МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное
образовательное учреждение высшего образования
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ

АНАЛИТИЧЕСКИЕ И ЧИСЛЕННЫЕ МЕТОДЫ РЕШЕНИЯ ЭКСТРЕМАЛЬНЫХ ЗАДАЧ

Лабораторный практикум

Санкт-Петербург
2021

СОДЕРЖАНИЕ

ПРЕДИСЛОВИЕ

Целью выполнения лабораторных работ является закрепление теоретического материала по курсу «Методы оптимизации» и получение практических навыков решения экстремальных задач с использованием математических программных средств.

По результатам выполнения работы студентом оформляется отчет.

Отчет должен содержать титульный лист, соответствующий образцу, размещенному на сайте университета.

При оформлении текста отчета следует соблюдать требования ГОСТ 7.32 – 2017, размещенные на сайте университета. В соответствии с этими требованиями текст набирается шрифтом Times New Roman кеглем не менее 12, строчным, без выделения, с выравниванием по ширине; строки разделяются полуторным интервалом. Абзацный отступ должен быть одинаковым и равным по всему тексту 1,25 см. Поля страницы: верхнее и нижнее – 20 мм, левое – 30 мм, правое – 15 мм. Полужирный шрифт применяется только для заголовков разделов и подразделов. Разрешается использовать компьютерные возможности акцентирования внимания, применяя шрифты разной гарнитуры.

Основную часть отчета следует делить на разделы в соответствии с пунктом «Содержание отчета». Разделы должны иметь порядковую нумерацию в пределах всего текста. Заголовки разделов следует печатать с абзацного отступа с прописной буквы, полужирным шрифтом, без точки в конце, не подчеркивая. Если заголовок состоит из двух предложений, их разделяют точкой. Переносы слов в заголовках не допускаются. Каждый раздел основной части отчета начинают с новой страницы.

Страницы отчета следует нумеровать арабскими цифрами, соблюдая сквозную нумерацию по всему тексту. Титульный лист включают в общую нумерацию страниц; номер страницы на титульном листе не проставляют. Номер страницы проставляют в центре нижней части листа без точки.

На все рисунки должны быть ссылки в тексте (например, …в соответствии с рисунком 1). Рисунки следует нумеровать арабскими цифрами сквозной нумерацией. Рисунки могут иметь наименование и пояснительные данные (подрисуночный текст).

На все таблицы должны быть ссылки в тексте. Таблицы следует нумеровать арабскими цифрами сквозной нумерацией. Наименование таблицы следует помещать над таблицей слева, без абзацного отступа.

Полностью оформленный отчет допускается к защите. Для успешной защиты отчета обучающемуся необходимо уметь аргументированно обосновывать приведенные в отчете утверждения и выводы.

ВВЕДЕНИЕ

Под *оптимизацией* в математике понимают задачу о нахождении экстремума (минимума или максимума) некоторой функции.

Оптимизационные задачи встречаются во многих прикладных областях: при проектировании технических объектов, в задачах управления, экономики, логистики и других. В качестве примера можно привести задачи о поиске наикратчайшего или наискорейшего пути, о поиске формы сосуда с максимальным объемом, о поиске наиболее экономически выгодного управляющего воздействия на объект.

Для поиска строгого математического решения оптимизационной задачи необходимо записать ее формулировку в виде некоторой функции, экстремум которой нужно найти. Эта функция $J = f(\mathbf{x})$ называется *критерием* или *целевой функцией*.

Для решения различных классов оптимизационных задач разработаны различные методы оптимизации [1, 2].

В зависимости от задачи целевая функция J может быть как от одного аргумента, так и от нескольких (в последнем случае $\mathbf{x}$ – вектор). В соответствии с числом аргументов целевой функции выделяют методы *одномерной* и *многомерной* оптимизации. Методы одномерной оптимизации предназначены для поиска экстремума функций одного аргумента, многомерные – для целевой функции нескольких аргументов.

Различают методы *условной* и *безусловной* оптимизации. Методы условной оптимизации применяются для поиска экстремума критерия $J = f(\mathbf{x})$ при дополнительном ограничении, например, в виде равенства $g(\mathbf{x}) = 0$. Методы безусловной оптимизации используют, если ограничения для целевой функции отсутствуют.

Аналитические методы оптимизации позволяют получить решение задачи в символьном виде. Однако многие оптимизационные задачи из-за сложности формулировки критерия не имеют аналитического решения, или поиск этого решения слишком трудоемок. В таких случаях используют аппарат *численных* методов оптимизации, позволяющих найти решение с некоторой точностью ε.

По виду целевой функции и ограничений задачи оптимизации и методы их решения делятся на два класса: *линейного программирования* (целевая функция и ограничения являются линейными функциями) и *нелинейного программирования*.

Исходя из требований к гладкости и наличию у целевой функции частных производных, могут быть выделены: *методы нулевого порядка* (*прямые*), требующие вычислений только целевой функции в точках приближений, *методы первого порядка*, требующие вычисления первых частных производных целевой функции, *методы второго порядка,* требующие вычисления вторых частных производных целевой функции. Методы более высоких порядков применяются редко.

По принципу поиска экстремума все методы можно разделить на: *детерминированные*, в которых движение к экстремуму осуществляется из некоторой начальной точки, и *стохастические* (случайные), согласно которым в процессе поиска экстремума осуществляются случайные переходы. Как привило, последние используются для численного поиска глобального экстремума многоэкстремальных функций.

Лабораторная работа
МНОГОМЕРНАЯ БЕЗУСЛОВНАЯ ОПТИМИЗАЦИЯ

Цель работы: ознакомиться с известными аналитическими и численными детерминированными методами многомерной безусловной оптимизации, получить опыт решения многомерной оптимизационной задачи, приобрести навыки использования математических программных средств для решения многомерных экстремальных задач.

1. Теоретические сведения

Задача многомерной безусловной оптимизации заключается в поиске экстремума функции от нескольких переменных:

$$J = f(x_1, \dots, x_n) \to extr.$$

Более удобной в ряде случаев является векторная форма записи этой задачи:

$$J = f(\mathbf{x}) \to extr,$$

где $\mathbf{x} = [x_1, \dots, x_n]$ – вектор переменных.

1.1. Аналитическая многомерная оптимизация

Так же, как и в одномерном случае, аналитический метод решения многомерных задач на безусловный экстремум гладких функций опирается на *лемму Ферма*. В соответствии с этой леммой поиск экстремума функции следует производить на множестве стационарных точек этой функции. Стационарной называется точка, в которой все частные производные функции равны нулю.

Для поиска стационарных точек функции нескольких переменных $J = f(x_1, ..., x_n)$ необходимо составить систему уравнений, приравняв нулю частные производные этой функции по каждой из переменных:

$$\begin{cases} \dfrac{\partial J}{\partial x_1} = 0, \\ \dots \\ \dfrac{\partial J}{\partial x_n} = 0. \end{cases} \qquad (1)$$

Вектор, составленный из частных производных функции $f(\mathbf{x})$, называют градиентом и обозначают $\nabla f(\mathbf{x})$. Таким образом, систему (1) можно записать в виде

$$\nabla f(\mathbf{x}) = 0.$$

Решением этой системы будет множество стационарных точек. Напомним, что стационарными точками являются точки максимумов, минимумов, а также так называемые седловые точки (в случае функции одной переменной седловую точку называют точкой перегиба). Для функции $J = f(\mathbf{x}) = f(x_1, ..., x_n)$ вид стационарной точки $\mathbf{x}^* = (x_1^*; \ldots; x_n^*)$ однозначно связан со знакоопределенностью гессиана функции в этой точке:

- если гессиан $\mathbf{H}(f(\mathbf{x}^*)) > 0$, то $\mathbf{x}^*$ – точка минимума;
- если гессиан $\mathbf{H}(f(\mathbf{x}^*)) < 0$, то $\mathbf{x}^*$ – точка максимума;
- если гессиан $\mathbf{H}(f(\mathbf{x}^*))$ не знакоопределен и не вырожден ($\det \mathbf{H}(f(\mathbf{x}^*)) \neq 0$), то $\mathbf{x}^*$ – седловая точка.

Напомним, что *гессианом* (или матрицей Гессе) функции $J = f(x_1, ..., x_n)$ называется матрица, составленная из вторых производных функции J:

$$\mathbf{H}(J) = \begin{bmatrix} \dfrac{\partial^2 J}{\partial {x_1}^2} & \cdots & \dfrac{\partial^2 J}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial^2 J}{\partial x_n \partial x_1} & \cdots & \dfrac{\partial^2 J}{\partial {x_n}^2} \end{bmatrix}.$$

Напомним, что матрица называется *положительно* (*отрицательно*) *определенной*, если все ее собственные числа положительны (отрицательны).

Если все вторые производные функции J, не стоящие на главной диагонали матрицы $\mathbf{H}(J)$, непрерывны, матрица $\mathbf{H}(J)$ является симметричной. В этом случае для определения знакоопределенности матрицы можно воспользоваться критерием Сильвестра.

Согласно *критерию Сильвестра*, симметричная матрица

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{12} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{2n} & \cdots & a_{nn} \end{bmatrix}$$

– является *положительно определенной* ($\mathbf{A} > 0$), если все ее угловые миноры положительны:

$$a_{11} > 0, \begin{vmatrix} a_{11} & a_{12} \\ a_{12} & a_{22} \end{vmatrix} > 0, ..., \begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{12} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{2n} & \cdots & a_{nn} \end{vmatrix} > 0;$$

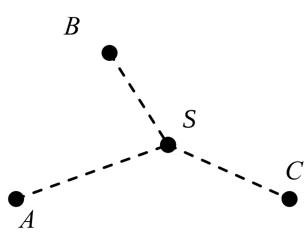
– является *отрицательно определенной* ($\mathbf{A} < 0$), если все ее угловые миноры четного порядка положительны, а нечетного порядка – отрицательны:

$$a_{11} < 0, \begin{vmatrix} a_{11} & a_{12} \\ a_{12} & a_{22} \end{vmatrix} > 0, \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{12} & a_{22} & a_{23} \\ a_{13} & a_{23} & a_{33} \end{vmatrix} < 0, \begin{vmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{12} & a_{22} & a_{23} & a_{24} \\ a_{13} & a_{23} & a_{33} & a_{34} \\ a_{14} & a_{24} & a_{34} & a_{44} \end{vmatrix} > 0, ...;$$

– иначе – матрица **A** *не знакоопределена*.

Пример 1. *Задача Штейнера*.

Требуется связать три деревни *A*, *B* и *C* дорогами так, чтобы их общая протяженность была минимальной (рис. 1).



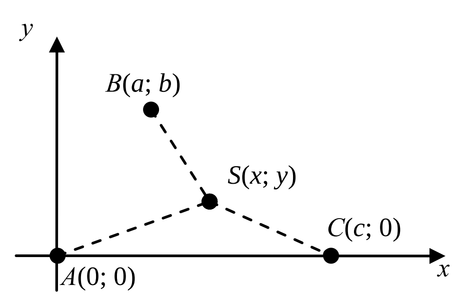


Рис. 1. Задача Штейнера

Задача была предложена Пьером Ферма и получила свое название по имени исследовавшего ее немецкого геометра Якоба Штейнера (1796 – 1863).

Формулировка Ферма. Для заданных трех точек найти такую четвертую, что если из неё провести три отрезка в данные точки, то сумма этих трех отрезков даст наименьшую величину.

Аналитическое решение. Как и при решении задачи Герона, введем систему координат так, чтобы максимально упростить решение. Расположим начало координат в точке A, а ось абсцисс проведем через точку C, как это показано на рис. 1, справа. По условию задачи, известны координаты точек A (0; 0), B (a; b) и C (c; 0). Пусть a = 1, b= 3 и c = 5. Требуется найти координаты точки S (x; y) такие, что сумма длин отрезков AS, CS и BS будет минимальной:

$$J = AS + CS + BS \to \min. \tag{2}$$

Запишем J как функцию неизвестных координат точки S. Для этого выразим длины отрезков AS, CS и BS через их координаты (по теореме Пифагора):

$$AS = \sqrt{x^2 + y^2},\ BS = \sqrt{(x-a)^2 + (b-y)^2},\ CS = \sqrt{(c-x)^2 + y^2}.$$

Подставляя эти выражения в (2), получим критерий оптимизации:

$$J = f(x,y) = \sqrt{x^2 + y^2} + \sqrt{(x-a)^2 + (b-y)^2} + \sqrt{(c-x)^2 + y^2} \to \min.$$

Для поиска стационарных точек функции найдем ее частные производные и, приравняв каждую производную нулю, получим систему уравнений:

$$\begin{cases} \dfrac{\partial f(x,y)}{\partial x} = \dfrac{x}{\sqrt{x^2+y^2}} + \dfrac{x-a}{\sqrt{(x-a)^2+(b-y)^2}} + \dfrac{x-c}{\sqrt{(c-x)^2+y^2}} = 0, \\ \dfrac{\partial f(x,y)}{\partial y} = \dfrac{y}{\sqrt{x^2+y^2}} + \dfrac{y-b}{\sqrt{(x-a)^2+(b-y)^2}} + \dfrac{y}{\sqrt{(c-x)^2+y^2}} = 0. \end{cases}$$

Для решения этой системы уравнений при a = 1, b= 3 и c = 5 воспользуемся возможностями *Symbolic Math Toolbox* пакета *MATLAB.*

```
syms x y a b c real %объявление символьных переменных
fxy=sqrt(x^2+y^2)+sqrt((x-a)^2+(b-y)^2)+sqrt((c-x)^2+y^2);%цел. ф-я
fxy=subs(fxy,{a,b,c},{1,3,5});
dfx=diff(fxy,x);dfy=diff(fxy,y);%частные производные
sp=solve(dfx==0,dfy==0,x,y);%решение системы уравнений
```

```
xp=simplify(sp.x),yp=simplify(sp.y)%координаты стационарной точки
```

Аналогичные возможности предоставляет библиотека *sympy* языка *Python*.

В результате получим стационарную точку функции J с координатами:

$$(x;y)=\left(\frac{1+\sqrt{3}}{2};\frac{9-\sqrt{3}}{6}\right).$$

Определим вид полученной стационарной точки. Для этого построим матрицу Гессе и рассчитаем ее значение в стационарной точке.

```
H=hessian(fxy,{x,y});%Гессиан
H=double(subs(H,{x,y},{xp,yp}));
```

Получим:

$$\mathbf{H}=\begin{bmatrix} 0.7929 & -0.0860 \\ -0.0860 & 0.5636 \end{bmatrix}.$$

Гессиан симметричен и все его угловые миноры положительны:

$$0.7929>0,\ \det\mathbf{H}>0.$$

Таким образом, согласно критерию Сильвестра, матрица **H** в стационарной точке положительно определена, следовательно, стационарная точка является точкой минимума.

Ответ. При a = 1, b= 3 и c = 5 для построения оптимальной сети дорог нужно каждую из деревень A (0; 0), B (1; 3) и C (5; 0) соединить прямой дорогой с точкой

$$S\left(\frac{1+\sqrt{3}}{2};\frac{9-\sqrt{3}}{6}\right)\approx(1.366;1.211).$$

Схема полученной оптимальной сети дорог на плоскости xy изображена на рис. 2.

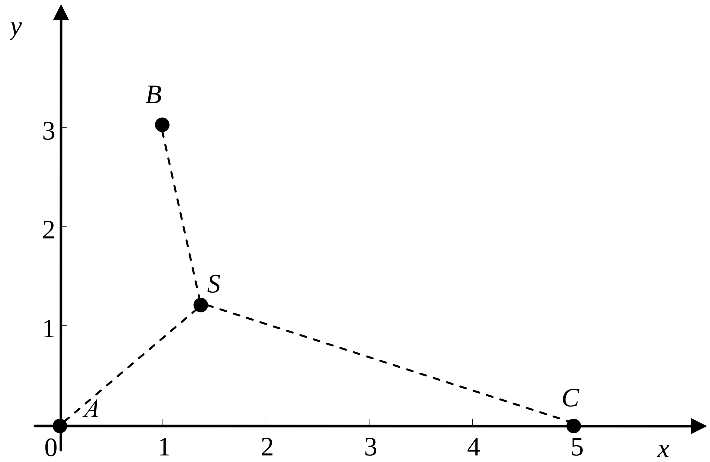


Рис. 2. Решение задачи Штейнера

Точка, являющаяся решением задачи Штейнера, то есть точка плоскости, сумма расстояний от которой до вершин треугольника минимальна, называется *точкой Ферма* или иногда – *точкой Торричелли*.

Геометрическое построение. Как видно, аналитические расчеты координат точки Ферма являются достаточно трудоемкими, однако согласно теореме Торричелли, существует простой геометрический способ построения точки S. Для геометрического построения S на каждой из сторон треугольника ABC нужно построить равносторонний треугольник (ABC', BCA' и CAB' соответственно). Если все углы треугольника ABC не превосходят 120°, точка пересечения прямых AA', BB' и CC' является точкой Ферма S (рис. 3). Если один из углов треугольника ABC больше 120°, точка S совпадает с вершиной тупого угла.

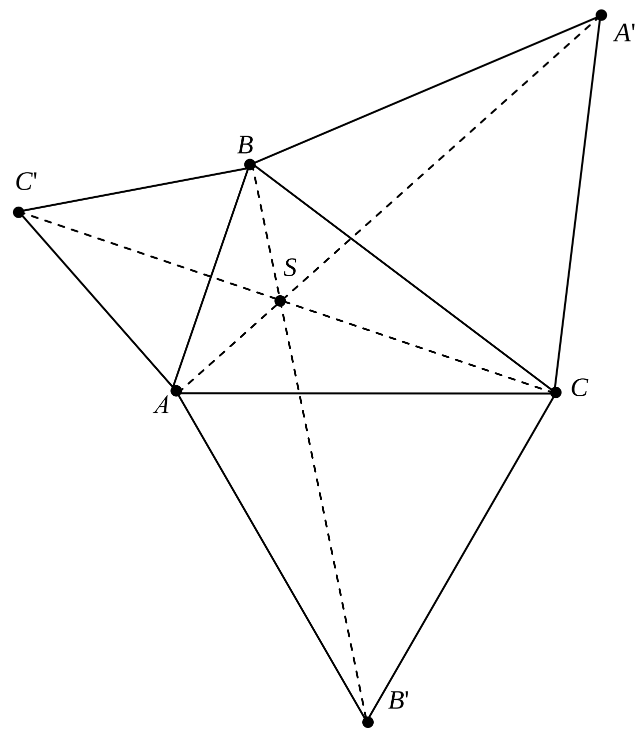


Рис. 3. Геометрическое построение точки Ферма

Экспериментальное построение. Точку Ферма можно построить и с помощью следующего физического эксперимента. На горизонтальной плоскости в точках A, B и C сделаем отверстия. В каждое отверстие пропустим нить. Верхние концы нитей свяжем, а к нижним концам нитей привяжем грузы одинаковой массы. После установления в получившейся системе равновесия, узел окажется в точке Ферма S.

1.2. Численная многомерная оптимизация

В случае если аналитический поиск решения многомерной экстремальной задачи невозможен или нецелесообразен, применяют численные методы многомерной оптимизации (см. раздел 2.2.3 в [1]).

Метод Хука-Дживса (ускоренная модификация)

Наиболее простым методом многомерной оптимизации нулевого порядка является метод покоординатного спуска [1], суть которого заключается в поочередном выполнении одномерной оптимизации по каждой из координат. Этот метод был предложен Карлом Фридрихом Гауссом, поэтому иногда его называют методом Гаусса. Одним из недостатков метода покоординатного спуска является низкая скорость сходимости при «овражистом» рельефе целевой функции.

Ускорить поиск экстремума для «овражистых» функций в некоторых случаях позволяет идея, предложенная в 1961 году *Р. Хуком и Т. А. Дживсом* [1]. Оригинальный алгоритм этого метода описан в [1]. Здесь рассмотрим его ускоренную модификацию.

Отличие ускоренной модификации метода Хука-Дживса от метода покоординатного спуска заключается в добавлении в конце каждой итерации еще одного действия, а именно: минимизации функции в направлении совершенного шага покоординатного спуска (то есть вдоль прямой, соединяющей точку до начала итерации и точку, полученную по окончании итерации покоординатного спуска).

Этап минимизации функции вдоль координатных осей называется этапом *исследующего поиска*, так как при этом определяется наиболее перспективное направление спуска. Этап минимизации вдоль полученного направления называется этапом *поиска по образцу*.

Рассмотрим пример минимизации функции модифицированным методом Хука-Дживса.

Пример 2. Дана функция:

$$f(x, y) = (x - y)^2 + \left(\frac{x + y - 10}{3}\right)^2.$$

Требуется найти минимум $f(x,y)$ с точностью $\varepsilon = 0.0001$ на интервале $x \in [0; 10]$, $y \in [0; 10]$.

Начальные установки. Выберем в качестве начальной левую верхнюю точку области поиска $(x_0; y_0) = (0; 10)$.

Итерация 1.

Исследующий поиск полностью совпадет с *итерацией 1* метода покоординатного спуска. По окончании исследующего поиска мы перемещаемся из точки $(x_0; y_0) = (0; 10)$ в точку $(x_1; y_1) = (9.0000; 8.2000)$. $e_x = 9$, $e_y = 1.8000$.

Поиск по образцу. Так как не достигнута заданная точность ($e_x > \varepsilon$ и $e_y > \varepsilon$), выполняем спуск на величину a вдоль прямой, соединяющей точки (x_0; y_0) и (x_1; y_1) (рис. 4).

Из рис. 4 следует, что координаты точки (x_2; y_2), полученной в результате спуска будут равны:

$$x_2 = x_1 + a\cos\alpha = x_1 + a\frac{x_1 - x_0}{d},$$

$$y_2 = y_1 + a\sin\alpha = y_1 + a\frac{y_1 - y_0}{d},$$

где $d = \sqrt{\left(x_1 - x_0\right)^2 + \left(y_1 - y_0\right)^2}$.

Величину шага a вдоль прямой нужно выбрать так, чтобы попасть в точку с самым минимальным значением функции на этой прямой, то есть решить задачу:

$$f(x_1 + b(x_1 - x_0), y_1 + b(y_1 - y_0)) \to \min_b,$$

где $b = \frac{a}{d}$.

Векторная форма записи этой задачи имеет вид:

$$f(\mathbf{x}_1 + b(\mathbf{x}_1 - \mathbf{x}_0)) \to \min_b.$$

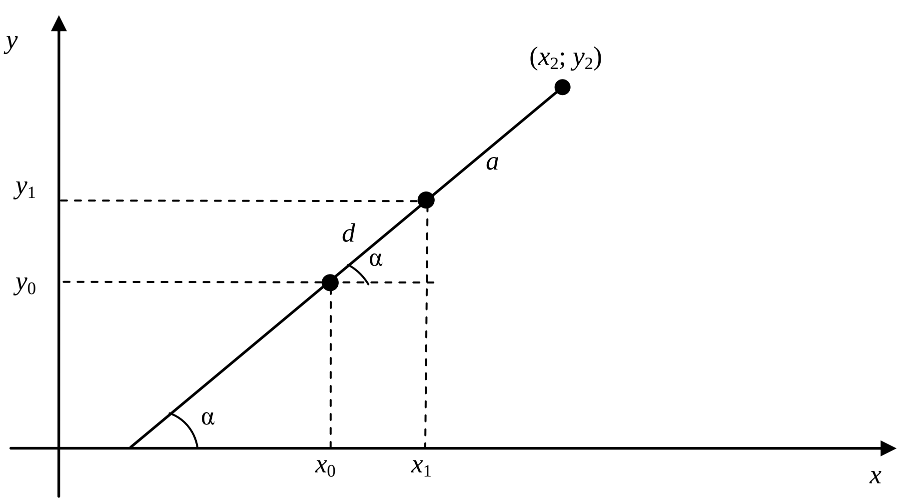


Рис. 4. Поиск по образцу

В нашем случае эта задача имеет вид:

$$f(9.0000 + b(9.0000), 8.2000 + b(8.2000 - 10)) \to \min_b.$$

Ее нужно решить с точностью ε = 0.0001 из начальной точки $b = 0$:

```
b=fminsearch(@(b)((x1+b*(x1-x0))-(y1+b*(y1-y0))).^2 +...
(((x1+b*(x1-x0))+(y1+b*(y1-y0))-10)/3).^2,0,...
optimset('TolX',10^(-4)));
```

Получаем ответ: $b = -0.1176$, следовательно, в конце поиска по образцу мы перемещаемся в точку $(x_2; y_2) = (x_1 + b(x_1 - x_0); y_1 + b(y_1 - y_0)) = (7.9414; 8.4117)$.

Выполним переобозначение: $(x_0; y_0) = (x_2; y_2)$.

Итерация 2.

Исследующий поиск. Решаем задачу $f(x, 8.4117) \to \min$ на интервале $x \in [0; 10]$ с точностью ε = 0.0001:

Поучаем $x_1 = 7.7294$, $e_x = 0.2120$.

Решаем задачу $f(7.7294, y) \to \min$ на интервале $y \in [0; 10]$ с точностью ε = 0.0001:

Поучаем $y_1 = 7.1835$, $e_y = 1.2282$.

Так как не достигнута заданная точность ($e_x > \varepsilon$ и $e_y > \varepsilon$), выполняем поиск по образцу.

Поиск по образцу. Решаем задачу $f(7.7294 + b(7.7294 - 7.9414), 7.1835 + b(7.1835 - 8.4117)) \to \min$ с точностью ε = 0.0001 из начальной точки $b = 0$.

Получаем: $b = 0.1833$, следовательно, $(x_0; y_0) = (7.6905; 6.9584)$.

Далее продолжаем итерации, пока не выполнится условие: $e_x < \varepsilon$ и $e_y < \varepsilon$. В результате получим ответ: $x_m = 4.9998 \pm 0.0001$, $y_m = 4.9998 \pm 0.0001$.

Всего для достижения заданной точности потребовалось 29 раз решать задачу одномерной оптимизации (рис. 5).

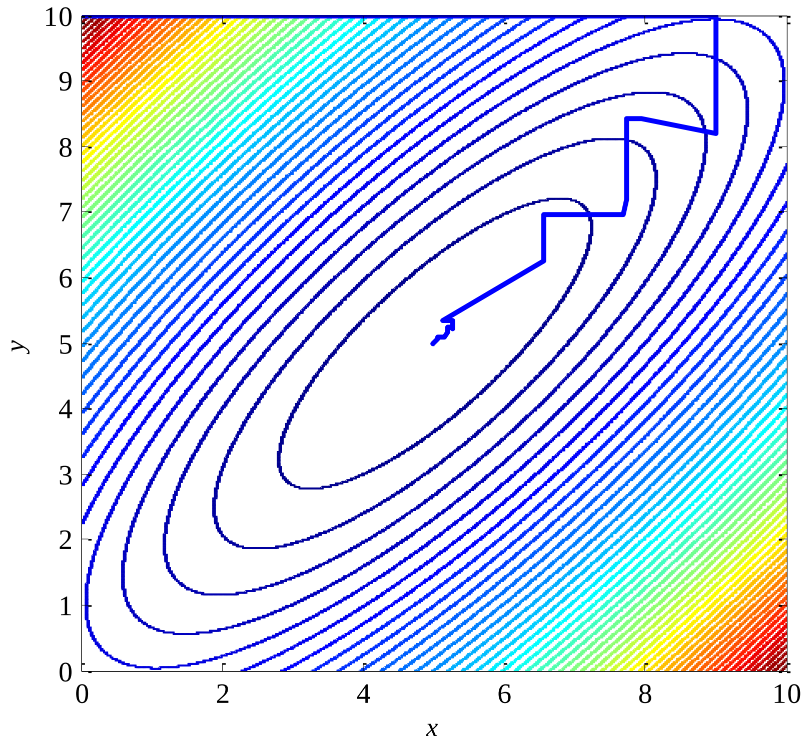


Рис. 5. Поиск минимума методом Хука-Дживса

Алгоритм Хука-Дживса (ускоренная модификация) для поиска минимума функции $f(\mathbf{x})$ на заданном интервале $\mathbf{x}\in[\mathbf{A};\ \mathbf{B}]$ с точностью $\boldsymbol{\varepsilon}$ можно описать следующим образом.

Входные данные: $f(\mathbf{x})$ – функция от n переменных $x_1, \ldots, x_n$; $\mathbf{A}$, $\mathbf{B}$ – левые и правые границы интервала поиска по каждой из переменных, $\boldsymbol{\varepsilon}$ – точность.

Выходные данные: $\mathbf{x}_m$ – точка минимума функции.

Шаг 1. Задать $\mathbf{x}_0 \in [\mathbf{A};\ \mathbf{B}]$.

Шаг 2. Задать $k = 0$.

Шаг 3. $k = k + 1$.

Шаг 4. Найти x_{1k} – точку минимума функции

$$f\left(x_k\right) = f\left(x_1, \ldots, x_n\right)\Big|_{x_i = x_{0i};\ i=1,\ldots,n;\ i\neq k}$$

на интервале $x_k\in[A_k;\ B_k]$.

Шаг 5. Вычислить $e_k = |x_{1k} - x_{0k}|$.

Шаг 6. Если $k < n$: перейти к шагу 3.

Шаг 7. Если $\mathbf{e} < \boldsymbol{\varepsilon}$: $\mathbf{x}_m = \mathbf{x}_1$, завершить алгоритм.

Шаг 8. Найти b – точку минимума функции $f(\mathbf{x}_1 + b(\mathbf{x}_1 - \mathbf{x}_0))$.

Шаг 9. $\mathbf{x}_0 = \mathbf{x}_1 + b(\mathbf{x}_1 - \mathbf{x}_0)$, перейти к шагу 2.

В случае если целевая функция на исследуемом интервале не унимодальна, то есть имеет несколько локальных экстремумов, при оптимизации возникает задача поиска глобального экстремума.

Глобальная оптимизация

Различают стохастические и детерминировано-стохастические численные методы глобальной оптимизации.

Детерминированно-стохастические численные методы глобальной оптимизации

Наиболее очевидная идея поиска глобального минимума состоит в поочередном поиске локальных минимумов целевой функции и выборе наименьшего среди них. При этом для поиска локальных минимумов можно применять какой-либо известный метод оптимизации унимодальных функций (детерминированный), каждый раз начиная оптимизацию из разных начальных точек. Начальная точка каждый раз может выбираться случайным образом (стохастически).

Алгоритм 1 глобальной оптимизации «basing-hopping» для поиска минимума функции $f(\mathbf{x})$ на заданном интервале $\mathbf{x} \in [\mathbf{A}; \mathbf{B}]$ с точностью ε можно описать следующим образом.

Входные данные: $f(\mathbf{x})$ – целевая функция, $\mathbf{A}$, $\mathbf{B}$ – левая и правая граница интервала поиска, ε – точность, $t_{\max}$ – максимальное число неудачных попыток.

Выходные данные: $\mathbf{x}_m$ – точка глобального минимума функции.

Шаг 1. Задать счетчик $t = 0$.

Шаг 2. Выбрать случайное $\mathbf{x}_0 \in [\mathbf{A}; \mathbf{B}]$.

Шаг 3. Любым детерминированным методом найти $\mathbf{x}_m$ – точку минимума функции $f(\mathbf{x})$ на интервале $\mathbf{x} \in [\mathbf{A}; \mathbf{B}]$ из начальной точки $\mathbf{x}_0$ с точностью ε.

Шаг 4. Выбрать случайное $\mathbf{x}_0 \in [\mathbf{A}; \mathbf{B}]$.

Шаг 5. Любым детерминированным методом найти $\mathbf{x}^*$ – точку минимума функции $f(\mathbf{x})$ на интервале $\mathbf{x} \in [\mathbf{A}; \mathbf{B}]$ из начальной точки $\mathbf{x}_0$ с точностью ε.

Шаг 6. Если $f(\mathbf{x}^*) < f(\mathbf{x}_m)$, $t = 0$, $\mathbf{x}_m = \mathbf{x}^*$, перейти к шагу 4.

Шаг 7. $t = t + 1$.

Шаг 8. Если $t > t_{\max}$: завершить алгоритм.

Шаг 9. Перейти к шагу 4.

Алгоритм завершается, когда $t_{\max}$ раз не удается получить локальный минимум, меньший найденного ранее.

В ряде случаев глобальный минимум можно найти быстрее, чем с помощью алгоритма *«basing-hopping»*, если каждый раз выбирать начальную точку, в которой значение целевой функции меньше, чем в найденной на предыдущей итерации точке локального минимума. В этом случае локальная оптимизация каждый раз позволит получить минимум меньший, чем на предыдущей итерации.

Алгоритм 2 глобальной оптимизации «random-flop» для поиска минимума функции $f(\mathbf{x})$ на заданном интервале $\mathbf{x} \in [\mathbf{A}; \mathbf{B}]$ с точностью ε можно описать следующим образом.

Входные данные: $f(\mathbf{x})$ – целевая функция, $\mathbf{A}$, $\mathbf{B}$ – левая и правая граница интервала поиска, ε – точность, $t_{\max}$ – максимальное число неудачных попыток.

Выходные данные: $\mathbf{x}_m$ – точка глобального минимума функции.

Шаг 1. Выбрать случайное $\mathbf{x}_0 \in [\mathbf{A}; \mathbf{B}]$.

Шаг 2. Задать счетчик $t = 0$.

Шаг 3. Любым детерминированным методом найти $\mathbf{x}_m$ – точку минимума функции $f(\mathbf{x})$ на интервале $\mathbf{x} \in [\mathbf{A}; \mathbf{B}]$ из начальной точки $\mathbf{x}_0$ с точностью ε.

Шаг 4. Выбрать случайное $\mathbf{x}_0 \in [\mathbf{A}; \mathbf{B}]$.

Шаг 5. Если $f(\mathbf{x}_0) < f(\mathbf{x}_m)$, перейти к шагу 2.

Шаг 6. $t = t + 1$

Шаг 7. Если $t > t_{\max}$, завершить алгоритм.

Шаг 8. Перейти к шагу 4.

Алгоритм 2 завершается, когда $t_{\max}$ раз не удается выбрать точку, в которой значение целевой функции меньше, чем в точке минимума, найденного ранее.

Во многих методах глобальной оптимизации локальная детерминированная оптимизация не применяется вовсе, они являются стохастическими. Например, одним из простейших стохастических методов глобальной оптимизации является метод случайного перебора, при котором в случайном порядке выбираются точки из области поиска и запоминается точка с наименьшим значением целевой функции.

Среди стохастических методов оптимизации широкое распространение получили методы, основанные на моделировании природных явлений. Большинство из них относится к классу так называемых роевых алгоритмов, моделирующих поведение группы связанных объектов (муравейника, пчелиного роя, стаи птиц и др.).

Алгоритм пчелиной колонии

Классическим представителем роевых алгоритмов является алгоритм пчелиной колонии (*artificial bee colony optimization*, *ABC*), предложенный в 2005 году. Алгоритм основан на моделировании процесса поиска нектара пчелиным роем.

Поиск нектара осуществляют специальные пчелы-разведчики. Обнаружив место с большим количеством медоносов, разведчики возвращаются в улей и передают информацию о найденном месте. В указанные разведчиками направления для сбора нектара из улья вылетают пчелы-фуражиры. В места, где обнаружено больше нектара, вылетает больше фуражиров, в места с меньшим количеством нектара – меньше. Фуражиры собирают нектар и проводят уточняющую разведку в окрестности сбора, информацию о которой также сообщают в улей.

Алгоритм ABC для поиска глобального минимума функции $f(\mathbf{x})$ на заданном интервале $\mathbf{x} \in [\mathbf{A}; \mathbf{B}]$ с точностью ε можно описать следующим образом.

Входные данные: $f(\mathbf{x})$ – целевая функция, $\mathbf{A}$, $\mathbf{B}$ – левая и правая граница интервала поиска, ε – точность, $t_{\max}$ – максимальное число неудачных попыток, s – количество разведчиков, B – количество «отличных» участков, G – количество «хороших» участков, b – количество фуражиров, отправляемых на каждый «отличный» участок, g – количество рабочих пчел, отправляемых на каждый «хороший» участок, R – размер окрестности участка.

Выходные данные: $\mathbf{x}_m$ – точка глобального минимума функции.

Шаг 1. Выбрать s случайных точек из интервала $[\mathbf{A}; \mathbf{B}]$ и записать их в множество $\mathbf{X}$.

Шаг 2. Выбрать $\mathbf{x}_m$ – точку из $\mathbf{X}$ с наименьшим значением функции $f(\mathbf{x})$.

Шаг 3. Задать счетчик $t = 0$.

Шаг 4. Выбрать из $\mathbf{X}$ точки $\mathbf{x}_1, \ldots, \mathbf{x}_B$ с наименьшим значением функции $f(\mathbf{x})$ и записать эти B точек в множество $\mathbf{X_B}$.

Шаг 5. Для каждого $\mathbf{x}_i$ $(i \in \{1, \ldots, B\})$ выбрать b случайных точек из интервала $[\mathbf{x}_i + R; \mathbf{x}_i - R]$. Записать полученные Bb точек в множество $\mathbf{X_b}$.

Шаг 6. Выбрать из $\mathbf{X} \backslash \mathbf{X_B}$ точки $\mathbf{x}_1, \ldots, \mathbf{x}_G$ с наименьшим значением функции $f(\mathbf{x})$ и записать эти G точек в множество $\mathbf{X_G}$.

Шаг 7. Для каждого $\mathbf{x}_i$ $(i \in \{1, \ldots, G\})$ выбрать g случайных точек из интервала $[\mathbf{x}_i + R; \mathbf{x}_i - R]$. Записать полученные Gg точек в множество $\mathbf{X_g}$.

Шаг 8. Выбрать $s - B - G$ случайных точек из интервала $[\mathbf{A}; \mathbf{B}]$ и записать их в множество $\mathbf{X_s}$.

Шаг 9. $\mathbf{X} = \mathbf{X_B} \cup \mathbf{X_G} \cup \mathbf{X_b} \cup \mathbf{X_g} \cup \mathbf{X_s}$.

Шаг 10. Выбрать $\mathbf{x}^*$ – точку из $\mathbf{X}$ с наименьшим значением функции $f(\mathbf{x})$.

Шаг 11. Если $f(\mathbf{x}^*) < f(\mathbf{x}_m)$, $\mathbf{x}_m = \mathbf{x}^*$, перейти к шагу 3.

Шаг 12. $t = t + 1$

Шаг 13. Если $t < t_{\max}$, перейти к шагу 4.

Шаг 14. Если $R < ε$, завершить алгоритм.

Шаг 15. $R = R/2$, перейти к шагу 3.

Записью вида $\mathbf{X} \backslash \mathbf{X_B}$ обозначается разность множеств; $\mathbf{X_B} \cup \mathbf{X_G}$ – объединение множеств.

В алгоритме *ABC* радиус окрестности, в котором осуществляется уточняющий поиск, уменьшается вдвое после того, как t_{max} раз не удается получить точку, значение целевой функции в которой меньше, чем на предыдущей итерации. Когда радиус R окрестности становится меньше заданной точности ε, алгоритм завершается.

Пример глобальной оптимизации

Пример 3. Дана функция:

$$f(x, y) = (x - y)^2 + \left(\frac{x + y - 10}{3} \right)^2.$$

Требуется найти глобальный минимум $f(x,y)$ с точностью ε = 0.0001 на интервале $x \in [0; 10]$, $y \in [0; 10]$.

Рассмотрим решение поставленной задачи с помощью алгоритма дифференциальной эволюции [1].

Сгенерируем начальную популяцию на интервале $x \in [0; 10]$, $y \in [0; 10]$. В соответствии с рекомендациями она должна содержать от 10 до 20 точек. Пусть N = 10.

Получим, например, следующий случайный набор точек (рис. 6, табл. 1)

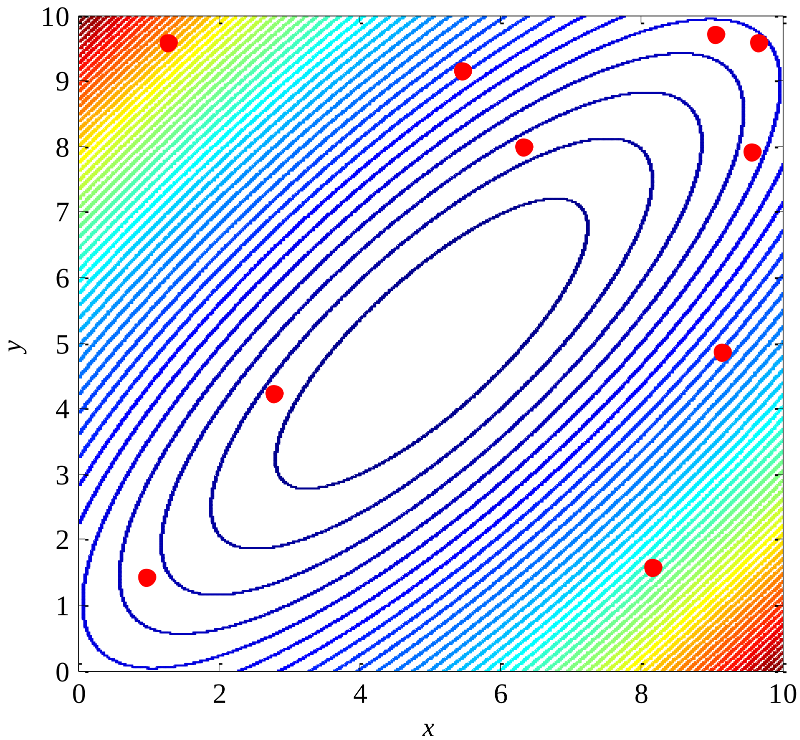


Рис. 6. Начальная популяция.

Таблица 1

x	8.1472	9.0579	1.2698	9.1337	6.3235	0.9754	2.7849	5.4688	9.5750	9.6488
y	1.5761	9.7059	9.5716	4.8537	8.0028	1.4188	4.2176	9.1573	7.9220	9.5949

Установим значения силы мутации $F = 0.5$ и вероятности мутации $P = 0.8$.

Итерация 1. Возьмем первую точку из популяции:

$$\mathbf{x}_1 = \begin{bmatrix} 8.1472 \\ 1.5761 \end{bmatrix}.$$

Случайным образом выберем среди оставшихся точек еще три, не совпадающие друг с другом, например:

$$\mathbf{x}_A = \begin{bmatrix} 9.0579 \\ 9.7059 \end{bmatrix}, \mathbf{x}_B = \begin{bmatrix} 9.5751 \\ 7.9221 \end{bmatrix}, \mathbf{x}_C = \begin{bmatrix} 2.7850 \\ 4.2176 \end{bmatrix}.$$

Вычислим точку-мутанта:

$$\mathbf{x}_M = \mathbf{x}_C + F(\mathbf{x}_A - \mathbf{x}_B) = \begin{bmatrix} 2.5264 \\ 5.1095 \end{bmatrix}.$$

Сгенерируем пару случайных чисел:

$$\mathbf{p} = \begin{bmatrix} 0.9340 \\ 0.6787 \end{bmatrix}.$$

Получим, что координата y наследуется от родителя $\mathbf{x}_M$, а координата x от родителя $\mathbf{x}_1$:

$$\mathbf{x}_T = \begin{bmatrix} 8.1472 \\ 5.1095 \end{bmatrix}.$$

Произведем отбор. $f(\mathbf{x}_T)$ = 10.4061, $f(\mathbf{x}_1)$ = 43.1879, следовательно, потомок $\mathbf{x}_T$ попадает в новую популяцию.

Затем те же действия повторим для второй точки:

$$\mathbf{x}_2 = \begin{bmatrix} 9.0579 \\ 9.7059 \end{bmatrix}.$$

$$\mathbf{x}_A = \begin{bmatrix} 2.7850 \\ 4.2176 \end{bmatrix}, \mathbf{x}_B = \begin{bmatrix} 9.1338 \\ 4.8538 \end{bmatrix}, \mathbf{x}_C = \begin{bmatrix} 5.4688 \\ 9.1574 \end{bmatrix}.$$

Точка $\mathbf{x}_A$ совпала с предыдущей $\mathbf{x}_C$ случайно.

$$\mathbf{x}_M = \mathbf{x}_C + F(\mathbf{x}_A - \mathbf{x}_B) = \begin{bmatrix} 2.2944 \\ 8.8393 \end{bmatrix}.$$

Для

$$\mathbf{p} = \begin{bmatrix} 0.6555 \\ 0.1712 \end{bmatrix}$$

обе координаты наследуется от родителя $\mathbf{x}_M$:

$$\mathbf{x}_T = \mathbf{x}_M = \begin{bmatrix} 2.2944 \\ 8.8393 \end{bmatrix}.$$

$f(\mathbf{x}_T)$ = 42.9780, $f(\mathbf{x}_1)$ = 8.9538, следовательно, потомок $\mathbf{x}_T$ в новую популяцию не попадет и вторая точка не изменится.

Те же действия повторим для оставшихся восьми точек. В результате получим новое поколение (табл. 2).

Таблица 2

x	8.1472	9.0579	4.9755	9.1337	6.3235	8.1560	2.7849	8.6602	9.5750	8.6632
y	5.1095	9.7059	9.2852	5.3882	8.0028	6.1182	4.2176	8.1576	7.9220	9.3687

Итерация 2. Выполнив те же действия, что и на итерации 1, над новым поколением, получим следующее поколение точек (рис. 7, табл. 3).

Таблица 3

x	8.1472	9.0579	9.7186	10.1760	6.3235	7.2886	2.7849	8.6602	9.5750	7.0345
y	5.1095	9.7059	9.2852	9.1674	8.0028	7.8133	4.2176	8.1576	7.9220	8.1980

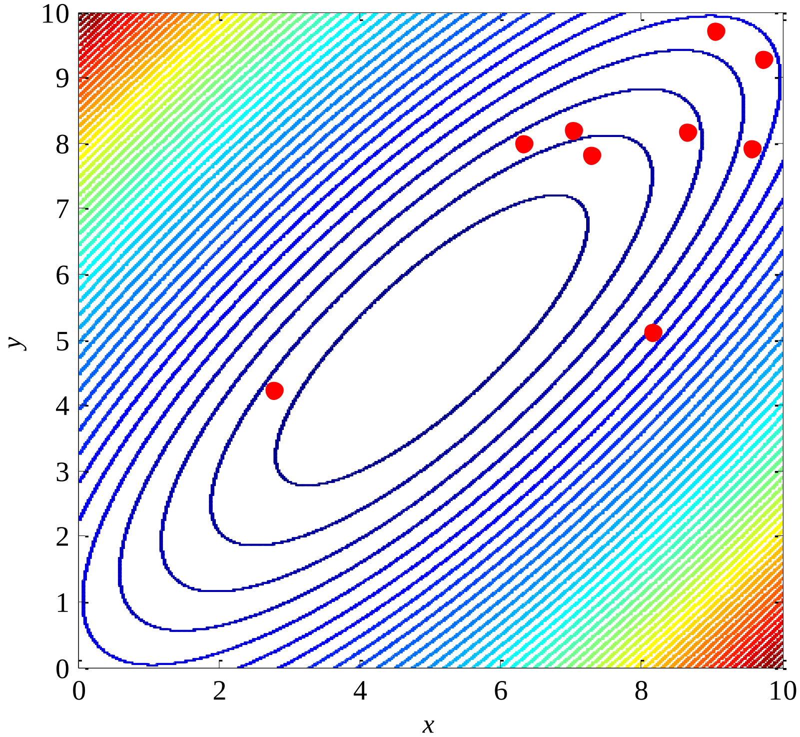


Рис. 7. Популяция в третьем поколении.

Заметим, что после второй итерации точки сместились в сторону минимума.

Будем продолжать итерации, пока разница между соответствующими координатами точек не станет меньше точности ε = 0.0001.

Расположение точек на седьмой итерации приведено на рис. 8.

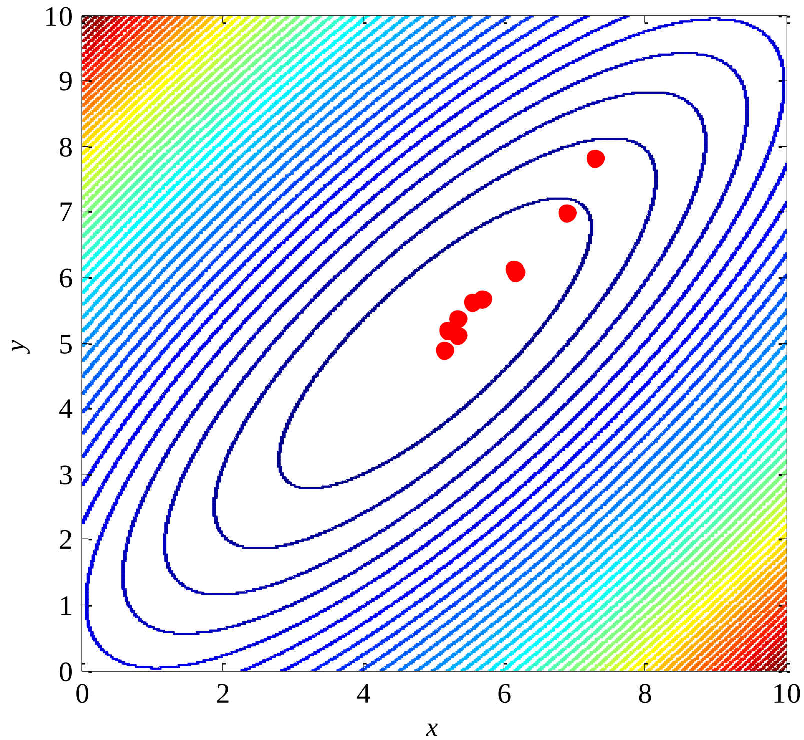


Рис. 8. Популяция в восьмом поколении.

В результате за 51 шаг получим популяцию, состоящую из одной точки с координатами (5.0001; 5.0001) и девяти точек с координатами (5.0000; 5.0000).

В пакете *MATLAB* графически отобразить процесс изменения популяции во времени можно с помощью команды *set*:

```
p=plot(Pnts(:,1),Pnts(:,2),'.r','EraseMode','xor','MarkerSize',20);
while (max(Pnts)-min(Pnts))>0.0001
    ...% мутация, скрещивание, отбор
    set(p,'Xdata', Pnts(:,1),'Ydata', Pnts(:,2));
    pause(1);
end
```

2. Задание по работе

Требуется решить оптимизационную задачу аналитически и одним из численных методов в соответствии с вариантом:

детерминированные методы

a) методом Хука-Дживса (см. раздел 2.2.3 в [1]);

b) модифицированным методом Хука-Дживса (см. раздел 1.2 в практикуме);

c) методом Розенброка (вращающихся координат) (см. раздел 2.2.3 в [1]);

d) методом Ньютона (см. раздел 2.2.3 в [1]);

e) методом BFGS (см. раздел 2.2.3 в [1]);

f) методом DFP (см. раздел 2.2.3 в [1]);

детерминированно-стохастические методы

g) методом «basing-hopping» (см. раздел 1.2 в практикуме);

h) методом «random-flop» (см. раздел 1.2 в практикуме);

стохастические методы

i) методом DE (см. раздел 2.2.3 в [1]);

j) методом ABC (пчелиной колонии) (см. раздел 1.2 в практикуме).

Численное решение задачи необходимо реализовать в виде программы на языке *MATLAB* или *Python* (по выбору обучающегося) и произвести анализ полученного решения. Для одномерной оптимизации рекомендуется использовать функции одномерной оптимизации языка *MATLAB* или *Python*.

Задача А. Серый Волк находится в точке *A* и хочет оживить Ивана-Царевича, находящегося в точке *B*. Оба они находятся между реками с мертвой и живой водой, текущими параллельно друг другу. Как Волку добраться до Ивана кратчайшим путем, набрав по дороге сначала живой, а затем мертвой воды (рис. 9 *а*)?

Задача Б. Серый Волк находится в точке *A* и хочет оживить Ивана-Царевича, находящегося в точке *B*. Оба они находятся между реками с мертвой и живой водой, вытекающими из одного ключа под углом 45º друг к другу (рис. 9 *б*). Как Волку добраться до Ивана кратчайшим путем, набрав по дороге сначала мертвой, а затем живой воды?

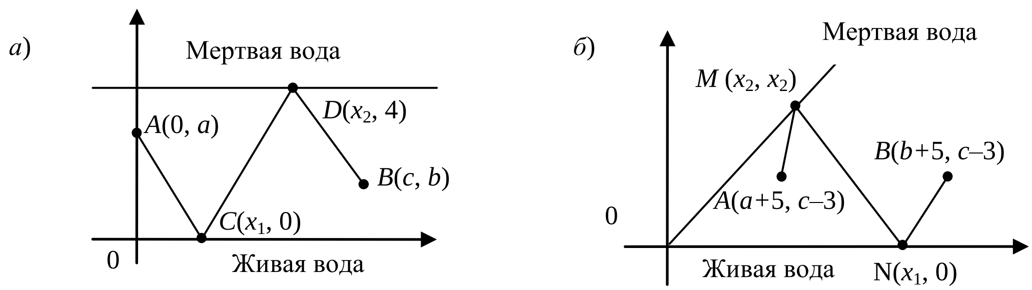


Рис. 9. Поясняющие рисунки к задачам

3. Порядок выполнения работы

Выполнение работы состоит из двух частей: аналитического решения задачи, решения задачи заданным численным методом с помощью его программной реализации.

При аналитическом решении задачи необходимо:

1) на основании условия задачи вывести целевую функцию;
2) найти стационарную точку функции методом Ферма;
3) проверить, что градиент функции в найденной точке равен нулю;
4) по знакоопределенности гессиана функции в найденной точке убедиться, что найденная стационарная точка является точкой минимума;
5) построить полученную оптимальную траекторию движения (рис. 9 с точными координатами).

При программной реализации решения на языке *MATLAB* или *Python* необходимо:

1) написать скрипт или функцию, реализующую решение задачи заданным численным методом оптимизации;
2) построить график линий уровня целевой функции и нанести на него найденную с помощью программы точку экстремума;

3) найти разность ответов, полученных при аналитическом и численном решении;

4) оценить время выполнения программы (например, с помощью функции *timeit*);

5) на график(и) линий уровня целевой функции нанести

   a. *для детерминированных методов*: точки, полученные после каждой одномерной оптимизации, и соединить их прямыми линиями (как на рис. 7);

   b. *для детерминированно-стохастических методов*: точки, полученные после каждой итерации, и соединить их прямыми линиями;

   c. *для стохастических методов*: точки популяции, полученные после выборочных итераций (как на рис. 6-8).

4. Содержание отчета

Отчет по лабораторной работе должен содержать следующие разделы.

1. Цель работы.
2. Задание по работе.
3. Описание оптимизационной задачи.

В данном разделе необходимо привести: формулировку задачи, поясняющий рисунок, вывод целевой функции и выбор интервала поиска для численного решения.

4. Аналитическое решение задачи.

В данном разделе необходимо описать поиск стационарной точки методом Ферма и определение вида стационарной точки.

5. Программная реализация решения.

В данном разделе необходимо привести: текст программы с комментариями; графики линий уровня целевой функции, на которых проиллюстрирован процесс поиска (см. п. 5 в порядке выполнения программной реализации) и найденный экстремум.

6. Вывод.

Пример вывода: «В результате выполнения работы было получено аналитическое и численное решение поставленной оптимизационной задачи. Аналитический ответ: …. Для поиска численного решения методом … была разработана программа на языке …. Полученный с помощью программы ответ равен … и с заданной точностью … совпадает с аналитическим. Объем программы составляет … строк. Среднее время выполнения программы составляет … мс».

5. Контрольные вопросы

1. В чем состоит основная идея метода покоординатного спуска?
2. В чем состоит основная идея метода Хука-Дживса?
3. В чем состоит основная идея метода Розенброка?
4. В чем отличие псевдо-ньютоновских методов от метода Ньютона?
5. Что нужно изменить в алгоритме покоординатного спуска, если необходимо осуществить поиск максимума?
6. Для заданной функции найдите экстремум методом Ферма. Выполните два шага покоординатного спуска из точки (10; 10), в качестве одномерного метода используя метод Ферма. Постройте на плоскости все найденные точки.

   a) $f(x,y)=2x^2+2y^2-2xy-4x-6y;$

   b) $f(x,y)=x^2+xy+(y-3)^2;$

   c) $f(x,y)=2x-2y+x^2+xy+y^2;$

   d) $f(x,y)=xy+2x^2+3y^2;$

   e) $f(x,y)=2x^2+2y^2+xy-4x-6y;$

   f) $f(x,y)=11x^2+3y^2+6xy+3(x-y)-22;$

   g) $f(x,y)=x^2+\frac{xy}{2}+(y-3)^2;$

   h) $f(x,y)=2x^2+3y^2-4xy-4(x-y)+4;$

   i) $f(x,y)=2xy+2x^2+y^2.$

7. Для заданной функции из п. 5 рассчитайте градиент в точке (–1; –1) и выполните оптимизацию в направлении вектора градиента методом Ферма.
8. Дан контурный график унимодальной функции двух аргументов. Изобразите путь покоординатного спуска.
9. Найдите численное решение с помощью функций многомерной оптимизации *MATLAB* или *Python* и сравните его с решением, найденным с помощью вашей программы, по точности и времени. Используйте следующие функции. Для *MATLAB*: *fminunc* (BFGS) и *fminsearch* (симплекс-метод Нелдера-Мида). Для *Python*: *scipy.optimize.minimize* с указанием *method='Nelder-Mead'* и *method='BFGS'*.
10. Приведите геометрическое решение задачи.
11. С помощью разработанной программы найдите минимум функции
    a) $f(x,y)=(a-x)^2+b(y-x^2)^2$;
    b) $f(x,y)=(x^2+y^2-2ax-2by+a^2+b^2)e^{x+y}$;
    c) $f(x,y)=(x^2+y^2-2ax-2by+a^2+b^2)e^{x}$;
    d) $f(x,y)=\dfrac{1}{a+(x-1)^2+(y-b)^2}$;
    e) $f(x,y)=\cosh(ax+by)+\cosh(x-1)$.
12. Проверьте знакоопределенность матрицы:
    a) $\begin{bmatrix}1 & 2\\ 2 & -1\end{bmatrix}$;
    b) $\begin{bmatrix}1 & 3 & 2\\ 3 & 5 & 2\\ 2 & 2 & 3\end{bmatrix}$.

6. Варианты заданий

№ варианта	1	2	3	4	5	6	7	8	9	10
Задача	А	Б	А	Б	А	Б	А	Б	А	Б
Метод	a	b	c	d	e	f	g	h	i	j
a	1	2	3	1	2	3	1	2	3	1
b	2	3	1	2	3	1	2	3	1	2
c	4	5	6	7	8	4	5	6	7	8

№ варианта	11	12	13	14	15	16	17	18	19	20
Задача	А	Б	А	Б	А	Б	А	Б	А	Б
Метод	j	i	h	g	f	e	d	c	b	a
a	2	3	1	2	3	1	2	3	1	2
b	3	1	2	3	1	3	1	2	2	3
c	4	5	6	7	8	4	5	6	7	8

№ варианта	21	22	23	24	25	26	27	28	29	30
Задача	А	Б	А	Б	А	Б	А	Б	А	Б
Метод	a	b	c	d	e	f	g	h	i	j
a	3	1	2	3	1	2	1	1	3	3
b	2	3	1	2	3	1	3	3	2	1
c	5	6	7	8	5	6	7	8	5	6

№ варианта	31	32	33	34	35	36	37	38	39	40
Задача	А	Б	А	Б	А	Б	А	Б	А	Б
Метод	j	i	h	g	f	e	d	c	b	a
a	2	3	1	2	2	3	1	2	1	3
b	3	1	2	3	1	2	3	1	3	1
c	4	5	6	7	4	5	6	7	4	5

Список литературы

1. Мироновский Л. А., Соловьева Т. Н., Шинтяков Д. В. Численные методы и оптимизация : учеб. пособие. СПб.: ГУАП, 2017. 147 с.

2. Тихомиров В. М. Рассказы о максимумах и минимумах. 2-е изд., испр. М.: МЦНМО, 2006. 200 с.