

13 Решение задач оптимизации

В этой главе будут рассмотрены возможности средства пакета Scilab для решения оптимизационных задач.

13.1 Поиск минимума функции одной переменной

В качестве простейшей оптимизационной задачи рассмотрим поиск локального минимума функции одной переменной.

ЗАДАЧА 13.1.

Найти минимум функции $f(x) = x^4 + 3x^3 - 13x^2 - 6x + 26$.

Решение задачи начнем с построения графика функции (см. листинг 13.1 и рис. 13.1).

```
x=-5:0.1:1;
```

```
y=x.^4+3*x.^3-13*x.^2-6*x+26;
```

```
plot(x,y);
```

```
xtitle('График функции f(x)=x^4+3*x^3-13*x^2-6*x+26','X','Y');
```

```
xgrid();
```

Листинг 13.1. Построение графика функции

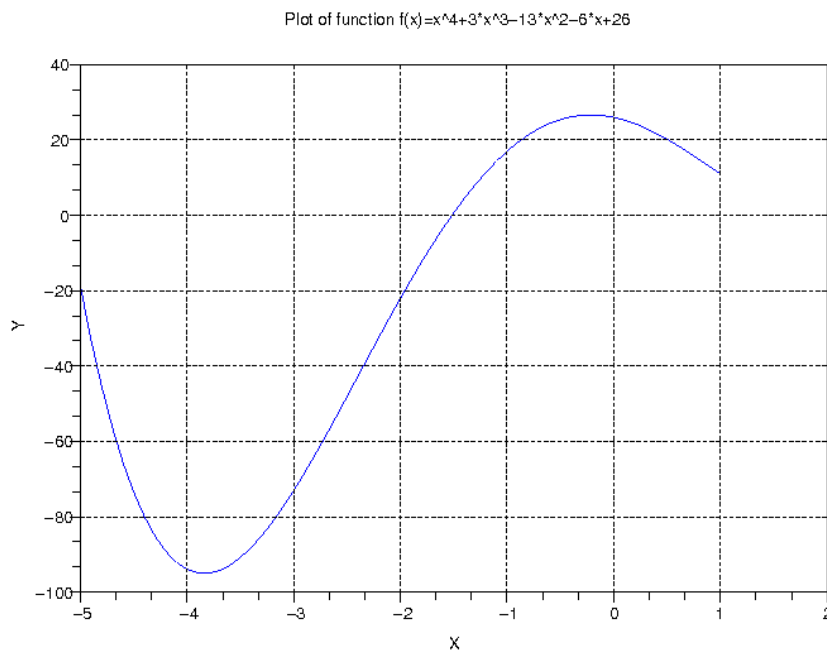


Рисунок 13.1. График функции $f(x) = x^4 + 3x^3 - 13x^2 - 6x + 26$

Построив график функции видно, что функция имеет минимум в районе точки -4. Для нахождения более точного значения минимума функции в Scilab служит функция

```
[f, xopt]=optim(costf, x0),
```

которая предназначена для поиска минимума любой функции, $x0$ — вектор столбец начальных приближений длиной n , функция *constf* определяет функцию, минимум которой ищется.

Функция возвращает минимум функции (f) и точку в которой функция достигает этого значения ($xopt$).

Главной особенностью функции *optim* является структура функции *costf*, которая должна быть следующей:

```
function [f,g,ind]=costf(x,ind)
//Функция costf должна возвращать функцию f, ее градиент g.
//f - функция от вектора неизвестных x, минимум которой ищется
f=gg(x);
// g - градиент функции f (вектор частной производной f по x),
g=numdiff(gg,x);1
endfunction
```

Для функции одной переменной в качестве f возвращается функция минимум которой ищется, в качестве функции g ее производная.

Если возвращаемое сформированной функцией *costf* значение *ind* равно 2, 3 или 4, то функция *costf* обеспечивает поиск минимума, т.е. в качестве результата функции *optim* возвращается f и $xopt$. Если $ind=1$, то в функции *optim* ничего не считается, условие $ind<0$ означает, что минимум $f(x)$ не может быть оценен, а $ind=0$ прерывает оптимизацию.

Вообще говоря, значение параметра *ind* является внутренним параметром для связи между *optim* и *costf*, для использования *optim* необходимо помнить, что параметр *ind* должен быть определен в функции *costf*.

Таким образом при использовании функции *optim* необходимо сформировать функцию *costf*, которая должна возвращать минимизируемую функцию f и ее градиент (производную).

Рассмотрим использование функции *optim* для поиска минимума $f(x)=x^4+3x^3-13x^2-6x+26$. Как видно из графика (см. рис. 13.1) минимум функция достигает в районе $x_{min} \approx 4$. На листинге представлено использование *optim* для поиска минимума функции одной переменной на примере $f(x)=x^4+3x^3-13x^2-6x+26$.

```
//Функция fi, в которой будет формироваться функция f и ее
// производная g.
function [f,g,ind]=fi(x,ind)
//Функция f, минимум которой ищется.
f=x^4+3*x^3-13*x^2-6*x+26
//Функция g - производная от функции f.
g=4*x^3+9*x^2-26*x-6
endfunction
// Начальное приближение точки минимума.
y0=-2;
//Для поиска точки минимума (xmin) и значения функции (fmin) в
// ней - вызов функции optim.
[fmin,xmin]=optim(fi,y0);
```

Листинг 13.2. Поиск минимума функции одной переменной.

Ниже представлены представлены полученные результаты поиска минимума

```
-->fmin
fmin =
- 95.089413
-->xmin
xmin =
- 3.8407084
```

Аналогично можно найти минимум любой другой функции одной переменной, главной задачей является проблема правильного выбора точки начального приближения. Но это проблема не пакета Scilab, а математическая проблема.

¹ Функция numdiff описана в седьмой главе.

13.2 Поиск минимума функции многих переменных

При нахождении минимума функции многих переменных функцию *costf* необходимо построить таким образом, чтобы входными данными в нее были значения вектора неизвестных x и параметра *ind*. Функция *costf* должна зависеть не от нескольких неизвестных, а от одного массива (вектора) неизвестных.

В случае функции многих переменных структура функции *costf* должна быть такой

```
function [f,g,ind]=costf(x,ind)
```

```
//f - функция от вектора неизвестных x, минимум которой ищется
f=gg(x);
```

```
// g - градиент функции f (вектор частной производной),
```

```
g=numdiff(gg,x);
```

```
endfunction
```

В качестве примера рассмотрим поиск минимума функции Розенброка $f(x,y)=100(y-x^2)^2+(1-x^2)^2$. График функции Розенброка представлен на рис. 13.2.

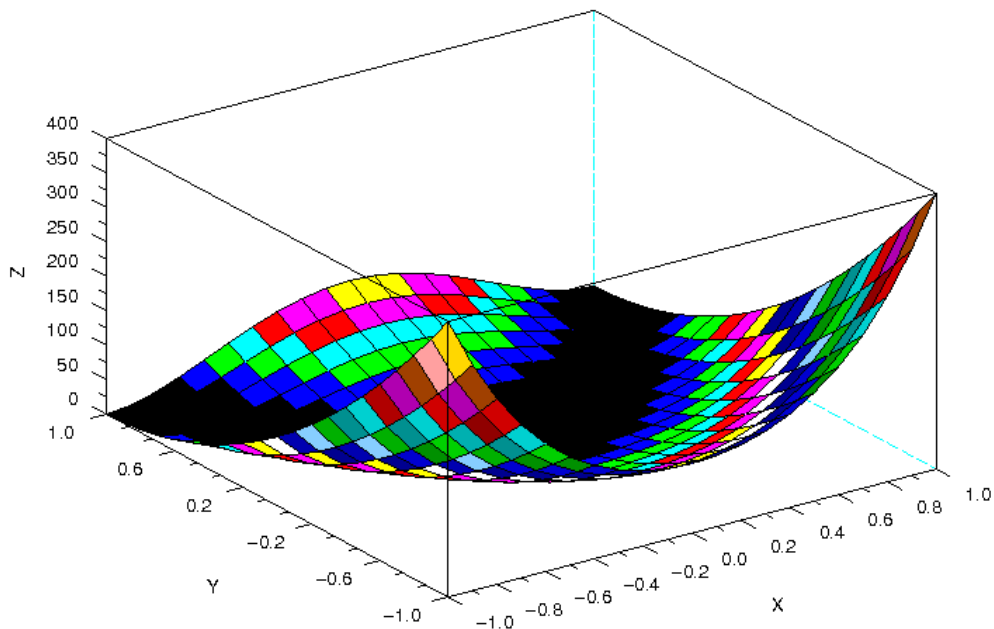


Рисунок 13.2. График функции Розенброка

Как известно функция Розенброка имеет минимум в точке (1,1) равный 0. В виду своей специфики функция Розенброка является тестовой для алгоритмов минимизации. Найдем минимум этой функции с помощью функции *optim* (листинг 13.2).

```
// Начальное приближение x0
```

```
x0=[-2;2]
```

```
//Функция Розенброка
```

```
function y=gg(x)
```

```
//Обратите внимание, здесь x - массив из двух неизвестных.
```

```
y=100*(x(2)-x(1)^2)^2+(1-x(1))^2;
```

```
endfunction
```

```
//Формирование функции cst, возвращающей функцию Розенброка и
```

```
//ее градиент.
```

```
function [f,g,ind]=cst(x,ind)
```

```
f=gg(x);
g=numdiff(gg,x);
endfunction
// Вызов функции optim
[f,xopt]=optim(cst,x0)
```

Листинг 13.3.

Ниже представлены результат поиска минимума функции Розенброка с помощью функции `optim`.

```
x0 =
-2.
 2.
xopt =
 0.9999955
 0.9999910
f =
 2.010D-11
```

13.3 Решение задач линейного программирования

Еще одной часто встречающихся в практике оптимизационных задач является задача линейного программирования. Знакомство с задачами линейного программирования начнем на примере задачи об оптимальном рационе.

Задача об оптимальном рационе. Имеется четыре вида продуктов питания: *П1*, *П2*, *П3*, *П4*. Известна стоимость единицы каждого продукта c_1, c_2, c_3, c_4 . Из этих продуктов необходимо составить пищевой рацион, который должен содержать не менее b_1 единиц белков, не менее b_2 единиц углеводов, не менее b_3 единиц жиров. Причем известно, в единице продукта *П1* содержится a_{11} единиц белков, a_{12} единиц углеводов и a_{13} единиц жиров и т.д. (см. таблицу 13.1).

Таблица 13.1. Содержимое белков, углеводов и жиров в продуктах

Элемент	белки	углеводы	жиры
<i>П1</i>	a_{11}	a_{12}	a_{13}
<i>П2</i>	a_{21}	a_{22}	a_{23}
<i>П3</i>	a_{31}	a_{32}	a_{33}
<i>П4</i>	a_{41}	a_{42}	a_{43}

Требуется составить пищевой рацион, чтобы обеспечить заданные условия при минимальной стоимости.

Пусть x_1, x_2, x_3, x_4 - количества продуктов *П1*, *П2*, *П3*, *П4*. Общая стоимость рациона равна

$$L = c_1 x_1 + c_2 x_2 + c_3 x_3 + c_4 x_4 = \sum_{i=1}^4 c_i x_i \quad (13.1)$$

Сформулируем ограничение на количество белков, углеводов и жиров в виде неравенств. В одной единице продукта *П1* содержится a_{11} единиц белков, в x_1 единицах - $a_{11}x_1$, в x_2 единицах продукта *П2* содержится $a_{21}x_2$ единиц белка и т.д.

Следовательно общее количество белков во всех четырех типов продукта равно $\sum_{j=1}^4 a_{jl}x_j$ и должно быть не больше b_1 . Получаем первое ограничение

$$a_{11}x_1 + a_{21}x_2 + a_{31}x_3 + a_{41}x_4 \leq b_1 \quad (13.2)$$

Аналогичные ограничения для жиров и углеводов имеют вид:

$$a_{12}x_1 + a_{22}x_2 + a_{32}x_3 + a_{42}x_4 \leq b_2 \quad (13.3)$$

$$a_{13}x_1 + a_{23}x_2 + a_{33}x_3 + a_{43}x_4 \leq b_3 \quad (13.4)$$

Принимаем во внимание, что x_1, x_2, x_3, x_4 положительные значения, получим еще четыре ограничения

$$x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0 \quad (13.5)$$

Таким образом задачу о рациональном рационе можно сформулировать следующим образом: найти значения переменных x_1, x_2, x_3, x_4 , удовлетворяющие системе ограничений (13.2) - (13.5), при которых линейная функция (13.1) принимала бы минимальное значение.

Задача об оптимальном рационе является задачей линейного программирования, функция (13.1) называется функцией цели, а ограничения (13.2) - (13.5) системой ограничений задачи линейного программирования.

В задачах линейного программирования функция цели функция цели L и система ограничений являются линейными.

В общем случае задачу линейного программирования можно сформулировать следующим образом. Найти такие значения x_1, x_2, \dots, x_n , удовлетворяющие системе ограничений (), при которых функция цели L (13.7) достигает своего минимального (максимального) значения

$$\sum_{j=1}^n a_{ij}x_j \leq b_i, i=1, \dots, m, \quad (13.6)$$

$$x_i \geq 0$$

$$L = c_1x_1 + c_2x_2 + \dots + c_nx_n = \sum_{i=1}^n c_ix_i \quad (13.7)$$

Для решения задач линейного программирования в Scilab предназначена функция *linpro* следующей структуры

$$[x, kl, f] = \text{linpro}(c, A, b[, ci, cs][, k][, x0])$$

Здесь c - массив (вектор-столбец) коэффициентов при неизвестных функции цели, длина вектора n совпадает с количеством неизвестных x , A - матрица при неизвестных из левой части системы ограничений, количество строк матрицы равно количеству ограничений m , а количество столбцов совпадает с количеством неизвестных n , b - массив (вектор-столбец) содержит свободные члены системы ограничений, длина вектора m , ci - массив (вектор-столбец) размерностью n содержит нижнюю границу переменных $(ci_j \leq x_j)$, если таковая отсутствует, указывают [], cs - массив (вектор-столбец) длиной n , содержит верхнюю границу переменных $(cs_j \geq x_j)$, если таковая отсутствует, указывают [], k - целочисленная переменная используется, если в систему ограничений кроме неравенств входят и равенства, в матрице они будут находиться в k первых строках, оставшиеся l строк займут неравенства,

то есть $m=k+l$, x_0 - вектор-столбец начальных приближений длиной n .

Функция `linpro` возвращает массив неизвестных x , минимальное значение функции f и массив множителей Лагранжа kl [].

Рассмотрим использование функции `linpro` на примере решения следующей задачи линейного программирования.

ЗАДАЧА 13.2.

Найти такие значения переменных x_1, x_2, x_3, x_4 при которых функция цели L

$$L = -x_2 - 2x_3 + x_4$$

достигает своего минимального значения и удовлетворяются ограничения:

$$3x_1 - x_2 \leq 2$$

$$x_2 - 2x_3 \leq -1$$

$$4x_3 - x_4 \leq 3$$

$$5x_1 + x_4 \geq 6$$

$$x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0$$

Обратите внимание, что в четвертом ограничении присутствует знак \geq . Для приведения системы ограничений виду (13.6) необходимо четвертое уравнение умножить на -1. Решение задачи 13.1 представлено на листинге 13.4.

```
c=[0;-1;-2;1];
A=[3 -1 0 0;0 1 -2 0; 0 0 4 -1; -5 0 0 -1];
b=[2;-1;3;-6];
ci=[0;0;0;0];
[x,kl,f]=linpro(p,A,b,ai,[])
```

Листинг 13.4.

Полученные значения представлены на листинге 13.5.

f=

2.

kl=

0.

0.

0.

0.

0.0909091

1.0909091

0.5454545

0.4545455

x=

1.

1.

1.

1

Листинг 13.5.

Рассмотрим решение еще одной задачи линейного программирования.

ЗАДАЧА 13.3.

Туристическая фирма заключила контракт с двумя турбазами на одном из черноморских курортов, рассчитанных соответственно, на 200 и 150 человек.

Туристам для осмотра предлагается дельфинарий в городе, ботанический сад и походы в горы.

Составить маршрут движения туристов так, чтобы это обошлось возможно дешевле, если дельфинарий принимает в день 70 организованных туристов, ботанический сад 180, а в горы в один день могут пойти 110 человек.

Стоимость одного посещения выражается таблицей 13.2.

Таблица 13.2.

Турбаза	Дельфинарий	Ботанический сад	Поход в горы
1	5	6	20
2	10	12	5

Для решения задачи введем следующие обозначения:

x_1 число туристов первой турбазы, посещающих дельфинарий;

x_2 число туристов первой турбазы, посещающих ботанический сад;

x_3 число туристов первой турбазы, отправляющихся в поход;

x_4 число туристов второй турбазы, посещающих дельфинарий;

x_5 число туристов второй турбазы, посещающих ботанический сад;

x_6 число туристов второй турбазы, отправляющихся в поход.

Составим функцию цели, заключающуюся в минимизации стоимости мероприятий фирмы:

$$Z = 5x_1 + 6x_2 + 20x_3 + 10x_4 + 12x_5 + 5x_6.$$

Руководствуясь условием задачи, определим ограничения:

$$x_1 + x_4 \leq 70;$$

$$x_2 + x_5 \leq 180;$$

$$x_3 + x_6 \leq 110;$$

$$x_1 + x_2 + x_3 = 200;$$

$$x_4 + x_5 + x_6 = 150.$$

Кроме того, количество туристов, участвующих в мероприятиях не может быть отрицательным: $x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0, x_5 \geq 0, x_6 \geq 0$.

Решение задачи представлено в листинге 13.6. В массиве x будут храниться значения x_1, x_2, x_3, x_4, x_5 и x_6 . В матрице коэффициентов A , первые две строки отражают равенства системы ограничений, поэтому включен параметр $k=2$. Вектор ci представляет нижнюю границу неизвестных, то есть отражает тот факт, что они не могут быть меньше нуля. Если присвоить результат вычислений функции вектору из трех компонент, то первая из них будет содержать вектор неизвестных x , вторая — множители Лагранжа kl , а третья — минимальное значение функции цели f .

$$Z = [5; 6; 20; 10; 12; 5];$$

$$A = [1 \ 1 \ 1 \ 0 \ 0 \ 0;$$

$$0 \ 0 \ 0 \ 1 \ 1 \ 1; 1 \ 0 \ 0 \ 1 \ 0 \ 0; 0 \ 1 \ 0 \ 0 \ 1 \ 0; 0 \ 0 \ 1 \ 0 \ 0 \ 1];$$

$$b = [200; 150; 70; 180; 110];$$

$$ci = [0; 0; 0; 0; 0; 0];$$

$$k = 2;$$

$$[x, kl, f] = \text{linpro}(Z, A, b, ci, [], k);$$

Листинг 13.6.

Полученное в Scilab решение задачи представлено на листинге 13.7

```
f=
    2120.
kl=
    0.
    0.
   -20.
    0.
   -1.
    0.
   -6.
  -11.
    1.
    0.
    6.
x=
    30.
   170.
    0.
   40.
    0.
   110.
```

Листинг 13.7.

Авторы надеются, что разобрав примеры из этой главы, читатель сможет использовать Scilab для решения своих оптимизационных задач.

Этой главой мы завершаем первое знакомство с пакетом и надеемся, что свободно распространяемый пакет Scilab поможет читателю решать математические задачи, возникающие в его практической деятельности.