OPTIMIZACION DE ESTRUCTURAS MEDIANTE ALGORITMOS GENETICOS

José Antonio Pérez Ruy-Díaz
Alfonso Alvarez Martínez

1. Introducción

El diseño de una estructura, y en general de cualquier obra de ingeniería, lleva implícito la búsqueda de una solución óptima, es decir, la que cumple las diversas condiciones de funcionalidad y seguridad con un coste mínimo. Resulta, por consiguiente, comprensible el interés que siempre ha existido por la aplicación de métodos de optimización a esta tarea. Dichos métodos han sido muy variados: analógicos y numéricos, diseñados para resolver problemas específicos o generales basados en la Programación Matemática [BEL85][HAF86].

Conviene de todas formas matizar que el concepto de «optimización» no puede ser interpretado en términos matemáticos exactos, ya que los diferentes parámetros que intervienen en el problema no se conocen con precisión absoluta, sino en términos de búsqueda de una solución satisfactoria. Esta razón, unida a las dificultades de aplicación de los métodos rigurosamente matemáticos, ha frenado su empleo sistemático en la ingeniería.

Con el desarrollo de las nuevas tecnologías, se ha abierto la posibilidad de realizar planteamientos alternativos para resolver la deseada optimización [ARO86]. Un método especialmente interesante es el denominado **Algoritmo Genético** basado en los sistemas LCS (Learning Clasiffier Systems), empleados en Inteligencia Artificial. Tal herramienta fue empleada por el autor de este artículo en su Tesis Doctoral, dirigida por el catedrático D. Alfonso Alvarez Martínez [PER90], dedicada a la optimización de presas arco y su posible ejecución con rollcrete.

Este artículo describe el método y desarrolla un ejemplo sencillo, consistente en una presa de gravedad recta, a modo de ilustración.

2. Planteamiento General de un problema de optimización

La optimización del diseño de una estructura se puede plantear en términos de una minimización condicionada de la siguiente manera:

Supóngase una estructura cuya forma depende total o parcialmente de un conjunto de parámetros $b=(b_1, b_2, ..., b_{n-1}, b_n)$, sometida a la acción de un sistema de cargas estáticas. Supóngase igualmente que se tiene un modelo de Elementos Finitos de la estructura con matriz de rigidez K, para el que las cargas toman la forma de un vector F. En general, K y F dependen de la forma de la estructura y, por lo tanto, del vector b, con lo que las ecuaciones de equilibrio se pueden escribir:

$$K(b)\delta = F(b) \quad [1]$$

Estas ecuaciones permiten calcular los desplazamientos δ de los nudos para las cargas y a partir de ellos las tensiones σ en los distintos elementos. Puesto que las ecuaciones están expresadas en función de b, tanto δ como σ serán funciones de b igualmente y se tendrá:

$$\delta = \delta(b) \quad \sigma = \sigma(b) \quad [2]$$

Por otra parte, se tienen que cumplir ciertas condiciones relativas a la forma, seguridad y resistencia. Dichas condiciones dependen de la geometría de la estructura, las tensiones y/o los desplazamientos y se pueden expresar de la forma:

$$\Phi_i(b,\delta,\sigma) \leq 0 \quad i=1,\ldots,m \quad [3]$$

y teniendo en cuenta [2], simplemente

$$\Phi_i(b) \leq 0 \quad i=1,..,m$$

Estas condiciones no bastan para determinar completamente el valor del vector b. En general puede existir un número indefinido de combinaciones b_i que las cumplen. El planteamiento de optimización añade una condición más del tipo:

$$\min \Phi_0(b,\delta,\sigma) \quad [4]$$

con el fin de reducir el conjunto de soluciones a una sola. Se trata de la minimización de una función dependiente directa o implícitamente de la geometría. Por ejemplo, el volumen, aunque puede ser de cualquier otra magnitud ligada al coste de la estructura o al estado tensional. Substituyendo en la condición los valores dados por [2] para δ y σ, se tiene:

$$\min \Phi_0(b)$$

con lo que se llega al planteamiento de optimización conocido:

«Hallar el vector b que minimiza la función:

$$\Phi_0(b)$$

y cumple:

$$\Phi_i(b) \leq 0, \; i = 1,2,\ldots,m$$

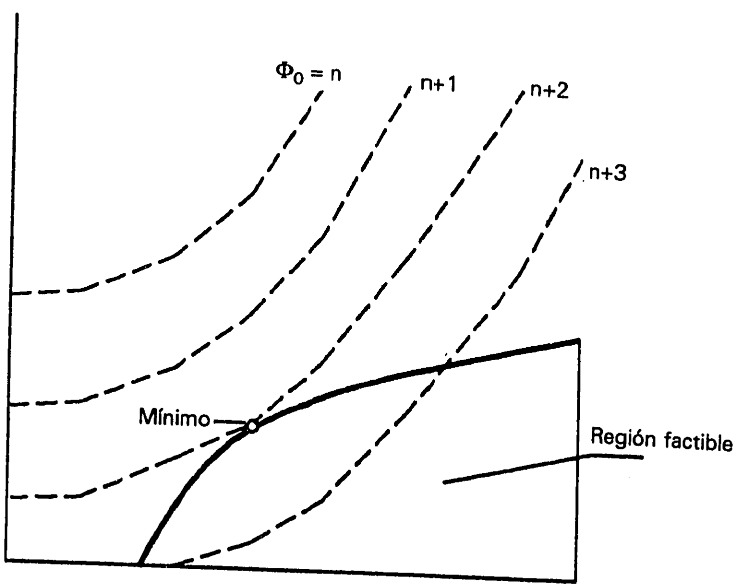


Fig. 1. Representación geométrica de un problema de optimización.

siendo $b \varepsilon R^n$ el vector de variables incógnita del problema, Φ_0 una función de coste, Φ_i la expresión analítica de ciertas condiciones que se tienen que cumplir».

Idénticos resultados se habrían obtenido para un sistema dinámico con sólo substituir [1] por $K(b)\delta + C(b)\delta' + M(b)\delta'' = F$, siendo C y M las matrices de amortiguamiento y masa respectivamente.

El conjunto de condiciones Φ_i delimita una región de R^n llamada espacio de soluciones o *región de factibilidad.* El problema consiste en hallar un *punto factible* que minimice la función de coste también llamada *función objetivo.*

Desde el punto de vista matemático el planteamiento resulta sencillo, aunque no siempre se puede decir lo mismo de la resolución del problema. Hay que hacer notar que no se han impuesto restricciones de ningún tipo a las funciones Φ_0 y Φ_i y que, además, no hay ninguna razón, para asegurar que la región de factibilidad sea siempre convexa. Por consiguiente pueden existir múltiples mínimos locales y no hay garantía de solución única. Pero, con ser importante este aspecto, existe una dificultad adicional nada despreciable consistente en que tanto Φ_0 como las Φ_i son funciones implícitas del vector de desplazamientos, dependiente a su vez del vector de variables de diseño b a través de las ecuaciones de equilibrio. En esas condiciones no resulta fácil la aplicación de los métodos de cálculo usuales.

A las anteriores consideraciones se pueden añadir otras derivadas de un planteamiento más próximo a la ingeniería. La primera de ellas, que no siempre se conocen todas las variables que intervienen en un problema, con lo que resulta vano exagerar la importancia de la exactitud. La segunda, relacionada con la anterior, que el concepto de óptimo puntual es ideal y que, en la realidad, resulta más apropiado hablar de áreas o dominios de soluciones casi óptimas. Por último, que el ingeniero tiene un conocimiento de experto que le permite acotar el dominio de las soluciones, por lo que frecuentemente la tarea de diseñar una estructura consiste en elegir como punto de partida algunas soluciones aceptables, combinarlas y comprobar resultados.

Este último aspecto, tan parecido al proceso de aprendizaje, es una buena razón para intentar abordar el problema mediante los Algoritmos Genéticos que se describen a continuación.

3. Introducción a los Algoritmos Genéticos

Los Algoritmos Genéticos son técnicas de optimización en los sistemas de aprendizaje. La forma de actuar de tales sistemas consiste en partir de un cierto estado e ir generando series de soluciones (clasificadores) cuyos efectos se contrastan con otras de resultados (mensajes). La efectividad de los mensajes se evalúa y con este dato se producen nuevas soluciones que se adaptan a cada circunstancia.

El planteamiento clásico de optimización mediante un algoritmo genético es:

$$\text{Max } \Phi(x_1, x_2, \ldots, x_n)$$

donde Φ(x) es una función positiva. Dejando aparte el hecho trivial de que se trata de una maximización en lugar de una minimización, los algoritmos genéticos (A.G.) se diferencian de los procedimientos normales de optimización en cuatro puntos fundamentales:

1. Los A.G. trabajan con codificaciones de los parámetros en lugar de con los propios parámetros.
2. Los A.G. parten de una población de soluciones, no de una única.
3. Los A.G. sólo usan los valores de la función objetiva, nunca de las derivadas u otra información auxiliar.
4. Los A.G. utilizan reglas de transición probabilísticas, no deterministas.

Los algoritmos genéticos requieren la codificación de los parámetros que caracterizan una solución en forma de una cadena finita (*cromosoma*) de caracteres de un alfabeto dado (*genes*). Así, cada solución se identifica mediante su cromosoma.

Supóngase, por ejemplo, que se trata de maximizar la función Φ(x) en el intervalo [0,31]. Con un método tradicional se emplearía el valor de x en los cálculos para hallar el valor máximo de la función. En un A.G., el primer paso consiste en codificar el parámetro x en forma de una cadena de longitud finita, por ejemplo su valor binario (considerando sólo los valores enteros de x). De este modo, todo valor entero de x vendrá representado por una cadena de cinco cifras binarias y un código binario se referirá a un valor de x entero. Si en lugar de Φ(x) se tuviera Φ(x,y), la codificación de las variables se haría de forma similar y el cromosoma correspondiente a un punto (x,y) sería la concatenación de los códigos binarios de x e y.

En la mayoría de los métodos de optimización iterativos, la solución se va desplazando desde un punto del espacio de decisión hasta el siguiente mediante una regla de transicción que lo determina. Un A.G. trabajan con una serie de puntos simultánemente (una población de soluciones), analizando sus características individuales e intentando extraer consecuencias globales que le permitan buscar una serie de soluciones mejores. Volviendo al ejemplo, el A.G. podría haber comenzado con un conjunto de soluciones elegidas de cualquier forma, por ejemplo:

01101

11000

01000

10011

A partir de esta población se generarían sucesivas poblaciones aplicando ciertas técnicas.

La mayor parte de los métodos de optimización requieren una gran cantidad de información auxiliar para poder trabajar correctamente. Por ejemplo, los métodos de gradiente necesitan derivadas calculadas analítica o numéricamente) para poder avanzar hacia el máximo; otros procedimientos de búsqueda local como las técnicas combinatorias de optimización [LAW76] necesitan acceder a un sinnúmero de parámetros. En cambio, los A.G. lo único que necesitan es el valor de la función objetiva asociado con cada código. No obstante, si se desea, puede emplearse información auxiliar mediante las técnicas denominadas Algoritmos Genéticos de Conocimiento Dirigido.

Los A.G. usan reglas de transición probabilísticas para guiar la búsqueda. Para alguien familiarizado con las técnicas deterministas, esto puede parecer extraño, pero no debe de ser interpretado como si se usara simplemente un «cara o cruz» para avanzar. Los algoritmos genéticos, emplean la elección al azar entre soluciones que tienen una probabilidad de mejorar la función objetiva.

La mecánica de un algoritmo genético es sorprendentemente simple, ya que sólo consiste en copiar y mezclar códigos. Para ello se define un conjunto de operaciones que generen nuevas poblaciones a partir de la inicial. El procedimiento se repite hasta alcanzar un cierto nivel de convergencia.

La mecánica de un algoritmo genético es sorprendentemente simple, ya que sólo consiste en copiar y mezclar códigos

Los operadores más usuales son:

1. Selección
2. Cruce
3. Mutación

La *Selección* es un proceso mediante el cual se elige un conjunto de soluciones de acuerdo con el valor de la función objetivo (El fuerte se impone al débil). Las mejores soluciones tendrán más probabilidades de contribuir con uno o más representantes (hijos) en la siguiente generación.

El operador selección se puede implementarse de muy diversas formas. La más fácil consiste en elegir con probabilidad proporcional a los valores de la función objetiva.

El *Cruce* es un operador que se aplica en dos pasos. En primer lugar, los elementos procedentes de la selección se emparejan aleatoriamente. Posteriormente, cada pareja procede a cruzarse intercambiando información genética y dando lugar a dos hijos. Con ello se obtienen nuevas soluciones que reemplazan a los más débiles (Los fuertes sobreviven).

Supóngase que el azar ha decidido el emparejamiento de las cadenas (01101) y (1100), entonces se elige al azar un número k (p.e k=4) y el resultado del cruce es:

$$\text{Nueva cadena}_1 = \text{Subcadena}_{11} + \text{Subcadena}_{22} = (0110) + (0) = 01100$$
$$\text{Nueva cadena}_2 = \text{Subcadena}_{21}\ \text{Subcadena}_{12} = (1100) + (1) = 11001$$

En cuanto a la *Mutación*, es un operador que actúa sobre los genes de los cromosomas conforme se van transfiriendo a la descendencia en el proceso de cruce. Su acción consiste en introducir cambios al azar en los mismos. La mutación es una protección frente a la pérdida irrecuperable de material genético de una forma prematura, pues los procesos de selección y cruce pueden destruir información valiosa.

Con todo, la mutación juega un papel secundario en los algoritmos genéticos al igual que sucede en la naturaleza. La práctica demuestra que los mejores resultados se obtienen con un número pequeño de mutaciones, pues, de lo contrario el algoritmo degenera en un paseo aleatorio.

Un procedimiento como el descrito puede sorprender a primera vista, pero, a poco que se reflexione, se puede entender la lógica subyacente:

- La Selección *acota* el espacio de búsqueda. Si no existiera más que este operador y el algoritmo volviera a seleccionar al azar dentro del nuevo intervalo de soluciones, ello bastaría en funciones unimodales para acercarse a la solución.
- El Cruce *recombina* los valores codificados de los parámetros, a la busca de una mezcla óptima. El hecho de que la combinación se haga con códigos binarios no debe confundir sobre el fin último: reconocer combinaciones optimas de los valores de las variables del problema.
- En cuanto a la Mutación, puede interpretarse como un *rearranque* limitado del proceso.

Con todo, el buen funcionamiento del Cruce es fundamental para la convergencia del algoritmo y gran parte de la justificación formal de la misma gira alrededor de dicho operador. Si genes individuales o en pequeños grupos influyen decisivamente en el valor de la solución, el cruce no destruirá significativamente el mensaje que los padres transmiten a los hijos y, por consiguiente, se avanzará hacia la solución (Hipótesis de los bloques de Holland) [HOLL75]. Si, por el contrario, fuera relevante la combinación de un gran número de genes contíguos, no se podría garantizar la convergencia.

Aunque la Hipótesis de Holland es bastante restrictiva (como corresponde a una condición suficiente de convergencia), se puede demostrar que se cumple cuando el grado de la función a optimizar no es elevado. Este es el caso del ejemplo que se desarrolla a continuación y el de otros más complejos de optimización de presas a poco que se cuide el planteamiento [PER90].

4. Prototipo de Algoritmo Genético

Veamos ahora un problema sencillo, con solución analítica, con el fin de ilustrar el funcionamiento de un algoritmo genético y comprobar su eficacia.

Planteamiento

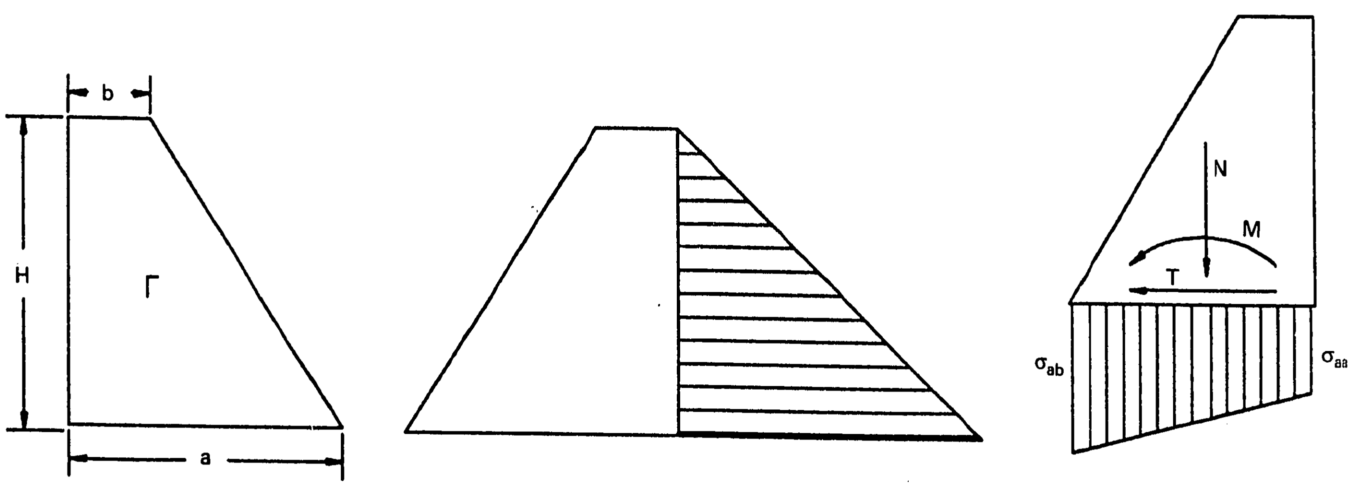
Supongase, que se prentende determinar el perfil de una presa de gravedad de altura H conocida y paramento agua arriba vertical, siendo la anchura de la base (a) y de la coronación (b) incógnitas, de forma que la presa resista a unas condiciones normales de carga como se indica en la figura 2.

Si se admite como válida la reducción de la estructura a una ménsula, suponiendo que la distribución de tensiones en la base es lineal, para una carga de peso propio y empuje hidrostático con el agua al nivel de la coronación se tiene:

$$\sigma_{ab}(h) = -\Gamma h b^2/a^2 - h^3/a^2$$
$$\sigma_{aa}(h) = -\Gamma h(1 + b/a - b^2/a^2) + h^3/a^2$$

Siendo $\sigma_{aa}(h)$ y $\sigma_{ab}(h)$ las tensiones agua arriba y agua abajo a una profundidad h, a y b la anchura de la base y la coronación, y Γ el peso específico del hormigón.

Figura 2.



Para que sea aceptable, la sección de la presa habrá de cumplir algunas condiciones:

■ En cuanto a resistencia, por ejemplo, que la presa esté libre de tracciones. De [22] se deduce que la tracción máxima se producirá en el talón de la presa, por lo que suponiendo que la subpresión toma un valor igual a la presión hidrostática, esta condición se puede escribir:

$$\sigma_{aa}(H) + H \leq 0$$

■ En cuanto a estabilidad, que la presa no deslice, es decir que la fuerza de rozamiento tendrán que superar a la tengencial:

$$N'tg\Phi \,/\, T \geq K$$

siendo N ́el esfuerzo axil efectivo tras restar la subpresión, Φ el ángulo de rozamiento del cimiento y K el valor del coeficiente de seguridad.

Por consideraciones evidentes de diseño, el ancho de coronación tendrá que ser menor que la base y se puede suponer igualmente que tiene que ser mayor que un mínimo, es decir:

$$b_{min} \leq b \leq a$$

Substituyendo los valores de las tensiones en función de las variables a y b dados por [22], las condiciones anteriores quedarán:

$$\Phi_1(a,b) \equiv b_{min} - b \leq 0$$

$$\Phi_2(a,b) \equiv b - a \leq 0$$

$$\Phi_3(a,b) \equiv Ha^2 + H^3 - \Gamma H(a^2 + ab - b^2) \leq 0$$

$$\Phi_4(a,b) \equiv KH^2 - (\Gamma H(a+b) - Ha)tg\Phi \leq 0$$

La figura 3 muestra las condiciones dibujadas en el plano de ejes de coordenadas (a,b). Se indica con sombreado la zona en que se cumplen las mismas.

Para tomar una decisión entre todas las secciones que cumplen con los requisitos impuestos se necesita algún criterio adicional. Se supone que se elige la de menor superficie. Se tendrá entonces una condición de la forma:

$$\min \Phi_0(a,b) = (a+b)H/2$$

Ahora bien, $\Phi_0(a,b) = (a+b)H/2$ es una recta que intersecta a igual distancia a los ejes a y b y cuya distancia al origen de coordenadas es la superficie de la sección. Por lo tanto, si por un punto cualquiera del cuadrante positivo (a+,b+) se hace pasar una paralela a la anterior, su distancia al origen será proporcional a la superficie correspondiente. Entonces, la sección de menor superficie de todas las que cumplen las condiciones será la correspondiente al punto cuya recta esté más próxima al origen. Este no es otro que el punto de tangencia al contorno. Trazando la tangente al mismo como se ve en la figura, se obtendrían los valores a y b buscados.

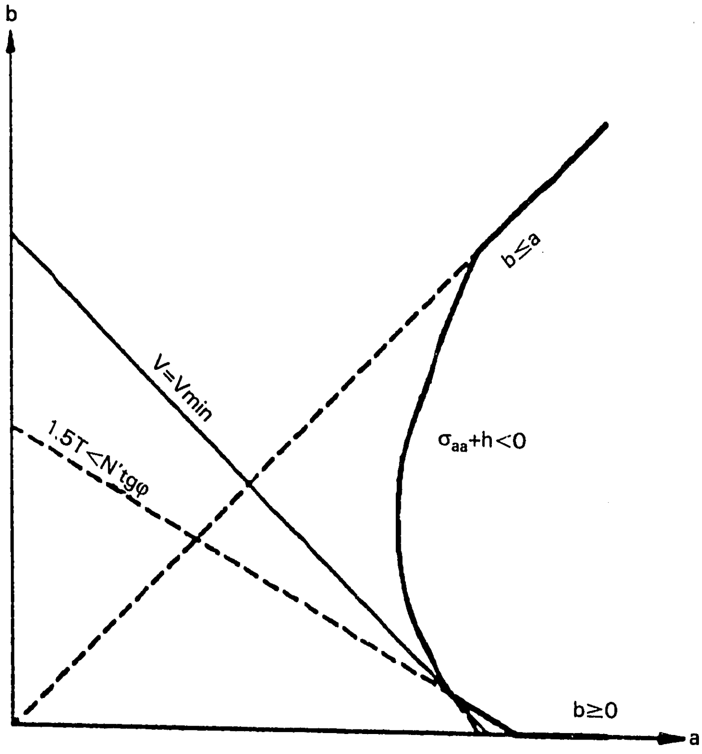


Figura 3.

Aplicando lo anterior al problema:

$H = 90$ m

$\Gamma = 2.35$ T/m3

$tg\Phi = 1.2$

$K = 1.5$

$b_{min} = 0$

se obtiene como solución a= 70.3 b=10, que satisface las ecuaciones [23] y dan una sección mínima de 3613.5 m^2.

Algoritmo Genético

Veamos ahora como se resuelve el problema mediante un Algoritmo Genético:

Los individuos en este caso son modelos de presas, definidos mediante sus parámetros a y b, cada uno de los cuales tiene un valor según su volumen (magnitud que se pretende minimizar) y el cumplimiento de las condiciones indicadas en [23].

Una forma de construir una función de valoración con las funciones a optimizar $\Phi_0(a,b)$ y los condicionantes $\Phi_i(a,b)$ es el modelo cuadrático de penalización de Courant:

$$F(a,b) = \Phi 0(a,b) + \sigma\Sigma(\Phi_i^+(a,b))^2 \quad [25ñ$$

donde $\Phi_0(a,b)$ y $\Phi_i(a,b)$ vendrán dadas por las expresiones [23] y [24]. El factor σes un coeficiente de penalización y σ_i^+ representa la medida del incumplimiento de la condición $\Phi_k \leq$. Así, $\Phi_k^+ = \Phi_k$ si se viola la condición y $\Phi_k^+ = 0$ en caso contrario.

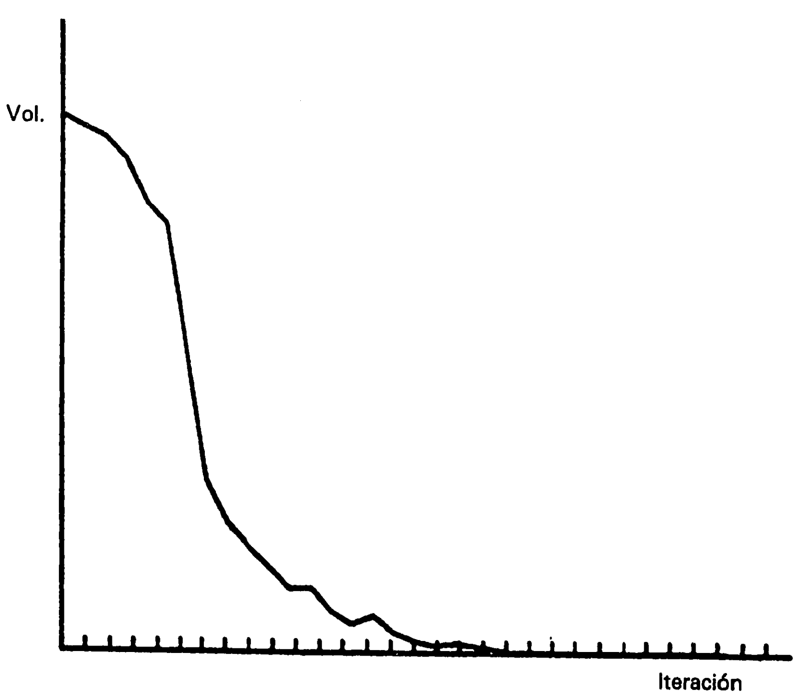
No hay que olvidar que el planteamiento de un algoritmo genético es de maximización, en lugar del clásico de minimización y que la función de valoración tiene que ser siempre positiva. Para lograrlo, basta con considerar una cota superior de la función que incluya los valores máximos del volumen de la presa y las penalizaciones, f_{max}, y emplear finalmente:

$$f(a,b) = f_{max} - (\Phi_0 + \sigma\Sigma(\Phi_i^+)^2) \quad [26]$$

Siguiendo la descripción hecha del algoritmo genético simple, es preciso asignar un código genético a cada una de las presas. Como ya se indicó, se puede construir un cromosoma mediante cifras binarias que representen el valor binario de forma unívoca. Para este ejemplo el código se puede construir concatenando dos números binarios de seis cifras, uno indicando las dimensiones de la base en el rango comprendido entre 26 y 90 metros y el otro las de la coronación en el rango 0 a 64.

La población inicial se ha elegido al azar para aumentar la dificultad. En la práctica podría haber sido elegida con un criterio de diseñador experto.

Figura 4.



En la figura 4 se muestra la evolución del algoritmo partiendo de 30 modelos distintos, empleando una probabilidad de cruce $p_c = 0.95$ y una probabilidad de mutación $p_m = 0.001$. Como se puede comprobar, el algoritmo converge al cabo de 25 iteraciones, habiéndose obtenido valores razonablemente próximos al óptimo al cabo de 10.

En la tabla adjunta se muestran los valores máximo, mínimo y medio de la función de valoración de la población para las generaciones 0 (inicial), 1, 10, 20 y 25 (final).

Como se puede ver, se comienza con una población generada al azar cuya valoración media es 1655 y en la generación 10ª, ya se ha alcanzado un valor medio de 3035. Este valor va creciendo asintóticamente hasta 3542 en la generación 25, momento en el que se alcanza la convergencia.

En la última generación se tiene una población prácticamente uniforme con parámetros a=70.7 b=10.1. La superficie para la misma es S=3636, lo que significa un error sobre el óptimo analítico del 0,6 %.

Generación	maxval	minval	medval
0	3.4473e+003	0.0000e+000	1.6555e+003
1	3.4930e+003	0.0000e+000	1.9388e+003
10	3.5633e+003	2.1077e+003	3.0354e+003
20	3.5633e+003	0.0000e+000	3.3841e+003
25	3.5633e+003	3.1714e+003	3.5492e+003

El número de individuos (presas) diferentes estudiados fue de 270 sobre un total de 2^{12}=4096 casos posibles, algo menos del 7 %, cifra exagerada por las condiciones extremas del experimento: población al azar, excesivo número de iteraciones, etc. y por tanto suceptible de mejora.

5. Conclusiones

Se ha podido comprobar que la mecánica de un Algoritmo Genético se puede aplicar a la optimización de una estructura. A pesar de su apariencia sencilla, los operadores de selección, cruce y mutación encuentran un valor próximo al óptimo muy rápidamente, tras examinar una pequeña fracción de todas las soluciones posibles.

Aunque se trataba de un algoritmo simplificado, el rendimiento obtenido para el ejemplo propuesto puede considerarse aceptable, tanto en el resultado como en el esfuerzo de cálculo realizado. Aplicando conocimientos previos de diseño y criterios de optimización más realistas se podrían haber obtenido mejores resultados.

En resumen, se trata de una herramienta fácilmente aplicable a la resolución de un cierto número de problemas de ingeniería que merece ser tenida en cuenta y de la que cabe aún esperar mejores resultados.■