

COLABORACIONES EN MATEMÁTICAS

OPTIMIZACIÓN GLOBAL DE FUNCIONES CONTINUAS NO DIFERENCIABLES

INTRODUCCIÓN

Optimizar el uso de algún recurso natural, económico o social es uno de los problemas de mayor interés del mundo moderno. Basta ojear cualquier publicación periódica para comprobar la presencia de estos problemas en casi todas las áreas de la vida actual. En algunos casos, la dificultad de esa optimización está en definir un proceso que modelice los problemas. En otros, sobre todo si los modelos aplicados están descritos por funciones, la dificultad está en obtener el valor óptimo y los puntos en donde se alcanza ese óptimo.

Estos últimos casos son reconocidos como problemas de maximizar o minimizar una función y se estudian en casi todas las etapas formativas de un estudiante. Los problemas de optimización son tratados desde la etapa de Enseñanza Secundaria, por ello suponemos que todos los estudiantes universitarios están capacitado para obtener el valor mínimo de una función; por ejemplo, de la función:

$$f(x) = \cos(x).$$

Un primer problema puede ser dar respuesta la siguiente cuestión: Determinar el valor mínimo de la función:

$$f(x) = |\cos(x)|.$$

La respuesta obliga a tratar con un conjunto de puntos aislados y no acotado de valores de x en los que se alcanza ese mínimo.

Una cuestión análoga a la anterior consiste en minimizar la función:

$$f(x_1, x_2) = \left| \sin \left(\sqrt{x_1^2 + x_2^2} \right) \right|.$$

Este caso puede presentar alguna dificultad para el estudiante universitario, sobre todo, si no realiza la siguiente sustitución $\rho^2 = x_1^2 + x_2^2$. Este cambio facilita la tarea de obtener el valor mínimo de la función $f(\rho, \theta) = |\sin(\rho)|$, pero emerge una nueva dificultad: Al determinar el conjunto de puntos: (ρ, θ) o (x_1, x_2) donde se alcanza ese valor mínimo, se obser-

va que dicho conjunto está compuesto por una familia, no acotada, de circunferencias concéntricas.

Es evidente que al incrementar la dimensión del espacio, en la cuestión análoga:

$$f(x_1, \dots, x_n) = \left| \sin \left(\sqrt{x_1^2 + \dots + x_n^2} \right) \right|,$$

el conjunto de mínimos se hace más intratable.

Esta dificultad, el obtener el conjunto de mínimos, permanece aunque se restrinja a una simple caja (por ejemplo, $[-2, 2] \times \dots \times [-2, 2]$), el dominio sobre el que está definida la función.

Al analizar las cuestiones anteriores, se observa que la función de cada problema no es diferenciable sobre su dominio de definición. Las técnicas basadas en el Cálculo Diferencial no son directamente utilizables en estos casos. Este hecho implica la utilización de algoritmos no diferenciables para su resolución exacta o resolución aproximada. No es raro que los modelos utilizados empleen funciones que no son diferenciables para tratar situaciones de la vida diaria.

También podemos preguntarnos: ¿Qué sucede si se realizan algunas modificaciones elementales en las expresiones de las funciones anteriores? Por ejemplo, al tratar de minimizar la función:

$$f(x_1, x_2) = \left| \sin \left(0,5 + 9,5\sqrt{x_1^2 + x_2^2} \right) \right|.$$

considerada la restricción $x_1 \in [0, 2]$, $x_2 \in [0, 2]$.

En este trabajo presentamos una respuesta a los problemas mencionados mostrando un algoritmo de resolución. Se trata de una variante del *Cubic Algorithm* de la Ref. [1], presentada en la Ref. [2], para tratar problemas de optimización global de funciones continuas que están restringidas a una caja.

El lector interesado puede obtener el código MAPLE de dicha variante en la Ref. [2], que fue desarrollado para funciones de n variables y este tipo de optimización. En el código se incorporan varios módulos con aplicaciones directas como la determinación de las raíces de una ecuación polinómica, el cálculo de autovalores, el cálculo de las frecuencias y de las raíces de la ecuación de frecuencias, y la resolución de sistemas de ecuaciones no lineales, tanto para el caso en que haya menos ecuaciones que incógnitas, como en el caso contrario en que haya más ecuaciones que incógnitas. Es evidente que estas aplicaciones son tratadas como *problemas de optimización global*. El código genera aproximaciones numéricas y aproximaciones gráficas del conjunto completo de puntos óptimos.

EL PROBLEMA

Se trata de resolver el problema de encontrar el valor mínimo global:

$$s^0 = \min f(x), x \in \bar{X} \subset \mathbb{R}^n, \quad (1)$$

en la n -caja (n -rectángulo):

$$\bar{X} = \{x \in \mathbb{R}^n : a_i \leq x_i \leq b_i, i=1, \dots, n\}, \quad (2)$$

y establecer el conjunto de mínimos globales:

$$X^0 = \{x \in \bar{X} : f(x) = s^0\}. \quad (3)$$

En este caso, está claro que $a_i < b_i, \forall i$, y la barra superior sobre un conjunto indica que se trata de un conjunto cerrado. Además, supondremos que la función $f(x)$ es Lipschitziana sobre \bar{X} , es decir:

$$|f(x) - f(x')| \leq A \|x - x'\|, \forall x, x' \in \bar{X}. \quad (4)$$

El problema de obtener el valor máximo de $f(x)$ es equivalente a determinar el valor mínimo de $-f(x)$.

Observación: Si se conoce la mínima constante de Lipschitz sobre \bar{X} :

$$L = \max \|\nabla f(x)\|; \quad x \in \bar{X}, \quad (5)$$

entonces se elige $A = L$ en (4). Generalmente, la constante L de (5) es desconocida o difícil de calcular.

En muchos casos sólo se sabe que la función es computable e incluso se desconoce su fórmula. En tales casos, un subconjunto de \bar{X} puede tener un valor de L menor que el de (5) sobre todo \bar{X} . Así pues, no asociamos A en (4) con L en (5) y la consideramos como una simple cota de las pendientes de $f(x)$ en \bar{X} , es decir, consideramos el ángulo:

$$\alpha = \arctan A; \quad 0 < \alpha < \pi/2. \quad (6)$$

El lector puede probar que si se encuentra (4), entonces la pendiente de $f(x)$ en alguna sección plana de \bar{X} que pasa por x, x' no es mayor que α . En la Tabla 1 se indican distintas cotas de pendientes para tratar en los casos más habituales de una función de pendiente acotada.

| Tabla 1 | | | | | | | | | |
|----------|----|--------------|-----|------------|-----|-----|-----|-----|-----|
| α | 0° | 30° | 45° | 60° | 63° | 71° | 76° | 78° | 80° |
| A | 0 | $1/\sqrt{3}$ | 1 | $\sqrt{3}$ | 2 | 3 | 4 | 5 | 6 |

Para $A > 3$ debería aplicarse un cambio de escala para poder asegurar un tiempo de cálculo razonable en la ejecución del algo-

ritmo. Por ejemplo, en lugar de tratar la función $f(x)$ se trataría la función $f(x)/10$ puesto que ambas funciones tiene el mismo conjunto de mínimos.

TRANSFORMACIÓN DEL PROBLEMA AL CUBO UNIDAD

Al utilizar la transformación afín:

$$x_i = a_i + (b_i - a_i) z_i, \quad i=1, \dots, n, \quad (7)$$

el problema planteado en (1) y (2) se convierte en otro problema dentro del cubo unidad, \bar{U} , cubo orientado según los ejes, de longitud de lado $c = 1$ y con un vértice en el origen de coordenadas. En este caso, la nueva función es:

$$g(z) = f[a+(b-a)z]; \quad z \in \bar{U} = \{z \in \mathbb{R}^n : 0 \leq z_i \leq 1\}, \quad (8)$$

expresión en la que se ha utilizado notación vectorial.

El problema, que tratamos de solucionar algorítmicamente, es determinar el valor mínimo global s^0 :

$$s^0 = \min g(z) = \min f(z); \quad z \in \bar{U}, x \in \bar{X}, \quad (9)$$

y el conjunto de puntos mínimos U^0 :

$$U^0 = \{z \in \bar{U} : g(z) = s^0\}. \quad (10)$$

Una vez obtenido este conjunto, al aplicar la transformación inversa de (7) al conjunto de (10) se obtienen el conjunto de soluciones de (9) contenido en la caja inicial.

Observación: El código Maple de la Ref. [1] (págs. 13-15), que desarrolla el algoritmo de aproximación de las soluciones, requiere los siguientes datos de entrada: la dimensión del espacio n , la función f y la caja de (1) y (2), para calcular una aproximación (s^0, X^0) de (9) y (3). El conjunto de mínimos $X^0 \subseteq \bar{X}$ puede estar constituido por una variedad contenida en \bar{X} , que puede ser uno o varios puntos, un conjunto numerable de puntos, una línea, una superficie, etc.

Al utilizar un algoritmo iterativo, éste produce una aproximación (s^*, X^*), con la precisión deseada $\eta > 0$, tal que:

$$s^0 \leq s^* \leq s^0 + \eta; \quad X_0 \subseteq X^*; \quad (11)$$

$$s^0 \leq f(x) \leq s^0 + \eta; \quad \forall x \in X^*.$$

Además, se tiene convergencia:

$$\eta_m \rightarrow 0; \quad s_m^* \rightarrow s^0; \quad X^0 = \bigcap_m X_m^*, \quad (12)$$

si $m \rightarrow \infty$, ver Ref. [1], págs. 13-15. Así pues, la salida del pro-

grama está constituida por el número s^* y el conjunto X^* quasi-rectangular (unión de rectángulos cerrados cuyos interiores son disjuntos), que es presentado como una gráfica (si $n = 2$) o una tala de puntos.

EL ALGORITMO

El *Cubic Algorithm* de la Ref. [1] (págs. 7-24) es un algoritmo iterativo muy simple. Para el cubo unidad $\bar{U} \subset R^n$ de (8) elegimos el origen como punto representativo el cubo o nodo (*grid point*). En estas condiciones el algoritmo cúbico (CA) puede ser descrito de la forma siguiente:

Primer Paso: Elegido un número natural $N \geq 2$ (que denominamos *razón de subdivisión*), se realiza una partición de \bar{U} en N^n subcubos del mismo tamaño, \bar{C}_i , tales que: $\cup \bar{C}_i = \bar{U}$, $C_i \cap C_j = \emptyset \quad \forall i \neq j$ (entendiendo que $C_i = \text{interior } \bar{C}_i$). Los lados de cada cubo \bar{C}_i son paralelos a los ejes, la longitud de cada lado es $1/N$ y el diámetro de cada cubo es \sqrt{n}/N .

Una vez elegido un valor A_g (una cota de la pendiente de $g(z)$ de (8), ver Tabla 1), se define la constante de eliminación de subcubos \bar{C}_i , que denominamos ahora \bar{C}_i^1 , como:

$$r_1 = A_g \frac{\sqrt{n}}{N} \quad (13)$$

Al mover el subcubo $\bar{C}_{i_0}^1$ de nodo (*grid point*) $z_{i_0}^1 = 0$ en el origen hasta coincidir con cada \bar{C}_i^1 uno por uno, se define un nodo $z_i^1 \in \bar{C}_i^1$ para cada \bar{C}_i^1 (este proceso es llamado *translated grid generator* en la Ref. [1], pág. 9), de manera que para actuar sobre el conjunto de subcubos basta actuar sobre el conjunto de los nodos elegidos. Así pues, seleccionados los N^n nodos z_i^1 , se calcula el valor de la función sobre ellos, es decir, todos los valores $g(z_i^1)$, y se determina el valor mínimo:

$$s_1 = \min g(z_i^1), \quad 1 \leq i \leq N^n. \quad (14)$$

Queda claro que ese valor s_1 proporciona una primera aproximación del valor mínimo global s^0 de (9).

Al considerar (4) para la función $g(z)$ de (8), se concluye que la variación de $g(z)$ sobre cada \bar{C}_i^1 está acotada por r_1 :

$$\begin{aligned} \text{Var}_{z \in \bar{C}_i^1} g(z) &= \max_{z, z' \in \bar{C}_i^1} |g(z) - g(z')| \leq \\ &\leq A_g \max_{z, z' \in \bar{C}_i^1} \|z - z'\| = A_g \frac{\sqrt{n}}{N} = r_1. \end{aligned} \quad (15)$$

Esto significa que cada \bar{C}_i^1 , para el cual:

$$g(z_i^1) - s_1 > r_1, \quad (16)$$

donde z_i^1 es el nodo de \bar{C}_i^1 , no contiene un mínimo global y podría ser eliminado de la lista de subcubos donde se buscan los mínimos.

Una vez eliminados todos los subcubos que verifican (16), se obtiene un conjunto cuasi-cúbico (no necesariamente conexo) compuesto con los restantes subcubos cerrados:

$$\bar{K}_1 = \{z \in \bar{C}_i^1 : g(z_i^1) - s_1 \leq r_1, 1 \leq i \leq N^n\}, \quad (17)$$

el cual es una primera aproximación del conjunto U^0 de (10) y, su imagen mediante la transformación afin inversa es una primera aproximación del conjunto X^0 de (3).

Pasos Sigüientes. Se realiza la partición de cada $\bar{C}_i^1 \subset \bar{K}_i$ de la misma forma que se hizo con U , se obtiene otra familia de subcubos \bar{C}_i^2 , y se elige el nuevo conjunto de nodos $\{z_i^2\}$ de la misma forma que se mostró en el paso anterior.

Se define la segunda constante de eliminación: $r_2 = r_1/N$, y se repite el proceso descrito en el Primer Paso pero reemplazando z_i^1, r_1, s_1 por z_i^2, r_2, s_2 .

Este proceso se reitera las veces necesarias hasta que es suficientemente pequeña la constante de eliminación correspondiente $r_m = r_{m-1}/N = A_g \sqrt{n}/N^m$ y concluye la iteración.

Si la última iteración es m , entonces la precisión lograda al aproximar el conjunto de mínimos con \bar{K}_m es:

$$\eta_m = 3r_m = 3A_g \sqrt{n}/N^m = 3A_g \sqrt{n} \varepsilon_m, \quad (18)$$

donde $\varepsilon_m = 1/N^m$ es la longitud del lado de cada subcubo contenido en \bar{K}_m .

Recíprocamente, si se quiere encontrar una aproximación del valor mínimo global s^0 , dado un valor η , entonces el proceso reiterativo pararía en la iteración m tal que:

$$m = 1 + \left[\lg_N(3A_g \sqrt{n}/\eta) \right] = 1 + \left[\frac{\ln(3A_g \sqrt{n}/\eta)}{\ln N} \right], \quad (19)$$

donde $[x]$ denota la parte entera de x .

El algoritmo proporciona al valor $s^* = s_m$ y el conjunto $U^* = \bar{K}_m$, el cual está representado por el conjunto de nodos $z_j^m \in \bar{K}_m$. Esos puntos z_j^m se transforman en puntos x_j^m de la caja inicial, usando (7), que representan al conjunto X^* de (11) constituido por pequeños rectángulos cuyos nodos correspondientes son los puntos $\{x_j^m\}$:

$$X^* = \bigcup_j X_j^*, \quad X_j^* = \left\{ x = (x_1, \dots, x_n) : x_{ji}^m \leq x_i \leq x_{ji}^m + \frac{b_i - a_i}{N^m} \right\}. \quad (20)$$

La regla de parada del algoritmo está establecida por el Lema 3 de la Ref. [3] (págs. 949-950), que dice:

$$0 \leq s_m - s^0 \leq r_m, \quad -r_m \leq g(z) - s_m \leq 2r_m, \\ g(z) - s^0 \leq 3r_m, \quad \forall z \in \bar{K}_m.$$

Observación. El valor de la constante A_g elegida en (13) es de importancia crítica. Si es conocida la mínima constante de Lipschitz L_g para la función $g(z)$ sobre \bar{U} , y en (13) tenemos $A_g \geq L_g$, entonces el algoritmo no elimina mínimos y se resuelve el problema con todas las garantías. Pero, si $A_g < L_g$ (lo cual puede suceder dado que L_g puede ser desconocida), entonces el algoritmo podría eliminar algunos mínimos, e incluso todos (ver Ref. [1], págs. 69-74). Además, valores grandes de A_g implican convergencia lenta e incremento del tiempo de cálculo.

El algoritmo minimiza a $g(z)$ de (8), y no a $f(x)$ de (1). Denotando $g(z) = \partial g / \partial z_i$, $f(x) = \partial f / \partial x_i$, $c_i = b_i - a_i$, de (8) se obtiene:

$$\|g(z)\| = \left(\sum_1^n g_i^2 \right)^{1/2} = \left(\sum_1^n c_i^2 f_i^2 \right)^{1/2}, \quad (21)$$

de donde, para un valor L dado en (5) se tiene que:

$$L \min_i c_i \leq \min_{z \in \bar{U}} \|\nabla g(z)\| \leq L \max_i c_i. \quad (22)$$

Así pues, para un valor adecuado A de la Tabla 1, el correspondiente valor A_g de (13) podría ser elegido teniendo en cuenta la siguiente desigualdad:

$$A \min_i c_i \leq A \leq A \max_i c_i, \quad c_i = b_i - a_i. \quad (23)$$

Esta incertidumbre es inherente al algoritmo en los casos de valor desconocido o valor muy elevado de la constante de Lipschitz $\min \|\nabla g(z)\|$ para $z \in \bar{U}$.

UN EJEMPLO

El problema a resolver en dos variables es:

$$\min f(x_1, x_2) = \left| \sin \left(0,5 + 9,5 \sqrt{x_1^2 + x_2^2} \right) \right|; \quad x_1, x_2 \in [0,2]. \quad (24)$$

Este problema tiene el valor mínimo global $s^0 = 0$ y un conjunto continuo de mínimos globales. En la Figura 1 se presenta una aproximación gráfica de tal conjunto solución y en la Figura 2 otra aproximación de la gráfica de la función para poder hacer comparaciones. Al cambiar la precisión $\eta > 0$, puede observarse cómo cambian las respuestas gráficas del conjunto solución convirtiéndose en curvas cuando $\eta \rightarrow 0$.

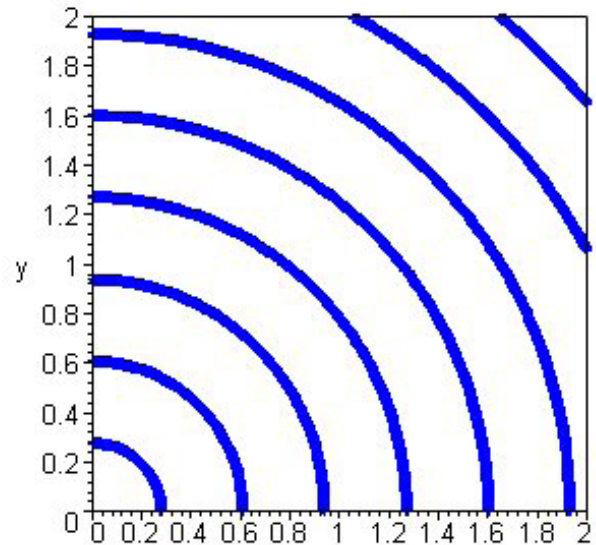


Figura 1. Solución gráfica del algoritmo.

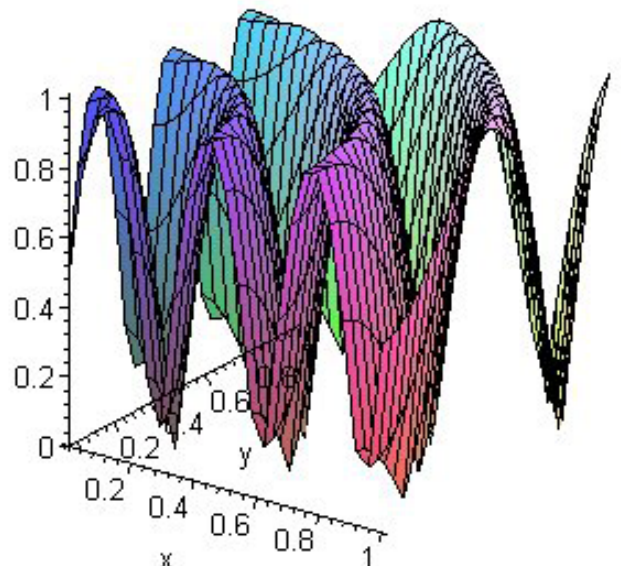


Figura 2. Gráfica de la función (24).

DOS APLICACIONES

Aplicación 1: Encontrar todas las raíces reales de un polinomio de grado k con coeficientes reales:

$$P_k(x) = x^k + a_1 x^{k-1} + \dots + a_k, \quad (25)$$

se reduce a resolver el problema:

$$\min |P_k(x)|, \quad (26)$$

donde: $-1 - \max |a_j| \leq x \leq 1 + \max |a_j|$.

En el caso de que se desee encontrar todas las raíces, reales y complejas, entonces se considera la sustitución: $x = u + iv$, $i^2 = -1$, se determina la nueva representación del polinomio: $P_k(u + iv) = p(u, v) + iq(u, v)$ y se resuelve el problema en R^2 :

$$\min [|p(u, v)| + |q(u, v)|], \quad |u|, |v| \leq M. \quad (27)$$

donde:

$$u, v \in [-1 - \max |a_j|, 1 + \max |a_j|]. \quad (28)$$

```
"Aproximation of all roots", 80 x5 - 148 x4 - 260 x3 - 55 x2 - 340 x + 93 = 0,
" in the rectangle ", [[0, 1], [-1, 1]]
      Number_of_Iterations = 14
      Edge_of_smaller_subcube = 0.00006103515625
      Number_of_subcubes_remaining = 778
      Number_of_grid_points_epsilon - separated = 15
      The_global_min_value = fpX([0, 0])
[[0., -1.], [0.001098632812, -0.9997558594], [0.2470703125, -0.0002441406250],
 [0.2481689453, -0.001220703125], [0.2492675781, -0.002441406250],
 [0.2492675781, -0.0002441406250], [0.2503662109, -0.002685546875],
 [0.2503662109, -0.0004882812500], [0.2514648438, -0.001708984375],
 [0.2525634766, -0.0004882812500], [0.2487792969, 0.001953125000],
 [0.2498779297, 0.001953125000], [0., 0.9990234375], [0.001098632812, 0.9992675781],
 [0.2509765625, 0.001708984375]]
```

Figura 3. Aproximación de las raíces reales de un polinomio.

Aplicación 2: Encontrar las soluciones de un sistema de ecuaciones de varias variables reales:

$$\begin{cases} h_1(x_1, \dots, x_n) = 0 \\ \dots \\ h_s(x_1, \dots, x_n) = 0 \end{cases} \quad (29)$$

en la n -caja: $\bar{X} = \{x \in R^n : a_i \leq x_i \leq b_i, \quad i=1, \dots, n\}$, (30)

se reduce a resolver:

$$\min \|h(x)\|_1 = \min (|h_1(x_1, \dots, x_n)| + \dots + |h_s(x_1, \dots, x_n)|), \quad (31)$$

donde $h(x) = [h_1(x), \dots, h_s(x)]$.

with Box Constraints and Applications: A Maple code. Mathematical and Computer Modelling, Vol. **38**, 77-97 (2003).

- Galperin, E.A.: *Global optimization in problems with uncertainties*. Journal of Nonlinear Analysis, Vol. **47**, 941-952 (2001).

Miguel Delgado Pineda

Dpto. de Matemáticas Fundamentales

REFERENCIAS

- Galperin, E.A.: *The Cubic Algorithm for Optimization and Control*. NP Research Publ., Montreal, 1990.
- Delgado Pineda, M. and Galperin, E.A.: *Global Optimization in R^n*

```
"Aproximation of the solution ", [x12 + x22 - x3, x1 + x2 + x3 - 1] = 0, " in the box ",
[[ -1, 1], [-1, 1], [-1, 1]]
      Number_of_Iterations = 7
      Edge_of_smaller_subcube = 0.007812500000
      Number_of_subcubes_remaining = 4536
      Number_of_grid_points_epsilon - separated = 32
      The_global_min_value = [-0.3374023438, 0.8906250000]
```

Figura 4. Resolución aproximada de un sistema.