

Одномерная минимизация функции

Владимир Руцкий, 3057/2

24 марта 2009 г.

1 Постановка задачи

Требуется найти с наперёд заданной точностью минимум одномерной функции $f(x)$ на заданном отрезке $[a, b]$:

$$\min f(x), \quad x \in [a, b],$$

используя *метод золотого сечения*.

Дана конкретная функция $\sin \frac{1}{x}$, и задан отрезок $[0.15, 0.6]$.

2 Исследование применимости метода

Алгоритмы поиска минимума одномерной функции $f(x)$ на отрезке $[a, b]$, использующие только значения функции в некоторых точках (исследуемый метод золотого сечения в их числе), применимы, только если заданная функция является *унимодальной*, а именно

$$\exists! x^* \in [a, b] : \begin{cases} \forall x_1, x_2 \in [a, b] : & x^* \leq x_1 \leq x_2 \Rightarrow f(x^*) \leq f(x_1) \leq f(x_2) \\ \forall x_1, x_2 \in [a, b] : & x^* \geq x_1 \geq x_2 \Rightarrow f(x^*) \geq f(x_1) \geq f(x_2) \end{cases}.$$

3 Описание алгоритма

Общая идея поиска минимума унимодальной функции на отрезке заключается в том, что отрезок с минимумом можно разбить на части и гарантировано определить в какой части находится минимум, тем самым позволив сузить область для поиска минимума.

В случае метода золотого сечения отрезок, на котором имеется минимум, $[a, b]$ делится в отношении *золотой пропорции*. *Золотая пропорция* — это деление непрерывной величины на части в таком отношении, что большая часть относится к меньшей, как большая ко всей величине.

$x \in [a, b]$ делит $[a, b]$ золотым сечением, если

$$\begin{cases} x - a > b - x & \frac{x-a}{b-x} = \frac{b-a}{x-a} = \varphi \\ x - a < b - x & \frac{b-x}{x-a} = \frac{b-a}{b-x} = \varphi \end{cases},$$

т.е. для любого непустого отрезка найдутся две точки, делящие его в отношении золотой пропорции.

Величина отношения φ определена единственным образом, и равна $\frac{\sqrt{5}+1}{2}$.

Поиск минимума унимодальной функции методом золотого сечения можно описать алгоритмом 1:

Алгоритм 1 FindMinimumByGoldenSectionSearch. Поиск минимума унимодальной функции.

Вход: Исследуемая функция f ; отрезок $[a, b]$, содержащий минимум; точность ε , с которой требуется найти минимум.

Выход: Точка x^* — аргумент функции в которой достигается минимум с точностью ε .

```
{ Инициализируем первоначальное деление отрезка  $[a, b]$  на три отрезка золотым сечением. }
 $\alpha := \frac{1}{\varphi}$  { с данной величиной далее будет удобно работать }
 $i := 0$  { счетчик итераций }
{  $[a_i, b_i]$  — отрезок содержащий минимум на  $i$ -м шаге. }
 $a_i := a$ 
 $b_i := b$ 
{  $\lambda_i < \mu_i$  — точки делящие отрезок на  $i$ -м шаге золотым сечением. }
 $\lambda_i := a_i + (1 - \alpha)(b_i - a_i)$ 
 $\mu_i := a_i + \alpha(b_i - a_i)$ 
{ Подразбиваем отрезок пока его длина не станет меньше необходимой точности. }
while  $|a_i - b_i| \geq \varepsilon$  do
  if  $f(\lambda_i) \leq f(\mu_i)$  then
    { Минимум лежит на  $[a_i, \mu_i]$ , подразбиваем этот отрезок и продолжаем работу алгоритма. }
     $a_{i+1} := a_i$ 
     $b_{i+1} := \mu_i$ 
    { Из-за особенностей золотого сечения необходимо перевычислять лишь одну новую точку. }
     $\mu_{i+1} := \lambda_i$ 
     $\lambda_{i+1} := a_{i+1} + (1 - \alpha)(b_{i+1} - a_{i+1})$ 
  else
    { Минимум лежит на  $[\lambda_i, b_i]$ . }
     $a_{i+1} := \lambda_i$ 
     $b_{i+1} := b_i$ 
     $\lambda_{i+1} := \mu_i$ 
     $\mu_{i+1} := a_{i+1} + \alpha(b_{i+1} - a_{i+1})$ 
  end if
   $i := i + 1$ 
end while
{ Все точки на результирующем отрезке равны точке минимума с точностью  $\varepsilon$ , выбираем точку с меньшим значением функции из  $\lambda_i$  и  $\mu_i$ . }
if  $f(\lambda_i) \leq f(\mu_i)$  then
  return  $\lambda_i$ 
else
  return  $\mu_i$ 
end if
```

Приведённый алгоритм 1 может быть легко оптимизирован, если сохранять вычисленные значения функции f в точках α_i и μ_i , таким образом функция может быть вычислена значительно меньшее число раз. Для простоты эта оптимизация не приводится в алгоритме.

4 Код программы

Listing 1: Код программы решения задачи поиска минимума унимодальной функции

```
1 # golden_section_search.m
2 # Golden_section_search_algorithm.
3 # Vladimir Rutsky <altsysrq@gmail.com>
4 # 17.03.2009
5
6 precPows = [-3:-1:-6];
7 #precPows = [-3:-1:-3];
8 func = inline("sin(1./x)");
9 range = [0.15, 0.6];
10
11 function plotFunction( func, a, b, step )
```

```

12 | assert(a <= b);
13 |
14 | x = [a:step:b];
15 | y = func(x);
16 |
17 | plot(x, y, "-");
18 | endfunction
19 |
20 | function goldenSectionSearch( func, a, b, precision )
21 |     assert(a <= b);
22 |
23 |     figure();
24 |     hold on;
25 |
26 |     # Plotting function.
27 |     plotFunction(func, a, b, 1e-3);
28 |
29 |     # Searching minimum with golden section search (with visualization).
30 |     phi = (5^0.5 + 1) / 2;
31 |
32 |     originalRange = [a, b];
33 |
34 |     assert(abs(phi) > 1e-10);
35 |     x1 = b - (b - a) / phi;
36 |     x2 = a + (b - a) / phi;
37 |     y1 = func(x1);
38 |     y2 = func(x2);
39 |     while (abs(b - a) >= precision)
40 |         assert(a < x1);
41 |         assert(x1 < x2);
42 |         assert(x2 < b);
43 |         oldDist = b - a;
44 |
45 |         # Drawing.
46 |         yMax = max([func(a), func(x1), func(x2), func(b)]);
47 |         #yMax = max([func(x1), func(x2)]);
48 |         x = [a, x1, x2, b];
49 |         y = [yMax, yMax, yMax, yMax];
50 |         plot(x, y, "+k");
51 |
52 |         if (y1 <= y2)
53 |             b = x2;
54 |             x2 = x1;
55 |             x1 = b - (b - a) / phi;
56 |             y2 = y1;
57 |             y1 = func(x1);
58 |         else
59 |             a = x1;
60 |             x1 = x2;
61 |             x2 = a + (b - a) / phi;
62 |             y1 = y2;
63 |             y2 = func(x2);
64 |         endif
65 |
66 |         newDist = b - a;
67 |         assert(oldDist > newDist);
68 |     endwhile
69 |
70 |     if (y1 <= y2)
71 |         res = x1;
72 |     else
73 |         res = x2;
74 |     endif
75 |
76 |     resStr = sprintf("Minimum at x=%9.7f, with precision of %5.2e", res, precision)
77 |     title(resStr);
78 |
79 |     drawnow();
80 | endfunction
81 |
82 | # Iterating through precisions.

```

```
83 | prec = 10 .^ precPows;  
84 | for prec = prec  
85 |     goldenSectionSearch(func, range(1), range(2), prec)  
86 | endfor  
87 |  
88 | input("Press_enter_to_quit.");
```

5 Результаты решения

6 Возможные дополнительные исследования

7 Обоснование достоверности полученного результата