

Нижегородский государственный университет им. Н. И.
Лобачевского

Радиофизический факультет
Кафедра математики

Отчет по лабораторной работе:

БЕЗУСЛОВНЫЙ ЭКСТРЕМУМ

ВВЕДЕНИЕ В ЧИСЛЕННЫЕ МЕТОДЫ

Выполнил:

студент 421 группы

Зайцев Юрий

Проверил:

Кулинич Виктор Валентинович

Нижний Новгород

2006 год

Содержание

Введение	3
1 Постановка учебно-практической задачи	3
2 Описание алгоритма	3
2.1 Критерии окончания процесса	4
2.2 Получение итерационной формулы	4
2.3 Модификация Ньютона-Рафсона	5
3 Исследование применимости алгоритма	6
4 Результаты расчетов	9
4.1 Диалог с программой (отладочная функция)	10
4.2 Диалог с программой (исследуемая функция)	10
4.3 Оценка овражности исследуемой функции	10
5 Заключение	11
6 Приложения	11
6.1 main.m	12
6.2 myfunction.m	14
6.3 mygradient.m	14
6.4 hessian.m	14
6.5 invhessian.m	15
6.6 alpha.m	15
6.7 nextx.m	16
Список литературы	17

Введение

В данной работе ставилась цель нахождения безусловного экстремума функции двух независимых переменных итерационным методом Ньютона-Рафсона, исследования применимости данного алгоритма к решению поставленной задачи и, наконец, проведения анализа полученных при помощи ЭВМ результатов.

1 Постановка учебно-практической задачи

Итак, в ходе выполнения данной лабораторной работы необходимо было решить задачу безусловной минимизации функции

$$f(x, y) = xy + \frac{50}{x} + \frac{20}{y} \quad (1)$$

итерационным методом Ньютона-Рафсона. Кроме того, требовалось нарисовать картину линий уровня минимизируемой функции в окрестности предполагаемой точки экстремума и нанести на неё траекторию поиска минимума. Также, для наглядности, можно было изобразить в изометрии поверхность, определяемую выбранной функцией.

Область \mathcal{D} , в пределах которой выбирались начальные точки и производилось исследование применимости итерационного метода, в нашем случае была определена следующим образом:

$$M(x, y) \in \mathcal{D} = \{(x, y) \in R^2 \mid x \in [0.1, 10], y \in [0.1, 10]\}. \quad (2)$$

Конкретно, начальная точка для итерационного процесса была выбрана равной

$$M_0(x_0, y_0) = (7.5, 1.25). \quad (3)$$

2 Описание алгоритма

Подробно алгоритм построения итерационной последовательности рассматривается в [2]. Ниже приведем лишь краткое описание выбранного алгоритма (метод Ньютона-Рафсона, модификация “Б”) применительно к конкретной задаче.

2.1 Критерии окончания процесса

Для решения поставленной перед нами задачи необходимо построить итерационную последовательность векторов $\{x^k\}_{k=0}^{\infty}$. Оборвать процесс нужно тогда, когда появится уверенность, что последний из элементов последовательности близок к точке минимума в том или ином смысле. Очевидно, что в таком случае, критерием окончания процесса может служить неравенство $\|x^k - \bar{x}\| < \bar{\delta}$, однако, непонятно каким образом проверить этот критерий, если точку минимума \bar{x} как раз и необходимо найти. Поэтому, была выбрана комбинация из двух критериев:

$$\|\nabla f(x^k)\| < \varepsilon_{gf} \quad (\text{по градиенту функции } f(x)) \quad (4)$$

$$\frac{\|x^k - x^{k-1}\|}{1 + \|x^{k-1}\|} < \varepsilon_{af} \quad (\text{по аргументу функции } f(x)). \quad (5)$$

Значения параметров ε_{gf} и ε_{af} должны были выбираться заранее и быть одними и теми же для всех итераций. В нашем случае, для обеспечения приемлемой точности вычислений, были выбраны значения $\varepsilon_{gf} = 10^{-5}$ и $\varepsilon_{af} = 10^{-5}$.

Такой выбор обусловлен тем, что, к сожалению, в связи с означенным выше противоречием, на данный момент не разработан надежный и универсальный критерий окончания счета, гарантировавший бы получение решения задачи минимизации с требуемой точностью $\bar{\delta}$ (см. [4]).

2.2 Получение итерационной формулы

Получим теперь использовавшуюся в работе итерационную формулу метода Ньютона. Данный метод основывается на том предположении, что заданную функцию можно локально аппроксимировать квадратичной функцией. Запишем для функции $f(x)$ в окрестности точки x^k формулу Тейлора второго порядка:

$$f(x) = Q_k(x) + o(\|x - x^k\|), \quad \text{где } \|x - x^k\| \rightarrow 0 \text{ при } k \rightarrow \infty$$

$$Q_k(x) = f(x^k) + (\nabla f(x^k), (x - x^k)) + \frac{1}{2}(\nabla^2 f(x^k)(x - x^k), (x - x^k)).$$

Если гессиан $\nabla^2 f(x^k)$ (матрица, составленная из вторых производных) функции $f(x)$ положительно определен, то квадратичная функция $Q_k(x)$ гарантированно достигает глобального минимума в точке

$$x^{k+1} = x^k - [\nabla^2 f(x^k)]^{-1} \nabla f(x^k).$$

Отсюда несложно получить формулу для нахождения p^k — вектора, определяющего направление движения от точки x^k к точке x^{k+1} :

$$p^k = -[\nabla^2 f(x^k)]^{-1} \nabla f(x^k) \text{ при } \alpha_k = 1. \quad (6)$$

Приведем формулы для расчета градиента, гессиана и матрицы, обратной гессиану для функции (1):

$$\begin{aligned} \nabla f(x^k, y^k) &= \begin{pmatrix} y^k - \frac{50}{x^{k^2}} \\ x^k - \frac{20}{y^{k^2}} \end{pmatrix}; \quad \nabla^2 f(x^k, y^k) = \begin{pmatrix} \frac{100}{x^{k^3}} & 1 \\ 1 & \frac{40}{y^{k^3}} \end{pmatrix} \\ [\nabla^2 f(x^k, y^k)]^{-1} &= \begin{pmatrix} -\frac{40x^{k^3}}{x^{k^3}y^{k^3} - 4000} & \frac{x^{k^3}y^{k^3}}{x^{k^3}y^{k^3} - 4000} \\ \frac{x^{k^3}y^{k^3}}{x^{k^3}y^{k^3} - 4000} & -\frac{100y^{k^3}}{x^{k^3}y^{k^3} - 4000} \end{pmatrix} \end{aligned}$$

В нашем случае расчет значения вектора p^k производился по следующей формуле:

$$p^k = - \begin{pmatrix} -\frac{40x^{k^3}}{x^{k^3}y^{k^3} - 4000} & \frac{x^{k^3}y^{k^3}}{x^{k^3}y^{k^3} - 4000} \\ \frac{x^{k^3}y^{k^3}}{x^{k^3}y^{k^3} - 4000} & -\frac{100y^{k^3}}{x^{k^3}y^{k^3} - 4000} \end{pmatrix} \begin{pmatrix} y^k - \frac{50}{x^{k^2}} \\ x^k - \frac{20}{y^{k^2}} \end{pmatrix}. \quad (7)$$

2.3 Модификация Ньютона-Рафсона

Чтобы обеспечить сходимость метода для любого начального приближения x^0 , метод Ньютона модифицируют, вводя переменный множитель α_k , получая фактически метод спуска.

В данной работе использовалась одна из модификаций метода Ньютона: метод Ньютона-Рафсона. В этом методе направление спуска определяется формулой (6), а множитель α_k , регулирующий длину шага,

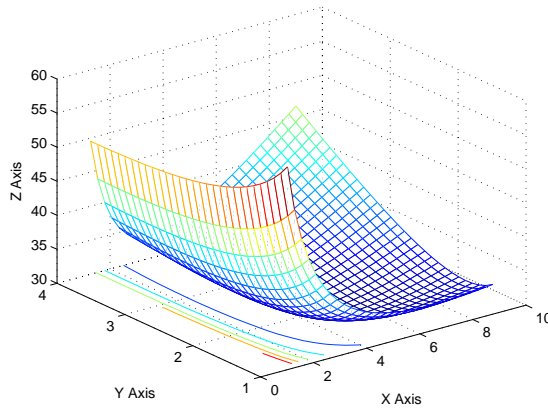


Рис. 1: Изометрия поверхности, определяемой функцией (1).

выбирается таким образом, чтобы выполнялось неравенство

$$f(x^k + \alpha_k p^k) \leq f(x^k) + \varepsilon \alpha_k (\nabla f(x^k), p^k),$$

где $\varepsilon \in (0, \frac{1}{2})$ — наперед заданная постоянная, одна и та же для всех итераций.

В данной работе было выбрано значение $\varepsilon = 0.1$.

Выбор значения α_k происходил как в градиентных методах с дроблением шага, а именно, следующим образом: положив сначала $\bar{\alpha} = \alpha$ проверялось неравенство

$$f(x^k + \bar{\alpha} p^k) < f(x^k). \quad (8)$$

Если оно выполнялось, брали $\alpha_k = \bar{\alpha}$, если нет — значение $\bar{\alpha}$ изменяли, домножая на фиксированный наперед заданный множитель λ , который в данной работе был выбран равным $\lambda = \frac{1}{2}$ и снова проверяли неравенство (8), продолжая до тех пор, пока оно не начинало выполняться.

В качестве начального значения α бралось $\alpha = 1$.

3 Исследование применимости алгоритма

Во-первых, для сходимости итерационного процесса необходима непрерывность функции в окрестности точки экстремума. Функция (1) при

$x, y > 0$, очевидно, является непрерывной, а, поэтому, непрерывность обеспечивается и в исследуемой области \mathcal{D} , см. определение области \mathcal{D} (2) и рис. 1.

Кроме того, сходимость метода Ньютона заведомо будет гарантироваться только при положительной определённости гессиана функции $f(x, y)$ (см. рассуждения выше). Воспользуемся для проверки этого условия критерием Сильвестра, гласящего, что для того, чтобы квадратичная форма, порожаемая данной матрицей была положительно определённой, необходимо и достаточно, чтобы все окаймляющие миноры этой матрицы были положительными. Гессиан H функции (1) и его миноры A_1 и A_2 имеют следующий вид:

$$H = \begin{pmatrix} \frac{100}{x^3} & 1 \\ 1 & \frac{40}{y^3} \end{pmatrix}; \quad A_1 = \frac{100}{x^3}; \quad A_2 = \frac{4000}{x^3 y^3} - 1$$

Очевидно, что при значениях $(x, y) \in \mathcal{D}$, которые использовались в итерациях (см. определение области \mathcal{D} (2) и путь на графике 3) оба минора A_1 и A_2 положительны, поэтому, согласно критерию Сильвестра, матрица Гессе для функции (1) в этой области положительно определена.

Для доказательства сходимости нашего итерационного метода воспользуемся следующей теоремой (см. [6]):

Теорема 1 *Если функция $f(x)$ ограничена снизу, а её градиент удовлетворяет условию Липшица¹, то какова бы ни была начальная точка итерационной последовательности x^0 ,*

$$\|\nabla f(x^k)\| \rightarrow 0, \text{ где } k \rightarrow \infty.$$

Если точка x^0 такова, что множество $K(x^0) = \{x : f(x) \leq f(x^0)\}$ ограничено, то последовательность $\{x^k\}$ сходится ко множеству $S = \{x : \nabla f(x) = 0\}$ стационарных точек функции $f(x)$, т.е.

$$\inf_{x \in S} \|x - x^k\| \rightarrow 0, \text{ где } k \rightarrow \infty.$$

¹Говорят, что функция $g(x)$ удовлетворяет условию Липшица на отрезке $x \in [a, b]$, если $(\exists N > 0)(\forall x_1, x_2 \in [a, b]) : |g(x_1) - g(x_2)| \leq N |x_1 - x_2|^\alpha$, $0 < \alpha \leq 1$. Число N называется константой Липшица.

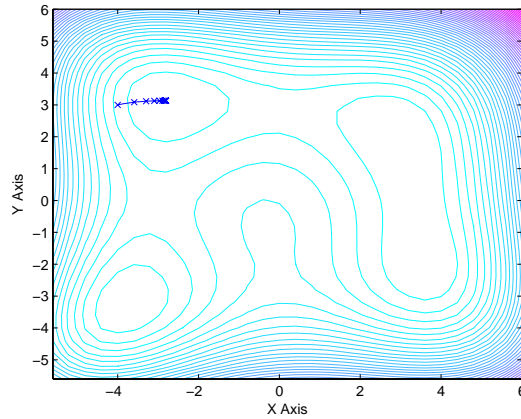


Рис. 2: Результат работы программы для отладочной функции.

Проверим эти условия для области \mathcal{D} (см. (2)). Выбор любой начальной точки из данной области (и, в том числе, точки (3)) приводит к сходимости итерационного процесса¹. Проверим условие ограниченности снизу в области \mathcal{D} функции (1):

$$f(M) = f(x, y) = xy + \frac{50}{x} + \frac{20}{y} > 0 \quad \forall M(x, y) \in \mathcal{D}$$

Как известно из курса математического анализа, непрерывная функция, имеющая непрерывные производные, в ограниченной области \mathcal{D} удовлетворяет условию Липшица. Наша функция имеет непрерывные в области \mathcal{D} производные любого порядка, поэтому градиент функции $\nabla f(x, y)$ также удовлетворяет условию Липшица. Таким образом, требования приведенной выше теоремы выполняются, а значит, итерационная последовательность, полученная использованным методом действительно сходится к точке минимума.

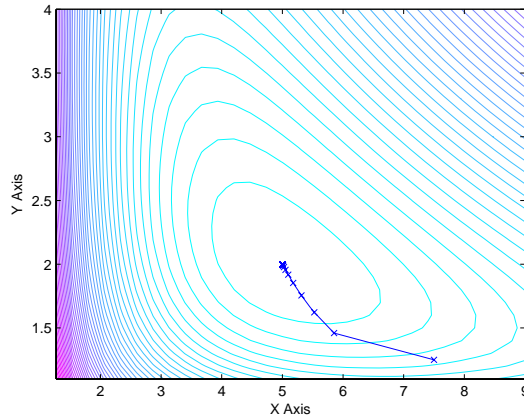


Рис. 3: Результат работы программы для исследуемой функции.

4 Результаты расчетов

В процессе написания, программа тестировалась на одной из отладочных функций, приведенных в методическом пособии [2]:

$$f(x, y) = (x^2 + y - 11)^2 + (x + y - 7)^2. \quad (9)$$

Эта функция, очевидно, удовлетворяет приведенной выше теореме о сходимости в своей области определения, и, при начальном значении $x^0 = (-4, 3)$, итерационная последовательность должна сходиться к точке минимума $\bar{x} = (-2.81, 3.13)$.

Результаты выполнения поставленной задачи иллюстрирует график на рис. 3, а на графике 2 приведены результаты работы программы для отладочной функции. Линии уровня показывают поведение функции в районе экстремума (минимума), а синяя ломанная линия представляет собой траекторию поиска решения итерационным методом Ньютона-Рафсона (крестиками обозначены значения x^k).

Ниже приведены распечатки вывода программы после двух запусков — с отладочной и исследуемой функцией соответственно.

¹Что, впрочем, ещё не означает, что выбор начальной точки вне этой области приведет к расходимости процесса.

4.1 Диалог с программой (отладочная функция)

Newton–Rafson Minimization

Start (X0, Y0): (-4, 3)

Precision (EpsX, EpsGF): (1e-005, 1e-005)

Speed (Eps): 0.1

Iterations: 25

Xmin = -2.805118e+000

Ymin = 3.131313e+000

F(Xmin, Ymin) = 2.688627e-013

4.2 Диалог с программой (исследуемая функция)

Newton–Rafson Minimization

Start (X0, Y0): (7.5, 1.25)

Precision (EpsX, EpsGF): (1e-005, 1e-005)

Speed (Eps): 0.1

Iterations: 21

Xmin = 5.000002e+000

Ymin = 1.999999e+000

F(Xmin, Ymin) = 3.000000e+001

4.3 Оценка овражности исследуемой функции

Найдя координаты минимума \bar{x} исследуемой функции (1), представляется возможным оценить её показатель овражности в окрестности этого минимума. По определению, под показателем овражности θ функции $f(x)$ в окрестности точки минимума \bar{x} подразумевают отношение наибольшего собственного числа матрицы Гессе $\nabla^2 f(\bar{x})$ к наименьшему. Чем больше этот показатель, тем более вытянутым и крутым является “овраг” поверхности уровня $f(x)$ в окрестности \bar{x} и тем медленнее сходятся в этой окрестности градиентные итерационные методы.

Оценим показатель овражности функции (1). Вспомним найденное

аналитическое выражение для $\nabla^2 f(x)$ и подставим в него найденное значение экстремума $M_{min}(\bar{x}, \bar{y}) = (5, 2)$:

$$\nabla^2 f(\bar{x}, \bar{y}) = \begin{pmatrix} \frac{100}{\bar{x}^3} & 1 \\ 1 & \frac{40}{\bar{y}^3} \end{pmatrix} = \{\bar{x} = 5, \bar{y} = 2\} = \begin{pmatrix} \frac{4}{5} & 1 \\ 1 & 5 \end{pmatrix}.$$

Собственные числа этой матрицы нетрудно найти:

$$\lambda_1 = \frac{29}{10} + \frac{1}{10}\sqrt{541}; \quad \lambda_2 = \frac{29}{10} - \frac{1}{10}\sqrt{541}.$$

Таким образом, коэффициент овражности

$$\theta = \frac{\lambda_1}{\lambda_2} = \frac{\frac{29}{10} + \frac{1}{10}\sqrt{541}}{\frac{29}{10} - \frac{1}{10}\sqrt{541}} \approx 9.103.$$

5 Заключение

В результате выполнения поставленной перед нами задачи можно с уверенностью выделить как достоинства, так и недостатки предложенного нам итерационного метода. К основным недостаткам метода Ньютона-Рафсона следует отнести необходимость вычисления вторых производных, а также тот факт, что сходимость заведомо гарантируется лишь при положительной определённости гессиана функции $f(x, y)$ (см. [6]).

Однако несмотря на это, если кроме удовлетворения требования $f(x, y) \in C^2$ и ограниченности сверху и снизу, для вторых производных функции дополнительно выполняется условие Липшица, метод Ньютона-Рафсона обладает квадратичной (!) скоростью сходимости в окрестности минимума функции (см. [5]).

6 Приложения

В приложениях приведем исходные тексты программы, написанные на языке Matlab. Отметим, что кажущаяся громоздкость связана с тем, что

процедуры вычисления функции, её градиента, матрицы Гессе и обратной ей матрицы вынесены в отдельные подпрограммы, чтобы упростить дальнейшее использование и модификации программы.

6.1 main.m

```
% main.m
% Newton-Rafson Minimization
clear all
fprintf(' \n\nNewton-Rafson Minimization\n\n');

% Function gradient and argument precision
epsx = 0.00001;
epsf = 0.00001;

% Epsilon used in alpha
eps = 0.1;

% Boundaries & number of [p x p]
p = 30;
% minx = 1; maxx = 9;
% miny = 1; maxy = 4;

% Test case
minx = -6; maxx = 6;
miny = -6; maxy = 6;

% Starting values
% x(:,1) = [7.5; 1.25];

% Test case
x(:,1) = [-4; 3];

k1 = max([epsx epsf]) * 2;
k2 = k1;
t = 1;
```

```

fprintf('Start (X0, Y0): (%g, %g) \n', x(:,1));
fprintf('Precision (EpsX, EpsGF): (%g, %g) \n', epsx,
    epsf);
fprintf('Speed (Eps): %g \n\n', eps);

while ((k1 > epsx) | (k2 > epsf))
    x(:, t + 1) = nextx(x(:, t));
    t = t + 1;
    k1 = norm(x(:, t) - x(:, t - 1)) / (1 + norm(x(:,
        t - 1)));
    k2 = norm(mygradient(x(:, t)));
end

fprintf('Iterations: %g \n\n', (t-1));
fprintf('Xmin = %e \n', x(1,t));
fprintf('Ymin = %e \n', x(2,t));
fprintf('F(Xmin, Ymin) = %e \n', myfunction(x(:,t)));

% Visualization
for i = 1:p
    for j = 1:p
        x1(i, j) = minx + (maxx - minx)/p*i;
        x2(i, j) = miny + (maxy - miny)/p*j;
        xc = [x1(i, j) x2(i, j)];
        z(i, j) = myfunction(xc);
    end
end

figure(1);
c = contour(x1, x2, z, 50);
xlabel('X Axis');
ylabel('Y Axis');
colormap(cool);
hold on;
plot(x(1,:), x(2,:), '-xb');
hold off;

figure(2);

```

```

%surf(z);
meshc(x1,x2,z);
%axis([minx maxx miny maxy 0 100])
xlabel('X Axis');
ylabel('Y Axis');
zlabel('Z Axis');

```

6.2 myfunction.m

```

% myfunction.m
function v = myfunction(x);

% Function

% Test case
v = (x(1)^2+x(2)-11)^2+(x(1)+x(2)^2-7)^2;

% My formula
% v = x(1)*x(2) + 50/x(1) + 20/x(2);

```

6.3 mygradient.m

```

% mygradient.m
function v = mygradient(x);

% Gradient

% Test case
v = [4*(x(1)^2+x(2)-11)*x(1)+2*x(1)+2*x(2)^2-14; 2*x
(1)^2+2*x(2)-22+4*(x(1)+x(2)^2-7)*x(2)];

% My formula
% v = [x(2)-50/x(1)^2; x(1)-20/x(2)^2];

```

6.4 hessian.m

```

% hessian.m
function v = hessian(x);
% Hessian

```

```
% Test case
v = [12*x(1)^2+4*x(2)-42 4*x(1)+4*x(2) ; 4*x(1)+4*x
(2) -26+12*x(2)^2+4*x(1) ];
```

```
% My formula
% v = [100/x(1)^3 1; 1 40/x(2)^3];
```

6.5 invhessian.m

```
% invhessian.m
function v = invhessian(x);
```

```
% Inverse Hessian
```

```
% Test case
v = [1/2*(-13+6*x(2)^2+2*x(1))/(-82*x(1)^2+36*x(1)^2*
x(2)^2+12*x(1)^3-26*x(2)+12*x(2)^3-4*x(1)*x(2)
+273-130*x(2)^2-42*x(1)) -(x(1)+x(2))/(-82*x(1)
^2+36*x(1)^2*x(2)^2+12*x(1)^3-26*x(2)+12*x(2)^3-4*
x(1)*x(2)+273-130*x(2)^2-42*x(1)) ; -(x(1)+x(2))
/(-82*x(1)^2+36*x(1)^2*x(2)^2+12*x(1)^3-26*x(2)
+12*x(2)^3-4*x(1)*x(2)+273-130*x(2)^2-42*x(1))
1/2*(6*x(1)^2+2*x(2)-21)/(-82*x(1)^2+36*x(1)^2*x
(2)^2+12*x(1)^3-26*x(2)+12*x(2)^3-4*x(1)*x(2)
+273-130*x(2)^2-42*x(1)) ];
```

```
% My formula
% v = [-(40*x(1)^3)/(-4000+x(1)^3*x(2)^3) (x(1)^3*x(2)
^3)/(-4000+x(1)^3*x(2)^3) ; (x(1)^3*x(2)^3)/(-4000+x
(1)^3*x(2)^3) -(100*x(2)^3)/(-4000+x(1)^3*x(2)^3) ];
```

6.6 alpha.m

```
% alpha.m
function var=alpha(x);
global eps

p = -invhessian(x) * mygradient(x);
```

```

g = mygradient(x);

a = 1;

i1 = 2; % random
i2 = 1; % i1 > i2 for the first iteration

while i1 > i2
    i1 = myfunction(x + a*p);
    i2 = myfunction(x) + a .* eps .* (g' * p);
    a = a / 2;
end

var = a;

```

6.7 nextx.m

```

% nextx.m
function xn = nextx(xp);

% Compute next X
xn = xp - alpha(xp) .* (invhessian(xp) * mygradient(xp)
    ));

```

Список литературы

- [1] Калиткин Н.Н., Численные методы. М.: Наука. 1978. 512 с.
- [2] Кулинич В.В., Новоженев М.М., Сумин В.И. Безусловный экстремум. Введение в численные методы. Методическое пособие, Нижний Новгород: изд. ННГУ им. Лобачевского, 2001 г. 24 с.
- [3] Самарский А. А., Введение в численные методы. М.: Наука. 1987. 288 с.
- [4] Васильев Ф. П., Численные методы решения экстремальных задач. М.: Наука, 1988.
- [5] Пшеничный Б.П., Данилин Ю.М. Численные методы в экстремальных задачах. М.: Наука, 1975.
- [6] Моисеев Н.Н., Иванчиков Ю.П., Столярова Е.М. Методы оптимизации. М.: Наука, 1978.