Estructuras Aeroespaciales

Práctica 1

Alumnos: Pedro López Sancha, Yi Qiang Ji Zhang
Profesor: Daniel Yago, Juan Carlos Cante

Ingeniería en Tecnologías Aeroespaciales
Escuela Superior de Ingeniería Industrial, Aeroespacial y Audiovisual de Terrassa
Universitat Politècnica de Catalunya

24 de febrero de 2020

Índice

1. Introducción

Esta práctica tiene como objetivo el uso de cálculo matricial para la determinación de las fuerzas y desplazamientos de las estructuras así para obtener el estado tensional de cada elemento de barra, ya si sea tracción (+) o compresión (−).

2. Definición del problema

En la Figura 1 se muestra el esquema de la estructura analizada. Esta esta formada por 16 elementos de barra un nodo en cada extremo. Para esta primera aproximación, se considerará que el problema es bidimensional, por tanto, cada nodo tiene 2 grados de libertad asociados.

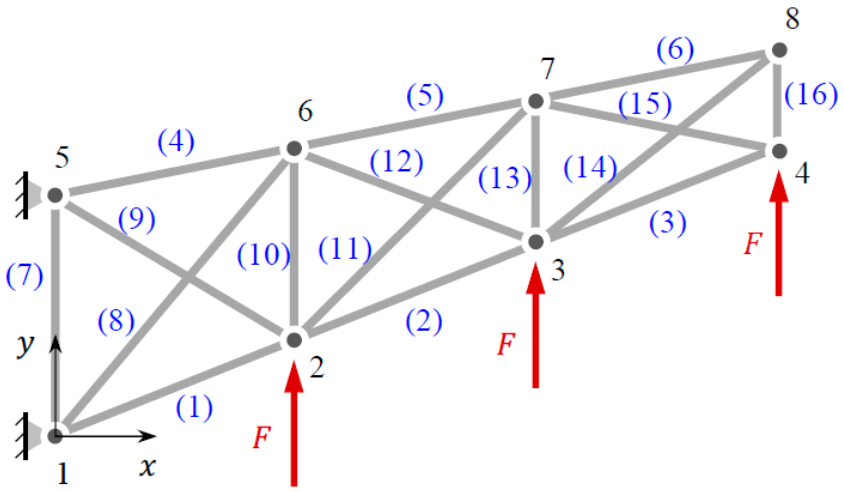

Figura 1 Esquema de la estructura analizada

Las posiciones de los nodos 1 y 5 están fijas y una fuerza F actúa en los nodos 2, 3, 4 en sentido vertical positivo. Todas las barras son del mismo material, caracterizado por un módulo de Young, área (cuadrada) y una densidad determinada. Los datos se muestran a continuación:

Fuerza	$F = 750$ N
Módulo de Young	$E = 70000$ MPa
Área	$A = 72$ mm^2
Densidad	$\rho = 2200$ kg $\cdot$ m^{-3}

Tabla 1

3. Resolución del Problema A

3.1. Tabla de los desplazamientos de cada nodo obtenido numéricamente en la parte A

En la primera parte, se pide encontrar una relación I entre la numeración de los GDOFs (*Global Degrees of Freedom*) con la numeración nodal de la estructura.

La relación existente entre los grados de libertad globales y la numeración nodal sigue la siguiente serie:

$$\begin{aligned} 1 \text{ Dimensión} &= \{n\} \\ 2 \text{ Dimensiones} &= \{2n-1, n\} \\ 3 \text{ Dimensiones} &= \{3n-2, 3n-1, n\} \end{aligned}$$

Donde n es el nodo en numeración global.

En primer lugar, los datos del problema se recogen en el archivo **main_01.m**. Se incluye en este archivo todas los parámetros intrínsecos a la estructura. Esto es, parámetros como el módulo de Young E, densidad ρ, etc, para cada una de los elementos de barra. Asimismo, es aquí donde también se especifica la matriz de coordenadas de cada nodo x, la matriz de fuerzas externas F_{data}, la matriz de los nodos con condiciones prescritas fix_{Nod}, así como las matrices de conectividad del material T_{mat}.

3.2. Función en MATLAB para la generación de la matriz de conectividad

Seguidamente, se hace uso de la función del *script* **connectDOFs**. Esta función tiene como parámetros de entrada: n_{el} (número total de elementos de barra), n_{nod} (número de nodos por elemento, n_i (número de grados de libertad por nodo y la matriz de conectividad nodal T_n.

La salida de la función es la matriz de conectividad de DOFs T_d.

$$(T_d)^T = \begin{pmatrix} 1 & 3 & 5 & 9 & 11 & 13 & 1 & 1 & 3 & 3 & 3 & 5 & 5 & 5 & 7 & 7 \\ 2 & 4 & 6 & 10 & 12 & 14 & 2 & 2 & 4 & 4 & 4 & 6 & 6 & 6 & 8 & 8 \\ 3 & 5 & 7 & 11 & 13 & 15 & 9 & 11 & 9 & 11 & 13 & 11 & 13 & 15 & 13 & 15 \\ 4 & 6 & 8 & 12 & 14 & 16 & 10 & 12 & 10 & 12 & 14 & 12 & 14 & 16 & 14 & 16 \end{pmatrix}$$

3.3. Creación de la función para generar las matrices de rigidez para cada elemento de barra

A continuación, empleando la función **computeKelBar** cuyas entradas son n_d (dimensión del problema), n_{el} (número total de elementos), x (matriz de coordenadas), T_n (matriz de conectividad nodal), mat (matriz de propiedades del material) y $Tmat$ (matriz de conectivdad de materiales).

Esta función emplea los parámetros anteriores y calcula la matriz de rigidez de cada elemento.

3.4. Creación de la función para generar la matriz de rigidez global

Seguidamente, una vez encontrados las matrices de rigidez elementales, se procede al proceso de ensamblaje de tales matrices. Por consiguiente, es lógico que los parámetros de entrada sean n_{el}, n_{dof} (número total de grados de libertad), $n_{el_{dof}}$ (número de DOFs por elemento) y las tablas T_d y K_{el} calculadas ya en los apartados anteriores. Par su implementación se emplea la función **assemblyKG**

3.5. Creación de una función en Matlab para generar el vector de fuerza global

De manera análoga, el vector de fuerza está descrito en coordenadas locales. Luego, se deben pasar a coordenadas globales. Como es de esperar, la entrada F_{data} la cual es la matriz de datos de las fuerzas externas y n_i y n_{dof}. La función **computeF** tiene como salida el vector de fuerzas en coordenadas globasles de cada DOF.

3.6. Creación de una función en Matlab para obtener los vectores de grados de libertad libres y prescritos así como también el vector de desplazamientos

Por lo que se refiere a la función cuyo objetivo es aplicar las condiciones iniciales, de pasan como parámetros n_i, n_{dof} y la matriz $fixNod$ cuyos índices expresan los nodos, direcciones y magnitud en la que algun desplazamiento está prescrito. Efectivamente, se usa la función **applyCond** para definir los nodos con desplazamientos prescritos y libres.

3.7. Creación de una función en Matlab para revolver los sistemas de ecuaciones obtener los desplazamientos nodales y las reacciones en los DOFs prescritos

El siguiente punto trata de la resolución del sistema de ecuaciones. Esto es, obtener los los vectores de desplazamientos y las reacciones. Para ello, se precisa la función **solveSys**. Como parámetros de entrada se recogen los resultados del apartado anterior y se añade la matriz de rigidez global K_G junto con el vector de fuerzas externas F_{ext}. Dentro de esta función además, se dará a conocer el número de condición de la matriz K_{LL}, parámetro que indica la precisión de los resultados teniendo en cuenta la que se debe calcular la inversa de la matriz K_{LL}.

Valores de número de condición cercanos a la unidad indican un resultado correcto. Sin embargo, como se tratará posteriormente en el apartado 4, en algunos casos particulares, el hecho de eliminar barraras induce a que la estructura no sea hiperestática sino que pasa a ser un mecanismo. En efecto, esto se refleja en el número de condición de la matriz K_{LL} el cual se incrementa a valores $>> 1$.

3.8. Creación de una función en MATLAB para el cálculo de las deformaciones (ϵ) y tensiones (σ) de cada barra

Finalmente, el último paso es hallar las deformaciones ϵ y tensiones σ de cada elemento de barra. En particular, este paso recoge los parámetros intrínsecos de cada elemento de barra cuyas propiedades

se recogen en las matrices mat y T_{mat}. A fin de ello, la función **computeStrainStressBar** permite el cálcul de dichas deformaciones y tensiones.

Una vez hechos todos los pasos anteriores, ya se puede estudiar la estructura. El siguiente paso se trata de observar qué elementos son los más conflictivos (las barras a las cuales están sometidas a una mayor tensión o carga) y qué barras no ejercen ayuda alguna (elementos de barra que tengan tensión nula).

3.9. Resultados

Nodo	**DOF**	**Desplazamiento** (mm)
1	1	0,000
	2	0,000
2	3	0,221
	4	0,633
3	5	-0,056
	6	2,003
4	7	-0,680
	8	3,794
5	9	0,000
	10	0,000
6	11	-0,492
	12	0,624
7	13	-0,963
	14	1,991
8	15	-1,374
	16	3,782

Tabla 2 Desplazamiento en cada grado de libertad (x e y) para cada nodo

Elemento	**Tensión** (MPa)
1	57.176
2	32.732
3	11.135
4	-49.436
5	-26.548
6	-7.162
7	0
8	14.704
9	-16.355
10	-1.595
11	12.264
12	-14.033
13	-2.802
14	8.993
15	-10.543
16	-4.214

Tabla 3 Tensiones para cada elemento de barra

Nodo	**DOF Global**	**Reacción** (N)
1	1	-4500.0
	2	-2342.2
5	9	4500.0
	10	92.2

Tabla 4 Reacciones en los nodos con desplazamiento prescrito

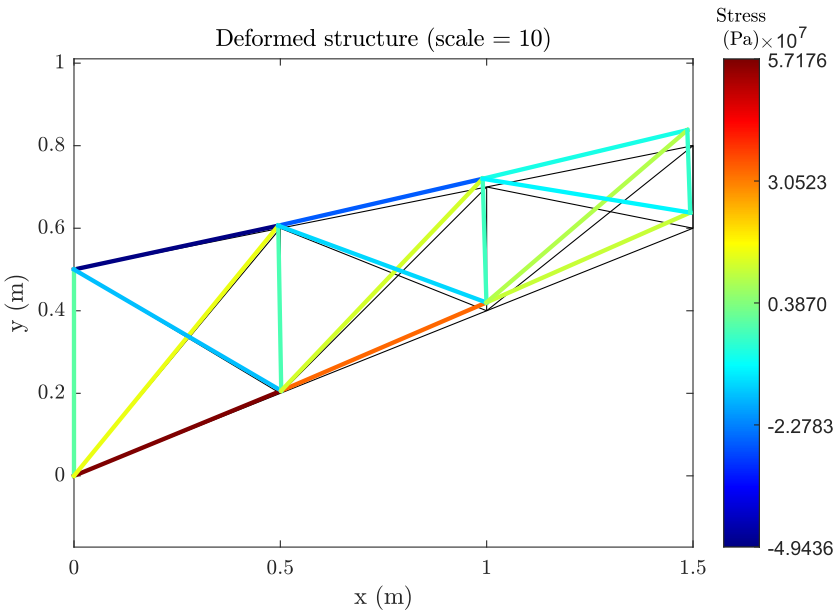

Figura 2 Representación de la estructura deformada y la tensión en cada barra en el problema A

4. Problema B1

4.1. Planteamiento del problema

En este problema se dispone de todo el material necesario para construir la estructura mostrada en la figura 1, además se considera que:

- Las cargas externas aplicadas y las condiciones de frontera son iguales que en el problema A. Es decir, se mantiene la dirección e intensidad de las fuerzas sobre los nodos 2, 3, y 4, y los desplazamientos prescritos en los nodos 1 y 5.
- Las propiedades de los materiales así como su sección, se mantienen.
- Las posiciones de los nodos se mantienen.

Se pretende mejorar la estructura, de manera que puede añadirse o eliminarse material, con las siguientes condiciones:

- Si se compra material adicional, el precio de compra será de 320 €/kg.
- Se se vende material sobrante, el precio de venta será de 46 €/kg.
- Las barras 1-6 y 16 no pueden eliminarse.

El objetivo es minimizar el coste de la estructura o bien maximizar los beneficios obtenidos vendiendo material sobrante. Se observa lo siguiente:

- La barra número 7 se encuentra entre dos nodos con desplazamientos prescritos iguales a cero en ambos grados de libertad. Esto implica que no existe desplazamiento relativo entre nodos, de manera que la barra 7 no se deforma y nunca trabajo, como puede apreciarse en la figura 2. Por este motivo, el elemento 7 siempre puede eliminarse de la estructura y venderlo como material sobrante.
- Sean i, j, $i \neq j$ dos nodos de la estructura, y la barra entre ellos $e = \{i, j\} = \{j, i\}$. Existen $\binom{8}{2}$ posibles barras en el problema, es decir, el conjunto

$$\{\{i, j\} \mid 1 \leq i \leq 7, i < j \leq 8\}$$

  En este conjunto existen barras que, dada la distribución de nodos en la estructura, atraviesan un nodo, de modo que no tienen sentido. Estas son: $\{1,3\}$, $\{1,4\}$, $\{2,4\}$, $\{5,7\}$, $\{5,8\}$, $\{6,8\}$. Existen también barras que cruzan la estructura en diagonal: $\{1,7\}$, $\{1,8\}$, $\{2,8\}$, $\{3,5\}$, $\{4,5\}$, $\{4,6\}$. Aparentemente podrían ser adecuadas para proporcionar rigidez, sin embargo si se colocan, impiden situar las barras verticales 10 y 13, así como varias otras barras diagonales. Por este motivo, no pueden añadirse nuevas barras, únicamente pueden eliminarse y, por consiguiente, solo puede venderse material adicional.

Atendiendo a las anteriores observaciones, el objetivo del problema es maximizar la venta de material consiguiendo una estructura que resista las restricciones impuestas. Las barras 8-15 son las susceptibles de ser eliminadas del problema.

Este problema puede plantearse como un problema de programación lineal. La programación lineal es la rama de la optimización numérica que se encarga del estudio de problemas donde se pretende maximizar o minimizar una función objetivo lineal, sujeta a restricciones lineales, así como a la implementación y desarrollo de algoritmos para resolverlo [1]. En particular, dentro de la programación lineal, existe la programación lineal entera, en la cual el óptimo que se busca debe ser un entero.

La estructura final puede representarse como un vector $v \in \{0,1\}^{16}$, donde $v_i = 0$ indica que la barra i no está presente en la estructura, mientras que $v_i = 1$ indica que esta es usada. La función objetivo del

problema es

$$z = \sum_{i=1}^{16} \rho_i L_i A_i \left(1 - v_i\right)$$

donde ρ_i, l_i y A_i son la densidad, longitud y sección de la barra i, respectivamente. Si $v_i = 0$, el producto $\rho_i L_i A_i (1 - v_i)$ proporciona la masa de la barra i, la cual podrá venderse puesto que no se incluye en la estructura. No es necesario incluir el precio de venta de 46 €/kg para el material, ya que maximizar la función lineal z equivale a maximizar la función $46z$. Finalmente la estructura debe actuar como estructura y no como mecanismo. Esta restricción puede modelizarse imponiendo un desplazamiento máximo en cada grado de libertad. Puesto que los desplazamientos obtenidos para el problema A son del orden de 10^{-3} m, el límite de desplazamiento impuesto puede ser de 10^{-2} m.

$$(PE) \begin{cases} \max\limits_{v \in \{0,1\}^{16}} \; z = \sum\limits_{i=1}^{16} \rho_i L_i A_i \left(1 - v_i\right) \\ \text{sujeto a:} \\ \qquad -10^{-2} < u_i < 10^{-2}, \quad 1 \leq i \leq n_{dof} \\ \qquad v_i = 1, \quad \forall i \in \{1, 2, 3, 4, 5, 6, 16\} \\ \qquad v_7 = 0 \end{cases}$$

Obsérvese que la restricción de desplazamiento sobre cada grado de libertad es lineal, puesto que el vector de desplazamiento u se calcula mediante un sistema lineal.

Sea $\mathcal{F}$ la región factible del problema (PE), es decir, el conjunto de vectores $v \in \{0,1\}^{16}$ que cumplen con las restricciones impuestas al problema. Puede "jugarse" con los índices 8-15, de modo que el problema el máximo número posible de soluciones es $|\mathcal{F}| \leq 2^8$. De este modo puede asegurarse que (PE) es un problema sin solución, o bien un problema con un único vector óptimo v^* o bien un problema con varios vectores óptimos. Nunca será un problema ilimitado, es decir, un problema en el que la función z pueda maximizarse sin límite.

4.2. Resolución del problema

Existen diversos algoritmos para la resolución de problemas de programación lineal entera. Destacan el algoritmo de Ramificación y Poda (*Branch & Bound*), el algoritmo de Planos de corte de Gomory y el algoritmo de Ramificación y Corte (*Branch & Cut*), [2].

El algoritmo *Branch & Cut* está implementado en MATLAB, sin embargo adaptarlo al presente problema es tedioso. Por ello y porque existen como mucho $2^8 = 256$ posibles soluciones, se aplicará fuerza bruta. No obstante, puede darse el caso de que la matriz K_{LL} usada para calcular el vector u_L sea singular o cercana a singular, lo cual daría lugar a desplazamientos sin sentido, del orden de 10^{12} m. Al revés la implicación también es cierta, si los desplazamientos no tienen sentido físico, la matriz será singular o cercana a singular. De esta forma, puede imponerse la restricción analizando cómo de cerca está la matriz a ser singular.

Para estudiar cómo de cerca está la matriz a ser singular se ha usado el número de condición μ asociado a la norma 2. Dado un sistema lineal $Ax = b$, el número de condición de la matriz A, $\mu(A)$, indica cómo varía la solución x cuando se introduce una perturbación δb en el vector b. Es decir, indica cómo de diferentes son los vectores x y $x + \delta x$, que son solución de $Ax = b$ y $A\left(x + \delta x\right) = b + \delta b$, respectivamente [3].

4.3. Nuevo código implementado

Se han creado diversas funciones, así como modificaciones del código principal:

- Se ha creado la función **solveSys2**, que es una ligera modificación de la función **solveSys**. En esta se calcula el número de condición de la matriz K_{LL} antes de la resolución del sistema lineal. Analizando todos los casos, se ha observado que si la solución tiene sentido físico, entonces $\mu(K_L L) \leq 10^3$, mientras que si no lo tiene, $\mu(K_L L) > 10^{10}$. Por ello se sigue la heurística de calcular la solución solamente si $\mu(K_L L) \leq 10^3$.
- Se ha creado la función **computeSells** que, dado un vector de solución v, calcula los beneficios obtenidos de las ventas del material sobrante.
- Se ha creado la función **optimizeB1** que calcula el óptimo de ventas. Ejecuta las funciones **connectDOFs**, **computeKelBar**, **assemblyKG**, **computeF**, **applyCond** y **solveSys2** para cada posible solución. Si la solución es factible, es decir, cumple las restricciones, ejecuta **computeSells** y **computeStrainStressBar**.
- Se ha creado un nuevo *main*, **main_B1**, que incorpora la función **optimizeB1**.

4.4. Resultados

La solución obtenida al problema es la siguiente [Figura 3]:

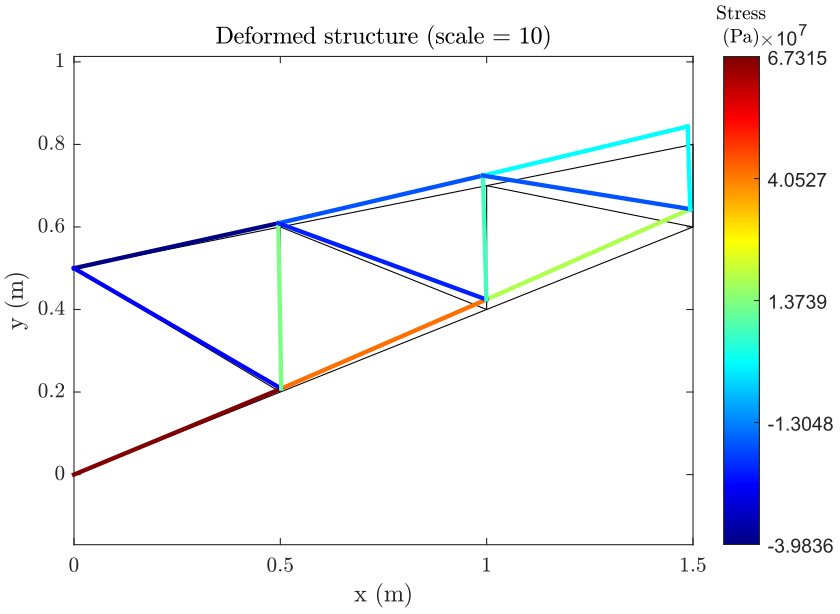

Figura 3 Representación de la estructura deformada y la tensión en cada barra en el óptimo del problema B1

5. Resolución del Problema B2

5.1. Planteamiento del problema

En el segundo planteamiento, análogamente al apartado 4, se dispone de todo el material necesario para la construcción de la estructura de la Figura 1.

5.2. Resolución del problema

En este segundo caso, sin embargo,se restringe una condición adicional pues el objetivo es garantizar la seguridad de la estructura:

- Se debe garantizar que la tensión máxima de tracción no debe superar los 50 MPa

El objetivo final es equivalente al anterior, minimizando los costes y maximizando los beneficios teniendo en cuenta el límite de tracción impuesto.

5.3. Nuevo código implementado

A diferencia del caso anterior, esta vez se implementa las mismas funciones anteriores difiriendo en una condición añadida. Para este apartado, se han generado los siguientes cambios:

- Se ha creado la función **optimizeB2**, que se trata de una ligera modificación del la función **optimizeB1**, el cual calcula el óptimo de ventas cumpliendo el límite de tracción.
- Se ha creado un nuevo *main*, **main_B2** que incorpora la función modificada **optimizeB2**.

5.4. Solución

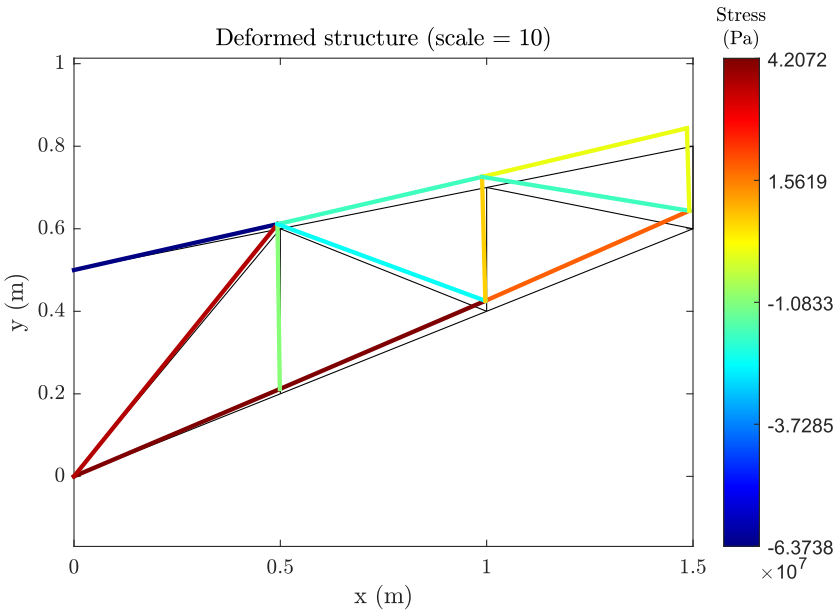

Figura 4 Representación de la tensión en cada barra en el problema B2

6. Conclusiones

A modo de conclusión, analizando los resultados obtenidos es conveniente estudiar en profundidad las gráficas finales y el coste final.

En el apartado A (sección 3), se intenta dar una visión general del estado tensional a la cual cada barra está sometida. Para ello, dado que se trata de una estructura hiperestática, se emplea el uso de cálculo matricial con el objetivo de simplificar el problema, generalizarlo así para facilitar y automatizar la resolución de ésta.

Estructura	Longitud (m)	Masa (kg)	Ventas (€)
A	8,305	1,316	0
B1	5,677	0,899	19,15
B2	5,875	0,931	17,71

Tabla 5 Longitud de las barras, masa y ventas obtenidas para cada estructura

La longitud total de la distribución de las barras en el apartado A es de $8,305$ m. El problema nos proporcionan las propiedades del material como su longitud, sección y densidad. Por lo que el cálculo del coste del material es sencillamente la masa total de la estructura por el precio por quilogramo de esta la cual es de $60,52$ €.

En el apartado B1, al extraer las barras [7, 8, 11, 14]. La longitud de la estructura final se ve reducido considerablemente con un ahorro de $2,628$ meter de diferencia, lo que se traduce en un ahorro de $19,15$ €. Cabe destacar que la gran ventaja de esta distribución es que no se cuenta con el elemento de barra [7] pues este en el ensayo analizado no ejercía ningún tipo de contribución a la estructura, lo que llevó a su eliminación de forma inmediata.

Por otra parte, el apartado B2 propone una limitación más teniendo en cuenta las mismas restricciones anteriores. En efecto, la solución del problema sigue permitiendo la supresión de los elementos de barra siguientes: [7, 9, 11, 14]. Analizando el coste final, la longitud de las barras es mayor con la cual cosa sobra menos material para la venta. De este modo, las ventas obtenidas son ligeramente inferiores.

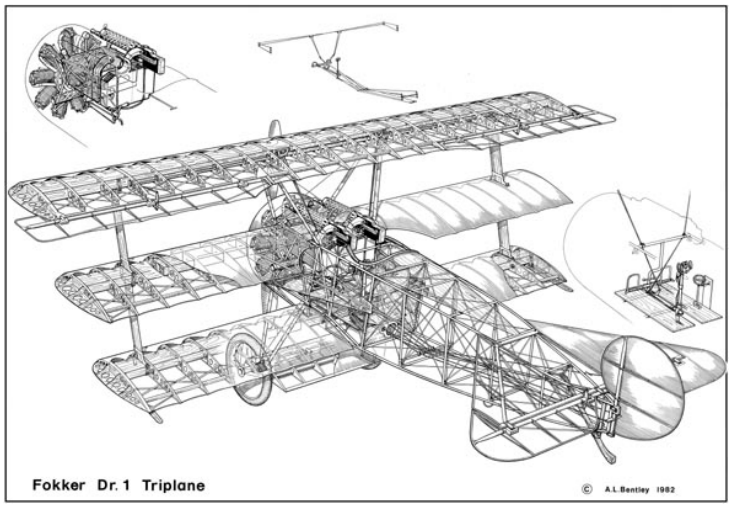

Figura 5 Estructura del Avión Fokker Dr I triplano usada en la WWI. A partir de [4].

Para concluir, una de las aplicaciones directas que hemos sacado de la estructura analizada tiene que

ver con el campo de la aeronáutica. La estructura 1 toma la forma de la parte de la estructura de braguero de una avioneta.

El caza 5 fue un avión de guerra alemán usada en la primera guerra mundial. La estructura del Dr. I era combinaba una construcción mixta muy resistente con un alto rendimiento aerodinámico. Los planos eran casi totalmente de madera, con acero en las juntas principales y en las uniones. Cada plano tenía una única caja de vigas, con largueros principales de madera dura ahusada (perforada según un patrón para aligerarla) unidos por costillas empotradas. Los bordes de fuga eran de cables. En la segunda imagen, se aprecia como la estructura del braguero presenta rasgos parecidos a nuestro problema.

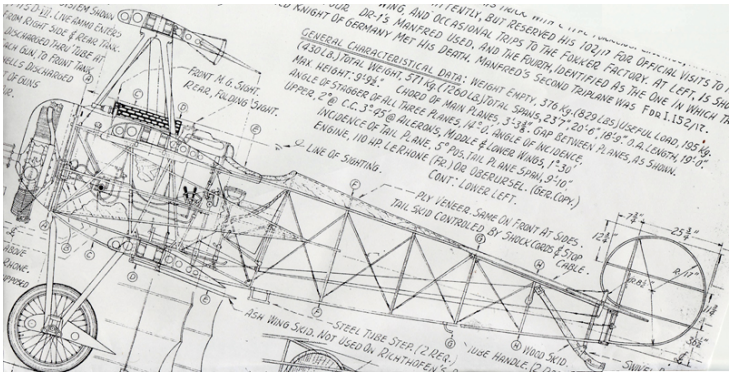

Figura 6 Estructura del braguero del Fokker Dr I. A partir de [5].

Referencias

[1] Mathematical Optimization Society. *About the Mathematical Optimization Society*. Feb. de 2020. URL: http://www.mathopt.org/?nav=about.

[2] Jordi Castro, Franciso Javier Heredia y Ma. Paz Linares. «Programació Matemàtica, Tema 3: Programació Lineal Entera». Transparencias de Programació Lineal Entera, de la asignatura de Programación Matemática impartida en la Facultad de Matemáticas y Estadística, curso 2018-2019. Feb. de 2020.

[3] José Tomás Lázaro, Mercè Ollé y Juan Ramón Pacha. «Teoria d'Àlgebra Lineal Numèrica». Teoria de la asignatura de Álgebra Lineal Numérica impartida en la Facultad de Matemáticas y Estadística, curso 2018-2019. Feb. de 2020.

[4] Anthony Fokker. *Fokker Dr I Triplane*. Feb. de 2020. URL: https://virtualpilot3d.net/page/2/.

[5] Anthony Fokker. *Fokker Dr I*. Feb. de 2020. URL: https://www.rcuniverse.com/forum/rc-scale-aircraft-169/8127853-scratch-built-fokker-dr-i-1-6-scale.html.