Tarea 3 - Optimización no Lineal

Camara Medina Cynthia Lilian
Reyes Zamora Ollin
Alanis González Edzon Omar
Mendoza Urrusquieta Jair Natael

20 de abril de 2023

Índice general

1. Teoría **2**
1.1. División de intervalos por la mitad 2
1.2. Método de la sección dorada 5
1.3. Comparación teórica 7

2. Programación **9**
2.1. Comparación numérica de algunos métodos de acotamiento del optimo . . . 9
2.1.1. Ejercicio 1 9
2.1.2. Ejercicio 2 13
2.1.3. Ejercicio 3 14
2.1.4. Conclusiones 15

3. Bibliografía **16**

1. Teoría

1.1. División de intervalos por la mitad

Hacer cinco iteraciones a mano 1 del método de división de intervalos por la mitad, visto en clase, para acotar el mínimo del siguiente problema: Minimizar

$$f(x) = 4x^2 + \frac{1}{x^4}$$

dentro del intervalo $[0.7, 2.5]$. Reportar los cálculos de cada etapa y los resultados numéricos $(a, b, L, x_m, f(x_m), etc)$. Indique cuál sería una aproximación del óptimo y estime la precisión.

Solución. $L = b - a = 2.5 - 0.7 = 1.8$

$x_m = \dfrac{2.5 + 0.7}{2} = 1.6$

• $ITERACION$ 1

$x_1 = a + \dfrac{L}{4} = 1.15 \qquad x_2 = b - \dfrac{L}{4} = 2.05$

$f(x_1) = f(1.15) = 5.8617$

$f(x_2) = f(2.05) = 16.866$

$f(x_m) = f(1.6) = 10.392$

$f(x_1) < f(x_m) \Rightarrow b = x_m = 1.6 \qquad x_m = x_1 = 1.15 \qquad a = a = 0.7$

$L = 1.6 - 0.7 = 0.9$

- $ITERACION$ 2

$x_1 = a + \dfrac{L}{4} = 0.925 \qquad x_2 = b - \dfrac{L}{4} = 1.375$

$f(x_1) = f(0.925) = 4.7884$

$f(x_2) = f(1.375) = 7.8422$

$f(x_m) = f(1.15) = 5.8617$

$f(x_1) < f(x_m) \Rightarrow b = x_m = 1.15 \qquad x_m = x_1 = 0.925 \qquad a = a = 0.7$

$L = b - a = 0.45$

- $ITERACION$ 3

$x_1 = a + \dfrac{L}{4} = 0.8125 \qquad x_2 = b - \dfrac{L}{4} = 1.0375$

$f(x_1) = f(0.8125) = 4.9352$

$f(x_2) = f(1.375) = 5.1686$

$f(x_m) = f(0.925) = 4.7884$

$f(x_m) < f(x_1)\ y\ f(x_m) < f(x_2) \Rightarrow a = x_1 = 0.8125 \qquad b = x_2 = 1.0375$

$L = b - a = 0.225 \qquad x_m = 0.925$

- $ITERACION$ 4

$x_1 = a + \dfrac{L}{4} = 0.86875 \qquad x_2 = b - \dfrac{L}{4} = 0.98125$

$f(x_1) = f(0.8675) = 4.7744$

$f(x_2) = f(0.98125) = 4.93$

$f(x_m) = f(0.925) = 4.7884$

$f(x_1) < f(x_m) \Rightarrow b = x_m = 0.925 \qquad x_m = x_1 = 0.86875 \qquad a = a = 0.8125$

$L = b - a = 0.1125$

- $ITERACION$ 5

$x_1 = a + \dfrac{L}{4} = 0.840625 \qquad x_2 = b - \dfrac{L}{4} = 0.896875$

$f(x_1) = f(0.840625) = 4.8291$

$f(x_2) = f(0.896875) = 4.76305$

$f(x_m) = f(0.86875) = 4.7744$

$f(x_2) < f(x_m) \Rightarrow a = x_m = 0.86875 \qquad x_m = x_2 = 0.896875 \qquad b = 0.925$

$L = b - a = 0.05625$

$$\therefore \; x_m = 0.896875$$

$\blacksquare$

1.2. Método de la sección dorada

Hacer 5 iteraciones a mano aplicando el método de la sección dorada al problema de la sección anterior. Escriba el procedimiento y los resultados. Compare los resultados de ambos métodos en cuanto a precisión del intervalo y número de evaluaciones de la función objetivo.

Solución. Minimizar $f(x) = 4x^2 + \dfrac{1}{x^4} \qquad [0.7, 2.5]$

$a = 0.7 \qquad b = 2.5 \qquad L = b - a = 1.8$

$$\tau = \frac{-1+\sqrt{5}}{2}$$

- $ITERACION$ 1

$x_1 = b - \tau L = 1.3875 \qquad a + \tau L = 1.8124$

$f(x_1) = f(1.3875) = 7.9704$
$f(x_2) = f(1.8124) = 13.2318$

$$f(x_1) < f(x_2) \Rightarrow \begin{cases} b = x_2 = 1.8124 \\ x_2 = x_1 = 1.3875 \\ a = 0.7 \\ L = b - a = 1.1124 \\ x_1 = b - \tau L = 1.1248 \end{cases}$$

- $ITERACION$ 2

$f(x_1) = f(1.1248) = 5.6854$
$f(x_2) = f(1.3875) = 7.9704$

$$f(x_1) < f(x_2) \Rightarrow \begin{cases} b = x_2 = 1.3875 \\ x_2 = x_1 = 1.1248 \\ a = 0.7 \\ L = b - a = 0.6875 \\ x_1 = b - \tau L = 0.9626 \end{cases}$$

- $ITERACION$ 3

$f(x_1) = f(0.9626) = 4.8711$
$f(x_2) = f(1.1248) = 5.6854$

$$f(x_1) < f(x_2) \Rightarrow \begin{cases} b = x_2 = 1.1248 \\ x_2 = x_1 = 0.9626 \\ a = 0.7 \\ L = b - a = 0.4248 \\ x_1 = b - \tau L = 0.8622 \end{cases}$$

- $ITERACION\ 4$

$f(x_1) = f(0.8622) = 4.7830$
$f(x_2) = f(0.9626) = 4.8711$

$$f(x_1) < f(x_2) \Rightarrow \begin{cases} b = x_2 = 0.9626 \\ x_2 = x_1 = 0.8622 \\ a = 0.7 \\ L = b - a = 0.2626 \\ x_1 = b - \tau L = 0.8003 \end{cases}$$

- $ITERACION\ 5$

$f(x_1) = f(0.8003) = 4.9996$
$f(x_2) = f(0.8622) = 4.7830$

$$f(x_1) > f(x_2) \Rightarrow \begin{cases} a = x_1 = 0.8003 \\ x_1 = x_2 = 0.8622 \\ b = b = 0.9626 \\ L = b - a = 0.1623 \\ x_2 = a + \tau L = 0.9006 \end{cases}$$

$$\therefore x^* \in [0.8622, 0.9006]$$

$\blacksquare$

1.3. Comparación teórica

Para los métodos división de intervalos por la mitad y sección dorada, deducir las fórmulas para determinar el número de iteraciones n que cada algoritmo requiere para acotar el óptimo en un intervalo de longitud menor o igual que ε. Considere como L a la longitud inicial del intervalo. Escriba claramente su razonamiento para obtener cada una de las fórmulas.

Método de división de intervalos por la mitad

A cada etapa del algoritmo, se borra exactamente la mitad de la longitud del intervalo de búsqueda. El punto medio de los intervalos subsecuentes es siempre igual a uno de los puntos previos $x_1, x_2 o x_m$. Por tanto, se requieren, cuando mucho, dos evaluaciones de la función a cada paso subsecuente.

Después de k evaluaciones de la función, el intervalo inicial de búsqueda será reducido a $\left(\frac{1}{2}\right)^{\frac{k}{2}} L$.

En consecuencia, el número de evaluaciones requeridas para obtener una precisión deseada ε puede calcularse resolviendo:

$$\left(\frac{1}{2}\right)^{\frac{k}{2}} L = (0.5)^{\frac{k}{2}}(b-a) \leq \varepsilon$$

Despejando k se tiene

$$k \leq \frac{2\log\left(\frac{b-a}{\varepsilon}\right)}{\log(2)}$$

con $\varepsilon > 0$

Por lo mencionado anteriormente, tenemos que:

$$n = 2k \leq \frac{4\log\left(\frac{b-a}{\varepsilon}\right)}{\log(2)}$$

Donde n corresponde al número de iteraciones necesarias para acotar el óptimo en un intervalo de longitud menor o igual que ε.

Método de la sección dorada

En este método, el intervalo se reduce a τ^{n-1}, con $\tau = 0.618$, después de k evaluaciones de la función objetivo. De tal forma, el número de evaluaciones de la función objetivo que se requieren para lograr una precisión deseada ε se calcula resolviendo (para k) la siguiente ecuación:

$$(0.618)^{k-1}L = (0.618)^{k-1}(b-a) \leq \varepsilon$$

Despejando k obtenemos:

$$k \leq \log_{\tau}\left(\frac{\tau\varepsilon}{b-a}\right)$$

con $\varepsilon > 0$

Como solo se requiere una evaluación de la función objetivo por iteración podemos deducir que

$$n = k \leq \log_{\tau}\left(\frac{\tau\varepsilon}{b-a}\right)$$

Donde n es el número de iteraciones necesarias para acotar el óptimo en un intervalo de longitud menor o igual que ε.

2. Programación

2.1. Comparación numérica de algunos métodos de acotamiento del optimo

2.1.1.

Programar en `Matlab` el método de búsqueda exhaustiva, el método de división de intervalos por la mitad y el método de la sección dorada, que vimos en clase, para acotar el óptimo del siguiente problema (sobre el intervalo [3, 5]):

Minimizar

$$f(x) = \frac{\sin x}{1 + x^2}$$

El intervalo inicial, y la precisión del intervalo final deben darse como parámetros de entrada. El intervalo final será el parámetro de salida. Debe incluirse una copia del código fuente (comentado) en el reporte de la tarea.

```
A = randi([0,100],[10,10]);
fprintf("Matriz A\n");
disp(A);

B = randi([0,100],[10,10]);
fprintf("Matriz B\n");
disp(B);

C = A + B;

fprintf("Matriz C = A + B\n");
disp(C);
```

Codigo Fuente

Búsqueda Exhaustiva

```
format long
clc;
clear;
global fcalls

%Ingresar los limites y el numero de iteraciones
a = input('Limite inferior: ');
b = input('Limite superior: ');
e = input('Tolerancia: ');


%Calcular la longitud del intervalo y el numero de iteraciones
L = abs(b-a);
n = 2*(L/e);

%Definir
Δx=(b-a)/n;
x1=a;
x2=x1+Δx;
x3=x2+Δx;
contador=1;
fcalls = 0;
l1 = L;

%Ciclo para iterar mientras el numero de veces solicitadas
while (l1 > e)
    if (f(x1)≥f(x2))&&(f(x2)≤f(x3))
        fprintf('El minimo se encuentra en el intervalo ...
            (%4.6f,%4.6f)\n',x1,x3);
        %Si se cumple esta condicion, no se ejecuta el resto del while
        break;
    else
        x1=x2;
        x2=x3;
        x3=x3+Δx;
            if x3>b
                fprintf('El optimo es %4.2f o %4.2f\n',a,b);
            %Si se cumple esta condicion, no se ejecuta el resto ...
                del while
            break;
            end
    end
    contador=contador+1;
    l1 = abs(x3-x1);
end

fprintf('Numero de iteraciones: %d\n',contador);
```

```
valor_optimo=(x1+x3)/2;
evaluacion=f(valor_optimo);

fprintf(['Valor optimo: %4.6f \n ' ...
    'Funcion evaluada en el valor optimo: %4.6f'], valor_optimo, ...
        evaluacion);

fprintf('Numero de llamadas a la funcion: %d\n',fcalls);

function y = f(x)
    global fcalls
    y = sin(x)/(1+x^2);
    fcalls = fcalls + 1;
end
```

División de Intervalos

```
format long
clc;
clear;
global fcalls

%Ingresar los limites y el numero de iteraciones
a = input('Limite inferior: ');
b = input('Limite superior: ');
e = input('Tolerancia: ');

%Calcular la longitud del intervalo y el numero de iteraciones
L = abs(b - a);
xm = (a+b)/2;

fcalls = 0;
iter = 1;

%Ciclo para repetir mientras no se alcance la precision
while L>e
    x1 = a + L/4;
    x2 = b - L/4;
    if f(x1) < f(xm)
        b = xm;
        xm = x1;
    elseif f(x2) < f(xm)
        a = xm;
        xm = x2;
    else
        a = x1;
        b = x2;
    end
    %Nueva distancia
    L = abs(b - a);
```

```
        fprintf('Iteracion %d:  Intervalo: (%f, %f)  Valor:  ...
            %f\n',iter,a,b,xm);
        disp(L);
            iter = iter +1;
    end

    valor_optimo=(a+b)/2;
    evaluacion=f(valor_optimo);

    fprintf(['Valor optimo: %4.7f \n ' ...
        'Funcion evaluada en el valor optimo: %4.8f\n'], valor_optimo, ...
           evaluacion);
    fprintf('Numero de llamadas a la funcion: %d\n',fcalls);


    function y = f(x)
        global fcalls
        y = sin(x)/(1+x^2);
        fcalls = fcalls + 1;
    end
```

Sección Dorada

```
    clc;
    clear;
    global fcalls

    %Ingresar los limites y el numero de iteraciones
    a = input('Limite inferior: ');
    b = input('Limite superior: ');
    e = input('Tolerancia: ');

    %Definir
    t = (-1+sqrt(5))/2;
    L = b - a;
    x1 = b - t*L;
    x2 = a + t*L;
    n = 1;
    fcalls = 0;
    fprintf('\n%d: [%d,%d], L = %d, x1 = %d, x2 = %d\nf(x1) = %d, f(x2) ...
       = %d\n',n,a,b,L,x1,x2,f(x1),f(x2))

    %Ciclo para repetir mientras no se alcance la precision
    while L>e
        if f(x1) < f(x2)
            b = x2;
            x2 = x1;
            L = b - a;
            x1 = b - t*L;
        else
```

```
            a = x1;
            x1 = x2;
            L = b - a;
            x2 = a + t*L;
        end
        n = n + 1;
        fprintf('\nIteracion %d: [%d,%d], L = %d, x1 = %d, x2 = ...
            %d\nf(x1) = %d, f(x2) = %d\n',n,a,b,L,x1,x2,f(x1),f(x2))
    end

    valor_optimo=(a+b)/2;
    evaluacion=f(valor_optimo);

    fprintf(['Valor optimo: %4.6f \n ' ...
        'Funcion evaluada en el valor optimo: %4.6f'], valor_optimo, ...
           evaluacion);

    fprintf('\nLlamadas a la funcion: %d\n',fcalls)

    function y = f(x)
        global fcalls
        y = sin(x)/(1+x^2);
        fcalls = fcalls + 1;
    end
```

2.1.2.

Ejecutar sus programas del inciso anterior, y reportar los resultados con una precisión de 0.05. El reporte debe incluir la salida de cada programa, o una parte de ella que muestre que el programa hace las iteraciones correctamente (p.ej. los valores de L o cómo se va reduciendo el intervalo). En particular, debe reportarse el valor óptimo encontrado $(x^* y f(x^*))$ como el punto medio del intervalo final.

```
Matriz A
    63    18    22    63    27    99    13    95    41    35
    35    73    37     2    25     6    21    68    60    98
   100    37     8    91    45    94    18    99    75    34
    22    84    64    80    22     1     4    77    58    89
    65    74    18    75    81    69    10    34    55    45
    61    57     4    82    99    79    62    66    58    41
    39    17    73    38     3    53    94    24    51    21
    14    96    35    62    54    89    35    29     8    12
     2    26    66    58     8    90    41    68    72    31
    42    93    38    53    81    63    99    53   100    73

Matriz B
    79    22    10    31    50    72    84    49    28    72
    70    27    18    18    43    62    32    70    23    86
     0    67    10    34   100    34    55    98    71    28
    85    48    49    21    81    94    98    33    63    73
    93    62    19    51    49    12    55    84    59    13
    77    23    90    91    90    73    33    74    66    84
     4    17    10    63    13    65    62    96     4    13
    38    83     4    10    39    84    36     3    35    59
    71    77    56    39    93    40    76    36    45    36
    73    94    78     5    92    75    41    66    24    81
```

```
Matriz C = A + B
   142    40    32    94    77   171    97   144    69   107
   105   100    55    20    68    68    53   138    83   184
   100   104    18   125   145   128    73   197   146    62
   107   132   113   101   103    95   102   110   121   162
   158   136    37   126   130    81    65   118   114    58
   138    80    94   173   189   152    95   140   124   125
    43    34    83   101    16   118   156   120    55    34
    52   179    39    72    93   173    71    32    43    71
    73   103   122    97   101   130   117   104   117    67
   115   187   116    58   173   138   140   119   124   154
```

2.1.3.

Ejecutar los programas anteriores para los siguientes valores de intervalos: [3,5], [3.5,5.5], [4,6], [-8,-6], [-8.5,-6.5], [-7,-9],[-2,0],[-1.5,0.5],[-1.3,0.7],[1,6]. Hacer una tabla, con la cantidad de veces que se calculó (i) la función objetivo, y (ii) el número de iteraciones en cada caso (i.e. (a) búsqueda exhaustiva, (b) div. de intervalos por la mitad, y (c) sección dorada) usando cada intervalo. Considere un tamaño de 10^{-12} para el intervalo final. Presentar en su reporte dicha tabla con los valores promedio y la desviación estándar sobre los 10 valores (de número de iteraciones y llamadas a la función objetivo) en cada caso. Presentar en su reporte dicha tabla con los valores promedio y la desviación estándar sobre los 10 valores en cada caso. Ejemplo: escribir en X el número de evaluaciones de f para cada caso:

```
x0 = 7;
h = 0.01;

f = @(x) 3*(x^3)+4*(x^2)+2*x;

d1 = (f(x0+h)-f(x0-h))/(2*h);
fprintf("Valor de primer derivada: ")
disp(d1)


d2 = (f(x0+2*h)-2*f(x0)+f(x0-2*h))/(4*(h^2));
fprintf("Valor de segunda derivada: ")
disp(d2)
```

```
Valor de primer derivada:   499.0003

Valor de segunda derivada:   134.0000
```

Precisión:	1.00E-12				
Busqueda Exhaustiva					
	Intervalo obtenido	Iteraciones	x*	f(x*)	Cálculos de la función objetivo
[3, 5]	[3.0006264 , 3.0006264]	125293834	3.0006264	0.014044	501175335
[3.5, 5.5]	-	-	-	-	-
[4, 6]	-	-	-	-	-
[-8, -6]	-	-	-	-	-
[-8.5, -6.5]	-	-	-	-	-
[7, 9]	-	-	-	-	-
[-2, 0]	-	-	-	-	-
[-1.5, 0.5]	-	-	-	-	-
[-1.3, 0.7]	-	-	-	-	-
[1, 6]	[1.000192 , 1.000192]	384,086,573	1.00019206	0.4207065	1,536,346,292
Promedio:		-			-
D. E. :		-			-

2.1.4.

Escriba sus conclusiones para toda esta sección. Verifique si los experimentos son consistentes con las fórmulas que obtuvo en la sección 1.3.

De las ejecuciones anteriores, vemos que el método de División de intervalos por la mitad es mejor, ya que obtenemos la misma precisión con menos iteraciones y menos evaluaciones de la función y, una vez más, queda descartado el método de búsqueda exhaustiva. Cabe mencionar que se empleó Arch como SO, lo que aceleró el proceso de iteraciones, sin embargo, por cada intervalo se tardó aproximadamente unos 40 minutos, resultando bastante ineficiente. Por otra parte, vemos que los resultados obtenidos en cuanto al número de iteraciones son congruentes con las fórmulas de la sección 1.3 En el método de búsqueda exhaustiva la eficacia de las fórmulas se había demostrado previamente. Para el método de división de intervalos, para los primeros 9 intervalos, todos con una longitud de 2, el valor resultante de n fue de 40.8631 mientras que para el último intervalo, el valor resultante fue de 42.1850. Aplicando la función parte entera (dado que no se pueden considerar iteraciones que no sean números enteros), los valores concuerdan respectivamente con 41 y 43. Para el método de Sección Dorada, en los primeros 9 intervalos, todos con una longitud de 2, el valor resultante de n fue de 59.8601 mientras que para el último intervalo, el valor resultante fue de 61.76. Aplicando la función parte entera (dado que no se pueden considerar iteraciones que no sean números enteros), los valores concuerdan respectivamente con 60 y 62.

3. Bibliografía

- n.d. UNIDAD Nº4 Métodos matemáticos de optimización no restringida Búsqueda unidimensional. [ebook] p.8. Available at: `<https://www.fio.unicen.edu.ar/usuario/cgely/q13-0/Apuntes/unidad4.pdf>` [Accessed 8 September 2022].
- Coello Coello, C., 2008. Optimización en Ingeniería Clase 2. [ebook] Ciudad de México. Available at: <http://delta.cs.cinvestav.mx/~ccoello/optimizacion/clase2-opt-2008.pdf> [Accessed 8 September 2022].
- Coello Coello, C., 2009. Optimización en Ingeniería Clase 3. [ebook] Ciudad de México. Available at: <http://delta.cs.cinvestav.mx/~ccoello/optimizacion/clase3-opt-2009.pdf> [Accessed 8 September 2022].