

Минимизация унимодальных функций методом оптисекции

Коднянко Владимир Александрович

доктор технических наук, профессор кафедры стандартизации, метрологии и управления качеством,
Сибирский федеральный университет. Россия, г. Красноярск. E-mail: kowlad@rambler.ru

Аннотация. Представлен названный оптисекцией новый метод минимизации унимодальных функций, основанный на оптимальном отсечении отрезков. Показано, что во всех случаях метод работает быстрее аналогичных классических методов своего семейства – метода бисекции и метода золотого сечения. При минимизации строго унимодальных функций оптисекция работает на 15–25 % быстрее метода золотого сечения и в 1,5–2 раза быстрее метода бисекции. Метод находит минимум монотонных функций за 5 обращений к ним. Функции-константы распознаются оптисекцией за 3 вычисления. При минимизации функций с плоским дном быстроедействие метода сопоставимо со скоростью распознавания монотонных функций.

Ключевые слова: метод бисекции, метод золотого сечения, метод оптисекции, унимодальная функция, быстроедействие метода.

Введение. Одномерной минимизации обычно подвергают унимодальные функции. Под унимодальной будем понимать функцию, отвечающую определению [4; 5], согласно которому на отрезке $G = [a, b]$ существуют делящие его на три сегмента числа α и β , $a \leq \alpha \leq \beta \leq b$, такие, что:

- 1) при $a < \alpha$ на отрезке $[a, \alpha]$ функция $f(x)$ монотонно убывает;
- 2) при $\beta < b$ на отрезке $[\beta, b]$ она монотонно возрастает;
- 3) на отрезке $[\alpha, \beta]$ функция постоянна и достигает своего минимума.

На практике чаще всего применяются численные методы семейства отсечения отрезков, суть которых состоит в последовательном отбрасывании на каждой итерации части отрезка неопределенности, способствуя тем самым его сжатию до тех пределов, пока не выполнится условие останова процесса оптимизации

$$b - a < \varepsilon,$$

где ε – малое положительное число, определяющее точность расчетов.

В семействе данных методов наибольшей известностью пользуются ставшие классическими *метод бисекции* (МБ) (другое название «метод дихотомии») и *метод золотого сечения* (МЗС) [4; 5; 7]. Качество таких методов определяется не только их надежностью, что свойственно обоим методам, но и быстроедействием, под которым понимают необходимое для решения задачи количество обращений к функции. Чем меньше таких обращений, тем быстрее работает метод.

На каждой итерации методу бисекции требуется два вычисления целевой функции, при этом отрезок неопределенности уменьшается в два раза. Второй метод только на первой итерации требует два ее вычисления, на остальных итерациях необходимо только одно. Хотя сокращаемый отрезок меньше, за счет этого быстроедействие метода золотого сечения оказывается выше.

Считается, что в данном семействе методов им нет конкурентов с точки зрения лучшего быстроедействия [7]. Действительно, метод золотого сечения не имеет ресурсов для улучшения. Метод бисекции можно было бы улучшить, если бы удалось довести количество обращений к функции до одного на каждой итерации или хотя бы уменьшить их. Однако, как считается, в рамках концепции метода такое улучшение невозможно. Тем не менее есть работы, в которых показано, что при определенных условиях количество обращений к функции можно уменьшить. Этот алгоритм назван методом экономной дихотомии [2; 3; 8].

В настоящей работе предложен новый метод отсечения отрезков, который на каждой итерации требует только одно вычисление функции. Метод быстро распознает монотонные функции и функции-константы, быстро работает при поиске экстремума функций с плоским дном ($\alpha < \beta$). Проведена оптимизация метода и выполнено сравнение методов с точки зрения их быстроедействия на примерах минимизации функций различного типа – монотонных, констант, функций с плоским дном и строго унимодальных функций.

1. Описание метода. В основу метода, который предназначен для нахождения абсциссы минимума унимодальной функции $f(x)$ на отрезке $G = [a, b]$ с точностью до e , положен принцип оптимального сечения отрезка в заданном отношении.

Для краткости по аналогии с бисекцией назовем метод оптисекцией.

В рамках метода сначала вычислим три точки $w[0]$, $w[1]$, $w[2]$ по формулам

$$w[i].x = a + (i+1)(b-a)/4, w[i].y = f(w[i].x), \quad i = 0, 1, 2. \quad (1)$$

Следующим шагом является проверка функции $f(x)$ на монотонность. Ниже приведен код логической функции *Mnt*, которая предназначена для такой проверки.

Логическая алгоритм-функция *Mnt*.

// Проверка функции на монотонность.

Входные данные:

a, b – границы отрезка, f – имя функции, e – точность расчетов, k – текущее количество обращений к функции,

$w[0]$, $w[1]$, $w[2]$ – три вычисленные точки.

Выходные данные:

x – решение, если функция монотонна, k – счетчик обращений к функции,

Result – значение функции *Mnt*: истина или ложь.

1. *Result* := false;

2. **if** ($w[0].Y < w[1].Y$) **and** ($w[1].Y < w[2].Y$) **then** $t[-1] := 1$ **else** $t[-1] := 0$;

3. **if** ($w[0].Y > w[1].Y$) **and** ($w[1].Y > w[2].Y$) **then** $t[1] := 1$ **else** $t[1] := 0$;

4. **if** $t[-1] = 1$ **then** // монотонное возрастание

begin $i := 0$; $g := a0$; $j := -1$; **end**

else

if $t[1] = 1$ **then** // монотонное убывание

begin $i := 1$; $g := b0$; $j := 1$; **end**

else exit; // монотонность не подтвердилась

5. **begin**

$z1.x := g$; $z1.y := f(g)$; $k := k+1$;

$w[1] := w[0]$; $w[1].y := w[2].y$; $w[2] := z1$;

end;

6. **if** $\text{Sign}(w[i].y - z1.y) = 1$ **then**

begin

$z2.x := g - j * e$; $z2.y := f(z2.x)$; $k := k+1$;

$w[1] := w[0]$; $w[1].y := w[2].y$; $w[2] := z2$;

if $\text{Sign}(z2.y - z1.y) = 1$ **then**

begin $x := g$; *Result* := true; **end**;

end;

Если ординаты трех упомянутых точек образуют монотонную последовательность – возрастающую или убывающую (блоки 2, 3), то алгоритм *Mnt* подвергает функцию $f(x)$ окончательной проверке на монотонность. Если такая последовательность возрастающая, то вычисляется значение $f(a)$. Если эта точка вписывается в гипотезу, то вычисляется значение $f(x+e)$. Если $f(x+e) \geq f(a)$, то в силу унимодальности функция $f(x)$ является монотонно неубывающей. В этом случае *Mnt* = true и решением задачи будет $x = a$. Аналогичны рассуждения для монотонно убывающей функции. Для нее решением будет $x = b$.

Таким образом, для установления факта монотонности функции $f(x)$ методу требуется пять обращений к ней.

Если функция не монотонна, то выполняется проверка условия

$$w[0].Y = w[1].Y = w[2].Y. \quad (2)$$

Если условие (2) выполнено, то в силу унимодальности функция $f(x)$ является константой, либо у нее сразу обнаружено плоское дно. В таких случаях решением будет абсцисса любой из этих точек. Если таковое не обнаружено при первых трех вычислениях функции, то не исключено, что плоское дно может быть обнаружено в процессе поиска на последующих итерациях.

Не прошедшие данные фильтры функции являются строго унимодальными с точкой текущего минимума и новыми границами отрезка

$$m = \underset{x}{\text{Min}}(w[i]), \quad i = 0, 1, 2, \dots, k-1,$$

$$a = w[0]x, b = w[2]x.$$

где k – количество обращений к функции.

Далее выполняется поиск, основанный на сечении отрезка в заданном отношении c .

Процедура *Ratio* поиска минимума функции $f(x)$ методом деления отрезка в заданном отношении

// Поиск минимума унимодальной функции

Входные данные:

a, b – границы отрезка, f – имя функции, e – точность расчетов, k – текущее количество обращений к функции, c – параметр деления отрезка в заданном отношении.

Выходные данные:

x – решение, если функция монотонна, k – счетчик обращений к функции,

repeat // цикл итераций

1. **if** $(m.x - a) > (b - m.x)$ **then** $p.x := c*a + (1.0 - c)*m.x$ **else** $p.x := c*b + (1.0 - c)*m.x$; // абсцисса новой точки
 $p.y := f(p.x)$; $k := k + 1$; // вычисление ординаты и обновление количества обращений к функции

2. **if** $m.y > p.y$ **then** // если найден новый минимум

begin

if $m.x < p.x$ **then** $a := m.x$ **else** $b := m.x$; // сжатие отрезка

$m := p$; $n := 1$;

end

else

begin

if $m.y = p.y$ **then** // найдена точка с повторившимся минимумом

begin

$n := n + 1$; // новое количество последних точек с одинаковыми ординатами

if $n = 3$ **then Goto** Fin; // если таких точки 3, то обнаружено плоское дно

end;

if $m.x < p.x$ **then** $b := p.x$ **else** $a := p.x$; // сжатие отрезка

end;

3. $e0 := \text{abs}(m.x) * e + 1.0e-10$;

until $\text{abs}(m.x - (a+b)/2.0) \leq 2.0 * e0 - 0.5 * (b-a)$; // конец цикла при выполнении условия сходимости

Fin: $x := m.x$;

Суть метода такова. Точка m текущего минимума функции делит отрезок $[a, b]$ на два сегмента $[a, m.x]$ и $[m.x, b]$. Из них выбирается тот, который длиннее, и он делится в отношении c , образуя на сегменте точку p вычислениями по одной из формул оператора 1. При $c = 0.5$ это соответствует бисекции, когда p будет расположена в середине наиболее длинного сегмента. При $c < 0.5$ она будет ближе к точке минимума, при $c > 0.5$ дальше от него. Представляется разумным располагать такую точку ближе к точке текущего минимума, то есть полагать $c < 0.5$. Однако слишком близко к ней располагать нельзя, ибо тогда процесс поиска решения может замедлиться. Это наводит на мысль, что существует оптимальное значение параметра c , которое для разных функций и отрезков неопределенности будет разным. Найти оптимальное значение параметра c можно на основании результатов анализа вычислительного эксперимента.

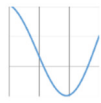
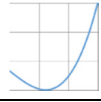
В качестве условия останова процесса использован оператор 3. Условие заимствовано из работы [6]. Ее автор предупреждает, что оно иногда может давать ошибку. В результате проведенных вычислительных экспериментов установлено, что такое иногда имеет место при минимизации монотонных функций. Однако рассматриваемый метод, как упомянуто выше, быстро находит решение для функций такого типа, поэтому при использовании данного условия останова ошибок он не должен давать.

2. Оптимизация метода по параметру c . Для определения оптимального значения параметра c были проведены вычислительные эксперименты на примерах минимизации 45-ти строго унимодальных функций без плоского дна. При этом для каждой функции варьировали значения границ a, b интервалов неопределенности так, что получилось около 1000 задач. Кроме того, эксперименты проводились для различных значений параметра точности e – от относительно больших, которые используются в большинстве практических расчетов, до весьма малых, которые обычно используются для исследования метода на надежность и вычислительную устойчивость.

Для проведения вычислений использовали среду Delphi, оперируя числами с 19–20 значащими цифрами [9]. Наиболее показательными оказались результаты, соответствующие весьма высокой точности расчетов $e = 10^{-10}$. Примеры расчетов приведены в таблице 1.

Таблица 1

Зависимости количества k обращений к функции $f(x)$ от параметра c

1	$f(x) = 1 + \ln(x) \cos[2(4 - x^2)], \quad G = [2.7, 3.1]$									
c	0.01	0.03	0.05	0.1	0.15	0.2	0.3	0.4	0.5	
k	116	71	45	40	38	35	39	47	55	
2	$f(x) = x^3 - 3x^2 - 5x + 8, \quad G = [1.0, 5.0]$									
c	0.01	0.03	0.05	0.1	0.15	0.2	0.3	0.4	0.5	
k	99	65	53	41	35	39	44	52	61	
3	$f(x) = 1.2 + \text{Max}[exp(-x), \cos(x), x^4, x^2], \quad G = [-0.6, 1.1]$									
c	0.01	0.03	0.05	0.1	0.15	0.2	0.3	0.4	0.5	
k	248	135	59	48	49	43	43	51	57	

Как видно из таблицы 1, при оптимизации функции 1 минимум количества k обращений к функции обнаружен при $c = 0.2$. Для второй функции оптимальным является $c = 0.15$. Для третьей функции оптимальны $c = 0.2$ и $c = 0.3$.

Аналогичные результаты с отклонением не более чем на 0.1 в ту или другую сторону от значения $c = 0.2$ имели место для большинства экспериментов. Каких-либо значительных отклонений от обнаруженных оптимальных значений не выявлено. Поэтому на практике можно использовать любое число из диапазона $c = [0.1, 0.3]$. Средним является $c = 0.2$, которое использовалось в качестве оптимального в дальнейших расчетах.

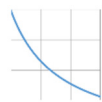
Отметим, что при относительно небольшой точности диапазон значений параметра несколько шире, но обнаруженное оптимальное значение остается прежним.

3. Сравнительный анализ быстродействия метода. Для сравнительного анализа быстродействия метода выполнены расчеты функций различных типов методами бисекции, золотого сечения и оптисекции.

В таблице 2 приведены данные для двух строго монотонных функций.

Таблица 2

Сравнительные данные о быстродействии методов при минимизации строго монотонных функций

Монотонная функция							
1	$f(x) = 29 + 16 / x, \quad G = [1.5, 4.5]$						
e	10^{-4}			10^{-9}			
Метод	МБ	МЗС	МОС	МБ	МЗС	МОС	
k	26	21	5	60	45	5	
2	$f(x) = 1.5 + \exp(x), \quad G = [1.5, 4.5]$						
e	10^{-4}			10^{-9}			
Метод	МБ	МЗС	МОС	МБ	МЗС	МОС	
k	34	26	5	66	49	5	

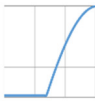
Из таблицы 2 видно, что при минимизации строго монотонных функций даже при относительно невысокой точности $e = 10^{-4}$ метод работает быстрее бисекции примерно в 5 раз, а золотого сечения в 4 раза. При высокой точности $e = 10^{-9}$ метод работает еще быстрее: в сравнении с бисекцией примерно в 12 раз и золотым сечением в 9 раз.

В таблице 3 приведены результаты минимизации трех функций с плоским дном.

Таблица 3

Сравнительные данные о быстродействии методов при минимизации функций с плоским дном

С ПЛОСКИМ ДНОМ							
1	$f(x) = 1, \quad G = [1.9, 3.1]$						
e	10^{-4}			10^{-9}			
Метод	МБ	МЗС	МОС	МБ	МЗС	МОС	
k	26	20	3	60	44	3	

2	$f(x) = 1.5 + \text{Max}[\cos(4 - x^2), 0.5]$, $G = [2.9, 3.2]$						
e	10^{-4}			10^{-9}			
Метод	МБ	МЗС	МОС	МБ	МЗС	МОС	
k	22	17	8	54	41	8	
3	$f(x) = 3 + \text{Max}[4\cos(x), -3]$, $G = [0.1, 4.9]$						
e	10^{-4}			10^{-9}			
Метод	МБ	МЗС	МОС	МБ	МЗС	МОС	
k	30	22	8	62	46	8	

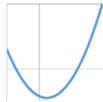
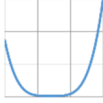
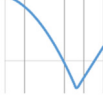
Как видно из таблицы 3, в сравнении с бисекцией и золотым сечением при минимизации функций с плоским дном метод демонстрирует существенно более высокое быстродействие. Для функции-константы в 7–20 раз, для функций с плоским дном в 3–7.

Приведенные примеры минимизации хотя и встречаются в практических задачах, и для них высокое быстродействие метода тоже важно, однако такие функции нетипичны: больший интерес представляют результаты минимизации строго унимодальных функций ($a < b$, $\alpha = \beta$). В таблице 4 приведены сравнительные результаты расчетов для таких функций.

Таблица 4

Сравнительные данные о быстродействии методов при минимизации строго унимодальных функций

строго выпуклая функция

1	$f(x) = 0.2 + (x - 1.5)^2 \quad G = [0.3, 3.2]$						
e	10^{-4}			10^{-9}			
Метод	МБ	МЗС	МОС	МБ	МЗС	МОС	
k	30	23	19	62	47	38	
2	$f(x) = 12 + 1000(x - 2.8)^4, \quad G = [1.5, 4.3]$						
e	10^{-4}			10^{-9}			
Метод	МБ	МЗС	МОС	МБ	МЗС	МОС	
k	28	22	17	60	46	24	
3	$f(x) = \text{Max}[x + 4, 6 \sin(x)], \quad G = [1.6, 1.9]$						
e	10^{-4}			10^{-9}			
Метод	МБ	МЗС	МОС	МБ	МЗС	МОС	
k	32	18	15	56	42	36	
4	$f(x) = 1.2 + \text{Max}[\exp(-x), \cos(x), x^4, x^2], \quad G = [-0.6, 1.1]$						
e	10^{-4}			10^{-9}			
Метод	МБ	МЗС	МОС	МБ	МЗС	МОС	
k	32	18	15	56	42	36	

Основной вывод, который вытекает из анализа представленных в таблице 4 данных, состоит в том, что во всех случаях при минимизации строго унимодальных функций оптимизация работает быстрее упомянутых классических методов. Анализ многочисленных расчетных данных показал, что независимо от характера минимизируемых функций – гладких (в таблице 4 это функции 1 и 2) или функций с изломами (функции 3, 4), оптимизация работает быстрее метода золотого сечения на 15–25 % и в 1,5–2 раза быстрее метода бисекции.

Закключение. В работе представлен названный оптимизацией новый метод минимизации унимодальных функций, основанный на оптимальном отсечении отрезков. В результате многочисленных вычислительных экспериментов по минимизации строго унимодальных функций показано, что метод работает быстрее самого быстрого из своего семейства метода золотого сечения на 15–25 % и в 1,5 – 2 раза быстрее метода бисекции. Метод оптимизации быстро находит минимум монотонных функций за 5 обращений к каждой из них. Функции-константы распознаются оптимизацией за 3 вычисления. Метод демонстрирует высокое быстродействие и при минимизации функций с плоским дном.

Список литературы

1. Аббасов М. Э. Методы оптимизации. СПб.: ВВМ, 2014. 263 с.
2. Коднянко В. А. Метод экономной дихотомии: программный код и сравнительная эффективность. Аллея науки. 2018. № 8.

3. Коднянко В. А. О вычислительной избыточности метода дихотомии и условной минимизации унимодальных функций методом экономной дихотомии. Системы и средства информатики. 2019. Т. 29. № 1. С. 164–173. DOI: 10.14357/08696527190113.
4. Уайлд Д. Дж. Методы поиска экстремума / пер. с англ. М. : Наука, 1967. 268 с.
5. Федоров В. В. Численные методы максимина. М. : Наука, 1979. 272 с.
6. Brent R. P. Algorithms for minimization without derivatives. Courier Corporation. 2013. 195 p.
7. Kiefer J. K. Sequential minimax search for a maximum. Proceedings of the American Mathematical Society. 1953. Vol. 4. No. 3. Pp. 502–506.
8. Kodnyanko V. A. Economical dichotomous search for minimizing one-variable functions. Radio Electronics, Computer Science, Control. 2019. No. 3. Pp. 34–39. DOI: 10.15588/1607-3274-2019-3-4.
9. Delphi 7 and Its IDE. URL: https://flylib.com/books/en/2.37.1/delphi_7_and_its_ide.html.

Minimization of unimodal functions by optisection method

Kodnyanko Vladimir Aleksandrovich

Doctor of Technical Sciences, professor of the Department of Standardization, Metrology and Quality Management, Siberian Federal University. Russia, Krasnoyarsk. E-mail: kowlad@rambler.ru

Abstract. A new method for minimizing unimodal functions based on optimal segment cutting, called optisection, is presented. It is shown that in all cases the method is faster than similar classical methods of the segment cutting family – the bisection method and the golden section method. When minimizing strictly unimodal functions, optisection is 15–25 % faster than the golden section method and 1,5–2 times faster than the bisection method. The method finds the minimum of monotone functions in 5 calls to them. Constant functions are recognized by optisection in 3 calculations. When minimizing functions with a flat bottom, the speed of the method is comparable to the speed of recognizing monotone functions.

Keywords: bisection method, golden section method, optisection method, unimodal function, speed of the method.

References

1. Abbasov M. E. *Metody optimizatsii* [Optimization methods]. SPb. VVM. 2014. 263 p.
2. Kodnyanko V. A. *Metod ekonomnoj dihotomii: programnyj kod i sravnitel'naya effektivnost'*. Alleya nauki [Method of economical dichotomy: program code and comparative efficiency. Alley of Science]. No. 8. 2018.
3. Kodnyanko V. A. *O vychislitel'noj izbytochnosti metoda dihotomii i uslovnoj minimizatsii unimodal'nyh funktsij metodom ekonomnoj dihotomii. Sistemy i sredstva informatiki* [On the computational redundancy of the dichotomy method and conditional minimization of unimodal functions by the economical dichotomy method. Systems and Tools of Informatics]. 2019. Vol. 29. No. 1. Pp. 164–173. DOI: 10.14357/08696527190113.
4. Wild D. J., *Metody poiska ekstremuma* [Methods of searching for an extremum] / transl. from English. M. Nauka (Science). 1967. 268 p.
5. Fedorov V. V. *Chislennye metody maksimina* [Numerical methods of maximin]. M. Nauka (Science). 1979. 272 p.
6. Brent R. P. Algorithms for minimization without derivatives. Courier Corporation. 2013. 195 p.
7. Kiefer J. K. Sequential minimax search for a maximum. Proceedings of the American Mathematical Society. 1953. Vol. 4. No. 3. Pp. 502–506. DOI: 10.1090/S0002-9939-1953-0055639-3.
8. Kodnyanko V. A. Economical dichotomous search for minimizing one-variable functions. Radio Electronics, Computer Science, Control. 2019. No. 3. Pp. 34–39. DOI: 10.15588/1607-3274-2019-3-4.
9. Delphi 7 and Its IDE. Available at: https://flylib.com/books/en/2.37.1/delphi_7_and_its_ide.html.

Поступила в редакцию: 15.04.2025

Принята к публикации: 19.05.2025