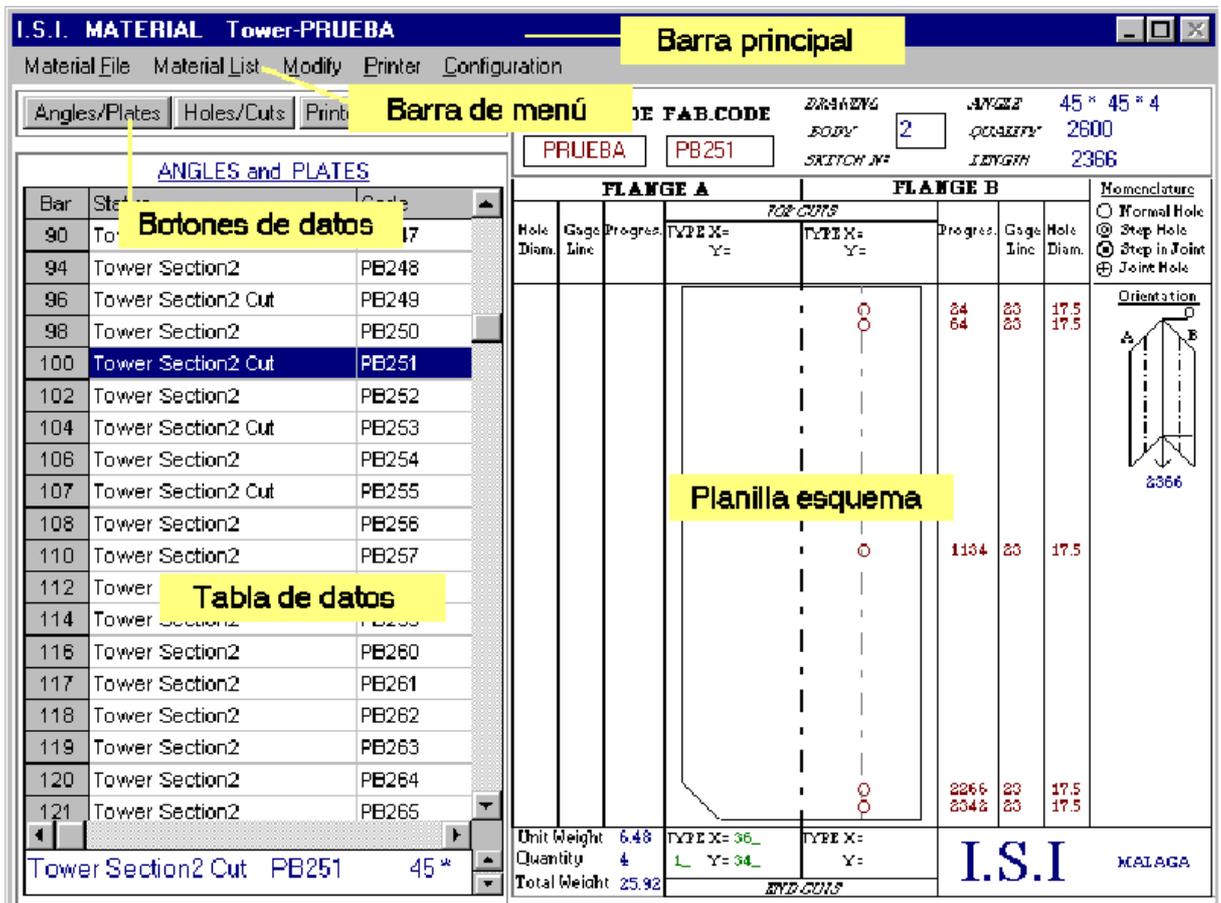


Ilustración 12-1. Comandos principales

- Barra principal : En la que se especifica el nombre del archivo de materiales que corresponde al proyecto o estructura de la misma denominación.
- Barra de menú : Conteniendo los comandos y opciones de carga, manipulación y configuración de MATERIAL.
- Botones de datos:

Angles/Plates : Visualiza la tabla de propiedades de los angulares y las chapas.

Holes/Cuts : Visualiza la tabla de propiedades de los taladros, puntos de soldadura, doblados y cortes de la barra o angular seleccionado.

Printer Tb. : Imprime la tabla de propiedades de todos los angulares y chapas del proyecto seleccionado.

Save : Guarda las propiedades de los angulares o las chapas seleccionadas.

Cuando se pulsa sobre alguno de los dos últimos botones, el programa preguntará por la aceptación o cancelación de la operación.

Se consideran características de angular o cartela a:

- de registro
 - Cuerpo o sección de la estructura al que pertenece
 - Código de fabrica con el que se le reconoce en el diseño
 - Longitud de angular en milímetros
 - Sección de barra entendiendo siempre ALA x ALA x ESPESOR en milímetros.
 - Dimensión de chapa entendiendo siempre ALTO x ANCHO x ESPESOR en milímetros.
 - Calidad del acero con el que se fabrican las piezas.
 - Cantidad de piezas
 - Peso unitario y total de las piezas
-
- Tabla de datos (o material): Presenta de forma tabulada la relación de la base de datos de material con todas las características. En ella se han de seleccionar los datos a presentar en la planilla esquema, a imprimir o a modificar. El recuadro de texto inferior se utiliza para modificaciones tal y como se explicará.

Se considera seleccionar a pulsar sobre una o varias filas de las tablas de presentación de material o elementos del material.

Son elementos de una barra o chapa:

- Gramil y progresiva de taladros
- Cortes de barra
- Doblados de barras y cartelas
- Deformaciones
- Puntos de soldadura

- Planilla esquema: Representa de forma gráfica la planilla de la barra o cartela seleccionada de la tabla de datos con todos los detalles de fabricación.

Esta representación, que es semejante al formato impreso, corresponde a la vista abatida.

En el esquema podemos apreciar los siguientes datos:

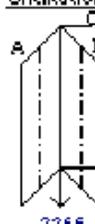
TOWER CODE		FAB.CODE		DRAWING	ANGLE	45 * 45 * 4		
PRUEBA		PB251		BODY	2	QUANTITY	2600	
				SKETCH #		LENGTH	2366	
FLANGE A				FLANGE B				Nomenclature <input type="radio"/> Normal Hole <input type="radio"/> Step Hole <input type="radio"/> Step in Joint <input type="radio"/> Joint Hole
Hole Diam.	Gage Line	Progres.	TOP CUTS		Progres.	Gage Line	Hole Diam.	
			TYPE X= Y=	TYPE X= Y=				
			2		24 64	23 23	17.5 17.5	Orientation 
					1134	23	17.5	
					2266 2342	23 23	17.5 17.5	
Unit Weight	6.48	TYPE X=	36_	TYPE X=				I.S.I MALAGA
Quantity	4	L	Y=34_	Y=				
Total Weight	25.92			4	END CUTS			

Ilustración 12-2. Planilla de Barra

1- Esquema de angular abatido con los taladros y cortes.

2 - Cabecera de alas y dimensiones de cortes superiores así como de los tipos codificados de taladros para tornillos.

Los cortes se definen:

- TYPE : Tipo de corte
- X e Y: Dimensión del corte horizontal y vertical en milímetros.

3 - Cabecera principal conteniendo los siguientes datos:

- Nombre de la estructura
- Código de fabricación
- Cuerpo o sección de la torre o estructura donde se localiza
- Sección del angular o dimensión de la chapa en milímetros
- Calidad de acero
- Longitud de barra en milímetros

4 - Cuadro de pesos, cantidades, rebajes y cortes inferiores.

5 - Tabla de:

- Progresivas. Distancia en ordenadas desde el origen (0) a la posición de los taladros en milímetros.
- Gramiles. Distancia en abscisas desde el origen (0) a la posición de taladros en milímetros.
- Diámetros. De los taladros en milímetros.

6 - Gráficos indicativos de orientación y en su caso de deformación de angular.

12.3.1 Opciones del Menú

Menú MATERIAL FILE :

- *Open Material File*: Presenta una ventana donde escoger un fichero de material para ser cargado y visualizado.
- *Reordering* : Permite reordenar en función de su código de fabricación el archivo de material cargado.
- *View complete Material*: El material cargado unifica las barras iguales. Esta opción deshace la asimilación y presenta la tabla de material barra a barra.

- *View assimilate Material*: Permite deshacer la opción anterior cargando el fichero de trabajo con las barras iguales asimiladas.
- *Close* : Cierra el programa

Menú *MATERIAL LIST*:

Visualiza la pantalla de la lista de materiales una vez cargado el archivo de material.

Menú *MODIFY* :

- *Additions* : Permite incluir nuevas barras, chapas y elementos en ambas.
- *Add Bars*: Opción para incluir nuevas barras
- *New* : Opción para incluir nuevas barras controlando todos sus elementos y propiedades que la constituyen.
- *Copy selected bar*: Adiciona a la base de datos la misma barra que se ha seleccionado a través de la tabla de material.
- *Copy mirror selected bar*: Adiciona a la base de datos la misma barra que se ha seleccionado a través de la tabla de material pero INVIRTIENDO las ALAS.
- *Add Plate*: Opción para incluir nuevas chapas controlando todos sus elementos y propiedades que la constituyen.
- *Add Holes*: Incluye taladros en una BARRA.
- *Add Cuts*: Incluye cortes en una BARRA.
- *Add Flange deformed*: Incluye deformaciones en una BARRA.
- *Add Bending* : Incluye doblados en una CHAPA.
- *Delete* : Permite eliminar elementos en barras y chapas.
 - *Delete Sketch* : Elimina la barra o chapa seleccionada.
 - *Delete Hole* : Elimina taladros en una BARRA.
 - *Delete Cuts*: Elimina cortes en una BARRA.
 - *Delete Bending*: Elimina doblados en una CHAPA.
- *Change Hole position*: Permite cambiar la posición (gramil) y diámetro de varios taladros a la vez.
- *Enlargement Bar*: Alarga una BARRA por su origen o su extremo final una longitud deseada.
- *Section Bar*: Modifica la sección de una BARRA.

Menú PRINTER :

- *Printer Sketch*: Imprime la planilla o planillas seleccionadas con el formato escogido a través de la configuración de salida.
- *ISI Sketch*: Imprime la planilla o planillas seleccionadas con el formato ISI de papel impreso.

Menú CONFIGURATION :

Visualiza la pantalla de configuración de salida de documentación.

12.3.2 Inicio del Programa

Para iniciar una sesión del programa pulsamos la opción *OPEN MATERIAL FILE* y aparece la siguiente ventana:

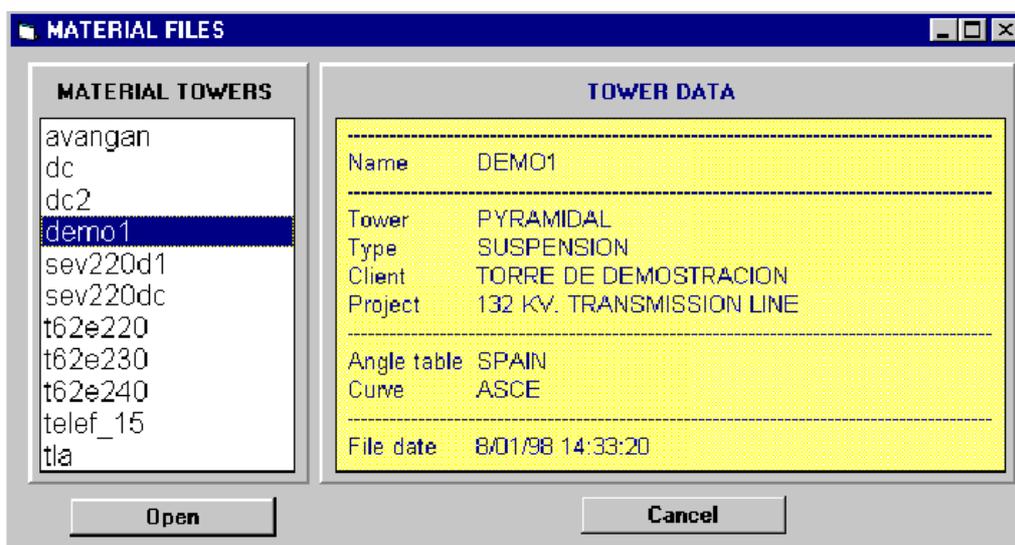


Ilustración 12-3. Elección de torre

Escogemos directamente el archivo/torre de material requerido y pulsamos para cargarlo.

La tabla de material presentará todos los registros de la base de datos y se activarán todos los controles de manipulación.

12.4 Gestión de Esquemas de Barras y Cartelas

12.4.1 Visualización

Para visualizar una barra o chapa será suficiente con pulsar sobre la fila de la tabla de material que la contiene. Automáticamente se representará la planilla con todas sus propiedades.

The screenshot shows the 'I.S.I. MATERIAL Tower-PRUEBA' application window. The 'Holes/Cuts' button is selected in the top menu. The main window displays two tables: 'HOLES, BENDING and FLANGE DEF.' and 'CUTS and CHANFER'. The 'HOLES' table has 5 rows, and the 'CUTS' table has 1 row. The 'Unit', 'Quan', and 'Total' labels are visible on the right side of the interface.

Bolt	Flange	Hole	Gage	Progress.	Type
1	(B)	17.5	23	24	1
2	(B)	17.5	23	64	1
3	(B)	17.5	23	1134	1
4	(B)	17.5	23	2266	1
5	(B)	17.5	23	2342	1

Cut	Flange	Type	Position	X	Y
1	(A)	1	Dw	36	34

Es posible presentar la tabla de taladros y cortes pertenecientes a la BARRA pulsando sobre el botón HOLES/CUTS tal y como se aprecia a continuación:

Se observan en esta figura las propiedades de la barra PB251 y que se corresponde a la planilla de barra mostrada anteriormente.

De los taladros obtenemos:

- El ala en el que se encuentra.
- El diámetro en milímetros.
- El gramil en milímetros.
- La progresiva en milímetros.
- El tipo de taladro.

Ilustración 12-4. Modificación de los datos de los taladros

De los cortes obtenemos:

- El ala en que se encuentra el corte.
- El tipo codificado del corte.
- La posición *Down*, *Up* o *Middle*.
- La X o ancho del corte sobre el eje de coordenadas.
- La Y o altura del corte sobre el eje de coordenadas.

Podemos también obtener los datos de topología o relación con el diseño gráfico de la estructura de una BARRA pulsando sobre la fila de la tabla de material, presentando una nueva tabla con los siguientes datos:

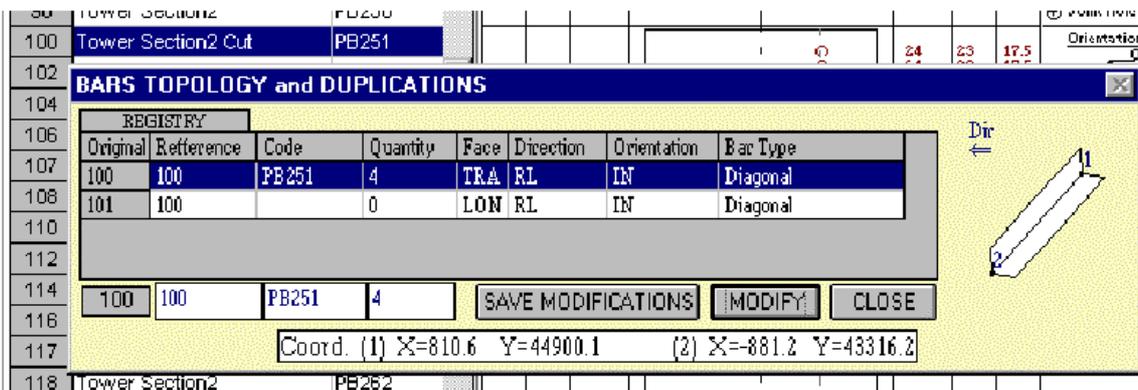


Ilustración 12-5. Formación de grupos de barras iguales

Datos de registro: Referidos al original (barra independiente) y a referencial (barras asimiladas como iguales).

- Code : Código de fabrica.
- Quantity : Cantidad de barras asimiladas como iguales.
- Face : Situación de la barra en la vista longitudinal (LON) o transversal (TRA) del diseño original de la torre.
- Direction : Dirección de la barra respecto al origen y hacia el extremo final en función de la vista del diseño y la posición de la barra en él. Puede ser de derecha a izquierda (RL) o de izquierda a derecha (LR).
- Orientation : Dirección de las alas de la barra en la torre respecto a la vista del diseño y que puede ser IN (alas hacia dentro) o OUT (alas hacia fuera).
- Bar Type : Nomenclatura de la barra.

Seleccionando una de las filas de esta tabla nos aparecerá un esquema orientativo a la derecha de la barra en cuestión, así como de los datos de coordenadas en el espacio de los vértices opuestos de la representación en el sistema gráfico.

En esta tabla también se pueden apreciar los botones MODIFY y CLOSE. El primero de ellos se utiliza para modificar el registro de referencia, el código y la cantidad, apareciendo otro botón (SAVE MODIFICATIONS) para guardar los cambios, y el segundo para ocultar la tabla.

Para que resulten visibles las modificaciones efectuadas es necesario volver a cargar el material. Estas modificaciones no serán efectuadas sobre el archivo de diseño de la estructura.

12.4.2 Modificación

MATERIAL permite modificar los elementos constitutivos de las estructuras calculadas y diseñadas por el Sistema de Torres para adecuar totalmente dichos materiales a las especificaciones. Para ello la aplicación utiliza tres métodos :

- Modificación a través de las cajas de texto de las tablas de material y de taladros.
- Modificación a través de las tablas de material directamente.
- Modificación a través de las opciones del menú.

1 - Modificación a través de las cajas de texto de las tablas de material

Al seleccionar una fila de la tabla de material o de taladros, en la caja de texto de la parte inferior aparece la propiedad o propiedades de la pieza como imagen de lo representado en la fila según se haya pulsado en una columna o en varias. Entrando con el cursor sobre la caja de texto, podemos modificar el valor

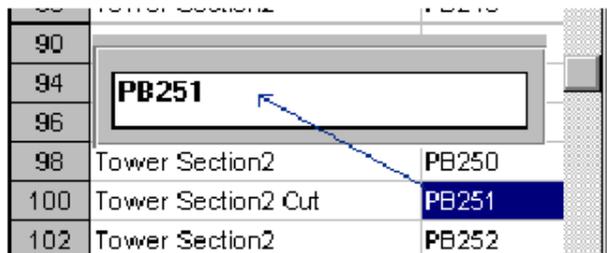
98	PB250	45 * 45 * 4	2600	2
100	PB251	45 * 45 * 4	2600	2
102	PB252	45 * 45 * 4	2600	2
104	PB253	45 * 45 * 4	2600	2

Tower Section2 Cut	PB251	45 *
--------------------	-------	------

Ilustración 12-6. Selección de Pieza

inscrito que quedará automáticamente actualizado en la fila seleccionada. Posteriormente se han de salvar los cambios pulsando sobre el botón *SAVE* del menú superior.

2 - Modificación a través de las tablas de material directamente.



90		
94		
96		
98	Tower Section2	PB250
100	Tower Section2 Cut	PB251
102	Tower Section2	PB252

Pulsando sobre una fila y una columna de la tabla de material nos aparece una caja de texto flotante donde podremos modificar el valor correspondiente a la fila y columna seleccionada. Presionando *ENTER* sin salir de la ventana de texto y

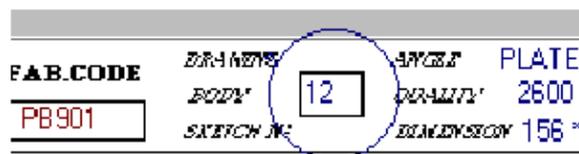
Ilustración 12-7. Cambio de Nomenclatura

posteriormente el botón *SAVE*, quedarán registradas las modificaciones.

Las modificaciones permitidas por estos dos métodos son las siguientes:

- Para las barras: Código de fábrica, longitud y cantidad
- Para las chapas: Código de fábrica, espesor y cantidad

Además podemos cambiar el cuerpo o sección de la torre en que se encuentra la barra o la chapa seleccionada sin más que cambiar el código del cuerpo en la ventana de texto titulada *BODY* en la cabecera principal de la planilla esquema, salvando los cambios posteriormente.



FAB.CODE	BARRENS	APCS	PLATE
PB901	BODY 12	QUALITY	2600
	SKETCH NO	QUANTITY	156 *

Ilustración 12-8. Cuerpo al que pertenece la pieza

3 - Modificación a través de las opciones del menú.

En la opción *MODIFY* del menú encontramos las siguientes posibilidades:

- Adicionar material, elementos o propiedades del material (*ADD*).
- Eliminar material, elementos o propiedades del material (*DELETE*).

12.4.3 Adición de elementos

Opción New : Opción para incluir nuevas barras controlando todos sus elementos y propiedades que la constituyen. Se presenta en pantalla la ventana :

Ilustración 12-9. Creación de barras nuevas

En ella observamos las cajas de texto en las que deberemos incluir los datos correspondientes a la nueva barra, siempre en orden descendente y pulsando para la aceptación de los datos integrados. El cuerpo o sección de la torre y el código de fabrica de la barra se pueden seleccionar de las listas desplegables correspondientes, de la misma manera que las secciones de la tabla de angulares del proyecto que se ha definido por defecto.

Las cuatro partes en que está dividida esta ventana son:

1. Cuadro superior o principal. Donde se aprecian los datos más genéricos de la barra.
2. Cuadro *BOLTS*. Donde se deben incluir todas las propiedades anteriormente mencionada de cada uno de los taladros de la barra. Estos datos son necesarios para la barra.
3. Cuadro *CUTS*. Donde se deben incluir todas las propiedades anteriormente mencionada de cada uno de los cortes de la barra. Estos datos son optativos para la barra.
4. Cuadro *GRAPHICS DATA*. Donde se deben incluir todos los datos concernientes a la situación y topología de la barra en el diseño gráfico como:

- Nivel o capa del dibujo
- Coordenadas X e Y del origen y X e Y del final de los vértices de la vista o proyección de esta en el diseño.
- Número de proyecciones (1 ó 2) según se represente la barra en la vista transversal, longitudinal o en ambas.
- Si tiene o no doblado.
- La orientación (IN/OUT) del ala en la vista.

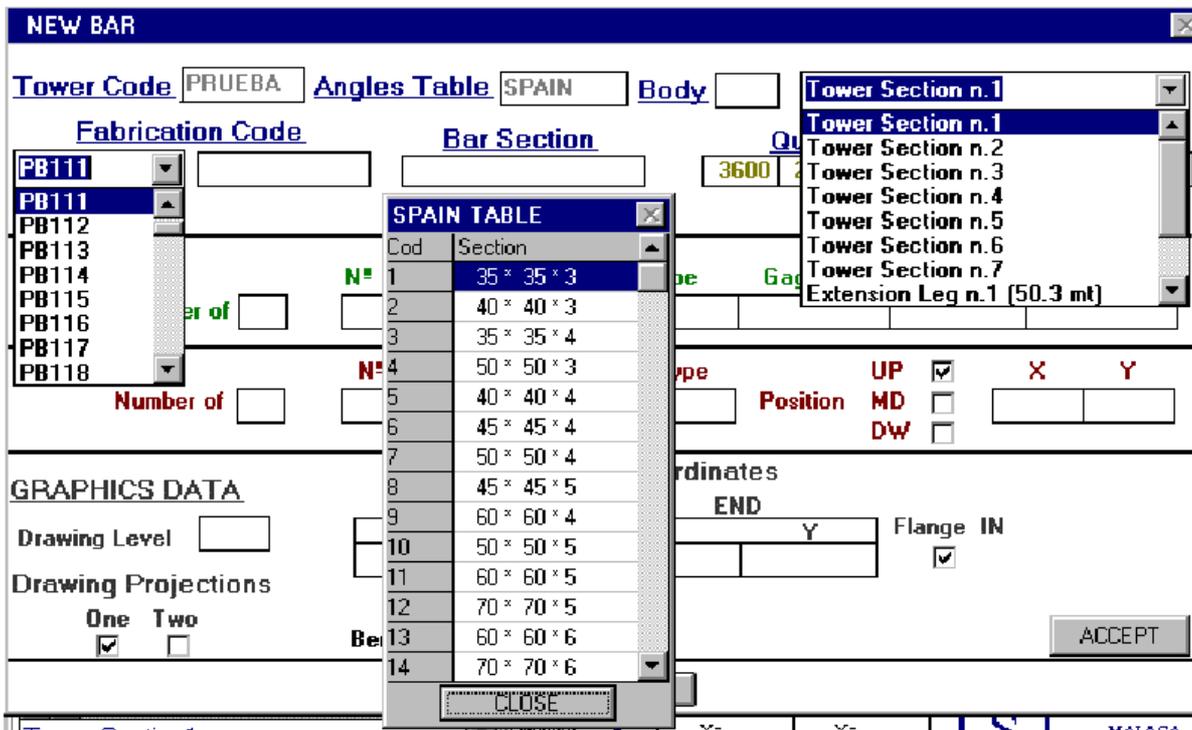


Ilustración 12-10. Menús desplegables para la elección de los datos

Para escoger la calidad del acero en la barra, ha de hacerlo de los tres valores que vienen por defecto en la especificación del proyecto. En el caso de la torre de ejemplo se observa en la penúltima figura los valores 3600 / 2600 / 0.

Para terminar la operación de integración de la nueva barra será suficiente con pulsar ACEPTAR e ir siguiendo las instrucciones de pantalla.

Copy selected bar: Adiciona a la base de datos la misma barra (con sus mismas propiedades) que se ha seleccionado a través de la tabla de material. Esta se situará al final de la base de datos como último registro.

Copy mirror selected bar: Adiciona a la base de datos la misma barra (con sus mismas propiedades) que se ha seleccionado a través de la tabla de material pero INVIRTIENDO las ALAS. Esta se situará al final de la base de datos como último registro.

Add Plate: Opción para incluir nuevas chapas controlando todos sus elementos y propiedades que la constituyen.

The 'New Plates' dialog box contains the following fields and values:

Name of Tower	Body	Code	Quantity	Thick
PRUEBA	1	PB999	1	6

Also enter bending line coordinates

N° Points	N°	X	Y
6	6	0	200

N° of Bolts	N°	Type	Diameter	X (gramil)	Y (porgres.)
2	2	1	17.5	175	50

N° of Bends	N°	Origin Point	End Point	Angle(rad)
1	1	2	5	60

CLOSE

Ilustración 12-11. Creación de chapas nuevas

Aquí al igual que con las barras observamos las cajas de texto en las que debemos incluir los datos correspondientes a la nueva chapa, siempre en orden descendente y aceptando por último los datos integrados.

Para los datos introducidos según la última figura obtendríamos una cartela tal y como se aprecia a continuación:

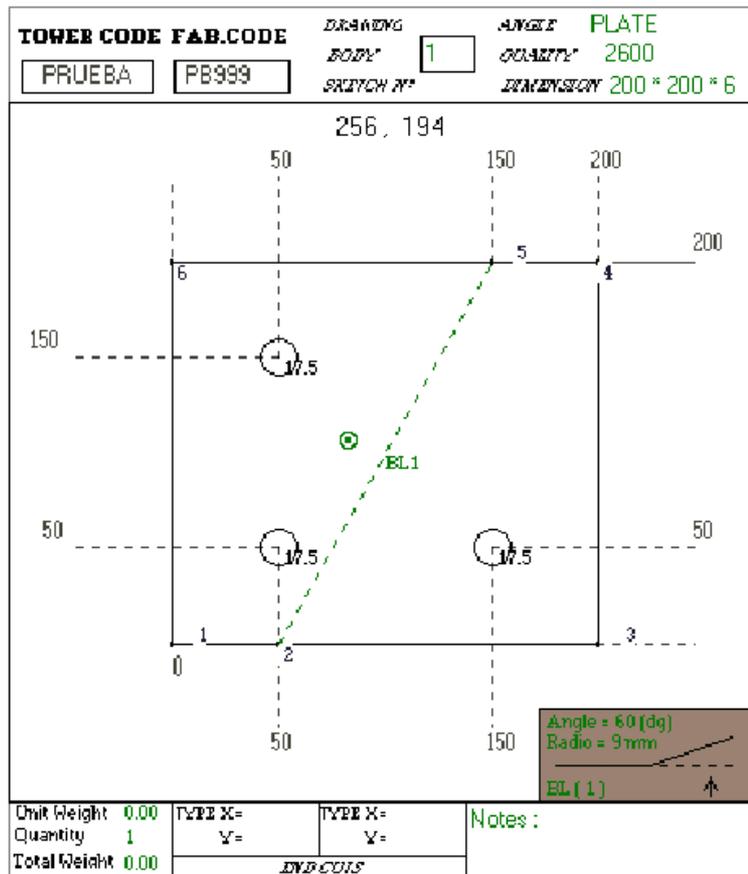


Ilustración 12-12. Doblado de chapas

Se observa la numeración de los puntos del contorno de la chapa (hasta 6) y como el doblado (BLn o Bending Line) se ha introducido del (2) al (5) con el ángulo de 60° detallado en la esquina inferior de la chapa.

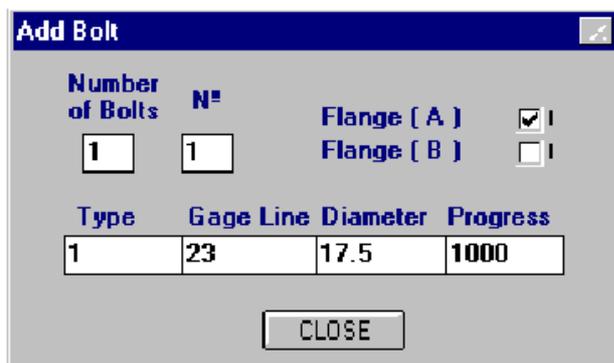


Ilustración 12-13. Incorporación de taladros nuevos a la barra

Para mayor comodidad del usuario es conveniente tener visible la tabla de taladros e ir comprobando en cada momento la actualización de estos.

Add Cuts: Incluye cortes en una BARRA.

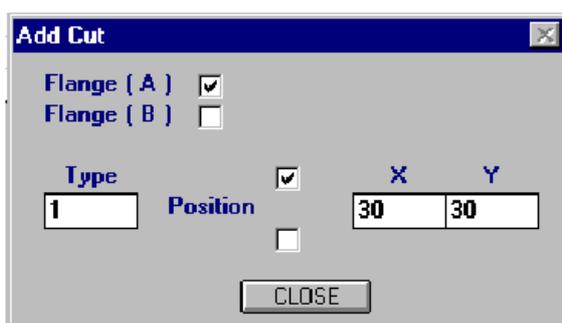


Ilustración 12-14. Colocación de nuevos cortes

El procedimiento para la inclusión de cortes es el mismo que el descrito en el apartado anterior.

Las características de tipo y posición se pueden observar en el cuadro de tipos de cortes sobre una barra del capítulo de fabricación.

Add Flange deformed: Incluye deformaciones en una BARRA.

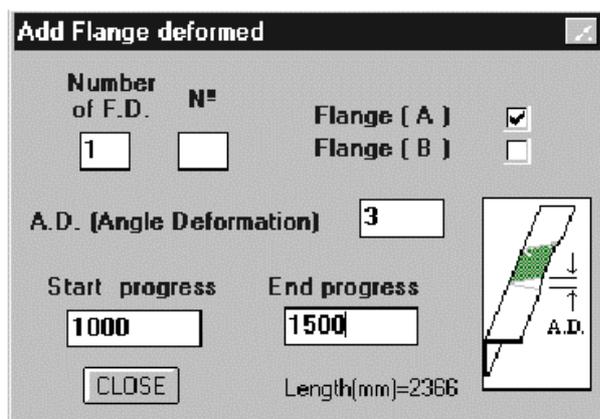


Ilustración 12-15. Doblado central de la barra

Para definir totalmente la deformación del ala de la barra se han de introducir el número de deformaciones, el ala a deformar (A) o (B), el ángulo de deformación o elevación del ala en milímetros y las dos progresivas que lo limitan.

Add Bending: Incluye doblados en una CHAPA.

En esta opción aparece una ventana en la que se han de entrar para una chapa

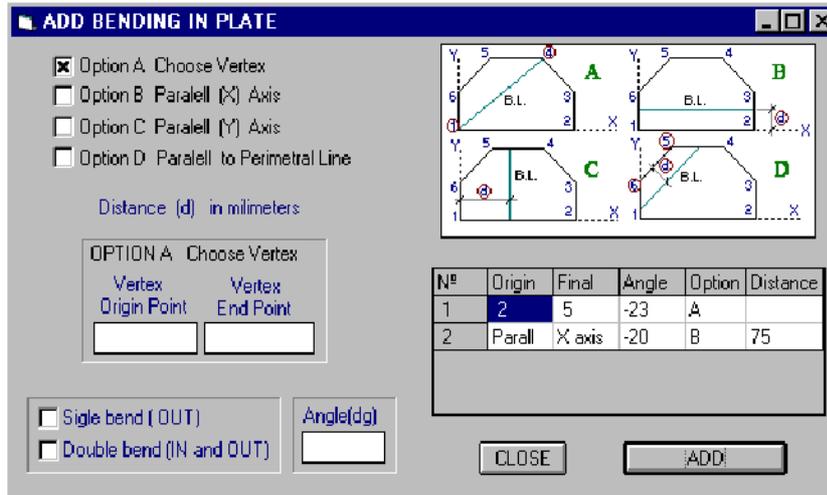


Ilustración 12-16. Elección del tipo de doblado de chapas

seleccionada los datos del doblado de forma similar a lo explicado en la adición de una nueva chapa.

12.4.4 Eliminar elementos

Delete Sketch : Elimina la barra o chapa seleccionada con sólo pulsar sobre la opción del menú.

Esta operación ha de hacerse con mucha seguridad para no perder definitivamente una planilla de la base de datos.

Delete Hole: Elimina los taladros seleccionados de la tabla de taladros en una BARRA.

Delete Cuts: Elimina cortes seleccionados de la tabla de cortes en una BARRA.

Delete Bending: Elimina los doblados de una CHAPA seleccionados de la tabla que se adjunta al pulsar sobre la opción de menú.

12.4.5 Otras modificaciones

Change Hole position : Permite cambiar la posición (gramil) y diámetro de varios taladros a la vez.

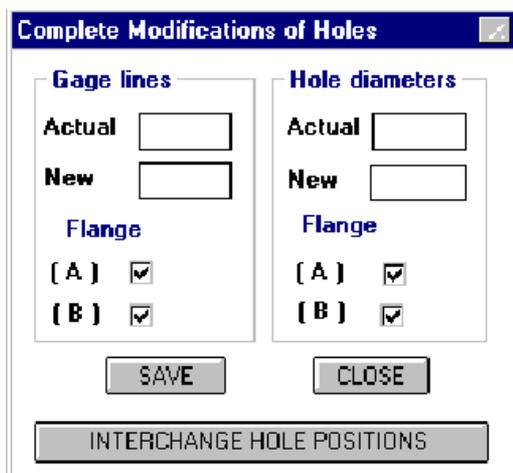


Ilustración 12-17. Cambio de gramiles

- Actual: Es el valor del gramil o el diámetro a cambiar.
- New: Es el valor por el que va a ser sustituido.

Se permite también el intercambio de la posición de los taladros respecto al ala de la barra mediante el botón al efecto. Para finalizar la operación ha de pulsar sobre SAVE.

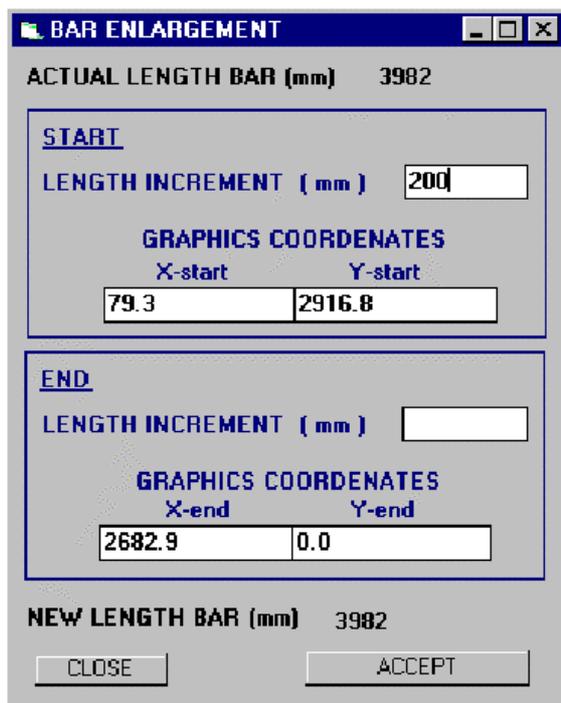


Ilustración 12-18. Alargamiento de extremos

Enlargement Bar : Alarga una BARRA por su origen o su extremo final una longitud deseada.

En el caso de la figura anterior se alargará 200 milímetros desde el extremo origen de la barra.

Section Bar:

Modifica la sección de una BARRA de la tabla de angulares de la especificación del proyecto.

SPAIN TABLE	
Cod	Section
1	35 * 35 * 3
2	40 * 40 * 3
3	35 * 35 * 4
4	50 * 50 * 3
5	40 * 40 * 4
6	45 * 45 * 4
7	50 * 50 * 4
8	45 * 45 * 5
9	60 * 60 * 4
10	50 * 50 * 5
11	60 * 60 * 5
12	70 * 70 * 5
13	60 * 60 * 6
14	70 * 70 * 6

Ilustración 12-19. Cambio de angular

12.5 Lista de Materiales

La lista de materiales es la expresión tabulada de todo el material contemplado en el proyecto, de manera que a través de la ventana que se visualiza podemos escoger y seleccionar (*SELECTED*) tanto los angulares y las chapas como la tornillería que se ajuste a la sección de la lista.

Las listas están por tanto estructuradas en secciones o cuerpos de la torre proyectada y sólo serán impresas las planillas y tornillos, previa visualización mediante el botón *SHOW OUTPUT LIST*, aquellas que se encuentren seleccionadas.

The screenshot shows the 'MATERIAL LIST' window with the following data:

Code	Material	Quality	Length	U/Weight	Quant	T/Weight	C)	R)	G)	Selection	
309	PB809	120 * 120 * 10	3600	514	9.35	4	37.42		*	*	SELECTED
311	PB810	45 * 45 * 4	2600	706	1.93	4	7.74			*	SELECTED
312	PB811	45 * 45 * 4	2600	913	2.50	4	10.01			*	SELECTED
313	PB812	45 * 45 * 4	2600	1363	3.74	4	14.94			*	SELECTED
314	PB813	45 * 45 * 4	2600	1540	4.22	4	16.88			*	SELECTED
315	PB814	45 * 45 * 4	2600	1618	4.43	4	17.73			*	SELECTED

Code	Material	Quality	U/Weight	Quant	T/Weight	C)	G)	Selection
PLATE	PB801	143 * 83 * 5	2600	0.470	2	0.930		SELECTED
PLATE	PB802	160 * 77 * 5	2600	0.480	2	0.970		SELECTED
PLATE	PB803	149 * 111 * 5	2600	0.650	2	1.300	*	SELECTED

Quantity	Diameter	Length	Quality	U/Weight	T/Weight	Selection
56	16	30	3000	0.138	7.73	SELECTED
92	16	35	3000	0.146	13.43	SELECTED
12	16	40	3000	0.154	1.85	SELECTED

Buttons: All Selection, Mostrar salida (ESP), Show output list (ENG), Total Weigth, CANCEL

Ilustración 12-20. Creación de listas de materiales

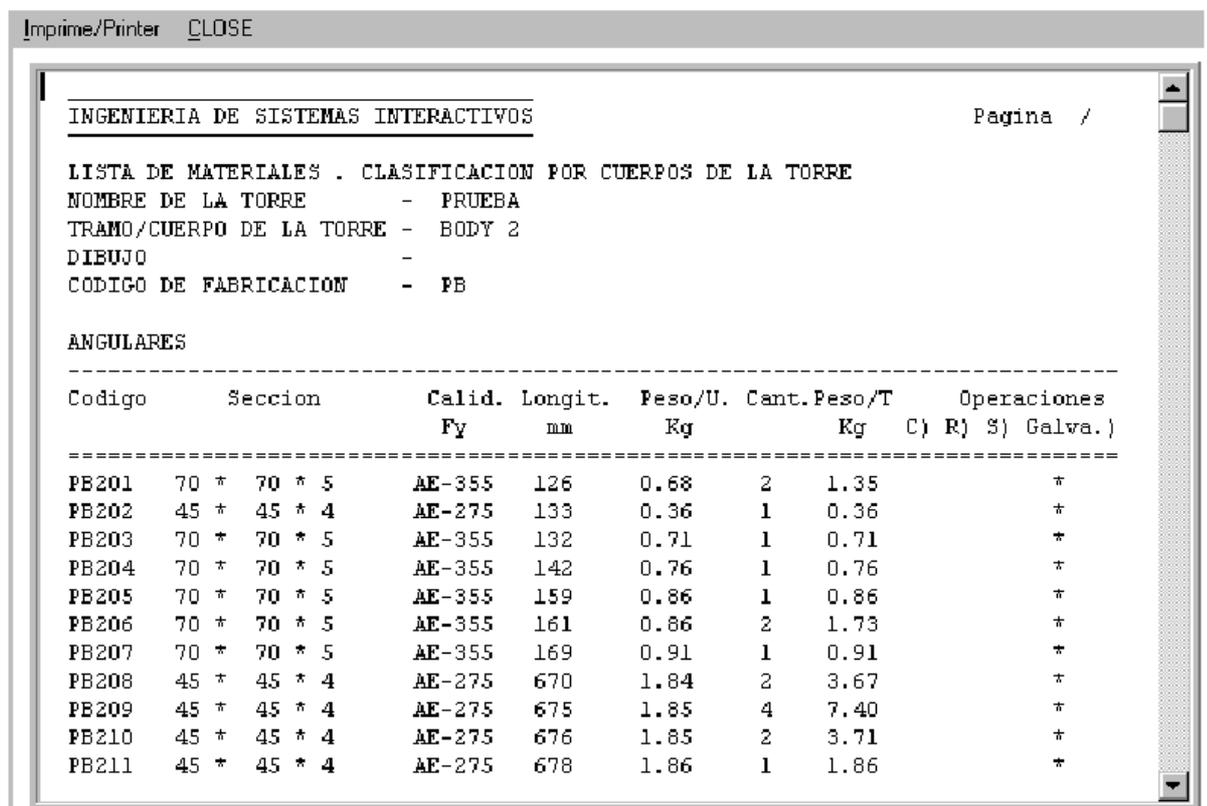
Para seleccionar las barras, chapas y tornillos podemos hacerlo en su totalidad pulsando *ALL SELECTION*, o mediante la selección manual con los botones *SEL BARS*, *SEL PLATES* y *SEL BOLTS*.

Si se desea modificar algún dato desde la lista de materiales lo hacemos pulsando sobre la fila del elemento y variamos en las casillas de texto bajo la tabla el dato en cuestión. Estos cambios, que sólo son posibles en las chapas y los tornillos, quedaran guardados pulsando la opción *SAVE* de la ventana.

En las barras podemos variar la propiedad de la cantidad presionando con el botón derecho del ratón sobre la fila de la barra e introduciendo el nuevo valor en la ventana de texto flotante que aparece.

Podemos adicionar y eliminar chapas y tornillos pulsando sobre la opción *NEW* y *DEL* de cada una de las tablas, definiendo posteriormente sus propiedades a través de las cajas de texto.

Para visualizar la lista de materiales tal y como se imprime será necesario pulsar *SHOW MATERIAL LIST*. Le aparecerá una pantalla tal y como ésta:



Imprime/Printer CLOSE

INGENIERIA DE SISTEMAS INTERACTIVOS Pagina /

LISTA DE MATERIALES . CLASIFICACION POR CUERPOS DE LA TORRE
 NOMBRE DE LA TORRE - PRUEBA
 TRAMO/CUERPO DE LA TORRE - BODY 2
 DIBUJO -
 CODIGO DE FABRICACION - PB

ANGULARES

Codigo	Seccion	Calid. Fy	Longit. mm	Peso/U. Kg	Cant.	Peso/T Kg	Operaciones C) R) S) Galva.)
PB201	70 * 70 * 5	AE-355	126	0.68	2	1.35	*
PB202	45 * 45 * 4	AE-275	133	0.36	1	0.36	*
PB203	70 * 70 * 5	AE-355	132	0.71	1	0.71	*
PB204	70 * 70 * 5	AE-355	142	0.76	1	0.76	*
PB205	70 * 70 * 5	AE-355	159	0.86	1	0.86	*
PB206	70 * 70 * 5	AE-355	161	0.86	2	1.73	*
PB207	70 * 70 * 5	AE-355	169	0.91	1	0.91	*
PB208	45 * 45 * 4	AE-275	670	1.84	2	3.67	*
PB209	45 * 45 * 4	AE-275	675	1.85	4	7.40	*
PB210	45 * 45 * 4	AE-275	676	1.85	2	3.71	*
PB211	45 * 45 * 4	AE-275	678	1.86	1	1.86	*

Ilustración 12-21. Salida de listas de materiales

En la barra de menús aparece la opción *PRINTER* para la salida de la lista a la impresora.

12.6 Configuración

Pulsando sobre la opción correspondiente de la barra de menús principal, podemos entrar en la pantalla de configuración de las cabeceras y datos de las planillas y listas de materiales.

CONFIGURATION

Sketch Title: **HIGHT TENSION LINE**

Tower Name: **PRUEBA** Tower Body: **BODY 2**

Sketch Notes: **[SEE MATERIAL LIST]**

Del Notes

Max. Holes only one face: **30** Material quality L: **AE-275** H: **AE-355**

Sketch Mode

ISI AVANGAN HELESA

Apply factor weight plate: **1.0** Print bending leg **ACCEPT**

Ilustración 12-22. Datos de salida de resultados

Mediante esta aplicación podemos configurar:

- Título del proyecto o planilla
- Nombre de la torre
- Nomenclatura de la sección o cuerpo de la torre.
- Notas y observaciones de la planilla
- Máximo número de taladros por cara de la barra y hoja de planilla (si el número de taladros es mayor que el fijado preparará la planilla en dos páginas)

- Nomenclatura de las calidades alta y baja del acero del material
- Selector de logotipos de las cabeceras de planillas
- Factor de aplicación del peso de las chapas referidos a la relación:
- área real / área máxima respecto a sus vértices
- Visualización del doblado en los montantes (SI/NO)

Para guardar las modificaciones se deberá pulsar ACEPTAR.

12.7 Documentación

La documentación proporcionada por MATERIAL se resume en :

- *Planillas de fabricación:* Impresas mediante la selección de las barras o chapas de la tabla de material y a través de la opción *PRINTER SCKETCHS* (para planillas con formato libre) o *ISI SCKETCHS* (para planillas con formato ISI de papel impreso) de la barra de menú.
- *Listas de materiales:* mediante la opción de la ventana de visualización previa de la lista de materiales antes descrita.

En ambos casos el sistema permite configurar el tipo de impresión deseada mediante la ventana de configuración de impresión.

13. Bibliografía

- [AA90] Arrillaga, J.; Arnold, C.P. "Computer Analysis of Power Systems". John Wiley, 1990.
- [AA99] R. Argüelles Álvarez, R. Argüelles Bustillo, F. Arriaga Martitegui, J. R. Atienza Reales."Estructuras de Acero: Cálculo, Norma Básica y Eurocódigo". Ed. Bellisco Ediciones Técnicas y Científicas. Madrid. España. 1999
- [Ac67] Acher, J. "Algebre lineaire et programmation lineaire". Dunod. Paris. 1967
- [AD00] Atwater, P.; DeHaan, J.; Romero, L. "Utilities Field Test Safety Grounds". Transmission & Distribution World. Denver. USA. 2000
- [AD97] Alexandrov, N., Dennis, J. E., Lewis, R. M., and Torczon, V., "A Trust Region Framework for Managing the Use of Approximation Models in Optimization," ICASE Report No. 97-50, 1997.
- [AG90] Arochena,H.E., Guerra,F. "Sistema de Optimización de Corte de Piezas Rectangulares". Revista Investigación Operacional, Vol.XI,No.3, Cuba. 1990
- [AH88] Allen, M. B., Herrera, I. y Pinder, G. F. " Numerical Modeling in Science and Engineering", John Wiley & Sons, 1988.
- [AH97] Ansari, N., and Hou, E. "Computational Intelligence for Optimization", Kuwer Academic Publishers Boston, USA. 1997
- [Ak89] Akin, J. E. "Application and Implementation of Finite Element Methods". Academic Press. Houston. Texas. USA. 1989.
- [AM01] Alberca, P.; Martín, D. "Métodos matemáticos: álgebra lineal y geometría". Aljibe. Málaga. España. 2001
- [AR86] Argüelles Álvarez, R. "Cálculo de estructuras". Edit. Fundación Conde del Valle de Salazar. Madrid . España. 1986.
- [AS88] American Society of Civil Engineers. "Guide for design of steel transmission towers". Edit. ASCE. New York. EE.UU. 1988.
- [Ba95] Bathe, K. J.: Finite Element Procedures, Prentice-Hall, Englewood Cliffs, 1995.

- [Ba00] Bastian, A. "Identifying Fuzzy Models Utilizing Genetic Programming", Fuzzy Sets and Systems, 113, 333-350, 2000.
- [BB66] Bergendal, G.; Brinck, I. "Lineär Algebra och Geometri". Hakan Ohlssons. Lund. Suecia. 1972
- [BB88] Brassard, G.; Bratley P. "Algorithmics. Theory and practice". Prentice Hall. Montreal. Canadá. 1988.
- [BC81] Becker, E. B.; Carey, G. F. ; Oden, J. T. "Finite Elements: An Introduction", Prentice Hall, 1981.
- [Be73] Bericua Llamas, A. "Truss, structural analysis program to design planar structures". Artículo. Sydney. Australia. 1973.
- [Be97] Berg. M. "Computational Geometry: Algorithms and Applications", Springer, 1997.
- [BF98] Burden, R.L.; Faires, J.D. "Análisis Numérico". Thomson International. México. 1998.
- [BM97] R. L. Brockenrough y F. S. Merritt . "Manual de Diseño de Estructuras de Acero". Ed. McGraw-Hill Santa Fe de Bogotá. Colombia. 1997
- [BN98] Banzhaf, W.; Nordin, P.; Keller, R.E. & Francone, F.D. "Genetic Programming: An Introduction: On the Automatic Evolution of Computer Programs and Its Applications", Morgan Kaufmann Publishers, ISBN: 155860510X, 1998.
- [BO94] BOE 313/1994. "Ley 40/1994 de 30 de diciembre de 1994, de ordenación del sistema eléctrico nacional". España. 1994
- [BO97] BOE 285/1997. "Ley 54/1997 de 28 de Noviembre de 1997, del Sector Eléctrico". España. 1997
- [Br85] Brassard, G. "Crusade for a better notation". Sigact News. ACM, vol. 17. Montreal. Canadá. 1985.
- [BR75] Bunch, J.R.; Rose, D.J. "Sparse matriz Computations". Academic Press. San Diego. USA. 1975
- [BS93] Bazaraa, M.; Sherali, H.; Shety, C. "Nonlinear Programming", Second Edition. Wiley, 1993.

[Bu94] Buckley, J.J. & Hayashi, Y. "Fuzzy Genetic Algorithm and Applications", Fuzzy Sets and Systems, 61, 129-136, 1994.

[BW93] Blanton, J.L., and Wainwright, R.L. "Multiple Vehicle Routing with Time and Capacity Constraints using Genetic Algorithms", Proc. 5th International Conference on Genetic Algorithms, Morgan Kauffmann Pub., San Mateo, California, pp. 452-459. 1993

[Ca95] Cammesa "Procedimientos para la programación de la operación, el despacho de cargas y el cálculo de precios". Tomos I y II. Buenos Aires, Argentina. 1995

[Ca01] Cadavid, J.D. "Construcción de torres para equipos de conversión de energía eólica". Revista de Aerodinámica 2001. Dpto. de Ingeniería Mecánica- Universidad de los Andes. Colombia. 2001

[CB99] Chandrupatla, T. R.; Belegundu, A. D. "Introducción al estudio del Elemento Finito en ingeniería", Prentice Hall, 1999

[CC95] Chen, P.; Chang, H. "Large-scale economic dispatch by genetic algorithm". 1995 IEEE/PES Winter Meeting , New York .

[CC98] Corz, A.; Carillo, S. "Aplicación del método de conos para el análisis acústico de grandes espacios". I Simposio Iberoamericano de Acústica (Lisboa). Publicación de la SEA. Málaga. España. 1998

[Ce94] Ceballos, F. J.. "Enciclopedia de Microsoft visual Basic". Edit. Ra-Ma.. Madrid. España 1994.

[Ch88] L. Checa, M. "Líneas de Transporte de Energía", Marcombo Boixareu Editores, 1988

[CH93] Chang, K. J., Haftka, R. T., Giles, G. L., and Kao, P. -J., "Sensitivity-Based Scaling for Approximating Structural Response," Journal of Aircraft, 30, 1993, pp. 283-288.

[CK76] Chang, L.; Korsh, J. "Canonical coin changing and greedy solutions". Journal ACM, vol. 23. USA. 1976

[CK01] Caccetta, L.; Kulanoor, A. "Algorithms for some hard knapsack problems" Kluwer Acad. Publ. 2001. Dordrecht. Holanda

- [CM96] Campos, V. y Mota, E. "Metaheurísticos para el CVRP", Actas del II Symposium de Ingeniería de los Transportes, Dep. de Transportes de la Univ. Politécnica de Madrid, (II):11-12. 1996
- [Co89] Cook, R. D.: Concepts and Applications of Finite Element Analysis, J. Wiley & Sons, New York, 1989.
- [Co95] Cook, R. D.: Finite Element Modeling for Stress Analysis, J. Wiley, New York, 1995.
- [CO83] Carey, G. F. ; Oden, J. T. "Finite Elements: A Second Course", Prentice Hall, 1983.
- [CO84] Carey, G. F. ; Oden, J. T. "Finite Elements: Computational Aspects", Prentice Hall, 1984.
- [CO98] Coz, J.J.; Ordieres, J. "Fundamentos para la modelización de la estructura". Manual de utilización del programa Solve. Cap. Escal3D. La Rioja.España. 1998
- [CP02] Carballido, J.; Ponzoni, I. y Brignole, D. "Un algoritmo genético basado en números difusos triangulares". Artículo. Bahía Blanca, Argentina. 2002
- [Cr86] Crisfield, M. A.: Finite Elements and Solution Procedures for Structural Analysis, Vol. 1: Linear Analysis, Pineridge Press, Swansea, UK, 1986.
- [CR90] Chowdhury, B. H. - Rahman, S. (1990) "A review of recent advances in economic dispatch". IEEE Trans. on Power Systems, Vol. 5, No. 4, pp. 1248-1259.
- [CW90] Conde, C.; Winter, G. "Métodos y algoritmos básicos del álgebra numérica". Reverte. Madrid. 1990
- [Da96] Davis, L. "Handbook of Genetic Algorithms". International Thompson Computer Press. Boston (USA). 1996
- [De90] Desrochers, M. "A Clasification Scheme for Vehicle Routing and Scheduling Problems". European Journal of Operational Research, 46 (1990), pp. 322-332. 1990
- [DG89] Díaz,P.,Guerra,F.Sistema de Optimización de Corte de Piezas Unidimensionales. Revista Investigación Operacional, Vol.X,No.2. Cuba 1989.
- [DG96] Diaz, A., Glover, F., Ghaziri H.M., González, J.L., Laguna, M., Moscato, P. y Tseng, F.T."Optimización Heurística y Redes Neuronales en Dirección de Operaciones e Ingeniería". Editorial Paraninfo S.A., Madrid. 1996

- [DX94] G. Dejuan, Z. Xiangzhou, "Modified Designs and Modified Computations". Computers and Structures, Vol 49, No 6. 1994.
- [EA97] Eby, D., Averill, R. C., Gelfand, B., Punch, W. F., Mathews, O., and Goodman, E. D., "An Injection Island GA for Flywheel Design Optimization" Invited Paper, Proceedings of the 5th European Congress on Intelligent Techniques and Soft Computing, Sept., 97.
- [FC96] Faulkenberry, L.M.; Coffey, W. "Electrical Power Distribution and Transmission". Prentice-Hall, 1996
- [Fe73] Feller, W. "Introducción a la teoría de probabilidades y sus aplicaciones". Wiley. Princeton. Madrid. España. 1973
- [FH96] Furuya, H., and Haftka, R. T., "Combining Genetic and Deterministic Algorithms for Locating Actuators on Space Structures," Journal of Spacecraft and Rockets, Vol. 33, No. 3, 1996, pp. 422-427.
- [FM96] Félez, J.; Martínez, M.L.; Cabanellas, J.M.; Carretero, A. "Fundamentos de Ingeniería Gráfica". Síntesis. Madrid. 1996.
- [Fo94] Fogel, D. B. "An introduction to simulated evolutionary optimization" IEEE Trans. on Neural Networks, Vol 5, No. 1, 1994, pp 3-14.
- [Fo02] Fonseca, J. "Ferramentas de Simulação em Mecânica: Elementos Finitos". Apostila da Pós-Graduação em Engenharia Mecânica. Universidade Federal do Rio Grande do Sul, em Porto Alegre, Rio Grande do Sul, Brasil. 2002.
- [FR84] Ford, B.; Rault, J.C.; Thomasset, F. "Tools, Methods and Languages for scientific and engineering computation". Elsevier. Oxford. Inglaterra. 1984.
- [Ga75] Gallagher, R. H. "Finite element analysis. Fundamentals". Edit. Prentice Hall. New Jersey. USA. 1975.
- [Ga93] M. Galante, "Un algoritmo genético simple para la optimización de estructuras planas articuladas". Revista Internacional de Métodos Numéricos, para Cálculo y Diseño en Ingeniería, Vol 9,2,1993.
- [Ga98] Garcia Castellanos, D. "Modelos numéricos 3D". Incluido en PhD Thesis *Desarrollo de modelos numéricos de flexión litosférica*. CSIC. Barcelona. España. 1998

[Gal98] Galán García, JL " Análisis vectorial para la Ingeniería. Teoría y problemas". Editorial Bellisco, 1998

[GC00] Gen, M. & Cheng, R. "Genetic Algorithms and Engineering Optimization", Wiley Series in Engineering Design and Automation, John Wiley & Sons, ISBN: 0471315311, 2000.

[Gh96] Ghaziri, H.M. "Redes Neuronales en Optimización Heurística y Redes Neuronales en Dirección de Operaciones e Ingeniería". Editorial Paraninfo S.A., Madrid, pp. 163-207. 1996

[Gi95] Gipe, P. "Wind turbine tower trends". John Wiley & Sons USA. 1995

[GK81] Greene, D. H.; Knuth, D. E. "Mathematics for the analysis of algorithms". Birkhauser. Boston. USA. 1981

[GK94] Graham, R. L.; Knuth, D. E.; Patashnik, O. "Concrete Mathematics". Addison-Wesley. Stanford University. USA. 1994

[Go74] Godement, R. "Álgebra". Tecnos. Madrid. España. 1974

[Go80] Gordon, G. "Simulación de Sistemas". Diana. México 1980

[Go89] Goldberg, D.E. "Genetic Algorithms in Search, Optimization and Machine Learning", Addison-Wesley Pub Co. 1989.

[Go01] González Rodríguez, C. "Una nueva formulación del algoritmo genético aplicado al diseño óptimo de estructuras" Artículo. Medellín. Colombia. 2001.

[GO97] Goodman, J.; O'Rourke, J."Handbook of Discrete and Computational Geometry" CRC Press, 1997

[Gö88] Gönen, T. "Electric Power Transmission System Engineering. Analysis and Design". John Wiley, 1988.

[GP93] D.E. Grierson, W.H. Pak, "Optimal Sizing, Geometrical and Topological Design Using Genetic Algorithm". Structural Optimization, 6, 1993.

[GS87] D.E. Goldberg, M.P. Samtani, "Engineering Optimization Via Genetic Algorithm", Proceedings of the Ninth Conference on Electronic Computation, Asce, 1987.

- [GT96] Goonatilake, S., and Treleaven, P. "Intelligent Systems for Finance and Business", John Wiley, 335p. 1996
- [GV85] J. Garcia de Jalon, F. Viadero, "A new Direct Method for the Simple and Efficient Reanalysis of Structures". Computers and Structures, Vol 21, No 5, 1985.
- [Ha72] Hanner, O. "Geometri och Lineär Álgebra". Gleerups. Lund. Suecia. 1972
- [HA75] Holland, J. H. Adaption in Natural and Artificial Systems. Cambridge, MIT Press. (1975)
- [HB96] Hernández, J.; Brignone, S.; Ruetsch, L.; Moitre, D.; Aromataris, L. "Planificación de la operación de sistemas de energía via optimización evolutiva". Anales del 7º Congreso Latinoamericano de Control Automático. Bs As, Argentina. 1996. Vol I, pp. 266 - 271.
- [He90] S.I. Hernandez, Métodos de diseño óptimo de estructuras, Colegio de Ingenieros de Caminos, Canales y Puertos, Aragón, 1990.
- [HH93] F. Hiroshi, R. Haftka, "Locating Actuators for Vibration Supression on Space Trusses by Genetic Algorithms". Proceedings of the 1993 ASM Winter Annual Meeting, 1993.
- [HK85] R.T. Haftka, M.P. Kamat, Elements of Stuctural Optimization, Martinus Nijhoff Publishers, 1985.
- [HL92] P. Hajela,, C.Y. Lin, "Genetic Search Strategies in Multicriterion Optimal Design". Stuctural Optimization, 4, 1992.
- [HL91] Hillier, F.S.; Lieberman, G. J. "Introduction to operations research" Edit. Mc Graw Hill. Standford. EE.UU. 1991.
- [HL97] Herrera, F.; Lozano, M. & Verdegay, J.L. "Fuzzy Connectives Based Crossover Operators to Model Genetic Algorithms Population Diversity", 92, 21-30, 1997.
- [Ho92] J.H. Holland, "Genetic Algorithms", Scientific American, September, 1992.
- [HO78] Hinton, E. ; Owen, D. R. J. "An introduction to the Finite Element Computations", Academic Press, 1978.
- [Hu87] Hughes, T. J. R.: Finite Element Method - Linear Static and Dynamic Finite Element Analysis, Prentice-Hall, Englewood Cli .s, 1987.

[Hu02] Huerta, J. “Modelado de Sólidos”. Apuntes de Diseño y Fabricación asistida por Ordenador. Universidad Jaume I. Castellón. España. 2002.

[HW00] Huang, Y.-P. & Wang, S.-F. “Designing a Fuzzy Model by Adaptive Macroevaluation Genetic Algorithms”, *Fuzzy Sets and Systems*, 113, 367-379, 2000

[HY96] Hajela, P., and Yoo, J., "Constraint Handling in Genetic Search Using Expression Strategies," *AIAA Journal*, Vo. 34, No. 11, 1996, pp. 2414-2420.

[IE96] Special Section on Genetic Algorithms for Industry. *IEEE Trans. on Industrial Electr.*, Vol 43, No 5. , (1996).

[IM90] IMEDEXSA (Industrias Mecánicas de Extremadura S.A.) . “Torres metálicas para líneas eléctricas de media tensión”. Manual. Cáceres. España. 1990.

[IS97] ISI (Ingeniería de Sistemas Interactivos Ltda.) . “Computer integrated transmission towers system”. Manual. Málaga. España. 1997.

[Je91] W.M. Jenkins, "Towards Structural Optimization Via the Genetic Algorithm", *Computers and Structures*, 40,pp, 1321-1327, 1991.

[Je92] W.M. Jenkins, "Plane Frame Optimum Design Environment Based on Genetic Algorithm". *J.of Structural Engineering*, November,1992.

[Je97] W.M. Jenkins, "On the Application of Natural Algorithms to Structural Design Optimization". *Engineering Structures*, V.19, No.4, 1997.

[JM99] Joyanes, L.; Muñoz, A. “Microsoft, Visual Basic 6.0”. Mc Graw Hill. Universidad Pontificia de Salamanca. España. 1999.

[JM00] Johnson, D. B.; Mowry, T. A. “Matemáticas Finitas”. Thomson International. México. 2000.

[Jo87] Joyanes Aguilar, L. “Metodología de la programación” Edit. Mc Graw Hill. Madrid. España. 1987.

[Ju95] Julstrom, B.A.”What Have You Done for Me Lately? Adapting Operator Probabilities in a Steady-State Genetic Algorithm”. *Proc. 6th International Conference on Genetic Algorithms*, Morgan Kauffmann Pub., San Mateo, California, pp. 81-87. 1995

- [JV85] J. Garcia de Jalon, F. Viadero, "A new Direct Method for the Simple and Efficient Reanalysis of Structures". Computers and Structures, Vol 21, No 5, 1985.
- [KG94] V.K. Koumosis, P.G. Georgiou, "Genetic Algorithms in Discrete Optimization of Steel Truss Roofs". J. of Computing in Civil Engineering, July, 1994.
- [Ki81] U. Kirsch, Optimum Structural Design, Mc.Graw-Hill, 1981.
- [Ki93] U. Kirsch, "Efficient Reanalysis for Topological Optimization". Structural Optimization, 6, 1993.
- [Ki95] Kirsch, U., "Improved Stiffness-Based First-Order Approximations for Structural Optimization", AIAA Journal, Vol. 33, No. 1, 1995, pp.143-150.
- [Ki02] Kipp, V.W. "Tower identification". Site Management & Technology. Wisconsin. USA. 2002
- [KI75] Kleinrock, L. "Queueing Systems". Wiley Interscience. Los Angeles. USA. 1975
- [KI89] Klein, R. "Concrete and Abstract Voronoi Diagrams" Springer-Verlag, 1989
- [KN96] Kodiyalam, S., Nagendra, S., and DeStefano, J., "Composite Sandwich Structural Optimization with Applications to Satellite Components," AIAA Journal, Vol. 34, No. 3, 1996.
- [KY95] Klir, G. & Yuan, B. "Fuzzy Sets and Fuzzy Logic, Theory and Applications", Prentice Hall, 1995.
- [LH96] Lee, J., and Hajela, P., "Parallel Genetic Algorithm Implementation for Multidisciplinary Rotor Blade Design," Journal of Aircraft, Vol. 33, No. 5, 1996, pp. 962-969.
- [LL92] Laporte, G., Louveaux, F., and Mercure, H. "The Vehicle Routing Problem with Stochastic Travel Times", Transportation Science, 26(3):162-170. 1992
- [Lo97] Louis, S.J. "Genetic Algorithms and Truss Design". Article. USA. 1997.
- [Ma86] Mason, J.C. "Basic Matrix Methods". Butterworth & Co. Londres. 1986
- [Ma93] T. Masters, Practical Neural Network Recipes in C++. Chapter 8, Academic Press, 1993

[Ma94] Man, K. F.- Tang, K.- Kwong, S. "Genetic Algorithms: Concepts and Applications" IEEE Trans. on Industrial Electronics, Vol 43, No. 5, 1996, pp 519-534.

[Ma96] Marques, O.A. "Eigensolvers and Applications in Finite Element Analyses". Advanced Solution Procedures on Innovative Computer Architectures (M. Papadrakakis and G. Bugeda, Eds.), 66-79, CIMNE Publications, 1996.

[Ma98] Marco García, J. "Fundamentos para el cálculo y diseño de estructuras metálicas de acero laminado". McGraw-Hill, España. 1998

[Ma00] Marco García, J. "Curso básico de cálculo y diseño de estructuras metálicas en ordenador". McGraw-Hill, España. 2000

[Mar00] Marcelo, B. "Corte Certo 2000. Sistema de optimización de cortes para chapas". Silicon Valley. USA. 2000

[MA96] Moitre,D.; Aromataris,L.; Hernández,J. "Evolución de los Mercados de Energía Eléctrica Argentino y Chileno". Sesiones Técnicas del Congreso: Transiciones Energéticas en México, Centro y Sudamérica. México, D.F., México, 1996.

[MA97] Moitre,D.; Aromataris,L.y otros. "Aplicación de Algoritmos Genéticos a la Coordinación Hidrotérmica de Corto Plazo". Artículo. México, D.F., México, 1997.

[MA01] Magnusson, P. C.; Alexander, G. C.; Tripathi, V. K.; Weisshaar, A. "Transmission lines and wave propagation". Ed. CRC Press,2001

[MB96] Moitre, D; Brignone, S.; Ruetsch ,L.; Maldonado, F.; Hernández, J.; Aromataris, L. "Técnicas emergentes en la optimización de sistemas de suministro de energía eléctrica". Ponencias del II Congreso Latinoamericano de Distribución de Energía Eléctrica, Viña del Mar, Chile.1996.

[Me91] Merino Córdoba, S. "Análisis de residuo mínimo" Tesis de Grado. Facultad de Ciencias Matemáticas. Universidad de Málaga. Dirección : Javier Erice. Málaga. España. 1991.

[Me95] Merino Córdoba, S. "Técnicas algorítmicas para almacenamiento" Artículo para la Secretaría General Técnica del Ministerio de Defensa. Málaga. España. 1994.

[Me01] Medina, I. "Algoritmos para la simulación de ordenación espacial". Proyecto fin de carrera dirigido por Salvador Merino. Málaga. España. 2001

- [Me02] Medina, J.R.. “Algoritmos genéticos para la optimización de redes de distribución”. Artículo. Valencia. España 2002.
- [Me03] Merino Córdoba, S. “Cálculo Inverso, Diagramas de Utilización”. Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería (para publicación en Diciembre) . 2003
- [Mer02] Merelo Guervós, J.J.. “Informática Evolutiva: Algoritmos Genéticos”. Artículo. Universidad de Granada. España 2002.
- [MF00] Mathews, J.H.; Fink, K.D. “Métodos Numéricos con Matlab”. Prentice Hall. Madrid. España 2000.
- [MG95] Miller, B. L., and Goldberg, D. E., "Genetic Algorithms, Tournament Selection and the Effects of Noise," IlliGAL Report No. 95006, University of Illinois at Urbana-Champaign, 1995.
- [MH96] Moitre, D; Hernández, J.; Ruetsch ,L.; Maldonado, F.; Aromataris, L. "Optimización del Despacho de un Sistema de Suministro de Energía Eléctrica combinando Técnicas Determinísticas y Estocásticas". Sesiones Técnicas del Congreso:Transiciones Energéticas en Mexico, Centro y Sudamérica, México,D.F., México. 1996.
- [MH97] Moitre, D.- Hernández, J.- Aromataris, L.- Rodriguez, G. “Programación de la Operación Óptima Económica de Sistemas Eléctricos de Potencia sujeta a restricciones energéticas: una aplicación de Algoritmos Genéticos”. RVP’97 IEEE Sección Mexico. Acapulco, Mexico. 1997.
- [Mi96] Michalewicz, Z. “Genetic Algorithms + Data Structures = Evolution Programs”, 3ra edición revisada, Springer Verlag, ISBN: 3540606769, 1996.
- [Mi97] Miller, B. L., "Noise, Sampling and Efficient Genetic Algorithms," PhD Dissertation, University of Illinois at Urbana-Champaign, 1997.
- [Mo86] Mole, R.H. “Cálculo Numérico”. Anaya Multimedia. Madrid. España. 1986
- [Mo88] Moreno Clemente, J. “Cálculo de líneas eléctricas aéreas de alta tensión”. Manual. Málaga. España. 1988.
- [MO01] Merino, S. y Ojeda, M. “Preliminares sobre la automatización del diseño de torres”. Congreso ACIIC2001. Cádiz. España. 2001

[MT90] Martello, S. y Toth, P. "Knapsack Problems" Edit. Wiley. Chichester. Inglaterra. 1990.

[MV99] Mata, M.; Vinacua, A.; Brunet, P. "Inclusión de metodos de ingeniería inversa en sistemas CAS/CAD para la optimización del Proceso de diseño de productos" CEIG'99 IX Congreso Español de Informática Gráfica, Jaen. 1999

[Na00] Navarro, J.A. "Estaciones de Ensayo. Historia y Funcionamiento". Documentación diversa y entrevista personal con el responsable de ensayos durante 30 años del Centro Europeo de Investigación y Ensayo de Torres. Madrid. 2000

[NC95] K. Nukulchai, Ch.Ch. Chung, L.Y. Chage, "Efficient Genetic Algorithms for Space Truss Optimization". Proceedings of the Fifth East Asia Pacific Conference on Structural Engineering and Construction, July, 1995.

[NH96] Nagendra, S., Haftka, R. T., Gurdal, Z., and Watson, L. T., " Improved Genetic Algorithms for the Design of Stiffened Composite Panels, " Computers and Structures, Vol. 58, No. 3, 1996, pp. 543-555.

[NK98] P.B. Nair, A.J. Keane. "Combining approximation concepts with genetic algorithm-based structural optimisation procedures". Proceedings of the 39th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Long Beach, CA, April 1998, AIAA 98-1912

[NK02] Nair, P. B., Keane, A. J., and Langley, R. S., "Improved First-Order Approximation of Eigenvalues and Eigenvectors," AIAA Journal, to appear.

[Ok92] Okabe, A. "Spatial Tessellations, Concepts and Applications of Voronoi Diagrams", John Wiley & Sons, 1992

[Ol70] Olesten, N.O. "Numerical control". Wiley interscience. La Jolla. California. USA. 1970

[Oñ92] Oñate, E. "Cálculo de estructuras por el método de los elementos finitos" Edit. CIMNE. Barcelona. España. 1992.

[OP92] Ottosen, N. S. ; Petersson, H. "Introduction to the Finite Element Method", Prentice Hall, 1992.

[Or94] O'Rourke, J. "Computational Geometry in C", Cambridge Univ. Press, 1994.

[Pe85] Pereira, M. "Optimal Scheduling of Hydrothermal Systems : An overview". IFAC Electric Energy Systems. Rio de Janeiro, Brazil. 1985.

- [Pe87] Peña Sánchez de Rivera, D. "Estadística, modelos y métodos". Alianza Universidad. Madrid. España. 1987
- [Pr92] Prieto, M. "Fundamentos geométricos del diseño en ingeniería". ADI. Madrid. 1992.
- [Ra73] Ras, E. "Teoría de líneas eléctricas". Ed. Marcombo Boixareu, 1973
- [RA92] Rifaldi, A.; Arnera, P.; Giacopinelli, A. "Diseño de Líneas Eléctricas". Apuntes de la Facultad de Ingeniería. Dpto. Electrotecnia Universidad Nacional de la Plata. Argentina. 1992
- [RAT] Reglamento de líneas eléctricas aéreas de Alta Tensión
- [RBT] Reglamento electrotécnico para baja tensión
- [RCE] Reglamento sobre centrales eléctricas, subestaciones y centros de transformación
- [RC02] Roberts, J.; Crandall, C.; Van Engelenhoven, J. "What is finite element analysis?". Ciras. Iowa State University. USA. 2002.
- [Re93] Reddy, J. N.: An Introduction to the Finite Element Method, McGraw-Hill Book Co., 1993.
- [Re95] AG Reinelt. "Discrete Optimization. Available Software and Data". 1995
- [RM02] R. Ramírez Borlado, C. Martínez Lasheras y R. Martínez Laceras. Prontuario de Estructuras Metálicas. Centro de Estudios y Experimentación de Obras Públicas (CEDEX).
- [RH97] Riubrugent, J.; Hernández, J.; Moitre, D.; Aromataris, L. "Control de Sistemas de Potencia: Aplicación de Algoritmos Genéticos a la Programación de la Operación". SIE'97. Santa Clara, Cuba., 1997.
- [Sa91] M.P. Saka, "Finite Element Applications of the Theorems of Structural Variation". Computers and Structures, Vol 41, No 3, 1991.
- [SB73] Szabo, B.; Babuska, I. "Finite Element Analysis", Prentice Hall, 1973.
- [SB94] Sheble, G. B.; Brittig, K. "Refined Genetic Algorithm - economic dispatch example". IEEE Trans. on Power Systems, Vol 10, No 1. , 1994. pp. 117-124.

- [Sc02] Schafer, B.W. "What is a buckling curve?. Elastic buckling by the finite strip method". Tutorials. Johns Hopkins Univ. Baltimore. USA. 2002.
- [Se85] Sewell, G. "Analysis of a Finite Element Method". Springer-Verlag. New York. USA. 1985.
- [SH96] Sobieszczanski-Sobieski, J., and Haftka, R. T., "Multidisciplinary Aerospace Design Optimization : Survey of Recent Developments," 34th AIAA Aerospace Sciences Meeting and Exhibit, AIAA Paper No. 96-0711, Reno, Nevada, 1996.
- [SU00] Sack, J.-R. ,Urrutia, J. "Hanbook of computational geometry" Elsevier Science Publishers B.V. North-Holland, 2000.
- [Th95] Thangia, S.R. "An Adaptative Clustering Method using Geometric Shape for Vehicle Routing Problems with Time Windows", Proc. 6th International Conference on Genetic Algorithms, Morgan Kauffmann Pub., San Mateo, California, pp. 536-543.1995
- [TY79] Timoshenko, S.; Young, D. H. "Elementos de resistencia de materiales". Montaner y Simón, Barcelona. España. 1979.
- [Wa91] Wadhwa, C.L. "Electrical Power Systems". Segunda Edición, John Wiley, 1991.
- [Wi85] Winter, P. "An algorithm for the Steiner problem in the Euclidean plane". Networks, vol. 15. 1985
- [Wi86] Wirth, N. "Algoritmos + estructuras de datos = programas" Edit. Del Castillo. Madrid. España. 1986.
- [Wi87] Winter, P. "Steiner problem in networks". Networks, vol. 17. 1987
- [Wi97] Widas, P. "Introduction to Finite Element Analysis". Virginia Tech Material Science and Engineering. USA. 1997
- [WJ84] Weaver, W.; Johnson, P.R. "Finite Element for Structural Analysis". Prentice Hall. Stanford University. USA. 1984.
- [WS93] Walters, D. C. - Sheble, G. B. (1993) "Genetic algorithm solution of economic dispatch with valve point loading". IEEE Trans. on Power Systems, Vol 8, No 3. ,pp. 1331-1352.

- [WS96] Whitley, D., Starkweather, T., and Shaner, D. "The Travelling Salesman and Sequence Scheduling: Quality Solutions using Genetic Edge Recombination". Handbook of Genetic Algorithms (L. Davis Ed.), International Thompson Computing Press, pp. 350-372. 1996
- [WW84] Wood, A. & Wollenberg, B. Power Generation, Operation, and Control. Wiley. 1984.
- [WY00] Wu, B. & Yu, X. "Fuzzy Modelling and Identification with Genetic Algorithm Based Learning", Fuzzy Sets and Systems, 113, 351-365, 2000.
- [Ya96] Yang, H.; Yang, P; Huang, Ch. "Evolutionary programming based economic dispatch for units with non-smooth fuel cost functions". 1995 IEEE/PES Winter Meeting , New York .
- [YS95] K.E Young, L.H. Soo, "A Study on Evolution Design Model Based on Genetic Algorithms". J.of the Architectural Institute of Korea, V.15, No.4, 1995.
- [YY95] Yang, H.- Yang. P- Huang, Ch. (1995) "Evolutionary programming based economic dispatch for units with non-smooth fuel cost functions". 1995 IEEE/PES Winter Meeting , New York .
- [Za65] Zadeh, L.A. "Fuzzy Sets", Inf. Control 8, 1965.
- [ZT94] Zienkiewicz, O. C.; Taylor, R. L. "El método de los elementos finitos" Edit. Mc Graw Hill. Barcelona. España. 1994.

14. Software Utilizado

AntWind

[AN00] ANDREW CORPORATION “AntWind. Version 1.0”. Programado por Allan Kennedy. Año 2000. EE.UU.

Software de libre distribución desarrollado por la compañía ANDREW para el cálculo de las fuerzas de viento y los momentos aplicado a las torres y a todo tipo de soportes de acero para antenas parabólicas tipo Andrews



[IS00] EMPRESA ISI “Torres de Comunicaciones de ISI”. Programado por ISI. Año 2000. España.

Software de libre distribución desarrollado por la compañía Ingeniería de Sistemas Interactivos Ltda.. para el diseño y cálculo de torres de comunicaciones, donde podemos establecer el entorno deseado y las antenas de mercado necesarias.



[IS01] EMPRESA ISI “Sistema de Cálculo, Diseño y Fabricación de Torres”. Programado por ISI. Año 2001. España.

Software privado desarrollado desde el año 1988 por la compañía Ingeniería de Sistemas Interactivos Ltda.. para el diseño, cálculo y fabricación de torres en general y en el que han sido aplicados los resultados expuestos en este trabajo.

Microstran

[MS97] EMPRESA EEC “MSTOWER & MAST, ver’ 5.00”. Programado por EEC. Año 1997. Inglaterra.

Software privado desarrollado por la compañía Engineering Systems Ltd. en Hertfordshire (Inglaterra) para el cálculo de torres de transmisión eléctrica y comunicaciones, Su entrada de datos es en modo texto y su versatilidad se debe a la elección de múltiples modelos de paneles, recuadros y crucetas, además del uso de sistemas de cálculo lineales (de primer orden) y no lineales (de segundo orden.)



[CA01] EMPRESA CADRE Analytic “CADRE Pro – Finite Element Structural Análisis. Versión 4.0”. Programado por CADRE. Año 2001. EE.UU.

Software privado desarrollado por la compañía CADRE Analytic en Washintong (EE.UU.) para el cálculo de estructuras planas y espaciales, permitiendo la entrada de datos de forma gráfica o en modo texto. Permite sistemas de cálculo lineales (de primer orden) y no lineales (de segundo orden.)

15. Índice Alfabético

- A**
- Álgebra, 5-1
 Al-Juwarizmi, 9-1
 Análisis
 Vectorial, 5-1
 Andrews, 7-18
 Argyris, 3-2
 ASCE, 2-5
 normativa, 3-24, 10-1
- C**
- CAD, 1-6
 Cadena de aisladores
 ecuaciones, 7-3
 CAE, 1-4
 Cálculo Inverso, 7-11
 Castigliano, 3-1, 3-7, 3-8
- Ch**
- Cholesky, 3-16
- C**
- Clough, 3-2
 Control Numérico, 5-7
 Courant, 3-2
 CPU, 9-4
 Cross, 3-2
- D**
- Desplazamiento
 Cono de, 3-28
 DIN
 normativa, 10-1
 Dutheil, 2-5
- E**
- EDS, 1-4
 Esbeltez, 2-10, 2-18
 Euler, 3-33
- G**
- Galante*, 4-2
 Gauss, 3-17
 Método de, 3-16
 por bloques, 3-16
 Seidel, Método de, 3-16
- Goldberg, 4-5, 4-14
 Greedy, 6-6
 Green, 5-34
- H**
- Hiperestructura, 7-11, 7-13
 Holland, 4-2
 Hooke
 Ley de, 3-4
 Hrenikoff, 3-2
- I**
- Inglete, 5-12
 ISI
 patentes, 7-14
 scketch, 12-24
 sistema, 1-7
- J**
- Jacobi, 3-16
 Jenkins, 4-1
- K**
- Kelsey, 3-2
- L**
- LNS
 diagrama, 7-10
 ecuaciones, 7-6
- M**
- Maney, 3-2
 Martin, 3-2
 Matriz
 de rigidez, 3-5
 Maxwell, 3-1
 McHenry, 3-2
 MEF, 3-3
 Mixtura algorítmica, 6-24
 Mohr, 3-1
- N**
- Navier, 3-1
 Newton
 Raphson, 7-11

O

Omega
 proyecto, 7-15
 tabla de pandeo, 2-5
Oñate, 3-4
Ostenfeld, 3-2

P

Patton, 3-2
Permutación
 algoritmo, 11-2

R

RAM, 3-17
Raphson
 Newton, 7-11
RAT, 1-5
Riemann, 5-34

S

Schwartz, 3-8

T

Taylor, 3-4
Timoshenko, 3-4, 3-8, 3-29
Topp, 3-2
TTE, 2-1
Turner, 3-2

U

UNE
 normativa, 10-1

V

Venant
 St., 3-1
Visual Basic, 3-19
Volta, 1-1
Von Mises, 8-5

Z

Zienkiewicz, 3-4

16. Tabla de Ilustraciones

Ilustración 2-1. Estación de Ensayos de Arak (Irán)	2-1
Ilustración 2-2. Diagrama Central del Sistema	2-3
Ilustración 2-3. Nomenclatura de Angular o Perfil en L.	2-4
Ilustración 2-4. Curva de Pandeo (o de Euler).....	2-5
Ilustración 2-5. Torre Piramidal.....	2-6
Ilustración 2-6. Torre Atirantada	2-6
Ilustración 2-7. Cabeza de torre Cara de Gato	2-6
Ilustración 2-8. Cabeza de torre de Ventana o Delta	2-6
Ilustración 2-9. Nomenclatura de las caras	2-7
Ilustración 2-10. Torre Piramidal Básica	2-8
Ilustración 2-11 Numeración de nudos	2-9
Ilustración 2-12. Panel Tipo 1	2-9
Ilustración 2-13. Panel Tipo 2.....	2-9
Ilustración 2-14. Panel Tipo 3.....	2-9
Ilustración 2-15. Panel Tipo 4.....	2-9
Ilustración 2-16. Panel Tipo 5	2-9
Ilustración 2-18. Recuadro Tipo 1	2-11
Ilustración 2-19. Recuadro Tipo 2	2-11
Ilustración 2-20. Recuadro Tipo 3	2-11
Ilustración 2-21. Recuadro Tipo 4	2-11
Ilustración 2-22. Recuadro Tipo 5	2-11
Ilustración 2-23. Recuadro Tipo 6	2-11
Ilustración 2-24. Cruceta Normal	2-12
Ilustración 2-25. Cruceta Invertida	2-12
Ilustración 2-26. Cruceta tipo Demonio para cable de tierra.....	2-12
Ilustración 2-27. Cruceta para cable de tierra	2-12
Ilustración 2-28. Cruceta especial tipo Danubio.....	2-12
Ilustración 2-29. Tipos de Patas y Extensiones de Torre.....	2-16
Ilustración 2-30. División de la torre en secciones para su fabricación	2-19
Ilustración 2-31. Tipos de Rellenos de Crucetas	2-21
Ilustración 2-32. Medidas de la cabeza de la torre Delta.....	2-23
Ilustración 2-33. Nudos en la cabeza de la torre delta.....	2-24
Ilustración 2-35. Conductores en la cabeza de la torre cara de gato.....	2-26
Ilustración 2-36. Medidas de la torre Atirantada	2-27
Ilustración 3-1. Numeración de nudos y barras	3-9
Ilustración 3-2. Método de Gauss por Bloques.....	3-15
Ilustración 3-5. Cambios de tangente.....	3-19
Ilustración 3-8. Cono de desplazamientos	3-27
Ilustración 3-9. Esquema de torre para concurso.....	3-34
Ilustración 5-1. Transformación de Outline a Plano de Fábrica	5-1
Ilustración 5-2. Distancias básicas en una barra	5-2
Ilustración 5-3. Junta a Solape	5-3
Ilustración 5-4. Junta a Tope.....	5-3
Ilustración 5-5. Dimensiones de un taladro	5-4
Ilustración 5-6. Distancias de seguridad	5-5

Ilustración 5-8. Diseño de Gramiles	5-7
Ilustración 5-10. Tipos de Cortes.....	5-11
Ilustración 5-11. Puntos principales de una barra	5-12
Ilustración 5-13. Ejemplos de Tipo 1	5-16
Ilustración 5-14. Nomenclatura de Tipo 1	5-16
Ilustración 5-15. Opciones de posición de barras en Tipo 1	5-16
Ilustración 5-16. Opciones de posición de barra 6 en Tipo 2.....	5-17
Ilustración 5-17. Nomenclatura de Tipo 2.....	5-17
Ilustración 5-18. Ejemplos de Tipo 2	5-17
Ilustración 5-19. Ejemplos de Tipo 3	5-18
Ilustración 5-20. Nomenclatura de Tipo 3.....	5-18
Ilustración 5-21. Opciones de barras exteriores en Tipo 3	5-18
Ilustración 5-22. Nomenclatura de Tipo 4.....	5-19
Ilustración 5-23. Ejemplos de Tipo 4	5-19
Ilustración 5-24. Opciones de barras exteriores en Tipo 4.....	5-19
Ilustración 5-26. Cogida de conductores sobre cruceta en punta	5-21
Ilustración 5-28. Compresión de datos	5-23
Ilustración 5-29. Modelo geométrico del Sistema Interactivo	5-24
Ilustración 5-30. Modelo geométrico de la Universidad de Vacon-Void	5-24
Ilustración 5-33. Peldaños	5-28
Ilustración 5-34. Acumulación de taladros.....	5-30
Ilustración 6-1. Algoritmo de Ordenación Decreciente	6-17
Ilustración 6-2. Algoritmo de Ordenación Creciente	6-18
Ilustración 6-3. Algoritmo de Ordenación Alternada.....	6-19
Ilustración 6-4. Algoritmo de Ordenación Mayorada	6-20
Ilustración 6-6. Carga de unidad de trabajo en unidad de almacenamiento.....	6-35
Ilustración 6-7. Puntos básicos de un rectángulo	6-36
Ilustración 6-8. Giros de una unidad de trabajo.....	6-37
Ilustración 7-1. Trazado y parámetros de una línea de transmisión eléctrica	7-2
Ilustración 7-2. Poliedro de utilización.....	7-12
Ilustración 7-4. Decisión booleana que indica el modelo de aplicación de viento.....	7-14
Ilustración 7-5. Anclaje de Antenas de ISI.....	7-16
Ilustración 7-6. Cálculo de Cargas por el método de Andrews.....	7-17
Ilustración 7-7. Elección de Antenas.....	7-18
Ilustración 7-8. Gráficos para Antenas Normalizadas.....	7-20
Ilustración 7-9. Entrada de Velocidad de Viento	7-20
Ilustración 7-10. Condiciones de carga	7-20
Ilustración 7-14. Entrada de datos para torres de telecomunicaciones.....	7-22
Ilustración 7-16. Presentación de antenas sobre la torre	7-25
Ilustración 7-17. Ángulos de Desvío y Desplazamientos.....	7-26
Ilustración 7-20. Fuerzas Axiales	7-27
Ilustración 7-21. Reacciones.....	7-28
Ilustración 8-5. Estación de ensayos con su centro de control abajo	8-10
Ilustración 8-6. Colocación de cables y cédulas de carga	8-11
Ilustración 8-7. Controles de una estación de ensayos.....	8-11
Ilustración 9-1. Tiempo de Ejecución de Algoritmos	9-4
Ilustración 12-1. Comandos principales	12-3
Ilustración 12-2. Planilla de Barra.....	12-5
Ilustración 12-3. Elección de torre.....	12-8
Ilustración 12-4. Modificación de los datos de los taladros	12-9

Ilustración 12-5. Formación de grupos de barras iguales	12-10
Ilustración 12-6. Selección de Pieza	12-11
Ilustración 12-7. Cambio de Nomenclatura	12-12
Ilustración 12-8. Cuerpo al que pertenece la pieza	12-12
Ilustración 12-9. Creación de barras nuevas	12-13
Ilustración 12-10. Menús desplegables para la elección de los datos.....	12-14
Ilustración 12-11. Creación de chapas nuevas.....	12-15
Ilustración 12-12. Doblado de chapas.....	12-16
Ilustración 12-13. Incorporación de taladros nuevos a la barra.....	12-17
Ilustración 12-14. Colocación de nuevos cortes	12-17
Ilustración 12-15. Doblado central de la barra	12-17
Ilustración 12-16. Elección del tipo de doblado de chapas	12-18
Ilustración 12-17. Cambio de gramiles.....	12-19
Ilustración 12-18. Alargamiento de extremos.....	12-19
Ilustración 12-19. Cambio de angular.....	12-19
Ilustración 12-20. Creación de listas de materiales	12-20
Ilustración 12-21. Salida de listas de materiales.....	12-21
Ilustración 12-22. Datos de salida de resultados.....	12-22

