

Diseño óptimo de celosías planas con criterios de estado límite último

Lluís Gil y Antoni Andreu

Departamento de Resistencia de Materiales y Estructuras en la Ingeniería
Universidad Politécnica de Cataluña
Módulo C1, Campus Norte
Gran Capitán s/n, 08034 Barcelona, España
Tel.: 34-93-205 70 16, Fax: 34-93-401 65 17
e-mail: lluis@cimne.upc.es, e-mail: tandreu@etseccpb.upc.es

Resumen

En este trabajo se presenta una metodología de cálculo óptimo de estructuras articuladas planas que permite obtener la forma geométrica y las secciones más convenientes de la estructura. Por lo tanto, se combina un problema de diseño de formas con otro de parámetros. La metodología presenta innovaciones en la combinación de ambos procesos de optimización y solventa las dificultades de trabajar con variables de muy distinta naturaleza. Se destaca el acierto del algoritmo para diseñar celosías que se adapten a la forma natural de trabajo para el problema considerado. Los resultados obtenidos en los ejemplos se comparan con puentes contruidos a principios de siglo, cuando la estructura metálica estuvo en boga y el ahorro del material repercutía sustancialmente en el coste. En el artículo también se detallan los aspectos de implementación en un código orientado a objetos en lenguaje C++ para el análisis y la optimización de estructuras.

OPTIMUM DESIGN OF PLANE TRUSSES WITH LOCAL FAILURE CRITERIA

Summary

This work presents a method for calculating an optimum topology and the optimum cross-sections for a plane truss structure under predetermined boundary conditions. It is combined the shape desing problem with the parameter design one. The method includes a new approach in combining these two problems and solves the difficulties that appear when trying to combine the optimization of variables of such different nature. It is remarkable that the results of the method produce geometries which naturally adapt to the external loads. These results are then compared to steel bridge constructions builded at the beginning of the century, when steel construction was widely used and saving material was of great importance. The paper also includes an implementation of object oriented C++ code for the analysis and the optimization of truss structures.

INTRODUCCIÓN

El diseño óptimo de estructuras de barras debe verse desde dos puntos de vista, por un lado el de obtener la mejor disposición geométrica de las barras y los nudos; y por otro lado el conseguir las secciones transversales más convenientes. Por ello, el problema de optimización combina un problema de formas y uno de parámetros. En este trabajo se van a seguir los planteamientos que se encuentran en las referencias^{3,9}. Planteamientos alternativos clásicos se encontrarán en la referencia⁶. En general, la forma de la estructura depende del criterio del ingeniero y en su diseño pesan aspectos económicos, constructivos, de inserción en el paisaje, etc. En cambio, las dimensiones de las barras dependen básicamente de criterios resistentes, económicos y funcionales.

Desde el punto de vista matemático, el problema de optimización se define de la forma siguiente. Se trata de obtener los valores de las variables de diseño

$$x = \{x_1, x_2, \dots, x_n\} \quad (1)$$

donde x_i representa las propiedades de las secciones estructurales, las coordenadas de las conexiones entre barras, etc.

De manera que se minimice una función objetivo que representa un criterio de diseño

$$f(x_1, x_2, \dots, x_n) \quad (2)$$

donde f puede ser el peso mínimo, el coste, etc.

Verificando unas restricciones del diseño

$$g_i(x_1, x_2, \dots, x_n) \leq 0 \quad i = 1, \dots, m \quad (3)$$

El planteamiento realizado en este trabajo pretende una optimización tanto de formas como de parámetros. Por consiguiente, un grupo de variables de diseño representa las coordenadas de los nudos mientras que otro grupo de variables representa las secciones transversales de las barras. A través de la modificación de las coordenadas de los nudos, se consigue una variación en la forma del diseño de la estructura, es decir, una modificación de ámbito global sobre la forma de trabajar de la estructura. Por el contrario, la modificación de secciones estructurales afecta únicamente de forma local a las barras. El hecho de que el espacio de las variables de diseño esté formado por entidades de distinta naturaleza que afectan de forma diferente al problema estructural representa un inconveniente para cualquier algoritmo de optimización³.

Como función objetivo se ha considerado el peso mínimo aunque desde el punto de vista de la ingeniería tal vez hubiera sido mejor considerar el coste. Sin embargo, de todos es sabido que hoy en día el coste no está directamente relacionado con la cantidad de material, sino que influyen otros factores locales como el precio de la mano de obra o la dificultad constructiva por citar ejemplos obvios. Por ello, para no particularizar excesivamente la función objetivo se ha considerado útil para los propósitos de este trabajo evaluar el peso. Por lo tanto, inicialmente se define la función objetivo como

$$w(x) = \sum_i^{\text{barras}} l_i A_i \gamma_i \quad (4)$$

donde l_i es la longitud de la barra, A_i el área de la sección y γ_i el peso específico del material. Nótese que se asume que las barras son de sección constante y directriz recta.

Finalmente, como restricciones al problema se considera la ecuación de estado que rige el comportamiento de la estructura, es decir, el cumplimiento de las ecuaciones de equilibrio y compatibilidad que en el caso de análisis lineal de estructuras se reduce a la conocida expresión

$$Ku = P \quad (5)$$

donde K es la matriz de rigidez de la estructura, u los desplazamientos incógnita y P el vector de fuerzas externas. Otro tipo de restricciones que afectan al problema de optimización es el valor de los esfuerzos máximos de trabajo en cada una de las barras solicitadas, de manera que

$$P_i < P_u \quad (6)$$

donde P_i es el valor de la solicitud axil de trabajo y P_u el axil máximo que puede resistir el material, tanto a compresión como tracción siguiendo el procedimiento que recoge el Eurocódigo de Cálculo y Proyecto de Estructuras Metálicas². En consecuencia, esta restricción permite verificar el estado límite último de la estructura analizando cada barra desde una posible inestabilidad local.

La forma óptima de la estructura, la que tiene un menor peso, se conseguirá desde un punto de vista resistente. Es decir, será aquella que permite que el material se aproveche al máximo cumpliendo las condiciones de equilibrio y compatibilidad.

EL ALGORITMO DE OPTIMIZACIÓN

El planteamiento del problema anterior conduce a intentar resolver un problema de optimización no lineal. La función peso (4) es fácil de evaluar, pero la dificultad estriba en verificar las restricciones, por un lado la ecuación de equilibrio (5) de la estructura y por otro las restricciones (6) en carga última que dependen de forma no lineal con respecto de las variables de diseño. Una estrategia bastante común para resolver el problema consiste en optimizar linealizando las ecuaciones en el entorno del diseño y utilizando un algoritmo, por ejemplo el simplex, para un problema lineal^{3,5,9}. En general, con este planteamiento de linealización, los diseños iniciales tienen que estar cerca del óptimo para asegurar la convergencia. Existen otros métodos^{1,7,8} pero tienen un acercamiento que no cubre el propósito de este trabajo.

Tratamiento de las restricciones

En este trabajo se utiliza la técnica de transformar un problema no lineal de minimización condicionada en otro de minimización no lineal pero incondicional a través de la técnica de la función penalty⁵. Para ello se define una función de penalización que pondera el grado de violación de las restricciones y penaliza, tal y como su nombre indica, el valor de las variables de diseño para acercarse al óptimo. En consecuencia, se define una función penalty mediante las restricciones y se introduce en la función objetivo original (función peso), de manera que la nueva función objetivo tiene en cuenta tanto el valor de aquello que se quiere minimizar, como el cumplimiento simultáneo de las restricciones en las variables de diseño.

En particular, para el problema presentado se define una función de penalización que pondera aquellas variables o parámetros restringidos que están superando su valor límite. La expresión general toma la forma siguiente

$$p(x) = \sum_i^{\text{barras}} \alpha_i (Q_i - 1)^2 + \sum_j^{\text{restricciones}} \beta_j (x_j - x_u)^2, \text{ siendo } Q_i = \frac{P_i(x)}{P_u} \quad (7)$$

Nótese que Q_i es el ratio entre el axil que solicita la barra y el máximo que puede ser resistido, por lo tanto depende implícitamente de las variables de diseño. Además dicho valor sólo toma efecto cuando $Q_i > 1$ porque es entonces cuando se está superando la restricción de resistencia límite en la barra e interesa penalizar dicha violación; en caso contrario se toma $Q_i = 1$. La variable x_j compara con su restricción máxima x_u y en principio, esta variable puede representar cualquier tipo de restricción, desde una inercia mínima en ciertas barras hasta el valor máximo de las coordenadas de algunos nodos de la estructura pasando por el módulo elástico. Sin embargo, para los propósitos de este trabajo se ha limitado al caso de coordenadas en ciertos nodos. Como en el caso anterior, si la variable no viola la restricción, el sumando de penalización se anula tomando $x_j = x_u$ en la función (7). Finalmente, α_i y β_j son coeficientes de peso que se escogen previamente y en función del problema con el que uno se enfrenta. Su valor es un tanto arbitrario siendo del orden de 10^3 para α_i y

10^{-3} para β_j en los ejemplos donde se ha probado la estrategia de optimización. Nótese que estos parámetros realizan una labor de homogeneización eliminando las desigualdades que provoca el hecho de trabajar con variables de naturaleza distinta que tienen órdenes de magnitud dispares.

Por consiguiente, la función objetivo que será sometida al proceso de minimización será la suma del peso de la estructura y de la función penalty

$$f(x) = w(x) + p(x) \quad (8)$$

siendo conocidas las expresiones a partir de (4) y (7). En el estudio se han utilizado otras funciones objetivo como

$$f(x) = w(x)[1 + p(x)] \quad (9)$$

Sin embargo, los resultados obtenidos no difieren entre ellos.

Dado que hay que optimizar en un doble sentido, la geometría y las secciones; se definen dos estrategias para optimizar en uno y otro sentido. Es importante desacoplar ambas optimizaciones para facilitar la convergencia. El tratamiento conjunto de informaciones dispares dificulta el hallar una solución como se señala en la referencia³.

El algoritmo de optimización mecánica

El algoritmo de minimización de las secciones es un algoritmo directo. Su funcionamiento se detalla a continuación. Inicialmente se dispone de un vector que representa una parte del espacio de variables de diseño, en particular

$$A = \{A_1, A_2, \dots, A_n\} \quad (10)$$

representa un vector con las diferentes secciones que tienen las barras de la estructura. Cuando se cumple el equilibrio y se verifica (5), se dispone de los esfuerzos axiles que solicitan las barras. Dado que las resistencias últimas de las barras (compresión o tracción) son conocidas, se puede verificar que se cumplen las restricciones que impone el Eurocódigo² en cargas últimas de todas y cada una de las barras. Se realiza una primera aproximación al estado límite último sustituyendo las secciones por valores más cercanos al caso crítico, según

$$A_i^{n+1} = A_i^n \frac{P^n}{P_u} \quad (11)$$

Se debe destacar que en el caso de que varias barras compartan el mismo tipo de sección y material, se adoptará para todas ellas una sección común que sea cota superior para que así ninguna de ellas rompa. Si se verifica el criterio de convergencia basado en la variación del peso

$$\frac{w(A^{n+1}, x)}{w(A^n, x)} \leq 0,95 \quad (12)$$

y además se verifica que con las nuevas secciones ninguna barra rompe, se detiene el proceso de optimización mecánica por substitución directa. A menudo sucede que muchas de las secciones sobrepasan los límites impuestos por las restricciones, en cuyo caso se debe sustituir el valor de dichas variables por sus límites.

Este proceso de optimización por substitución mecánica directa se realiza cada vez y de forma previa a la evaluación de la función objetivo (8). Por ello, a continuación, se evalúa la función penalty (7) dado que ya se dispone de los axiles que actúan en la estructura, y por

lo tanto se está en condiciones de calcular el peso, teniendo en cuenta tanto las restricciones que imponen las cargas últimas como los otros tipos de restricción, por ejemplo coordenadas de los nodos.

El algoritmo de optimización global

Una vez evaluada la función objetivo se procede a optimizar la geometría de la estructura. En un caso más general se pueden incluir en esta estrategia de optimización otro tipo de variables como el módulo elástico o restricciones en desplazamiento, pero siempre teniendo en cuenta que el tratamiento conjunto de variables de diferente naturaleza suele dar problemas. En este caso, el algoritmo se basa en la variante de Fletcher-Reeves^{3,9} del método de los gradientes conjugados.

Este método consiste en encontrar una dirección de avance dentro del espacio de diseño y actualizar el vector de variables de diseño según

$$x^{n+1} = x^n + \lambda s^n \quad (13)$$

donde x es el vector de variables de diseño relativas a coordenadas geométricas. El parámetro λ se obtiene mediante una minimización unidimensional y el vector s marca la dirección de avance. Las expresiones son de sobras conocidas y se pueden encontrar en la referencia³.

Una vez que están actualizados los valores de las variables de diseño, se deben verificar los criterios de convergencia siguientes

$$\|\nabla F(x^{n+1})\| \leq \varepsilon_1 \quad y \quad \|\lambda s^n\| \leq \varepsilon_2 \quad (14)$$

Si no se cumplen, se vuelve a iniciar la secuencia con los nuevos valores.

El esquema del algoritmo se muestra en la Figura 1 como diagrama de flujo del programa implementado.

Comportamiento del algoritmo implementado

El método de optimización de los gradientes conjugados ofrece un comportamiento bastante bueno cuando las variables contempladas son todas de la misma naturaleza, pero se muestra inestable cuando el vector de variables de diseño contiene magnitudes diversas, en concreto en este trabajo secciones transversales y coordenadas de nudos.

Por ello, para estabilizar el algoritmo se optó por considerar dos estrategias de optimización distintas, una para las coordenadas de los nudos y otra para secciones de barras. Las coordenadas se contemplan como variables sobre las que opera el método de gradientes conjugados. En cambio, las secciones se toman como variables implícitas que se optimizan mediante un método directo donde se compara la resistencia última de las barras y el axil solicitante.

La estrategia utilizada para mejorar los resultados y estabilizar el proceso de convergencia consiste en combinar los dos tipos de optimización. Esto se consigue introduciendo la optimización mecánica en el interior de la función objetivo tal como se ha explicado en el apartado anterior, de manera que la optimización geométrica actúa simultáneamente con una optimización mecánica. El inconveniente reside en que el proceso de cálculo es más costoso, ya que para evaluar la función objetivo el número de cálculos de la estructura aumenta porque el proceso de optimización mecánica se lleva a cabo en su interior. En definitiva, esta manera de crear una complicidad entre ambas estrategias de optimización conduce a una resolución más lenta pero mucho más fiable y que ofrece resultados lógicos desde una perspectiva estructural.

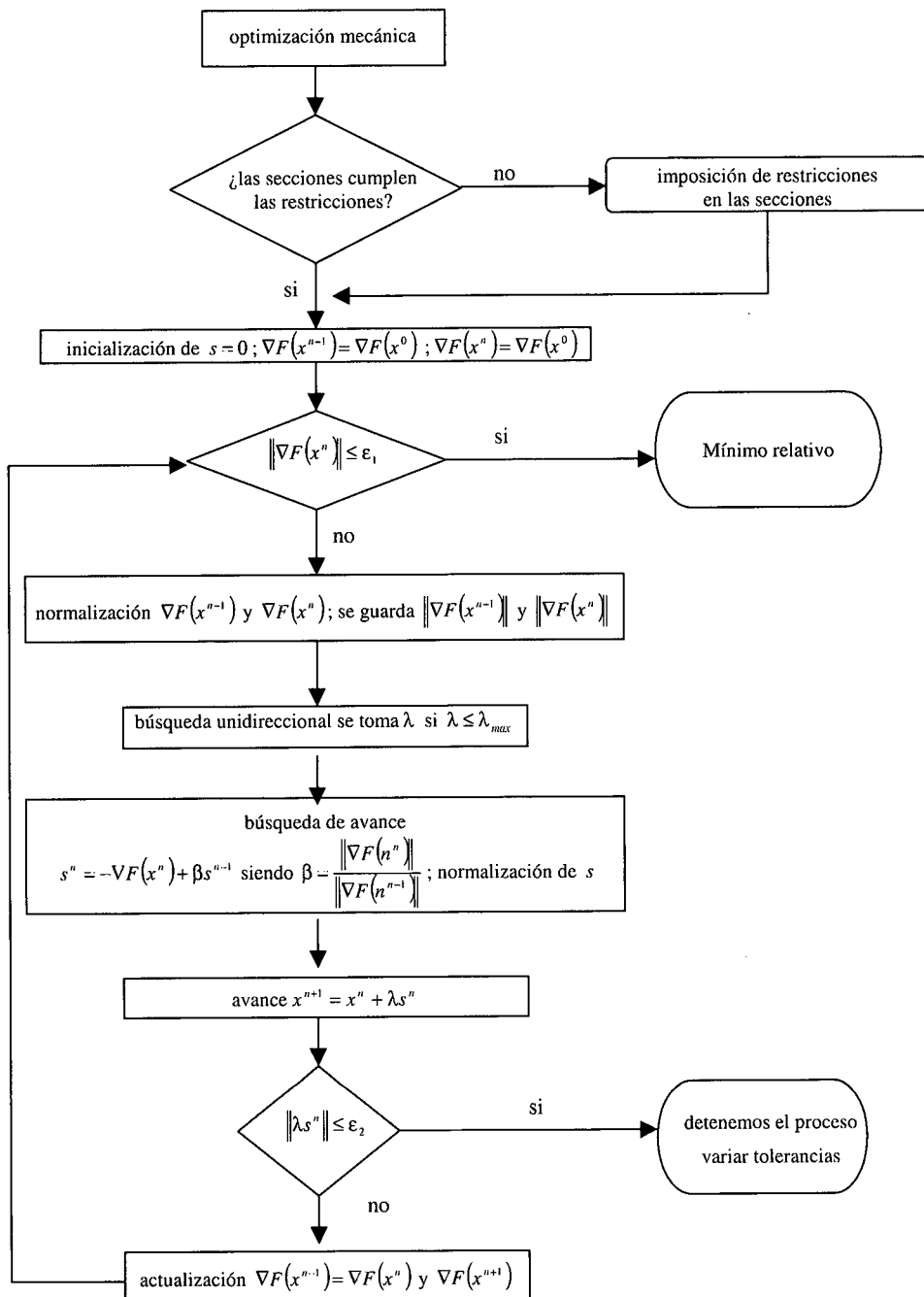


Figura 1. Algoritmo de optimización utilizado

El método utilizado requiere tres tipos de procesos iterativos. El primero consiste en la iteración global que renueva las variables, el segundo busca el mínimo de la función en la dirección del gradiente y finalmente otro que cambia las secciones. Este último es bastante fiable, pero los dos primeros tienen una cierta dosis de inestabilidad. Esta inestabilidad depende en gran manera de la semilla utilizada. Si la estructura inicial está muy lejos del óptimo, es decir, si muchas variables están fuera de la región de diseño, los parámetros resultantes de la búsqueda unidireccional λ no son en absoluto representativos y esto conduce a una divergencia segura. En general, los diseños que propone el algoritmo se alejan mucho unos de otros, las iteraciones producen grandes saltos dentro del espacio de variables de manera que nunca se llega a un acercamiento progresivo hacia un óptimo atractor. La manera de aumentar la estabilidad se basa en limitar el valor de dichos avances, de manera que al principio, al estar avance acotado, se evita que los nuevos diseños se den en lugares muy alejados. A medida que el proceso de convergencia avanza, los valores del diseño se tornan más representativos y además, a medida que se acerca al mínimo, el parámetro disminuye y la acotación se vuelve innecesaria. Esta limitación de λ se traduce en una mayor lentitud en la obtención del óptimo pero en un aumento de la estabilidad del algoritmo. Se ha comprobado que si se relaja la acotación y se llega a una convergencia del proceso, ésta se produce a una velocidad mayor pero en contrapartida diverge en un mayor número de casos. Esto es así porque el espacio de diseño es complejo y fácilmente se puede uno alejar del óptimo.

Adviértase que usando el método de la función penalty en el tratamiento de las restricciones, el mínimo relativo de la función objetivo no tiene porque coincidir con el mínimo relativo de la función peso. Esto es así porque la función objetivo contiene el peso y la penalización de las restricciones tal y como muestra la Figura 2. Únicamente coincidirá cuando el peso mínimo esté dentro de la región de diseño factible en cuyo caso la penalización se anula tal y como aparece en la Figura 3 y en consecuencia la función objetivo se iguala exclusivamente a la función peso.

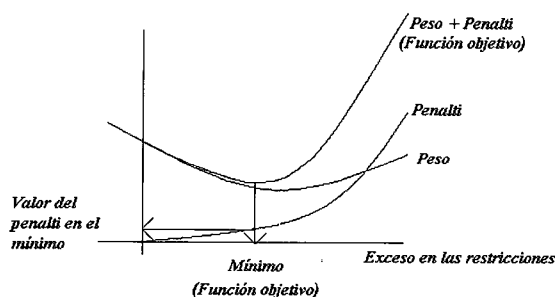


Figura 2. Los mínimos relativos de las funciones peso y objetivo no tienen porque coincidir

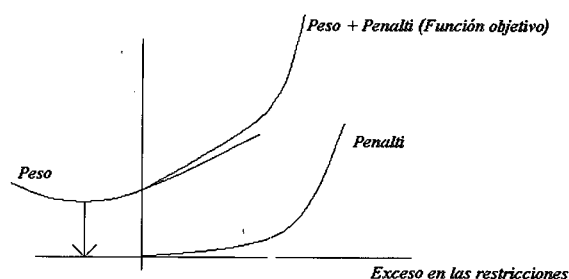


Figura 3. Sólo coincidirán en este caso

El hecho de contemplar las restricciones a través de una función penalty hace que la solución no tenga que respetar dichas limitaciones. Simplemente, se llega a un punto donde el módulo del gradiente se anula, lo que no implica en absoluto que el valor del penalty se anule y, en consecuencia, que se verifiquen las restricciones (Figura 2). Por lo tanto, la mejor solución suele ser un compromiso entre el valor del mínimo relativo y el cumplimiento de las restricciones.

El hecho de modificar las secciones a través de un algoritmo directo, donde se comparan las resistencias máximas de barras con los axiles que las solicitan, garantiza que las barras trabajarán en el límite. Esto facilita mucho la disminución del valor de la penalización

asociada a la rotura de las barras. Sin embargo, el inconveniente de utilizar este método para optimizar secciones es precisamente que sólo tiene en cuenta el criterio de rotura de las barras, y por lo tanto las demás restricciones sólo se podrán cumplir a través de las variables que se renuevan con el algoritmo de los gradientes conjugados. Relacionado con este hecho, también se ha observado un gran número de casos donde el proceso de convergencia se estanca y no avanza, pero sin que el gradiente llegue a anularse. Esto se atribuye a que las restricciones en secciones no intervienen de forma explícita en la función de penalización, y por lo tanto el avance en el algoritmo global puede verse coartado por las restricciones en las secciones. En este caso, se llega a un mínimo en la frontera de las variables de diseño (secciones), por lo que el gradiente no tiene porque anularse cuando las variables (coordenadas) no avanzan.

La implementación del código

El algoritmo de optimización así como el análisis de las estructuras de barras se han implementado en un código orientado a objeto. Esta nueva manera de programar ha tenido un fuerte impulso en la década de los 90 con el lenguaje C++ a pesar de que antes ya existían lenguajes orientados a objetos.

La programación orientada a objetos está basada en una nueva forma de organizar la información y de la interacción entre los diferentes datos que contiene. Se considera que los códigos desarrollados son más comprensibles y coherentes, de manera que facilita la reprogramación y la transferencia de código. La escritura de programas de análisis de estructuras mediante lenguajes poco o nada estructurados no presta atención a la naturaleza intrínseca de las magnitudes con las que trata. Es decir, sea cual sea la naturaleza propia de la información (desplazamientos, matrices de rigidez, matrices de cambio de coordenadas, etc.), ésta siempre se organiza bajo la misma estructura formal, básicamente vectores o matrices. Esta técnica de programación no es cómoda para una transferencia de código, pues la organización de los datos es arbitraria y la representación formal de la propia información no ayuda a identificar su contenido, el programador no sabe qué es qué ni cómo interacciona con otro tipo de datos. Este es un inconveniente leve cuando el programa es desarrollado íntegramente por un solo programador, pero se vuelve importante cuando se debe actuar sobre un código ajeno o cuando se va a trabajar en grupo.

La programación orientada a objetos tiende a organizar de forma natural la información, por ejemplo en un código orientado al objeto (CO_2) se puede diseñar una unidad de información llamada material que contenga un número real llamado *área*, otro llamado *young* y otro llamado *inercia*. A partir de la definición de este tipo de unidad de información se pueden crear tantas unidades materiales (objetos tipo material) como distintos materiales tenga el problema. En un lenguaje convencional el soporte para almacenar las propiedades del material de una barra se hace mediante una matriz con tantas filas como tipos de materiales y tantas columnas como tipos de propiedades tenga un determinado material. Aquí se aprecia el problema, en la programación de objetos se accede al área mediante la variable área del objeto material, en la convencional mediante el índice i, j de la matriz material, la claridad está fuera de toda duda.

La programación de objetos une los tipos de datos con las cosas que se pueden hacer con dichos datos, por ello cada objeto informa de forma natural no sólo de lo que contiene, sino también de lo que puede hacer. En la programación convencional la información va por un lado y aquello que se hace con ella por otro.

El lenguaje C++ tiene toda la potencia de la programación orientada a objetos y además la programación estructurada secuencial, por ello ha sido escogido como el instrumento más útil. Estas cualidades han permitido formular un código coherente que ha facilitado el desarrollo del programa de optimización. La información y funciones han sido agrupadas y jerarquizadas de una manera natural que facilita la comprensión y desarrollo del código.

Diversos autores han hecho aportaciones sobre la estructura de datos para una programación de objetos de un programa de cálculo de estructuras, cabe citar^{4,11}.

En este trabajo, la estructura de la información de los objetos encargados del cálculo matricial es:

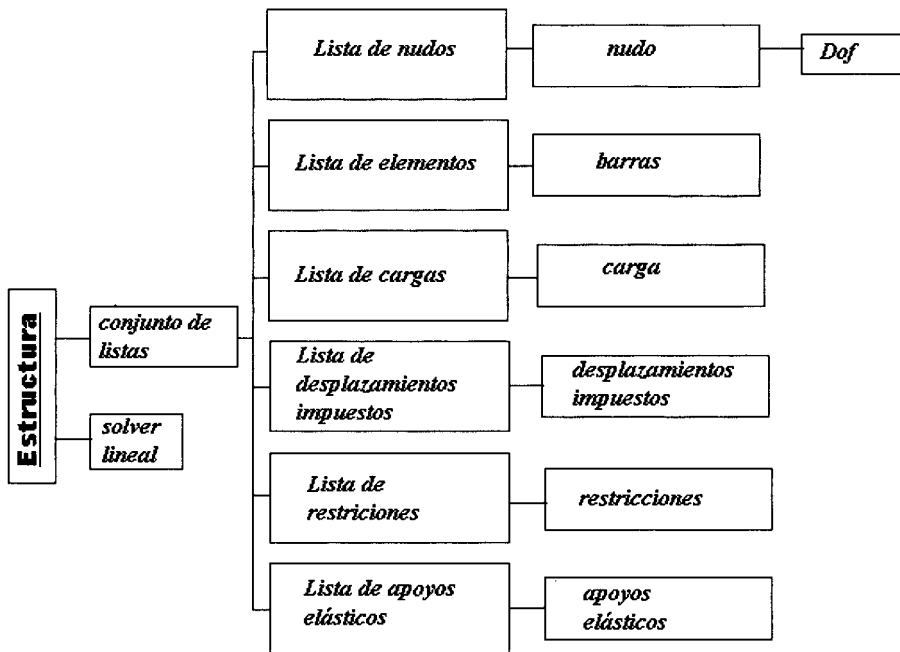


Figura 4. Organización de objetos para el análisis matricial

Además, la estructura de objetos del algoritmo de optimización es:

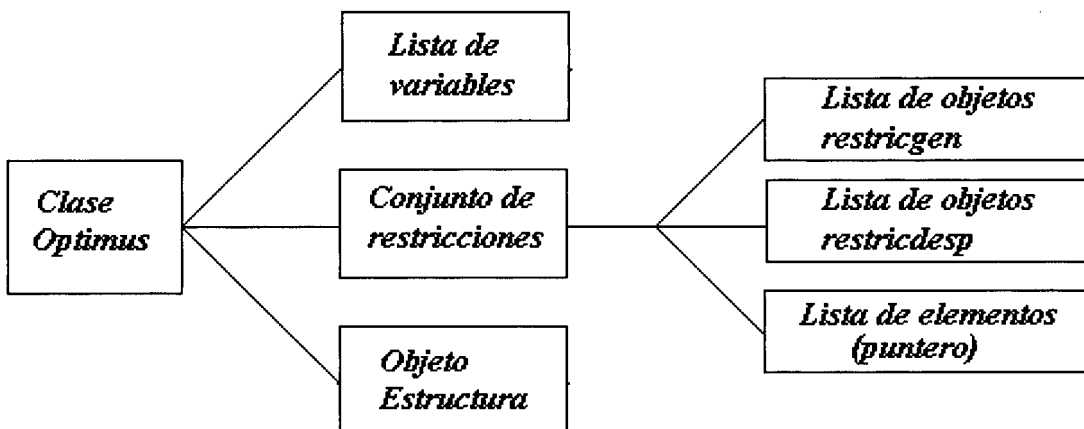


Figura 5. Organización de objetos para la optimización

Ejemplos de aplicación

A continuación se muestran algunos ejemplos que se han resuelto con el programa.

Ejemplo 1

En este ejemplo se desea optimizar una viga isostática cuyo diseño original se concibe como una celosía de canto constante con diagonales a 45° . Se toman como variables del problema las secciones transversales de cada pareja de barras simétricas, teniendo como límite de variación $0,5 < A_i < 60 \text{ cm}^2$. Asimismo, se toman como variables de diseño las coordenadas de los nudos, excepto los apoyos que se consideran fijos. Las acciones del problema son el peso propio y las cargas indicadas en la Figura 6.

Para el diseño original se necesitan unos 4900 kg de material para que ninguna barra rompa. Tras 45 iteraciones del método propuesto se obtuvo el siguiente diseño de la Figura 7 donde se aprecia claramente que los cordones adoptan un trazado que se adapta a la ley de flectores. También se observa que los montantes verticales se inclinan algo. En este caso, el peso final de la estructura optimizada es de 1754 kg.

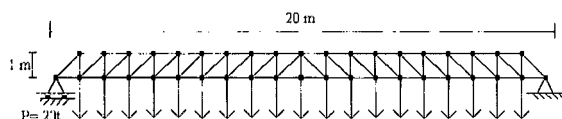


Figura 6. Estructura inicial

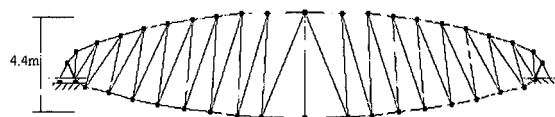


Figura 7. Estructura optimizada

Es destacable la similitud de la geometría resultante de la reparación realizada en el puente sobre el Trissana, en el trazado del ferrocarril del Arlberg (Austria). Hans Wittfoht¹⁰ comentaba sobre este puente: “*puede notarse con que atrevimiento, en parte excesivo, se aprovechó el material*”.

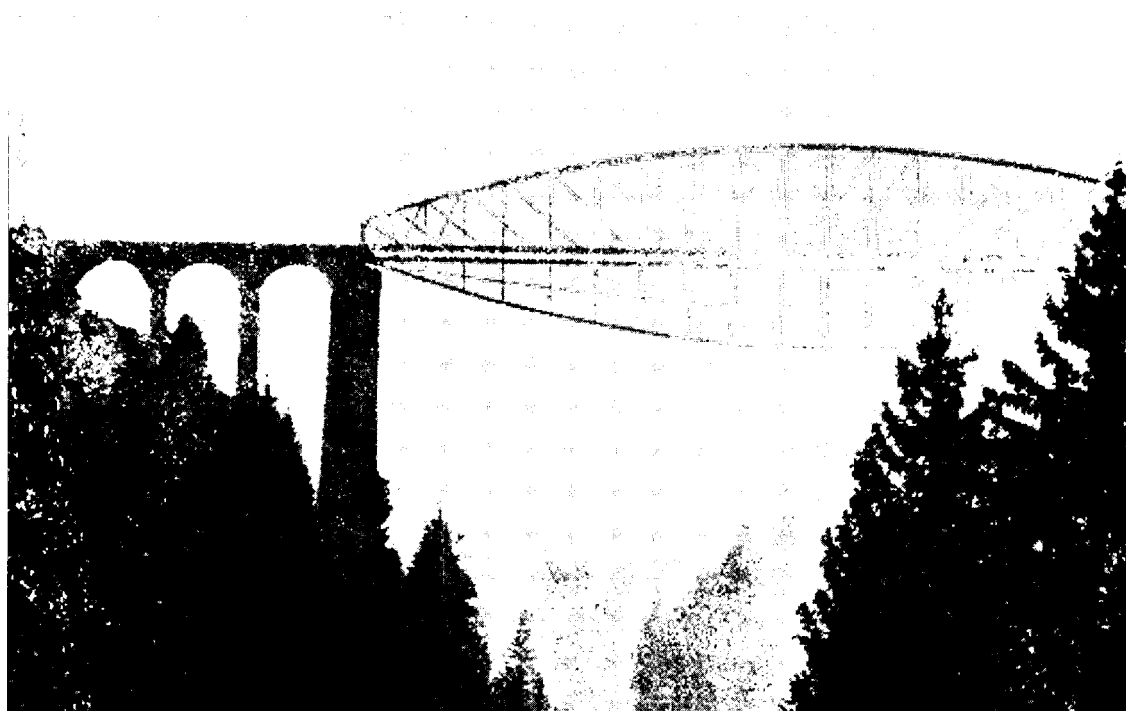


Figura 8. Puente de ferrocarril sobre el Trissana. Año de construcción 1884

Ejemplo 2

En este ejemplo se optimizará la misma viga del caso anterior pero variando las condiciones de contorno en el apoyo izquierdo (Figura 9). En concreto, se impide el desplazamiento horizontal y en consecuencia puede resistir esfuerzos horizontales.

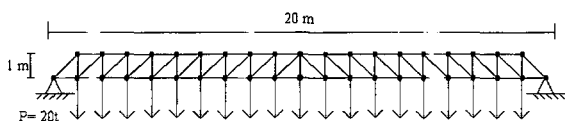


Figura 9. Estructura inicial

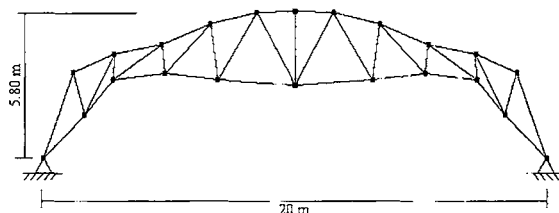


Figura 10. Estructura final

El peso del material necesario para que la estructura inicial no rompa es de unos 3300 kg. Después de 42 iteraciones se obtuvo el diseño óptimo con un peso de 8062 kg. En este caso se muestra que adoptar la forma de arco antifuncular (directriz parabólica) es la manera más eficiente de soportar las cargas externas gracias a que las condiciones de contorno tienen la capacidad de absorber reacciones horizontales (Figura 10). La magnitud de los esfuerzos axiales se muestra en la Figura 11. Obsérvese que las barras más solicitadas definen la directriz curva del arco, mientras que el resto de barras tiene una función secundaria de arriostramiento y rigidización.

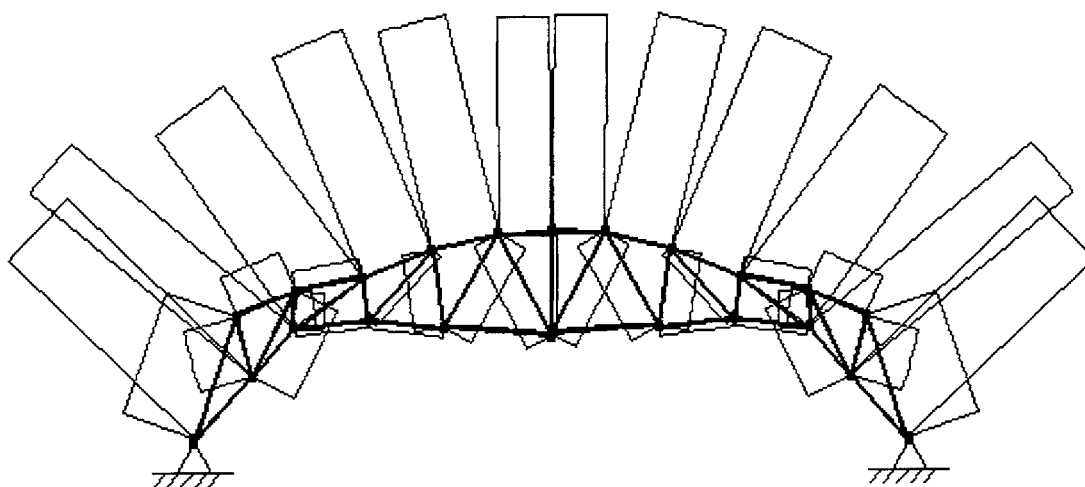


Figura 11. Ley de esfuerzos axiales de la estructura de la Figura 10

Ejemplo 3

El ejemplo que se propone a continuación optimiza una viga continua en celosía de infinitos vanos de longitud de 16 m y 1 m de canto. Se toman como variables del problema las secciones transversales de cada pareja de barras simétricas, teniendo como límite de variación $0,5 < A_i < 100 \text{ cm}^2$. Asimismo, se toman como variables de diseño las coordenadas de los nudos, excepto en los apoyos donde sólo podrán variar las coordenadas verticales. Las

acciones son el peso propio y una carga aplicada en los nudos inferiores. Si se asume que la viga es infinitamente larga, se puede modelizar a través de un vano, considerando que trabaja como biempotrado. El esquema estructural de este caso se encuentra en la Figura 12.

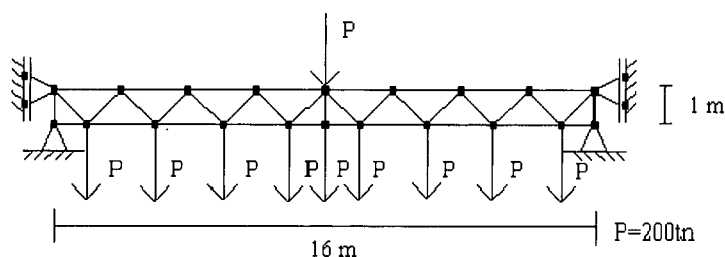


Figura 12. Esquema de cálculo del problema

La cantidad de material necesario para que no rompa la estructura bajo el diseño original es de unos 16700 kg. Tras 38 iteraciones se obtuvo un diseño optimizado como el de la Figura 13. La forma óptima para este esquema estructural consiste en adoptar dos tramos que funcionan como un voladizo y un tramo intermedio que trabaja como viga biapoyada cuyo canto se adapta la ley de flectores. Es destacable el estrangulamiento que de forma natural adopta el canto, formando una rótula justo donde el flector se anula para una ley de viga biempotrada. El canto en apoyos es de 5,4 metros, siendo superior al del centro de vano, de sólo 4,15 metros. Efectivamente, en una viga biempotrada el flector en el empotramiento es mayor que en el centro de vano. La estructura optimizada pesa 7102 kg.

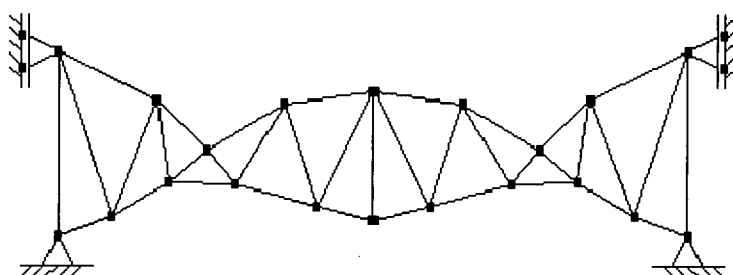


Figura 13. Forma óptima del problema

Si se adapta esta configuración para realizar una viga continua, se obtendría el aspecto de la Figura 14 que se puede comparar con el del *First of Forth*¹⁰. Puente de voladizos de inercia variable en apoyos y tramo isostático central también de inercia variable. El esquema estructural que se adoptó en el diseño de este famoso puente posiblemente estuvo condicionado por el proceso constructivo y no por el ahorro del material, pero es de admirar la enorme intuición estructural del ingeniero.

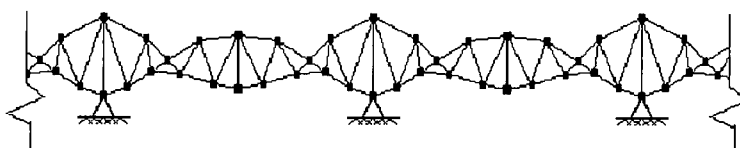


Figura 14. Esquema estructural optimizado formado por la continuidad de los vanos



Figura 15. Vista del puente First of Forth construido durante el período 1883-1890

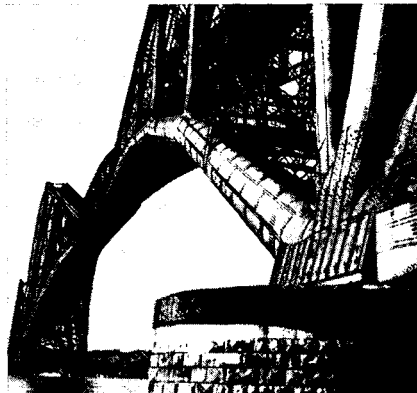


Figura 16. Detalle de la forma geométrica

CONCLUSIONES

La programación orientada a objetos PO_2 constituye un paso más en la evolución de los lenguajes y estrategias de programación. Su capacidad de organizar, tanto la información de datos como las funciones que operan sobre ellos, la convierte en una herramienta muy válida y operativa en el desarrollo de programas complejos.

La experiencia práctica demuestra que la optimización es una cuestión compleja y muy sensible a las características particulares del problema con que uno se enfrenta. En concreto se evidencia que el hecho de utilizar variables de diferente naturaleza (secciones y geometría) en un mismo algoritmo de optimización (método de los gradientes conjugados) produce muchos problemas de convergencia. Debido a la diferente naturaleza de las variables, así como el orden de magnitud muy diferente que existe entre ellas ocasiona una gran inestabilidad numérica. Para resolver esta cuestión es necesario combinar adecuadamente diferentes algoritmos de optimización en función de los diferentes tipos de variables que parametrizan el problema.

Se destaca que el método de la función penalty, para de considerar las restricciones, tiene una gran sencillez y generalidad. Esta ventaja conlleva dificultades en la convergencia e inseguridad a la hora de cumplir dichas restricciones. De hecho, el método se muestra muy sensible a los factores que escalan el valor de excedencia del parámetro limitado y el propio límite. Sin embargo, en el caso de que no se cumplan las restricciones impuestas, o bien que el algoritmo no converja, siempre se puede modificar el valor del parámetro de escala resolviendo el problema. En particular, las restricciones en coordenadas nodales se respetan colocando los coeficientes de peso adecuados, en caso contrario el proceso se vuelve numéricamente inestable. Hay que destacar que la metodología propuesta no crea ningún conflicto para las restricciones en variables seccionales porque se imponen automáticamente.

Sin ninguna duda, aquello que influye decisivamente a la hora de mejorar la solución y la convergencia del proceso es la elección de las variables más significativas del problema. Su variación influirá claramente en el diseño de la estructura.

El algoritmo propuesto en este trabajo se muestra altamente estable partiendo incluso de estructuras iniciales muy alejadas de la solución. De hecho las soluciones que se obtienen se adaptan perfectamente a los criterios intuitivos de la ingeniería. Se observa que la estructura optimizada tiene una forma geométrica homotética o relacionada con la ley de esfuerzos flectores de la estructura original. Por lo tanto, a medida que adaptamos la geometría de la estructura al camino natural que permite transmitir las cargas hacia los apoyos, se obtiene una estructura que trabaja mejor y que disminuye la cantidad de material para resistir el mismo esfuerzo.

No obstante, creemos que bajo la tendencia actual con costes de material decrecientes y costes de mano de obra en auge, la función peso dejará de ser representativa del coste de una estructura por lo que dejará de ser objeto de algoritmos de minimización. Esta tendencia será tanto más acusada en países avanzados y en sectores productivos como el nuestro, donde el diseño de una estructura suele ser un prototipo único. En cambio si que seguirá teniendo sentido usar como función objetivo el peso en piezas fabricadas automáticamente y en serie (prefabricación) donde la cantidad de material repercute significativamente en el coste del producto.

REFERENCIAS

- 1 M. Beckers y C. Fleury, "A primal-dual approach in truss topology optimization", *Computers and Structures*, Vol. **64**, pp. 77-88, (1997).
- 2 Eurocódigo 3, "Proyecto de estructuras de acero".
- 3 S. Hernández, *Métodos de diseño óptimo de estructuras*, Col. Senior, Colegio de Caminos, Canales y Puertos, (1990).
- 4 B.W.R. Forde, R.O. Foschi y S.F. Stierner, "Object-Oriented Finite Element Analysis", *Computers and Structures*, Vol. **34**, pp. 355-374, (1990).
- 5 P.E. Gill y W. Murray, *Numerical methods for constrained optimization*, Academic Press, (1974).
- 6 K.I. Majid, *Optimum design of structures*, Ed. Butterworth, (1974).
- 7 M. Ohsaki, "Simultaneous optimization of topology and geometry of regular plane truss", *Computers and Structures*, Vol. **66**, pp. 66-77, (1997).
- 8 J.E. Taylor, "On the prediction of topology and local properties for optimal trussed structures", *Structural Optimization*, Vol. **14**, pp. 53-62, (1997).
- 9 G.N. Vanderplaats, *Numerical optimization techniques for engineering design*, Ed. McGraw-Hill, (1983).
- 10 H. Wittfoht, *Puentes, ejemplos internacionales*, Gustavo Gili Ed.
- 11 T. Zimmermann, Y. Dubois-Plerin y P. Bomme, "Object-Oriented finite element programming: I Governing principles", *Computer Methods in Applied Mechanics and Engineering*, Vol. **98**, pp. 291-303, (1992).